Research Records

The Manuscript Compendium

F. Amane
Department of Physics

Last comprehensive review: May 18, 2025

Contents

Contents 2

List of Figures 8

Introduction i
What is in the book i
How much topic is covered? ii

1 Introduction to Artificial Intelligence 1
1.1 History of artificial intelligence 1
1.1.1 Roughly, artificial intelligence 2
1.1.2 The themes of AI 2
1.1.3 Precursor - From the ancient time to 18th century 3
1.1.4 The 1940s 4
1.1.5 The 1950s to 1960s 5
1.1.6 The first AI winter (1966-1973) 8
1.1.7 Knowledge-based systems - expert system (1969-1979) 9
1.1.8 Neural network, the return (1986) 13
1.1.9 Adoption of scientific method (1987) 13
1.1.10 Intelligent agents (1995) 15
1.1.11 The revolution of Internet (2001) 15
1.2 Defining intelligence 15
1.2.1 Historical accounts 15
1.2.2 The Descartes experiments 18
1.2.3 The economic of definiendum 19
1.3 Formalizing artificial intelligence - an attempt 19
1.4 Philosophical arguments against AI 22
1.4.1 Descartes's argument (1700s) 22
1.4.2 Categorization - Strong and weak AI (1991) 23
1.4.3 The Chinese Room Argument (1980s) 24
1.4.4 The Gödelian argument (1961) 26

I Preliminary 29

2 Foundational mathematics 31
2.1 Elementary logics 31
2.1.1 The Propositional logics 32
2.1.2 From frames to statements 36

2.1.3 Characterization . . . 38
2.1.4 Restricted variables . . . 38
2.2 Naive set theory . . . 39
2.2.1 De Morgan's theorem . . . 40
2.3 Formal set theory . . . 40
2.3.1 An informal view on sets . . . 42
2.3.2 Classes . . . 42
2.3.3 Axiomatizations . . . 42
2.4 Ordered pairs and relations . . . 44
2.4.1 Equivalence relations . . . 45
2.5 Functions . . . 46
2.6 Product sets, index notation . . . 48

3 Linear algebra 49
3.1 Notation and structure . . . 50
3.1.1 Array form . . . 50
3.2 Vectors . . . 52
3.2.1 Properties of vectors . . . 56
3.3 Matrix . . . 58
3.3.1 Special matrices, operations, and properties . . . 59

4 Probability and Statistics 63
4.1 What is probability? . . . 63
4.1.1 The fundamental principle of probability . . . 63
4.1.2 Subjectivity of probability . . . 63
4.2 Space of probability . . . 64
4.3 Combinatorial probability . . . 65
4.4 Axiomatic probability . . . 67
4.4.1 Axiom of probability . . . 67
4.4.2 Boole's inequality . . . 68
4.4.3 Conditional probability . . . 68
4.5 Random variables . . . 69
4.6 Expected values . . . 69
4.6.1 Variance . . . 71
4.7 Some distribution functions . . . 71
4.7.1 Bernoulli and binomial distribution . . . 71
4.7.2 Poisson distribution . . . 72
4.7.3 Uniform distribution . . . 73
4.7.4 Normal distribution . . . 73
4.7.5 Exponential distribution . . . 74
4.7.6 Distribution function of variables . . . 75

5 Metric spaces, measure theory 77
5.1 Metric theory . . . 77
5.1.1 Pseudometric spaces . . . 77
5.2 Measure . . . 78
5.2.1 Sigma-algebra . . . 79
5.2.2 A few sigma algebras . . . 79
5.2.3 Measure . . . 80

6 Concentration inequalities 83
6.1 What are concentration inequalities? . . . 83
6.2 Basic tools . . . 84
6.2.1 Markov's inequality . . . 84
6.2.2 Chebyshev's inequality . . . 85
6.2.3 McDiarmid's inequality . . . 85

II Theory 87

7 Classical mathematical modelling 89
7.1 The supposed goal of modelling . . . 89
7.2 Models, systems, and questions . . . 90
7.2.1 The modelling scheme . . . 91
7.2.2 Simulation . . . 92
7.2.3 System . . . 92
7.2.4 Conceptual and physical model . . . 93
7.3 Mathematical models . . . 93
7.3.1 Definitions . . . 94
7.3.2 State variables and system parameters . . . 95
7.3.3 The Problem-Solving Scheme . . . 96
7.3.4 The black-box interpretation . . . 97
7.4 Flavours of modelling definitions . . . 98
7.4.1 Phenomenology and Mechanistic . . . 98
7.4.2 Stationary and unstationary models . . . 101
7.4.3 Classification of models . . . 101
7.5 The don'ts of mathematical modelling . . . 104
7.6 Conclusion . . . 107
7.7 Appendix . . . 107
7.7.1 Linear programming . . . 107

8 Classical learning theory 109
8.1 Before the learning theory . . . 109
8.2 Classical theory . . . 110
8.2.1 Principles . . . 111
8.2.2 Concept and hypothesis . . . 111
8.2.3 Phenomenological aberration . . . 112
8.2.4 Estimation and approximation error . . . 117
8.3 Structural example of learning setting . . . 117
8.3.1 First section . . . 118
8.3.2 Second section . . . 118
8.3.3 Third section . . . 119
8.4 The problems within classical learning theory . . . 120
8.4.1 Data and the general setting . . . 120
8.4.2 Observational partitioning . . . 121
8.5 Learning criteria on time - PAC theory . . . 123
8.5.1 Classical PAC-learning . . . 124
8.6 Generalization bound for PAC-learning . . . 128
8.6.1 Consistent Learning . . . 128
8.6.2 Finite H, consistent hypothesis . . . 128

8.6.3 Examples . . . 130
8.6.4 Finite hypothesis sets H - inconsistent case . . . 130
8.7 (Agnostic) General PAC-learning . . . 132
8.7.1 Noise . . . 133
8.8 Occam's Razor . . . 134
8.8.1 Occam Learning and Succinctness . . . 134
8.9 Rademacher Complexity . . . 136
8.9.1 Rademacher descriptions . . . 137
8.9.2 Growth function . . . 142
8.10 Vapnik-Chervonenkis Theory . . . 143
8.10.1 Clarification of the linear halfspaces . . . 144
8.10.2 VC-dimension . . . 145

9 Introduction to classical connectionism 155
9.1 Biological inspiration . . . 155
9.2 The neuron model . . . 160
9.2.1 Principles and philosophy . . . 162
9.2.2 Class separation . . . 164
9.3 Notation . . . 165
9.4 Classical neuron template . . . 165
9.5 Class $\mathcal{N}_0$ simplex . . . 167
9.5.1 Chaining standard unit . . . 169
9.6 Class $\mathcal{N}_1$ simplex . . . 171
9.6.1 McCulloch-Pitts multivariate . . . 172
9.6.2 Minsky-Papert Perceptron . . . 174

10 General principles 179
10.1 Defining artificial intelligence . . . 179
10.1.1 Intelligence criterion . . . 180
10.1.2 The should not of defining AI . . . 181
10.1.3 Deferences in approach? . . . 181

11 The principle neural architecture 183
11.1 Neurons perspective . . . 183
11.1.1 Analysis . . . 186
11.1.2 Why we call it minimal . . . 187

III Drafts 189

12 Double Descent 191
12.1 Note . . . 191
12.2 Developing analysis . . . 191
12.2.1 Statistical learning theory . . . 191
12.2.2 Double descent . . . 191
12.3 Issues . . . 197
12.3.1 The messiness of analysis . . . 197
12.3.2 The ambiguity of analysis . . . 197
12.3.3 Stepping in the wrong direction . . . 198
12.3.4 Main problems . . . 198

12.3.5 Hypothesis . . . 198
12.3.6 A rather simple solution . . . 200
12.4 The perspective of modelling theory . . . 202
12.4.1 Structures of object . . . 202
12.5 The perspective of modified learning setting . . . 206
12.6 Bias-variance: A history . . . 207
12.6.1 Another approach - No-free-lunch . . . 209
12.7 Abstract . . . 211
12.8 Introduction . . . 211
12.8.1 Statistical Learning and Double Descent . . . 211
12.8.2 Relation to Graph Neural Network . . . 212
12.9 Outline . . . 213
12.10 Background . . . 213
12.11 Bias-variance tradeoff . . . 215
12.11.1 Defining the bias and variances . . . 215
12.11.2 Precursor (Geman, 1992) . . . 216
12.11.3 Formalism issues and uncertainty . . . 218
12.11.4 Approximation-Estimation tradeoff . . . 219
12.12 Double descent . . . 219
12.13 The break-off between theoretical and modern practice . . . 220
12.14 Preliminary experiments . . . 221
12.14.1 Polynomial model . . . 221
12.14.2 Support Vector Machine (SVM) . . . 221
12.15 Experiments . . . 223
12.15.1 Main result . . . 223
12.15.2 Analysis of GNN . . . 225
12.15.3 Experiment 1: Identifying bias-variance in GNN . . . 225
12.16 Conclusion . . . 226
12.17 Related works . . . 226

13 Deconstruction of Neural Network Architecture 227
13.1 Abstract . . . 227
13.2 Introduction . . . 227
13.3 Constructions . . . 228
13.3.1 Multilayer perceptron (MLP) . . . 229
13.3.2 Remark . . . 230

IV Appendix 231

Index 233

List of transfer functions 235
Classical transfer functions . . . 235
Hard limit (`hardlim[x]`) . . . 235
Symmetric hard limit (`hradlims[x]`) . . . 236
Linear family (`satlin[x]`, `satlins[x]`, `purelin[x]`) . . . 236
Sigmoid (`sigmoid[x]`) and log-sigmoid (`logsigmoid[x]`) . . . 237
Hyperbolic tangent (`tansig[x]`) . . . 239

Bibliography 241

List of Figures

1.1 Well, look at that. Can you guess if we are about to face another one soon enough? 9
1.2 The diagram of a typical expert system. Here, human involvement is fairly representative in the figure, and is illustratively indicating the overwhelming reliance on human touches. 11

3.1 An illustration of a vector in two-dimensional vector form in endpoints representation, and directional-magnitude representation. 53
3.2 The **parallelogram law** for vector addition of two vectors x and y on adjacent side. 55
3.3 Illustration of matrix multiplication. Taken from Andrilli and Hecker [2010]. . . . 59
3.4 Illustration of matrix multiplication for 2×2 shape. 60

6.1 Markov's inequality bounds the probability for the shaded region $\mathbb{P}[X \geq a]$ 84

7.1 The problem-solving scheme from the mathematical modelling perspective 97
7.2 There exists an unbreakable wall in the black-box condition - throwing a dart in blind, except perhaps it can be right. 98
7.3 A typical pendulum with degree 1, for parameter θ as angle, and a rod of length ℓ connecting the origin to the mass m. 99
7.4 With the question Q, you can ask everything, including the... not so pure one. . . 102
7.5 Plato's allegory of the cave by Jan Saenredam, according to Cornelis van Haarlem, 1604, Albertina, Vienna 105

8.1 An illustration of statistical learning theory on the evaluation of the risks and errors, during learning process. c' is presented in the 'orbital' vicinity around c, with its distance of certain metric define how 'accurate' the reconstruction from distribution can be. Of the hypothesis set $\mathcal{H}$, there exists the Bayes hypothesis h_B and an arbitrary 'random' hypothesis h, and their respective measure. 115
8.2 An illustration of the (supervised) statistical process. Phase III contains two parts: First is the evaluation $\nabla(h, c)$ according to the data $\mathcal{D}$, and second is the Update process to re-align c to the actual target. 118
8.3 **Conceptual representation of the sample set partitions and its effect on iterative process.** (1), for S_1, and of the specified ordering, h is able to make it to the optimal point compared to the actual concept c. The bubble around c is what we call irreducible error, intrinsic of the observational space. (2) for S_2, of the changing dataset, while of the same partition but also changing order, gives different volatile path, and perhaps suboptimal performance compared to the first dataset case. Note that they are the supposed *optimal path* of both S_1, S_2. If randomization is introduced, may suboptimal path will occur, and the result will differ. 122

8.4 Partitioning process and its error potential consideration. We assume each partition includes the irreducible error ϵ accompanied by the n partition, belongs to the furthest partitioning set. Within every increasing partition, for supposed distributed data (unordered data), the generalization risk is further decomposed. 123
8.5 Illustration of the notion of hyperplane in two and three dimensions. This can be extended to $n > 3$ dimension, but no figurative illustration can be found (or ever understood). Taken from Introduction to Statistical Learning using R Book Club by The R4DS Online Learning Community. 145
8.6 Illustration of a halfspace **region** created by a hyperplane on the side of the axis. If the halfspace is created of the unit frame hyperplane (aligning with the axis), then it is called a *normal space partitioning halfspace*. 145
8.7 VC-dimension of intervals on the real line. (a) Any two points can be shattered. (b) No sample of three points can be shattered as the $(+, -, +)$ labelling cannot be realized. Taken from Mohri et al. [2012]. 146
8.8 Unrealizable dichotomies for four points using hyperplanes in $\mathbb{R}^2$. (a) All four points lie on the convex hull. (b) Three points lie on the convex hull while the remaining point is interior. Taken from Mohri et al. [2012]. 147
8.9 Illustration of convex and non-convex set. The segment $[x, y]$ must be fully contained in the region of the set, otherwise it is not convex. 148
8.10 Illustration of (left-hand side) $d = 1$ Radon partition, and (middle and right-hand side) $d = 2$ Radon partition. More options are available as d increases. 151

9.1 The simplistic, schematic illustration of the structure of the biological neuron. . . 156
9.2 An illustration of Santiago Ramón y Cajal on the structure and design of a biological brain network. Many of these was made during his career. 158
9.3 Examples of the rich variety of nerve cell morphologies found in the human nervous system. Tracings are from actual nerve cells stained by impregnation with silver salts (the socalled Golgi technique the method used in the classical studies of Golgi and Cajal). Asterisks indicate that the axon runs on much farther than shown. Note that some cells, like the retinal bipolar cell, have a very short axon, and that others, like the retinal amacrine cell, have no axon at all. The drawings are not all at the same scale. Some more details about the jargon is the *retinal bipolar cells*, which are neurons that connect the outer retina to the inner retina, for processing layer (or projection neurons, where all information are relayed from this connection.); the *retinal ganglion cell, amacrine cells* are the same visual processing unit; Cerebellar Purkinje cells (a type of GABAergic neurons) uniquely determined for cerebella cortex (for processing large data, and coordinating functions like cognition and emotions.). Reused from Purves et al. [2004]. 159
9.4 **An illustrative example of the abstraction and categorization by 'size' of different components and constructs in the neuron model.** By the order of abstraction k, we assign a notion of size on different neural structure, by increasing complexity, and backward compatibility (described to be composed of previously defined objects). The first two stage for $k = 1, 2$ includes the standard basis components $\{n_{0,i}\}$ and the standard neuron class N_0, respectively. 161
9.5 **The standard minimal configuration of any neuron** $x \in \mathcal{N}_i$. We denote p, q for particular neuron input and output sequences. 166
9.6 Commutative diagram of the standard $\mathcal{N}_0(\mathbb{R})$ class. The in-out objects are denoted in blue, the operators are denoted in red, and the objective mass (parameters) are denoted in yellow. The procedure is then denoted of four successive processes of S_i, up to S_4. 167

9.7 **Illustration standard neuron class $\mathcal{N}_0$.** (a). We regard the component of the model as $\mathbf{M}$, consists of the mass and the operations $\mathbf{H}$. (b). Instead of considering the operation as subcomponent of the model structure, decomposition gives them separated with two types of operation - either *processing* operators (operations that prepare the parameters) or *transforming* operators (act on the prepared processing that it receives). 168
9.8 Chaining of multiple standard unit on each other. 169
9.9 Illustration of the chaining process and the nested function chaining between σ_i. . 170
9.10 Initial starting configuration for x_1, x_2, x_3 and their functionals $\sigma_1, \sigma_2, \sigma_3$ under the same initializer. 170
9.11 Sequential configuration for x_1, x_2, x_3 and their functionals $\sigma_1, \sigma_2, \sigma_3$ of the same initializer with $\mathbf{w} = (2.0, -2.5, 2.5)$. 171
9.12 **Illustrative simplified commutative diagram schematic of r-input neuron process unit.** The specific field dimension transition for a standard neuron of $\mathbb{R}$ is denoted specifically between transitions. 172
9.13 **Schematic of the AND and OR logical configuration.** The only change in their construction is that the criterion in the function is now different - from $|\mathbf{x}| = \Sigma(\mathbf{x})$ (which means all signals' sum must be equal to their absolute magnitude - in agree state), or $\Sigma(\mathbf{x}) \neq 0$ (as long as a single signal is active is enough). 173
9.14 Schematic of the NOT logical configuration. 173
9.15 Schematic of the NAND and NOR logical configurations. 174
9.16 Results and values of the predicate ψ_{circle} on various geometrical shape. The detail of what gives the criterion is not mentioned, however implicitly defined to be naturally encoded. Taken from Minsky and Papert [1988]. 175
9.17 Results and values of the predicate ψ_{convex} on various geometrical shape. While still being implicitly defined, computationally this predicate takes more complexity than the circle predicate. Taken from Minsky and Papert [1988]. 175

11.1 Minimal neuron structure 184
11.2 The compound structure construction. The same component can be seen, for n_i, n_o and M. Multiple consecutive components construct some components, and further outward. Also, we also reflect the complexity of $\mathcal{C}$ for a given architecture. 185

12.1 (a) A typical example of bias-variance tradeoff in a statistical dataset. (b) When graphed into a continuous notion, we gain the complexity-error graph. Notice that it specifically goes for the *test error*, which fits - the representative problem of prediction. 192
12.2 **Curves for training risk (dashed line) and test risk (solid line).** (a) The classical *U-shaped risk curve* arising from the bias-variance trade-off. (b) The *double descent risk curve*, which incorporates the U-shaped risk curve (i.e., the "classical" regime) together with the observed behaviour from using high capacity function classes (i.e., the "modern" interpolating regime), separated by the interpolation threshold. The predictors to the right of the interpolation threshold have zero training risk. Reproduced from Belkin et al. [2019a]. 193
12.4 **Left**: Test error as a function of model size and train epochs. The horizontal line corresponds to model-wise double descent–varying model size while training for as long as possible. The vertical line corresponds to epoch-wise double descent, with test error undergoing double-descent as train time increases. **Right** Train error of the corresponding models. All models are Resnet18s trained on CIFAR-10 with 15% label noise, data-augmentation, and Adam for up to 4K epochs. 196

12.3 **Left**: Train and test error as a function of model size, for ResNet18s of varying width on CIFAR-10 with 15% label noise. **Right**: Test error, shown for varying train epochs. All models trained using Adam for 4K epochs. The largest model (width 64) corresponds to standard ResNet18. Reproduced from Nakkiran et al. [2019]. 196
12.5 The representative order of representation and description. As of the name implied, in transition to a mathematical formalism and language, there must then exist a representation to each and every element of certain subject. The process of doing is this called *external encoding*, and is true also between portion of mathematical-encoded system to each other, if they are distinct. The reverse act is called again, *decoding*, and between mathematical subjects to each other might as well be called *internal encoding*, with respect to the mathematical language. 206
12.6 A conceptual illustration on the running flow of an n-layer GNN on particular structure of interest. Note that the data section itself has particular embedding structure on its own. 225

13.1 (a) Figure of the original organization of the biological model of the brain functions. (b) Specifically, note that it is specifically for optical case, but can be extended to others type. Furthermore, the layer between last A-unit and the response units, there exists a pattern of feedback loop. 229

1 The typical hard limit transfer function with fixed a, and fixed range for x in $[0, 1]$. 235
2 The typical hard limit transfer function with variable inhibition a, and fixed range for x in $[0, 1]$. 236
3 The typical symmetric hard limit transfer function with static inhibition a, and fixed range for x in $[-1, +1]$. As specified, this is the normal-extended range. 237
4 The saturating linear with linear region of $[0, 1]$. A smoother variation would be something like sigmoidal functions, that is. 237
5 The symmetric saturating linear with linear region of $[-1, 1]$, a positive-negative variation of the saturating linear. 238
6 The sigmoidal function channel 238
7 The logarithmic sigmoidal function channel. Notice that the range of `logsigmoid` is $[-\infty, 0]$, making it somewhat weird of a choice for a transfer function. 238
8 The hyperbolic tangent transfer function channel. 239

Introduction

This book serves a dual purpose - one is to record my understanding and knowledge about the topic, and related to it, and one is to develop my own theories and models - most likely original, more often than not it's not. Also, it will also include rough implementation, papers' ideas analysis, and else. So, a bag full of everything, I suppose. a dual purpose - one is to record my understanding and knowledge about the topic, and related to it, and one to directly record and write down my discovery, or rather, most of the time just analysis and designs.

It might sound as if this book is entirely self-sufficient, and also self-serving, yet it isn't. The text cannot take into account all preliminaries or requirements of understanding, yet. It is simply too large to take all in, because the field itself is very complex. Self-serving side, perhaps it can be broken down to $0-1$ loss function two (it's a joke), since it is perhaps too difficult to follow directly from the get-go, so I pretend to be the third-party narrative. With that, the book is not just about myself asking and answering questions, logging knowledge, but to also write and try to explain it as a perhaps to a 5-year-old, to somehow make it sounds not like listening to quantum mechanics.

Overall, I seriously hope that this book will help me myself, ultimately, and help anyone along the way, if they stumble upon this. Hopefully you will give me some exposure too (famous?), so, I am counting on you. With that said, artificial intelligence is a hoax. Only us can refute that.

What is in the book

The book is concerned of the main umbrella topic of **artificial intelligences**. Alongside with it are relevant theories and practical implementations that support said theoretical view. So, you would also expect to have *machine learning, mathematical modelling, a lot of mathematics, information theory, complexity analysis*, and more. Specifically, there will also be an entire large chapter on the formalism of the learning theory, in a rigorous sense, so many of the chapters would be there to reinforce this.

As it currently stands, the main top category is the parts. Specifically, there are three main parts. Part I on *theory* - the supporting theory, discussions, results and analysis. Part II on *advanced theory*, being called advanced just because it is my own implementation and theory in accordance. And part III on *implementation* and any necessary details on such - so, it can contain sections on the Python language itself (which is boring) - but also sections concerning deployment or rather typical construction of what has been established in the preceding part. Though there are plans to go for C++ implementation, high-intensity computation is perhaps not in the list for the current time (April 2024).

How much topic is covered?

Well, not so much, but I hope it will be enough to formalize and construct a formal treatment of a potentially modern approach to artificial intelligence theory. Of the latest revision, however (May 18, 2025), the book has been subsequently changed of its status as a comprehensive AI-based book, to a rather general book by its own standard. Though, it does not contrive the structure for literally everything, so mostly we will narrow down to either philosophy, computer science theory, artificial intelligence, mathematics, and physics.

Chapter 1. Introduction to Artificial Intelligence

This book is concerned with mainly the matter of **artificial intelligence**. This is a very broad term, a very misunderstood notion, and a very fruitful endearing venture.

Historically, artificial intelligence begins with the question of rethinking the origin of rationality: what gave rise to the rational process, the logical thoughts? Partaking on this problem in history was mostly in the field of the *philosophy of mind*, which is prominent during the Greek era of philosophical boom, the Renaissance, and the overall European intervention to the topic. For around almost all the existence of humanity, this issue, phrased and approached differently from the view of the human mind and body, philosophy dominated in the search for the origin of intelligence, even though said word is not even clearly defined, and is historically put in a much lower place than other notions or questions regarding the same concept. Only until the dawn of the 20th century, where the new computer has finally grown to be substantially more useful than previously primitive computer has been, that the change shifted from philosophical debate with no ends, to actual reverse engineering and implementation of similar type machines. Furthermore, with the advancements of technology and the Technological Revolution of the 18th century, neurology and other medical analysis field finally took part in discovering one's self - by accessing what is deemed the most important of separating man from the apes. And with that, comes the term *artificial intelligence* in its form.

This chapter serves as the introductory session to the notion of artificial intelligence. By doing this, we would look at the loose picture of the conceptual ideas, the philosophical approach to defining artificial intelligence, questions regarding what is intelligence and artificial as a whole, and a variety of debates surrounding the notion as it is. By doing this, perhaps we can come off to a conclusion on what the book is about, what it was aimed for, and what question would it partake and attempt to solve in the later section. For now, however, we take a role as the historian. Many have tried before, and many have failed before. So what is the history of the term artificial intelligence as a whole, and what comes even before it?

1.1 History of artificial intelligence

If we have somehow settled on the notion and importance of artificial intelligence in the question of finding intelligence, it is then imperative that we find for it a formal treatment and perhaps an entire field of research dedicated to structuring the underlying system of intelligence. The birth of artificial intelligence as a field, with the supporting argument using *computation system*, or computer in simplicity, started in a workshop of 1956, the *Dartmouth workshop*.

> *We propose that a 2-month, 10-man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions*

> *and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.*

And with this, came officially the field of artificial intelligence, in the formal sense.

For this historical account, we would like to take the inquiry mostly from et al. [2006], for a more detail analysis. More so can be found in *Artificial Intelligence - A modern approach* Russell and Norvig [2009]. More authoritative sources and literatures can be found and potentially replace most of the following section. Though I doubt about the brief historical account that it is necessary in doing so. Well, perhaps it is. It is recommended in such section, to review the work *The Quest for Artificial Intelligence (2010)* by Nils J. Nilsson, *Artificial Intelligence, Foundations of Computational Agents* by David L. Poole and Alan K. Machworth.

1.1.1 Roughly, artificial intelligence

The term artificial intelligence was first coined by John McCarthy in 1956, when he held the first academic conference on the subject. Though,argubaly, the issues were discussed, as the journey went far and wide before then, even in the time of Descartes in the 1700s when he conjectured the intelligent machine's existence. More recently, in Vannevar Bush's *As We May Think (July 1945)*, widely recognized to be visionary in some way or another, he proposed a system which amplifies people's own knowledge and understanding, with the aspect of machine assistance. Five years later, Alan Turing wrote a paper on the notion of machines being able to simulate human beings and the ability to do intelligent things, such as playing chess.

However one might want to say about intelligence and qualities of such, it is not of a single doubt that this endeavour has begun very soon, ever since the debate on the philosophical idea of what constitute the mind, the conjectures between knowledges and what constitutes as true knowledge, the various machines such that can say to lift the burden of the mortal human - promptly the industrial revolution, and the wishful thinking of reproducing one's self. Artificial intelligence now solely belongs to the landscape of the most powerful non-human structure we would have know, that is capable of a fraction of the abstract thoughts, being computers.

No one can refute a computer's ability to process logic. After all, we create it based on logic and components that is configured in logical sense. But to many, it is unknown, and misrepresented, too, that it can host the entity that can *think*. The dichotomy of defining the precise definition of think here is important, because there has been some strong opposition as to whether or not this notion is even possible. In the end, the definition of artificial intelligence is perhaps one thing to desire of. Yet we shall then not define it in its entirety, because as later will be discussed, to define artificial intelligence requires separating 'artificial' and 'intelligence'. As of the face of the Earth when those words are written, no one has been ever close to finding out the truth.

1.1.2 The themes of AI

The main advances over the past sixty years have been advances in search algorithms, machine learning algorithms, and integrating statistical analysis in to understanding the world at large. However, most of the breakthroughs in AI aren't noticeable to most people (figured, they are not so well-educated on the subject). Rather than talking machines in movies, AI is used in more subtle ways such as examining purchase histories and influence marketing decisions more than not.

What most people think of as 'true AI' hasn't experienced rapid progress over the decades. A common theme in the field has been to overestimate the difficulty of foundational problems. Significant AI breakthroughs have been promised in 10 years or less, begins earlier than the

time when Marvin Minsky's words of having human-level intelligent machine in 'fewer than 6 years' was cited. In addition, there is a tendency to redefine what intelligent means after machines have mastered an area or problem, which is troublesome to specifically correct what can be called intelligence.

In the field of AI, expectations seems to always outpace the reality. After decades of research, no computer has come close to pass the Turing Test, a figurative framework for defining and measuring intelligence, of a given system. Expert System have grown but have not become as common as human experts; and while we've built software that can beat humans at some games, open-ended games are still far from the mastery of computers. Is the problem simply that we haven't focused enough resources on basic research, as it created two AI winters historically, or is the complexity of AI one that we haven't yet come to grasp yet? And instead, like in the case of proposing playing chess, we focus much more on specialized problems, rather than understanding the notion of understanding? Perhaps, that is the case, that somehow diluted ourselves from seeing the real picture, by masking the false one with expectation to the real alignment.

1.1.3 Precursor - From the ancient time to 18th century

Throughout human history, people have used technology to model themselves. There is evidence of this from ancient China, Egypt, and Greece, bearing witness to the universality of this activity. Each new technology has, in its turn, been exploited to build intelligent agents or models of mind.

Throughout the history, the pursuit of intelligent model and the process of rationalism of the mind has been pursued by directly study it - or the *philosophy of mind*; and as you suspected, it is mostly dominated by philosophical research. In fact, the amount of backlogs and historical researches specifically in this type of philosophy, engulfed quite a lot of paperworks - more so than one should be able to consume in a year's calendar of afternoon tea time, that is. Perhaps it is more apparent to realize that the concept of *duality* - that is, the mind and the body is dual but inseparable part of being human, or its existence thereof, is also from such study. Well, it is best to say that if we stick too much to such philosophical form, this book might have never been written.

In a more diverged and recent years, Hobbes (1588-1679), who has been described by Haugeland as the "Grandfather of AI", espoused the position that thinking was symbolic reasoning, like talking out loud or working out an answer with pen and paper. The idea of symbolic reasoning was further developed by Descartes, Pascal, Spinoza, Leibniz, and others who were pioneers in the European philosophy of mind. The idea of symbolic operations became more concrete with the development of computers, and sooner or later, was tested out of light.

It is also at this point that I would also like to note about the altitude toward the philosophy of mind and its branches of analysis. While it is true, that sticking too much to said philosophy will dilute yourselves of the fundamental problems to solve about intelligence and the latter term sticking the word artificial in, there has always been the priori concepts and illustrations to the problem by itself. That comes of as nothing more but the philosophy of the old, in which even now we utilize. Perhaps more believable, is from the word of Dreyfus:

> *As I studied the RAND papers and memos, I found to my surprise that, far from replacing philosophy, the pioneers in CS had learned a lot, directly and indirectly from the philosophers. They had taken over Hobbes' claim that reasoning was calculating, Descartes' mental representations, Leibniz's idea of a "universal characteristic" - a set of primitives in which all knowledge could be expressed, – Kant's claim that concepts were rules, Frege's formalization of such rules, and Russell's postulation of logical atoms as the building blocks of*

> *reality. In short, without realizing it, AI researchers were hard at work turning rationalist philosophy into a research program.* (Hubert L. Dreyfus)

Though, he also directly argued that this might be more catastrophic than not to the AI researchers.

> *At the same time, I began to suspect that the critical insights formulated in existentialist armchairs, especially Heidegger's and Merleau-Ponty's, were bad news for those working in AI laboratories– that, by combining rationalism, representationalism, conceptualism, formalism, and logical atomism into a research program, AI researchers had condemned their enterprise to reenact a failure.* (Hubert L. Dreyfus, Heideggerian)

It is wise to use or at least consider the concepts and notions of the old. However, it is also imperative that there must be some other meanings to the word old that does not mean classic, especially from the philosophical point of view for a subject that is so diverged from the general guarantee, that correctness cannot be fully gauged. So, at the end, may it be in the quote itself. "Do listen to the old, and carry the journey of the young. But do not be so bold to pursuit the uncertain future, nor too reserved to glorify the lacking knowledge of the past."

1.1.4 The 1940s

In 1943, neurophysiologist (that's right, this job existed at the time) Warren McCulloch and logician Walter Pitts collaborated on a groundbreaking paper titled, "A Logical Calculus of the Ideas Immanent in Nervous Activity", published in the "Bulletin of Mathematical Biophysics." McCulloch and Pitts [1943]. The central aim of their work was to investigate the possibility of representing logical functions through the conception of what is then called the first formulation of an *artificial neuron*, which is fairly common in the neurophysiological field of the time, in which they adopted a model of simplified neuron structure. The details of the paper are pretty much, well, complex to have a look at, because it is aimed toward logical representation, which at the time, they chose to represent them in a fairly convoluted, difficult notational scheme. I mean, seriously, using symbolism of Language II (Carnap, 1938), Russell and Whitehead *Principia* (1927) and else is fairly not so nice for the reader, though arguably it is used for correctness. Though, we can still try decrypting the paper as it is. Actually, no, because it is pretty cumbersome. But we can try seeing what is the consequence of it.

McCulloch and Pitts illustrated, rather convincingly through symbolism and a network of assumptions and framework, that predicate logic can be represented by this framework of neurological units in a net. Using the binary logical approach (apparent in their assumptions for the calculus that 'the activity of the neuron is an *all-or-none* process') and of the static case for the network, logical operations can be executed, albeit in a fairly strict form. The descriptions of said network is also pretty intricate - they concerned of the problem for *learning*, as well as the non-linearity of the signalling process, that is, the nets with circles for excitation, being represented or at least conducted in the theory as recursive functions. But more important than such, is the apparent connection to the notion of the computational machine, or rather, the (Universal) Turing machine (UTM). They argued that their network model could theoretically simulate and computation performed by a Turing machine, which is, in its own, a surprise to be sure of.

Later on, based on this, Donald Hebb (1949) demonstrated a simple updating rule for modifying the connection strength between neurons can lead to new and more complex structure. His rule, now called **Hebbian learning**, remains an influential model to this day, despite the call of modern practitioner for it to be obsolete.

Hebbian learning

Two undergraduate students at Harvard, Marvin Minsky and Dean Edmonds, built the first, perhaps apparent neural network computer in 1950. Later, at Princeton, Minsky studied universal computation in neural network. Historical accounts of the time recited concerning opinion about whether if the subject of study can be called mathematics, but von Neumann reportedly said "If it isn't now, it will be someday".

There are many more historical records correlated to the development of the theory of intelligence, often patched up with artificial in the former. For example, W. Grey Walter (1947) demonstrated the development of the autonomous robots known as "tortoises", named Elmer and Elsie; which, in the same year, Norbert Wiener publishes *Cybernetics*, quite a work at the time. They are settled to showcase what Walter then termed "machina speculatrix" behaviour, encapsulating the essence of a contemplative machine, for what it means. In 1948, an interdisciplinary conference was held at Caltech in Pasadena, California, on the topics of how the nervous system controls behaviour and how the brain might be compared to a computer - the Hixon Symposium of Cerebral Mechanisms in behaviour, which is also considered to be quite substantial in the timeline. In 1945, Turing writes a pioneering, but unpublished paper of "Intelligent Machinery" later on published a paper of the same content in 1950, proposing also the Turing Test. While those are important and pretty much equally interesting, by historical content, I don't think we should be of concerned about them more than we have already given to the main matter.

1.1.5 The 1950s to 1960s

The 1950s is called the period of the *birth of intelligence*. Granted, this is perhaps because of the exact Dartmouth workshop that happened in summer of 1956, which ultimately comes of inviting John McCarthy, Marvin Minsky, Claude Shannon (the one whose works came off as information theory), Nathaniel Rochester, Trenchard More, Arthur Samuel, Ray Solomonoff Oliver Selfridge and more together, to think, and study about how machine can be made, to use language, form abstraction and concepts, or rather, as recited in *The Quest for Artificial Intelligence*, it comes from the dream of extracting oneself to a machine that is capable of simulating such. Though, perhaps the term *computational rationality* would have been equally, if not more reasonable of an expression, but seems like McCarthy resisted using such. If you are wondering yet why there can be the word computers at start, then it is because one of the main assumption in this event is that "...the conjecture that every aspect of learning or any other feature of intelligence can be in principle be so precisely described that a machine can be made to simulate it.", which inherently make it the target for the computational structure.

computational rationality

It is well in my interest (and certainly you maybe) that this step up is one of the fundamental assumption in which will shape what artificial intelligence to become in later years. Or rather, it is more reasonable this way, and is pretty obvious, of the transition from physical models and physical representation, to the more abstract, conceptual, mathematically expressed system in which one does not have to take into account the physical hardware, but only the abstract software itself. By removing physical binding, the theory of artificial intelligence by itself has focused on the modelling of the process inherent within specimens exhibiting intelligent, rather than focusing on replicating the specimen itself. As for once we said the Wright brother did not create the plane because they want to replicate the birds, it is similarly said that we want to create intelligence, artificially, without having to reconstruct the brain. Or, alien-case, some organ that is responsible for such.

In such session, it is recorded that Allen Newell and Herbert Simon have already proposed the notion of the Logic Theorist (LT) Newell and Simon [1956], claimed to "...have invented a computer program capable of thinking non-numerically, and thereby solved the venerable mind-body problem" - again, quite a strong claim. Though, it is also this program that is able

to prove most of the theorems in Chapter 2 of Russell and Whitehead's *Principia Mathematica*, which gave Russell quite the delight when the program had come up with a proof for one theorem that was shorter than the one in the original book. Though, on the flip side, the editors of the *Journal of Symbolic Logic* were less impressed by such notion.

Logic Theorist

While the workshop itself is not particularly successful, in the sense of giving any breakthroughs, it did introduce all the major figures to each other, and, by extension, formally start the journey of finding the dream of artificial intelligence. It's also those figures, that in 20 years time into the period that dominated the entire field. Interestingly, few of them will be directly against each other, and one particular will create the first winter of AI research.

The early year of AI were full of successes - in a limited way, and with it comes expectations. Given the primitive computers and programming tools of the time, even by the time of Turing where the formal system of computer science and encoding was formulated, while the structure and hardware allows the interpretation of computers to be perhaps no more than arithmetical machines - arguably, not quite so even since the dawn of World War II - it is astonishing whenever a computer did anything remotely clever, or perhaps certainly to encode the logical framework that itself was based upon, that many tried to relentlessly refuse to believe. The intellectual establishment, by and large, preferred to believe that "a machine can never do X", and then someone demonstrated that it can. This has been around for a long while, and even by today's time, it is still being done regardless. A consequence of this type of progress is, probably mentioned anyway, the reestablishment of what separate man and machines of year-by-year basis - the Turing test seems to further and further every year and then.

intelligent agent

Because we are using Norvig's text on artificial intelligence, the historical account can historically be separated into four central aspects of analysis. Though naive as it can be, and overvaluing as it might is, all those approaches to AI has been followed throughout the history, each by different people with different methods. This is perhaps one of the most fundamental rules when researching anything - to work on anything, one must have a definition and concept of such, for it to be anywhere ambiguous or descriptive as possible, and for controversial versus widely accepted in its sense.

Generally, Norvig Russell and Norvig [2009] separated in categories, into 4 main approaches.

- **Thinking Humanly**: Despite the description being rather arbitrary, it settled itself in the cognitive science field of study, or rather, incited of the *cognitive modelling paradigm*. The original idea is that if we are going to say that a given program thinks like a human, then what and how human thinks? We then need to get inside the actual workings of human minds, either through introspection, through analytical psychology, and observing the brain in action, which correlates to neuroscience. Once we gain all the sufficiency criteria for forming a theory about the mind, then comes the computer, in which case it is guaranteed to match certain trace of human process in it. Ultimately, this resulted in the interdisciplinary field of *cognitive science*, which brings together computer models from AI, and experimental techniques to construct testable theories of the human mind.

- **Acting Humanly**: The study of, "make(ing) computers do things at which, at the moment, people are better." This is expressed most prominently in the approach which leads to the *Turing Test*, proposed by Alan Turing (1950) to provide a satisfactory operational definition of intelligence. We perhaps want to emphasize on the word being operational definition. In the short term consideration, the test requires computer to possess the following capabilities: natural language processing, knowledge representation, automated reasoning, machine learning (to adapt and detect plus extrapolate patterns). By default, Turing's test deliberately avoided direct physical interaction between the interrogator

(human) and the machine, because, the assumption is that physical realization is unnecessary for intelligence. However, the more comprehensive test called **total Turing Test** also includes computer vision and robotics to the frame. This, in essence, also outlines most of the six disciplines that made up the modern research of artificial intelligence.

- **Thinking Rationally**: This approach, can be fairly cited to be the "laws of thought" general paradigm. Their principles started of the earlier day of Greek philosophy and those that codify "right thinking", that is, "irrefutable reasoning processes". These laws of thoughts were supposed to govern the operation of the mind, and thereby, initiated the field of logic. And by extension, comes to the point of initializing the study of logical AI, or more widely known to be the **Symbolic AI** school of thought. There are two main obstacles to this approach. The first being rather obvious - it does not scale well, for example, to state in formal terms the informal knowledge and actions, and write it in the language of logical notations. Second, logic is concrete, and thereby, not real life - that is, there is a difference between solving a problem "in principle" and solving it in practice. This makes them ineffective in the long run, and furthermore can be more than deceiving to be called good.

- **Acting Rationally**: This comes off as the theory of *intelligent agent*. An **agent** is just something that acts [1]. Computer agents are expected to operate autonomously, depends on the definition, perceive their environment, also depends on the interpretation, persist over a prolonged period of time, adapt to changes, and facilitate the notion of *goals*. A **rational agent** is then one that acts to achieve the supposed criteria, for the outcome of what is expected to be the target. The rational-agent approach has two supposed advantages. First, it is more general than the law of thoughts approach, because correct inference is just one of several possible mechanisms for achieving rationality. In such case, going small, but principle is perhaps more effective than trying to encapsulate everything into a more general notion of rationalization. Second, it is more amenable to scientific development than approaches based on human behaviour or human thoughts.

By now, there must be observed to be two schools of thought present in the study of artificial intelligence. One is concerned of the old-fashioned research of McCulloch and Pitts in 1943, ultimately resulted in the theory of **connectionism**, which aims to facilitate the smallest singleton component of human brain - the *neuron*, and its wider framework called the *neural network*. Hence, artificial intelligence in this sense aimed to create and understand the theory of artificial neuron and neural network. Prominent in this portion of research can be attributed to Winograd and Cowan (1963) for demonstrating how a large number of elements could collectively represent an individual concept, with a corresponding increase in robustness and parallelism (which, undoubtedly, later on became known as the technique of *encoding representation*). As mentioned before, Hebb's learning, Widrow-Hoff learning, presented in the **ADALINE system**, and Rosenblatt's **perceptron** are some of the illustrative groundbreaking formalism existed within this timeframe. Despite such, as we have seen above with the old figures of the Dartmouth workshop, there exists the entirely opposite in spirit as well as nature of the approach, led by Marvin Minsky, called the **symbolic conformation** approach, or more generally known to be the *Symbolic AI* paradigm. Works prominent in symbolic view includes the famous Newell and Simon (1976) **physical symbol system**, Herbet Gelernter's (1959) Geometry Theorem Prover, a bunch of other works conducted by Minsky's student in his lab, for example, the domain of **microworlds** in which James Slagle's Saint program (1963), Tom Evan's ANALOGY program

connectionism

[1] To be fair, this distinction is useful, because aside from agent, there is another model of the internal process that, that separate of thinking, and acting altogether.

(1968) and Daniel Bobrow's STUDENT program (1967) we realized. This approach, which relies on logic and symbolic approach to construct and formalize the thinking machine, led by Minsky, McCarthy, Papert, and a lot of the AI researching community of the time, went against the connectionism view. Minsky and Papert personally attacked the theory of artificial perceptron by their book *Perceptron (Minsky, Papert, 1969)*, with perhaps some coincidental timing and the rise of symbolism as the winner in the late 1960s, to then effectively "shut down" the direction of research toward the theory of connectionism. Though, one might ask, from the first glance, what is so different of the two?

In essence, the detail of connectionist AI postulates that learning of associations from data (with little to no prior knowledge) is crucial for understanding behaviour. This perhaps comes of as not so surprising for the camp of symbolic AI, as their theorists postulate and suggest that the intelligence that underlies human activity can be formalized, or must, as the string of logical conclusions and operations that, quoted, "By extension, one can see the process without questioning why it was there, rather than observe actions whilst blinded of the process that led to it". Recent debate, even to the current time, between the two AI paradigms has been prompted by advances in connectionist AI since the turn of the century that have significant applications, and furthermore, the technological evolution of the age of abundance of data - the internet. So, you might be better off comparing the two to be the debate between *empirical approach*, to the *intrinsic rationalization approach*, respectively. In the end, however, we need both. Goel [2022]

1.1.6 The first AI winter (1966-1973)

History suggested, or rather, indicated the overwhelmingly positive altitude, most of the time, from AI researchers about their predictions of the upcoming progresses. Herbert Simon in 1957 is often quoted:

> *It is not my aim to surprise or shock you — but the simplest way I can summarize is to say that there are now in the world machines that think, that learn and that create. Moreover, their ability to do these things is going to increase rapidly until—in a visible future — the range of problems they can handle will be coextensive with the range to which the human mind has been applied* (Herbert Simon, 1957)

Terms such as visible future can be interpreted in various ways, but Simon also made more concrete predictions, by the notion of 10 years for certain capabilities. These predictions came true (or approximately true), however, within 40 years rather than 10. Simon's overconfidence was due to the promising performance of early AI systems on simple examples. In almost all cases, however, these early systems turned out to fail miserably when tried out on wider selections of problems and on more difficult problems. That is to say, they particularly did not scale well.

In the 1970s, the entire AI field of study entered a period of time described as the "AI Winter". According to the AI Newsletter, the phrase was borrowed from the term "Nuclear winter", which originated from the Cold War perspective of nuclear war. Indeed, during the AI winter, commercial and scientific activities in AI declined dramatically. Research fund was ruled out, supports were cancelled, and overall the field grinds to a halt. Arguably, AI is still recovering from the winter that lasted nearly two decades, with another small subsequent winter period, the second AI Winter, happened just before the onset of 1990s.

The major cause of the AI winter can be attributed to the story of the government's decision to pull back on AI research, of missed promises and overreaching predictions. This is perhaps more apparent in the two infamous reports, specifically the Automatic Language Processing

Advisory Committee (ALPAC) report by U.S. Government in 1966, and the Lighthill report for the British government in the 1973.

I am not going to get into the details of such story, but they have the same shadow of impact - both did not deliver what was expected. Whilst the dream and predictions is absurd, they failed to deliver what they were tasked of, ultimately because of several factors. Observationally, it comes from mostly the problem with scalability, as previously argued, which is bad of those early model. More than such is the fallacy that most early program knew nothing of their subject matter, and only operates on clever, often simple syntactic manipulation. Again, we specify on the word simple. And perhaps more importantly, is the question of the illusion of unlimited computational power that AI researchers often made during the time.

The last difficulty that leads to this downfall is then perhaps of some fundamental limitations on the basic structures being used to generate intelligent behaviour. By the language of representation theory, Marvin Minsky and Papert's book Perceptron has demonstrated that perceptron model is pretty much poor in representing concepts or any structure of the underlying representation spaces. Although it did not consider the more complex system of multilayer perceptron, research funding for neural network research soon dwindled to almost nothing. Sometimes, people still attributed the backward crisis of AI research to such event, though it is rather not so convincing of an argument.

On a flip side, it also brings about to us an interesting ordeal of the hype cycle of AI research (Menzies, 2003)

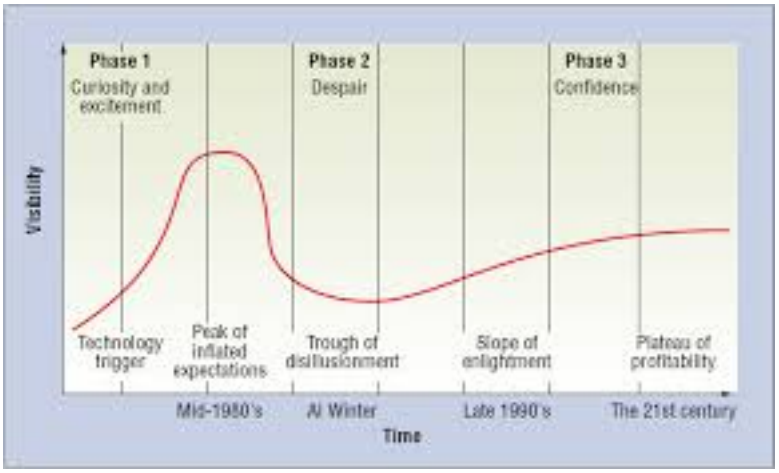


Figure 1.1: Well, look at that. Can you guess if we are about to face another one soon enough?

1.1.7 Knowledge-based systems - expert system (1969-1979)

During the first decade of AI research, the camp of either symbolic or connectionist AI has been to construct a general-purpose construct, in which deduction and reasoning can be made on such system to derive new situations or new knowledge. Such approaches have been called the *weak method*, Russell and Norvig [2009], because although it is general in some partial way of the logical facts, it cannot scale well to large or difficult problem instances. This comes off as the scaling issue in both the complexity of the task, and the expression of the system by

itself, in which reasoning might be applied in principle [2]. furthermore, philosophically, it ran into the problem that is analogous of the Chinese Room Argument - those reasoning systems are perhaps only symbolic manipulation, and is not actually intelligent. Naturally (speaking), the alternative to weak methods is to use more powerful, specialized knowledge that allows for larger reasoning steps and can more easily handle typically occurring cases in narrow area respectively. Think about it as the reduction of generality. This resulted in the domain called **expert system**, for which, if you followed from above, connectionist has already been phased out (apparently, at this point there are no funding so they have no jobs...), so mostly it is a mostly pure symbolic approach, just with added flavours.

expert system

Informally, expert system are computer programs aiming to model human expertise in one or more specific knowledge areas. They usually consist of three basic components: a *knowledge database* with facts and rules representing human knowledge and experience, an *inference engine* processing consultation and determining how inferences are being made, and an input/output interface for interaction with the user or usually, just the problem case itself.

knowledge database

inference engine

According to Metaxiotis and Samouilidis [2000], expert system can be characterized by:

- Using symbolic logic rather than numerical calculation (so no connectionism, as we said).
- The processing is *data-driven*.
- A knowledge database containing explicit contents of certain area of knowledge.
- The ability to interpret its conclusion in the way that is understandable to the user (the *explainability criteria*)

While most of the characterization can be said to be weirdly strict, except for the second one, the last characteristic is perhaps of very large interest within the modern framework of artificial intelligence, the requirement of **Explainable AI**. This sentiment (or requirement) is particularly stronger in the field of medical science and overall medical field, where explicit conclusion must be specified of its content and deduction - for example, why is this diagnosis true, and what factor is taken into account? This is perhaps reasonable - nobody wants to be that one guy anyway.

Historically, expert systems emerged in the early 1950s when the Rand-Carnegie team developed the general problem solver to deal with theorems proof, geometric problems and chess playing. About the same time, LISP, the later dominant programming language in AI and expert systems, was invented by John McCarthy in MIT.

Perhaps one of the main example of expert system can be traced back to the `DENDRAL` program (Buchanan et al., 1969), developed at Stanford by Ed Feigenbaum, Bruce Buchanan, and Joshua Lederberg. They teamed up to solve the problem pretty much very specific, and not so general as we have seen: inferring molecular structure from the information provided by a mass spectrometer. The input to the program consists of the elementary formula of the molecule, and the mass spectrum giving the masses of various fragments of the molecule generated when

[2] This can be further clarified. The aspect of complexity in the task is reasonable - with increasingly large system, comes increasingly many objects in action, many rules or patterns, many properties and aspect to consider, which makes it harder to design a working well system in such complex term using symbolic approach or phenomenologically, the search mechanism of connectionist. And second of all, formalizing them into symbolic logic, while the logical predicates and laws are the same, the entire problem setting and the system that use such logical system is entirely inoperable with scales. For the connectionist camp, at least up to said point, there are no evidences or ways to effectively search for the correct representation or dynamic system using their model of the computing neural unit.

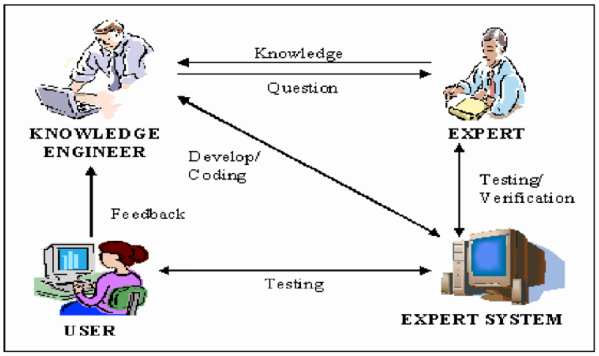


Figure 1.2: The diagram of a typical expert system. Here, human involvement is fairly representative in the figure, and is illustratively indicating the overwhelming reliance on human touches.

it is bombarded by and electron beam. The naive program generated all possible structures consistent with the formula, and then predicted what mass spectrum would be observed for each, comparing this with the actual spectrum.

This, by itself is pretty much intractable even for moderate-sized molecule, even with the modern standard computation (perhaps). So, instead, they opted for shortcuts - an informed shortcut that is - by utilizing *human specialized input*, or expert input. They used analytical chemist to know what they were looking for in the problem, and apply it to the rule system of the program. This approach, while seems to be pretty obvious, turned out to be fairly powerful and successful of the time. `DENDRAL` program was rendered successful, most is can be said is because

"*All the relevant theoretical knowledge to solve these problems has been mapped over from its general form in the [spectrum prediction component] ("first principles") to efficient special forms ("cookbook recipes").*" (Feigenbaum et al., 1971)

However, it is to be said that this type of approach is, arguably not artificial intelligence anymore, in the true sense of it. Rather, it is similar to a conditional, variable logic system in which the dynamics are specified by junctions in a box, where rules dictate conditions in which junction should a ball rolls up to certain path, and every structure has to be mapped, and configured by the designer himselfs. In a modern word, it is eerily similar to a proof machine, just with different goal in mind. However, it is sometime enough for progress. At least, though, we get to know how an isolated "dynamic system" might work, which will benefits us later on.

Eventually, more follows suite. There is then XCON, a computer hardware configuration system, MYCIN, a medical diagnosis system with about 450 rules that make it performs sometimes even better than junior doctors, and more. MYCIN, specially, contain the notion of certainty factor , which is used to reflect uncertainty in a diagnosis decision. Quite similar to fuzzy logic, I wonder anyone have tried using fuzzy logic style into it? The more epitome of natural language processing, under the lens of expert system, is `PROLOG`, a computational linguistic programming language to aid in the construction of this kind of new expert construct. Eventually, by some hilarious means, expert system were considered revolutionary to the point of threatening humanity by itself and its virtue of existence. Or rather, I would say for now, the onset of another winter of AI, toward which expectation reaches even further than before. How bad can it be of successes to be on the receiving end of those expectations:

certainty factor

> The success of these systems stimulated a near-magical fascination with smart applications. Expert systems were largely deemed as a competitive tool to sustain

> technological advantages by the industry. By the end of 1980s, over half of the Fortune 500 companies were involved in either developing or maintaining of expert systems. The usage of expert systems grew at a rate of 30% a year. Companies like DEC, TI, IBM, Xerox and HP, and universities such as MIT, Stanford, Carnegie-Mellon, Rutgers and others have all taken part in pursuing expert system technology and developing practical applications. Nowadays, expert system has expanded into many sectors of our society and can be found in a broad spectrum of arenas such as health care, chemical analysis, credit authorization, financial management, corporate planning, oil and mineral prospecting, genetic engineering, automobile design and manufacture and air-traffic control.

Seems to me, quite a lot. While I was bashing expert system back there, it is certain of its use, specifically with systems and generic scenario where not a lot of things change, and more so on the front of static system, unlike natural language where ambiguity is almost always present. It is at this point, that since then, AI becomes an industry on its own, and is not just the umbrella terms for the weird little computational model here and there. And, you guess it, the second winter hits.

The problems with expert systems

On the surface, it is good. However, if it is the end of the road, we wouldn't be here and talking about something else than expert system. There are then indeed, problems with how the expert system turned out to be - many unresolved technological issues and performance limitation that severely affect the development and implementation of expert systems. This prompted the *second AI winter*, arguably, which is even more damaging than the first one because of the rapid industrialization style, and the failed promises of various companies. The keys issues facing expert ssytem by then, are software, knowledge acquisition, handling uncertainty, and validation. Which, most of them can be grouped into the word *scalability*. And, added to the list which I stole from somewhere, there is also the *structural issues* with expert system, much like so in general for today's approach of ML.

First, is the issue called **software standards and interoperability**. It seems like at the time, a lot of people want to do things their way, and because of that, there are no general standards in expert system software, development methodology, protocols or infrastructure. This makes the expert system section totally specialized in not just goals, but also designs, implementation, architectures, and all, even though they are based of some languages like `PROLOG` or the more famous `Lisp`. Though, arguably, because of its fairly prominent utilization, there has been pushes for standardizing it like IEEE Standard, so, this line of argument might dissapear in the air later on.

Second, is the fact that knowledge acquisition in the context of expert system is very complex and dependent on human experts makes it both shallow and unscalable. Their facts are human-dependent structure, static and was articulated into a set of which does not reflect tha actual knowledge itself. This renders the expert system to be similar to a set of rules with added semantic, and with logical rules for its inner composition, rather than artificial intelligence. Furthermore, it also does not learn from experience, if one wants it to have the action of learning, since it only goes on to follow the set of rules and how they operate. Best of all, those knowledges are too shallow for any reasoning to be maded of extended semantic, as the interpretation is often hard to configure, and many times, because constructions and rules are in nature subjective of human experts, incompatibility is also an issue. This also covers the third point about the **handling of uncertain situations** or out-of-bound cases.

Many tried to fix it. For once, there exists the scheme of *Case-based reasoning* (CBR) theory

that focuses on solving new problems based on similar past problem solutions, which seems to be able to eliminate the complex task of maintaining rules and facts through the use of adaptive acquisition of problem-solving techniques. However, again, this is perhaps shallow as it can get, and added to the semantic, there are also biases, and the uncertainty of past experiences being compatible with new situation, by virtue of different situations and scenario. CYC project by Cycorp Inc. aims to assemble also, the processing of commonsense knowledge in which somewhat allegedly, CBR cannot handle. yet I think this is not so much of a fruitful pursuit, because even the notion of *common sense* is perhaps distinctively believed, even by a majorative mean. Such is also to say the surprisingly complex and unscalability that one would face if they try to encode common sense in such way.

The worse reason on the list, though, is the lack of **validation on system operations**. The quality of expert systems is often measured by comparing the results to those derived from human experts. However, there are no clear specifications in validation techniques. How to adequately evaluate an expert system remains an open question, although attempts have been made to utilize test cases develop by again, expert, to do the job. However, many believe (including me) that this will turn out to be worse.

1.1.8 Neural network, the return (1986)

What makes a system a good exhibition of artificial intelligence? Many said of different qualities, like in the way of symbolic AI, for the way and realization of universal logic calculi, or expert system and the tiresome knowledge banks acquisitions. But for many, one representative quality of being intelligent, is *the learning ability*, or the representative sufficiency of learning action. Interestingly, logic can also be learnt, but in a rather strict sense and not so confounding on its own. Connectionism, with their neural networks, however, has a different view.

IN the failure of expert system, and to the larger extend, symbolic computation in utilizing and mitigating their disadvantages in designs of the time, many turned to the hidden contender of the old neural network approach, which sought conformality and uniform of expression, in a dynamic way rather than the strict, universally required criteria of symbolic formation, for example. It might seem obvious that at some level, human manipulates symbols, of which by Terrence Deacon's book *The Symbolic Species* (1997) suggests that this is the defining characteristic of humans. Though, most ardent connectionists questioned whether symbol manipulation had any real explanatory role in detailed models of cognition. This question remains unanswered, but the current view is that connectionist and symbolic approaches are complementary, not competing. As occurred with the separation of AI and cognitive science, modern neural network research has bifurcated into two fields, one concerned with creating effective networks, and the other concerned with careful modelling of empirical properties and actual neurons and ensembles of neurons. Though, that is not to say that McCulloch and Pitts's neuron, and other at this point, *classical* neural network structures are still considered of principle to be the same. Many principles and 'axioms' have been removed, and many more have been added underway, except the only reasonable target - that the ability to reason or to be intelligent is somehow, dependent on those neurons.

1.1.9 Adoption of scientific method (1987)

By the virtue of Russell and Norvig [2009], it is said that of such date, there are deliberate efforts to centre or to ground the wing of those who fly high in the sky, respective to the way AI isolated itself from the large portion of the scientific world. By then, AI was perceived to be the earlier rebellion against the limitation of existing fields, and then was reincorporated into the frame by such forces. Furthermore, AI has also adopted the scientific method. To be accepted, hypotheses must be subjected to rigorous empirical experiments, and the results must

be analysed statistically for their importance. Though, with that said, there are some subtle point to argue and requisitioned.

Every science start with conceptual understanding and conceptual structures. The main goal of the earlier prospects with artificial intelligence is to understand the manifestation and formation of intelligent to be created. Hence, in such times, there are many theories that subjected itself to the category of conceptual framework. In fact, if we are looking at the current trend of defining and conceptualizing *Artificial General Intelligent* (AGI), it shows the same sign and way. This is not a problem, at all, that some might see the above shift and criticize the old progresses as costing us a fair amount of time and being rather too adventurous — to the point of isolationism. Though arguable agreeable, it is rather far from the truth.

There is also the problem with other fields of scientific research, for example, in their mention, information theory, stochastic modelling, optimization theory, and control theory. Partially, it can be of the reason that we ourselves, as human, do not believe or want to justify a computer full of simply binary bits and machines can be considered intelligent of quality similar to human, it is also believed that the essence of artificial intelligence cannot be fully captured using simply just, mathematics and rigours. Such is true of the time, when ideas are not tested out yet and many innovations and conceptual proposals are there, still waiting for someone to rule it out because of its relative insufficiency in both practical purpose, and partial correctness (because nobody knows if it is actually correct, or not). Only by the time that the infinite possibility of an entire new, controversial, and expansive field is narrowed down to the possibility within grasp, and the problems that was said to be present - for example, the reproducibility problem, was identified in the non-formal way of conducting researches. [3].

Then, there is also the fact that those fields, throughout the development of artificial intelligence in our timeline, is severely lacking the tools of which artificial intelligence requires of the time. Indeed, science does not revolve around AI, and many fields continue to progress in the timeframe of 50 years, since the 1940s to the 1990s, as we recall from the year 1987. That is an entire 50 years of development. Only when there are tools or some effective theories from other fields that see acceptance in adoption, by then, that artificial intelligence would want to take a turn toward those theories for obvious reason - to utilize them with the tools that they provide, that we can use - which obviously does not exist before. For example, try in the hand of information theory in the onset of artificial intelligence in the 1940s. Can you find the tools required of information theory that artificial intelligence conceptual of the time would be able to utilize, least of it that the most axiomatic principle of AI at the time of the Dartmouth workshop is that "intelligence can be observed and constructed computationally"? Probably not, at least from my view. Abandoning this fact, that sciences did, indeed, lack what is required or at least desired of AI research, is particularly frustrating, like as if other fields of 'mathematically rigorous' - which they are not, have all the tools necessary from the beginning.

Nevertheless, this marks an important point of the development of artificial intelligence research, after which we know what is the relatively usable and effective approach, which allows the theoretical analysis and actual formalization to be taken into place. Again, one might even suggest this is because of the scale ergonomic, and I would not bat an eye. This, in turn, tells much of the maturity of the field as a whole, and the focus of actual construction and application purposes. Nowadays, there are **hidden Markov models** (HMMs) for speech recognitions

[3] We use the word non-formal instead of non-mathematical, because a lot of theories, while being formal, aren't necessarily mathematically rigorous. Indeed, mathematical rigours are not required, and is often scuffed off the table in a lot of theories and fields. Even in physics, there are also the distinctively air about mathematics - if everything is mathematical already, then it is not physics - and it is true, that a lot of physics cannot be mathematically defined in a rigorous way, for it to not be labelled as another sub-field of mathematics. Plus, the fact that many sees language as logical, more so than mathematics, kind of proves the point.

modelling, and the principle of stochastic evolution stands as an effective analytical tool for uncertain process. Machine translation now have information theory into its aid, neural network coming back with their rigorous analysis, both in statistic and by mathematical analysis. And finally, recognizing the probabilistic possibility of neural network and a few learning theory of which utilize classical non-network models, probability theory were utilized heavily into the field, which even birthed the **Bayesian network** formalism.

1.1.10 Intelligent agents (1995)

Perhaps encouraged by the progress in solving the subproblems of AI - as we have seen by not tackle the hard general problem of AI first during the 1960s, researchers have also started to look at the "whole agent" problem again, though this time, with experiences of past failures. The work of Allen Newell, John Laird, and Paul Rosenbloom on SOAR (Newell, 1900, Laird et al., 1987) is the best-known example of a complete agent architecture. In a modern date, ChatGPT (Open AI, 2018) is one of the attempt to create a completely functional, albeit limited, semi-universal agent.

1.1.11 The revolution of Internet (2001)

1.2 Defining intelligence

Inside the term *artificial intelligence*, there is the word *intelligence*. A normal person will tell you that they are intelligent. But it just so happens that this notion of qualification is harder to define when one participates in the active action of finding it. So, what is it? This is the question we should take in.

Perhaps this is one of the most important question when encountering, or researching artificial intelligence. Intelligence is difficult to define, and is natural as a human concept, yet ambiguous as it can be. If one wants to tread on the road to artificial intelligence, for whatever it is, they must have a makeshift, for now, definition of artificial intelligence, and intelligence in mind. By then, for me myself, I chose to restrain myself from defining the exactness of intelligence, by rather justifying the act of working on artificial intelligence as actually reverse engineering intelligence by itself. The detail arguments? It will be in there.

1.2.1 Historical accounts

We start this section with a series of historical accounts. Shane, Marcus (2007)'s paper *A collection of intelligence* and Masahiro's (2023) *Descartes and Artificial Intelligence*[4] might be a great place to start this, since they provide a non-trivial amount of definitions and attempts already there, which serve us more as exhibition for observant in this section and beginning.

We list of the interesting definitions that has been given to artificial intelligence, and the question of what is intelligence. Interestingly, those definition are often phenomenological and "soft" rather than not, since a concrete definition cannot be achieved, or rather, be formalized within the right of the established theory.

> **R. J. Sternberg** ...I prefer to refer to it as 'successful intelligence.' And the reason is that the emphasis is on the use of your intelligence to achieve success in your life. So I define it as your skill in achieving whatever it is you want to attain in your life within your sociocultural context — meaning that people have different goals for themselves, and for some it's to get very good grades in school and to do well on tests, and for others it might be to become a very good basketball player or actress or musician.

[4] See more in Journal of Philosophy of Life Vol.13, No.1 (January 2023):1-4

D. K. Simonton ...certain set of cognitive capacities that enable an individual to adapt and thrive in any given environment they find themselves in, and those cognitive capacities include things like memory and retrieval, and problem-solving and so forth. There's a cluster of cognitive abilities that lead to successful adaptation to a wide range of environments.

H. Nakashima Intelligence is the ability to process information properly in a complex environment. The criteria of properness are not predefined and hence not available beforehand. They are acquired as a result of the information processing.

P. Voss ...the essential, domain-independent skills necessary for acquiring a wide range of domain-specific knowledge - the ability to learn anything. Achieving this with 'artificial general intelligence' (AGI) requires a highly adaptive, general-purpose system that can autonomously acquire an extremely wide range of specific knowledge and skills and can improve its own cognitive ability through self-directed learning.

Jensen, Huarte, Dearborn ...the ability to learn, the ability to understand, either principles, truths, facts, or common sense, to profit from experiences; the ability to comprehend, or the capacity to reason.

A. Anastasi Intelligent is functionally of multiple components combined.

J. Peterson ...a bunch of stimuli.

Humphreys ...the resultant of the process of acquiring, storing in memory, retrieving, combining, comparing, and using in new contexts information and conceptual skills.

Out of those definitions, we see two patterns, or the point of view in defining the operational definition of intelligence - one is the *top-down*, empirical approach, and one is again, the *ground-up* structural approach.

Two ways of definition

Out of those definitions, there are two kinds of defining the notion of intelligence, we call it the **top-down** and the **ground-up** approach. The top-down line of thought demonstrate, most of the time conjectures, the existence of intelligence as a whole, without finding the actual shell that contains it. If intelligence is *general*, then their implementation follows, but to a sufficient degree, it can be achieved everywhere. It guarantees, partially, of certain school of thoughts the generalizability of intelligence as the ground base to re-create such, which is characterized, often, by current machine learning discipline. This approach beside from guarantees such existence, also has the capability to 'test' a subject of being, 'intelligent'.

This is done by setting up agenda and criteria, of which the current theory serves as more of a black box for the actual 'machine' that contain it, but enough exhibitions fitting those criteria for intelligent. Fortunately, this also sets certain criteria for artificial intelligence to be specified so in the name. The Godel's argument is one of such example in this line of thoughts, theorized by J. R. Lucas (1961), Penrose (1994, 1989), and Benacerraf (1967), similar to the Chinese Room Argument (Searl, 1980). Coincidentally, the notion of **computationalism** is also formed out of this approach.

The *ground-up* approach of defining intelligence is simply the polar opposite: Instead of defining intelligence by criteria, they create machines or models that have intelligence seems to be the emergence behaviour from those model. That is to say, they define intelligence by not defining it but constructing it. Though, this type of approach still requires the intuitive feeling

of intelligence to figure out or identify such emerging signs of a growing construct, but it is more or less general, as it does not depend on certain opinion, or fixed high-level criteria to classify it.

Nevertheless, if one wants to work on notions or related measure to intelligence, or in general, any given topic, it is imperative to form any form of knowledge, barely so as the *definiendum*.

Problems with defining intelligence

A traditional viewpoint is that definition is a curse. Some says definition is only the typographical shortcut - mainly the opinion in maths - of convenience, certain idea consider definition to be the encapsulation of features. Most of the time, definition is even separated into different forms: either dictionary, stipulative, descriptive, explicative, or ostensive [5], of which none are mutually exclusive, but works rather by the mean of overlapping each other but retains certain uniqueness. In one such way, what we desire from the definition of intelligence is unclear.

A provisional definition can be obtained, however, at least from the subjective point of view, to at least defining intelligence by generalizing what others have done in the progress of realizing this:

Intelligence

Definition 1.2.1 (Intelligence). *Philosophically, intelligence of a subject S is the representation of that subject's probability to* ***think*** *of its internal structures, and* ***act*** *or not, based of such premise of thoughts.*

This definition is the type of descriptive definition. The word *think* requires a lot of times to work out, since it is not apparent how to think. Arguments about the state of think and its actual representative are various, and inconclusive. Furthermore in such, there is also the issue with the correlation between intelligence and consciousness, which is troublesome, and yet again controversial enough to guarantee itself another entire book dedicated still.

Perspectively, we would like the *identification of intelligence*. Note that, in our view, intelligence is an umbrella term, for now.

Intelligent

Definition 1.2.2 (Intelligent). *A subject S is considered intelligent if, for certain proxy of observations, fits the set of representation that correlates it to the notion of intelligence.*

Intelligence, most of the time, is rather dubious to look at. Generally, and incidentally, it is because of the concept itself is not so clear and rather ambiguous. What exactly rings in

[5] Clarification for each of those terms, and a general idea of such concept. A definition is made up of two parts: the definiendum and the definiens. The definiendum is the term that is to be defined, whereas the definiens is the group of words or concepts used in the definition that is supposed to have the same meaning as the definiendum.

A *dictionary definition* reports the existing meaning of a term, in one sense of this phrase, and aims to provide definitions that contain sufficient information to impart an 'understanding' of the term, whether this understanding involve operations and overall intuitions of such meaning.

Stipulative definitions imparts a meaning to the defined term, but involve no commitment that the assigned meaning agrees with prior uses (if any) of the term; basically, assigning new meanings to a term, whether the term has already got a meaning or not. *Descriptive definitions* on the other hand is similar to stipulative, but also aim to be adequate of existing usage.

Explicative definition offers neither descriptively nor stipulatively, but as explication, aim to respect some central uses of a term, but stipulative on others. This, nevertheless, can be understood as having both, in a sense. The central aspect, however, relies on the fact that explicative definition views identical function with different definiendum the same, as long as the definiens retains its core functions, and truth, for some case the consequences might differ from such, as long as the essential remains.

Finally, *Ostensive definitions* while depend on context and experience, its essential function is to enrich certain language, by the mean of limiting the windows of perception to certain object which entails shared attributes - so for example, 'let x be a number in $\mathbb{Z}$' while we know that there are many other integers. In fact, this way of defining is considered, for some thinkers to be the source of all primitive concepts (Russell, Whiteley). We would be using these notions of definitions in the future.

your brain when you heard of *intelligence*? The subject of which concerns the typical human brain, is the philosophy of mind, or rather, the famous one being **dualism**, or Cartesian dualism. And, such philosophy will always be philosophy - without implementation, and without actual realization of the underlying subject.

1.2.2 The Descartes experiments

Instead of defining intelligence by referring to oneself, or by exploring such notion by assuming properties of human, Descartes took a different approach - what is required to project such notion to another, perhaps inanimate being, to be equal to man? Arguably, this experiment also falls into the category of the thought experiment of artificial intelligence; however, it is more befitted being mentioned and argued in the section more on the intelligence side of the word. Under the Descartes interpretation, let's have a look at the conditional, and rather empirical approach to the question. It seems, and as it is, more closely to the **black box argument**, as I called it as such, to provide classification and justification of which relies solely on empirical test, but not knowing the true essence of what inside. Descartes [1950]

Descartes's argument start with the justification of the apparent reactive behaviours observed by human themselves, who at the time, was largely considered to be the only species capable of advanced rational thoughts and processes.

> *(I)f someone touched it (= the machine) in a particular place, it would ask what one wishes to say to it, or if it were touched somewhere else, it would cry out that it was being hurt, and so on. But it could not arrange words in different ways to reply to the meaning of everything that is said in its presence, as even the most unintelligent human beings can do. [Descartes, 1700]*

Here, Descartes argues that in order for human-like robots to acquire intelligence, they have to gain a universal capability to accurately react to any unknown situation that may happen in the environment. However, what machines can do is no more than to respond to a single situation one-on-one via a specific organ, hence, they cannot be considered to have a universal capability that even unintelligent human beings can enjoy.

Continuing, Descartes argues that those machines do not act on their knowledge, but the disposition of organs.

> *For whereas reason is a universal instrument that can be used in all kinds of situations, these organs need a specific disposition for every particular action. It follows that it is morally impossible for a machine to have enough different dispositions to make it act in every human situation in the same way as our reason makes us act.*

The argument is quite clear. Human is universal of the environment. Whereas machine is no more than a combination of abilities that are applicable only to certain situation that the creator could imagine when they built the automated machine.

From this, we can design a very specific test, under the assumption taken, and the principle argued, on determining between human and machine. Notice that the empirical approach of such does not specify **intelligence**; rather, it attempted to question the difference between the natural highest epitome of intelligence (human) to the analogous artificial construct (machine).

Suppose $A(i, o)$ is a machine capable of 'intelligent' act, in a sense, of which we would call it as 'action with reasons'. Then, under the argument, there exists a frame $M(A)$ of which encapsulate all capabilities of this machine. Unspecified of the correct notion of this frame, the

empirical test relies on the fact that A as a machine, is provided with in/out resources, of such that it is guaranteed of information received, and actions being capable. We state that $M(A)$ is finite, as the environment E of which A would be received is limitless. Or rather, suppose that $\{M_i(A)\}$ is the set of all given actions and reactions of which A can output, provided that it receives the information. Then, there exists an element $M_k(A)$ such that

$$\exists k \text{ s.c } M_k(A) \notin \{M_i(A)\} \forall A$$

meaning that there will always be certain parameters that is out of control for the given machine to effectively respond to such. Under such constraint, the machine would fall out of control, and inevitably fails to respond, hence render the machine useless under such circumstance. This, seems to be different, as far as Descartes saw, of the human behaviour.

Of all, Descartes *rejects* the possibility that there exists an artificial, man-made construct or machine that can be intelligent, to the degree of which human presents. By then, it lifts the omniscience of human intelligence to a higher ground, much higher than previously perceived to be of the mechanical machines.

1.2.3 The economic of definiendum

It is clear from the above discussion that intelligence is very hard to define. Not only it is hidden in the most complex structure of a biological organism - in fact, biologically, the brain, or its smaller form in living organism, is typically not *required until further evolution*.

In one way or another, we want to *define by creation*, if to say more than reverse engineering ourselves. By this, we assume that the existence of intelligence is composed of meaningful parts, of which comes up to certain levels, become the current intelligence that we enjoy.

Stands on such ground, it is sufficed to say, or rather, to observe and clarify that artificial intelligence does not curtail of an infeasible attempt to try constructing subject of intelligent without knowing intelligence. Rather, *artificial intelligence* is the *process* of finding intelligence, others than the philosophical ground of defining such. Philosophy works, but our desires differentiate from that. Certainly, one of the most important field of research along such line has been neuroscience, neurobiology, and the field of neurological research, aside from theoretical modelling of conceptual models. Understanding and observing the most advanced mechanism of intelligence as of date, is one way to study and reverse engineering intelligence. And with that, again, goes artificial intelligence.

1.3 Formalizing artificial intelligence - an attempt

What do we think of AI? What can be AI, what is perceived to be one, and how do we create them? The latter half of the word is hard, but, not so much for the first one, or *artificial*. We can have a sense of what being called artificial by examining it.

artificial

Definition 1.3.1 (Artificial). *We say an object is artificial only if it is not natural, or rather, it is* ***intentionally created*** *by meaning intentions, and not the general evolution of states, or the natural transformation of biology.*

Most of human's construct hence can be called as artificial. The definition above is stated so, is to generalize to certain notion of intended creation, for in such event if we ever discover alien lifeform, then their "AI" would not be argued against such to be not artificial because of the word's basic meaning related to human, and hence can be considered a stipulative definition in its stead.

Hypothesis 1.3.1. *The subject of intelligence, in one way called the working state, is different from the underlying mechanisms that host it, but dependent two-way.*

Hypothesis 1.3.2. *Intelligence on* ***machine***[a] *has itself the basic principles of* ***recursive representation*** *(i.e. with sufficient complexity, one system can host another of the same type in itself), and* ***relative abstraction*** *(i.e. the entire machine is supported by individual layers of components classified by certain metric, and the top layer are those that dictates such behavioural outcomes that is the subject of classification to either intelligent, or not).*

[a] We are using the **dictionary definition** for the machine here, as an apparatus using mechanical power and having several parts, each with a definite function and together performing a particular task.

In one way or another, an artificial intelligent subject can the theorized to consist of two things - the host, and the process. We call it as such for clarification, such is not to use the notion of *mind* and *body* for even now we have no conclusive decision on the matter named as said. The duality of the subject starts here, however, as we soon see.

Nevertheless, it is evident that while we abbreviated intelligence into only the container of necessitated traits considered so, the notion of artificial intelligence brings about a lot more than such. The previous principle assumption that we make indirectly agrees with the notion that the process and the host are dependent, yet distinct. This is, historically, against the Descartes' view on the metaphysical reality of the duality of mind and body. Yet, it seems not so contrived to think of it as the case, than not.

For the definition of artificial intelligence, or the construct that supports it, to make sense, we need to evaluate again, from what we have seen, what is even the term. As noted by definition on the notion of *artificial* in the preceding section, being *artificial* mostly comes of from the consideration of evolutionary processes - of which the interaction in the physical worlds, the biological worlds, and overall, anecdotally, of anything that is non-human of its (human) own capability to morph objects into an intended state - this is what normally resided to. Then, artificial intelligence refers to a set of observations, observable qualities deemed sufficiently of all intents and purposes intelligent, by any given constructs that is created artificially so.

This breaks down to the two conceptual ways to talk about artificial intelligence.

Conjecture 1.3.1 (Artificial intelligence). *Artificial intelligence is the classification for any such object of constructs sufficiently reflects those qualities that fit the standard of intelligence, of which also created* ***artificially*** *of intent and purposes (as reflected in definition of artificial).*

The second (conjecture) definition goes into a bit more detail on such term. We link artificial construct (a more generalized name) toward machines, for whatever the machine can be.

Conjecture 1.3.2. *Artificial intelligence refers to (a)****construct(s)*** *- of which consists of the* ***machine*** *and its* ***process****, for such that the machine supports the process to reflects the observed results quantified in one way or another, to be interpreted as intelligence by the construct that is standard for those terms. Those constructs however, are absent, or not, by choice, of the existential facility - or of either a rigid static facility of such - and hence artificially made.*

Arguably, the second definition is far more interesting than the first one. However, the claims, of such, can be hypothesized as perhaps not so ideal generalization:

Hypothesis 1.3.3. *The term artificial intelligence, generically, refers to the comparison between two actual constructs. If the current human - or us - are the ones evaluating certain constructs as intelligent, then it is*

equivalent to generalize human into a construct on its own, of sufficient analysis such that the comparison can be conducted.

This conjecture, in one way or another was called such is only because of its uncertainty by its own creator (myself) that it is true. Its purpose, is to justify and describe the thought processes thereof that leads to the discussion of *intelligence* as a concept, and *artificial intelligence* as a construct, whether the directed definition was raised in the correct orientation of the objected in focus or not. In such definition, however, it leaves much to desire - specifically, and especially of the relative tendency between two constructs - two of which for A, B to be them, the figured measure $\mathcal{M}(A)$ was applied on B such that $\mathcal{M}(A) > \mathcal{M}(B)$.

By analogy, it is of human as A, and the machine, or any given of present, per term, low-intelligence lifeform B. However, the inconsistency starts when we consider the act of evaluating itself. By such concept and measure would we be able to model the state of self-conscious to the point of self-reflection and evaluation, that gives rise to this comparison in the first place? However, in most of the discussion, the subject matter of consciousness should be ignored, or at least assumed negligible, for it to be too complex that an entire corpus would not be enough to rather conjure on such idea alone. For now, we assume this ability is achieved of only the conventional end-goal.

We shall, then, define loosely what we meant by a **construct** for completeness of the above conjecture.

Definition 1.3.2 (Construct). *A **construct** is a conceptual encapsulation of two components: The **machine** in broad term which houses the operational facility, and optionally, the existential facility of the construct, and the **process**, or rather, its **state(s)** of being. These two makes up a functional construct of interest.*

What do we mean by *existential facility*? This is the first step, or rather, the first application of the abstraction line of thoughts. Although the discovery of biological facts and rules are still insufficient to guarantee a desirable amount of admission that we truly understand the physical world or at least, the physical realization of living organism, virtually every living organism as of date needs certain standard basis on which its existences are guaranteed. For example, albeit the chance of taking this wrong is substantial, but there are, for example, the requirement for the existences of proteins for the functions of cells. So, in one way or another, There are layers of existential requirements for large constructs to be reasonably placed upon, and such treatment facilitates the relative *existential facility* that we recalled. Note, as written, qualifying to be called existential or not is relative, meaning for certain level of 'abstracted reasoning', the main existential facility might be different, and certainly might be of dynamic configuration than what would be expected.

So, in general, the entire continuum at present of the artificial intelligence concepts, lies within this framework that we would be considering.

Theorem 1.3.1 (Artificial framework). *An **artificial intelligence** framework considers the following aspect of exhibitive qualities:*

- *The set T of traits from any given subject reference A of which shall be (but conventionally be) called **intelligence** and their observables, their conditions, and their conceptual descriptions. This includes a **measure** takes either A, or similar concept of the same grade [a] denoted by $\mathcal{M}_A(A) : A \to T$.*
- *The set $\mathcal{C}$ of all **constructs**, as the generalization of the notion **machine** (however can be identical in nature), expressed in shorthand terms as the pair $(\mathcal{F}, P)$ of **facilities** and **processes**.*

Given $\mathcal{C}$ is the set of all constructs, its function might be assumed to hold the following rules, either for interpretability, or for better organization in the way it is structured:

1. ***Principle of abstraction**: For any given construct $C \in \mathcal{C}$, there exists a finite partition $\mathcal{P}_n(C)$ into n arbitrary grades, or **layers** of such it follows the exhibition of knowledge - for any components $c \in C$ of a layer $n-1$, the layering partition effectively removes its correspondence to n and its data availability to components and structure of layer n.*
2. ***Principle of externality**: For every component or construct, of any layers, there are always the relative functional partition between **external processes** and **internal process**, every such concepts stripped down to the last implementation is expressed by an input-output procedure.*
3. ***Conjecture of recursive immergence**: With sufficient complexity, or given of given layer, one system can host another of the same type in itself, for such to enable self-consideration and recursive loops.* [b]

*For the set of construct $\mathcal{C}$, partitioned by layers similar to its internal structure, there exists a finite measurable set σ-$\mathcal{C}$ such that the set is expressed by $\sigma\text{-}\mathcal{C}[\mathcal{M}_\sigma] = \{C_1, \ldots, C_n\}$ for n being such finite size. Then, with $A \in \mathcal{A}$ of the set of all subject reference, and $\mathcal{M}_\sigma \cong \mathcal{M}_A$ for $\mathcal{M}_\sigma(C_k) : (\mathcal{F}, P) \to T$, if $\mathcal{M}(\sigma\text{-}\mathcal{C}) \cap \mathcal{M}_A(A) \neq \varnothing$, or, rather by $\mathcal{M}_\sigma(\sigma\text{-}\mathcal{C}) \approx T_A$, then we call it **sufficiently intelligent**, and guarantees the existence of a given $C_k \in \mathcal{C} \supset \sigma\text{-}\mathcal{C}$ that satisfies the above notion.* [c]

[a] Definition lacking. Later on need to define what does this mean.

[b] This conjecture follows from the observation that any sufficiently strong system, will have the ability to construct a logical system of similar complexity to itself in its own operating space. The foremost example is the action of simulating a computer, inside a computer. By the *computational theory of mind*, if we take on the stance that computer to a given complexity is indeed intelligent being, then the conjecture follows.

[c] Note that however, this entire concept relies on the expanded order, perhaps analogically represented as a set funnel - then if $\mathcal{M}(\sigma\text{-}\mathcal{C}) \cap \mathcal{M}_A(A) \neq \varnothing$ but also, $\mathcal{M}_A(A) \subset \mathcal{M}(\sigma\text{-}\mathcal{C})$, then we would have to call it *above of the sufficient intelligent reference*. If this difference is over by a ratio $T_{\sigma\text{-}\mathcal{C}}/T_A = \lambda > 1$, we call it, if A is the human references, then we would call it **artificial super-intelligence**.

There are problems, of courses, and arguments against such construction made above, since it aligns and takes several aspects of old theories, beside the new content involved. Such problems must be addressed for the foundational basis to be considered broadly as much as possible, for the purpose of inclusion, and to define the practical limiting boundary onto current constructs. Additionally, from there, certain points and observations can be made, and new questions might be asked for problems that cannot be solved yet, for now. Arguing is the best source of thoughts, as the ancients predating us told of, rather true than not.

1.4 Philosophical arguments against AI

Philosophically, we have several considerations to make use of. Most of these comes of from historical remarks from ancient time (if to say 1700 is indeed ancient), and from 1950s onward, where capability of simulating models is possible. We would see what has been brought up about the topic of artificial intelligence, and especially, against the view of computationalism.

1.4.1 Descartes's argument (1700s)

Descartes left a lot of his works upon the topic that now is conceived to be related to the philosophy of artificial intelligence, ignoring only the historical namesake itself. Most of his works are of great interest on the 'artificial construct' of the time - the idea and society of the 17th century's Age of Enlightenment's *machina*. In the quest to understand his reasoning, for it holds values even now, and holds specific philosophical reasons in such discourse, we

will examine **Discourse on Method**, the one specific piece of work directly target this line of thoughts. One of the first target, will be the distinction between man and machine. Though this has been directly argued above, in the example of Descartes and the machina animated to life, it is worthy of mention that this has not been solved, yet. In fact, one can even interpret the Turing test as one kind of empirical boundary for Descartes' criterion - by setting out a boundary on which human-like reaction can take. Moreover, it presumes the justification and the criterion that we should interdict whenever a similar structure is proposed - would it be reactive enough as a man, years long ago predicted?

1.4.2 Categorization - Strong and weak AI (1991)

Boden's concept of artificial general intelligence resembles John Searl's "strong AI".

"Weak" AI can be defined as the form of AI that aims at a system able to pass not just the Turing Test (again, abbreviated as TT), but the Total Turing Test (Harnad 1991). In TTT, a machine must muster more than linguistic indistinguishability: it must pass for a human in all behaviours - throwing a baseball, eating, teaching a class, etc. According to Searl, "weak AI" is a computer that can behave as if it were thinking wisely.

"Strong AI" is then differently defined as a computer that actually thinks like humans. For a quote:

"*According to strong AI . . . the appropriately programmed computer really is a mind, in the sense that computers given the right programs can be literally said to understand and have other cognitive states.*" (Searl, 1991)

The theme of strong AI was frequently discussed in the late 20th century - however, it became clear that in order for a computer to be a strong AI, it must resolve various difficult problems. The most difficult philosophical problem was **the frame problem**.

The frame problem

The **frame problem** is the problem that an AI cannot autonomously distinguish important factors from unimportant ones when it tries to cope with something in a certain situation. The problem arises, for example, when we let AI robots operate in the real world. This problem was proposed by John McCarthy and Patrick J. Hayes in 1969. This is considered a philosophical problem that cannot be merely reduced to a technical problem. Historically, the problem is narrowly defined for the field of *logic-based artificial intelligence*. But it was taken up in an embellished and modified form by philosophers of mind, and given a wider interpretation, and hence, is since then applicable to almost all formal system that wishes to call themselves artificial intelligence. We will not cover all of it here, at least for now. For trusted literature, it is recommended to refer to the Stanford's article on the frame problem Shanahan [2016], and other literatures Gryz [2013], Seager [2010s (or older], Briggs [2014].

For now, the problem is unresolved, per Boden and specialists. Although there is no consensus about the definition of the frame problem, we could say that this is a problem centred around the question of how we can make an AI memorize the 'tacit knowledge' that almost all human adult can have in a given context.

Take an example of the normal cashier. Theoretically, and perhaps realistically speaking, a high schooler can be trained in a rather hasty fashion to do the job. While doing such work, there are many factors we take for granted. For example, during the process of using the computer to input the amount of cash required per transaction making, there might be a few terminal differences between different interface. The cashier knows that, and adapt to it. If she encounters an angry or hurry customer, the cashier can also act accordingly, without the consideration for the performance. "If the customer is in a hurry, then maybe I should use this or

that or skip this". Or even "If I push this button, then the trading screen appears". In fact, there are too much knowledge to be involved in such normal and particularly easy job. However, we do not have to input the knowledge that the stock market will affects the customer's money that you are indeed doing transactions, or the fact that if the customer comes in with a bag of money, then in some cases, there will be missing bills, simply because it is not concerned of such.

Considering this, it becomes clear that there is an infinite amount of knowledge the robot must memorize. Who can make such a list of knowledge, and how is it possible to make the robot memorize them? The reason why this happens is that, when a robot encounters a new situation that it has never experienced, it cannot autonomously judge what kind of coping would be important to itself and what kind of coping would not, and therefore it cannot adequately solve the problem it faces. It is interesting that humans seem to be able to solve this kind of problem.

According to Dreyfus Dreyfus [1979], for traditional AI to have the capacity to solve the frame problem, and become a true AI, it must become the Heideggerian AI, which incorporates Vorhandenheit (presence-at-hand) and Zuhandenheit (readiness-to-hand). Examining Rodney Brooks' robot architecture, he said that the robot respond only to fixed features of the environment, not to context or changing significance. But the robot 'continually referring to its sensors rather than to an internal world model'. Then, would the act of choosing, implementing a good enough sensor which have in itself the impending sensory power in need, and most importantly have the best data coverage be enough to mitigate this? Since even for human, the sensory power is limited, since our limbs, nervous system and cognitive realization systems are finite, but perhaps we have a major sensor of which brought us the power of solving the frame problem locally, as well as responding to 'context or changing significance'? There might be, but then the question is quite straightforward: What (part) of human exhibits such trait? A further concession if made would be even more in line: which construct would give rise to such desired adaptation in the absence of rigorous human benchmark, but for lower-intelligent animals as comparative instead? And even then, would it be reasonable to call computers no better than the smallest reptiles?

1.4.3 The Chinese Room Argument (1980s)

One of the most prominent critique for the philosophy of "strong AI" is the Chinese Room Argument. Accordingly,

> *CRA is based on a thought-experiment in which Searle himself stars. He is inside a room; outside the room are native Chinese speakers who don't know that Searle is inside it. Searle-in-the-box, like Searle-in-real-life, doesn't know any Chinese, but is fluent in English. The Chinese speakers send cards into the room through a slot; on these cards are written questions in Chinese. The box, courtesy of Searle's secret work therein, returns cards to the native Chinese speakers as output. Searle's output is produced by consulting a rulebook: this book is a lookup table that tells him what Chinese to produce based on what is sent in. To Searle, the Chinese is all just a bunch of — to use Searle's language — squiggle-squoggles.* (Searl, 1980s)

We denote O the observers (Chinese speaker), i, o being the input, output accordingly, and the rulebook P.

The argument of this experiment is simple - the Searle (in the box) is supposed to be everything a computer can be, and because he doesn't understand Chinese, no computer could

have such understanding. Searle is mindlessly moving around, and according to the argument, that's all computers do, fundamentally.

Nowadays, CRA, among AI practitioners, is generally rejected. Among these practitioners, there is John Pollock, who writes:

> *Once (my intelligent system) OSCAR is fully functional, the argument from analogy will lead us inexorably to attribute thoughts and feelings to OSCAR with precisely the same credentials with which we attribute them to human beings. Philosophical arguments to the contrary will be passé.* (Pollock, 1995)

Still, despite such argument, the relevance of CRA is actually more apparent than ever. The brute fact is that deeply semantic NLP is rarely even pursued, hence the proponents of CRA are certainly not the ones feeling some discomfort in the light of the current state of AI. Searl would rightly point to any of the success stories of AI, and still able to proclaim that what we want - of an artificially intelligent agent has not been made, and understanding has not been reach, and philosophically, we can't refute it.

An analysis of CRA

In one way or another, CRA infers with the notion of an *input-output* procedure. In such case, CRA clearly tells us that a simple, mundane notion of an input-output machine which 'does its tasks' would not be sufficient of receiving the clarification and qualification for being intelligent. Which, by all means and purposes, are true. The construct if given in such circumstances, of the thought experiments, represents, if we took out the part where the argument said Searl is supposed to be everything a computer can be, then it's true that such constructs are false in its claim that it can 'understand'. In fact, relatively simple, a given input-output mechanism, from what was observed from the outside, do not exhibit anything, and do not have the ability to even *think*, regardless that is valid of such. If we are to stand by Descartes arguments, then it is even more of the truth - the system in which the thought experiment was conducted provides it with no capabilities of any such.

Then what would be of the Chinese Room Argument that is worth it to dissect? Well, firstly, the claims of, at least in the acute interpretation - the man in the box is supposed to be everything a computer can be is *false*. IF we are to stand by our construction of facilities, then we are innately arguing about such facilities, and not the processes, and the underlying operations itself. Rather, we are complaining that the machine is not capable enough, which is true. But we also, to a given reference point, pointing to the **existential facilities** instead of the arbitrary, yet reasonable operational facilities instead for the comparison. And in fact, if we think of it in the layer construction, it makes more sense - each layer is classified given its arbitrary for now, an interpretation thereof. Such interpretation is contained for such layer, and hence cannot be thoroughly or at least in a glance, interpreted by lower-layer components. This construct offers a one-way restriction on the property of interpretability. In such case, the **System Reply** is partially right - the man inside can not be, by all means and purposes, understand what does it mean by even 'English', or 'Chinese'. And even if such English 'understanding' is embedded in the reasonable interpretable space of Searl in the room, then Chinese would appear to be entirely unknown, *unless* there exists a helper tool to resolve the situation of undefined operation. Indirectly, this prevents a Descartes situation from happens - an undefined situation with particular way to resolve.

But more than that, what constitute the notion of understanding? Taken only from the setting of the thought experiment, we cannot do but deceptively assume that understanding a language is similar to giving it certain ruleset to transfer from this word to others word only.

By that, converting from base-2 to base-10 works the same - you know how to do it, yet there are things that constitute the philosophical, higher-level notion of 'understanding' in such that allows you to actually understand the conversion - otherwise, the conversion is a blind matching. In fact, given the setting, would a conversion rulebook exists for such language? Language is, by itself, a very high-level concept. Translating from text to texts requires it not only to provide the definition matching, which could be reasonably identified by such notion like the rulebook mention, but the interpretation of the string is dictated by the logic of the language - the context in which it appears, the logical conformation that it contributes, the grammatical structure that makes sense of what is said and what is transferred, and else. Language itself, is a medium of information exchange, by one of its definition. A conversion does not constitute an understanding. And if the argument going back and forth is that Searl can somehow figure it out the patterns mean something, then it violates what I called the *principle of externality* - the 'lower components' up to a given point in which the *law of recursive immergence* does (not) apply, cannot implement its higher constructions. Then, the Chinese Room Argument can be interpreted, in somewhat meagre form, the argument against telling the current conformation as able to understand, while it is not.

The only thing here that is wrong, however, is the fact that Searl claims that it is all the computer can do. However, artificial intelligence, as for now, using this term since chapter 3 which is not yet here, is not a computer in its form. We say, however, for an *artificial intelligent subject with computers as its existential facilities*, CRA is partially right. But partially wrong since the comparison is limited to a form of internal structure in a well-formed system. That is to said - we need to create the (a) construct(s) that exceed(s) such argument. The problem is, how?

1.4.4 The Gödelian argument (1961)

In 1961, J. R. Lucas presents the Gödelian argument against the existence of a "strong" AI. His proof is based on Gödel theorem, which is stated as followed:

> *In any consistent system which is strong enough to produce simple arithmetic there are formulae which cannot be proved-in-the-system, but which we can see to be true. Essentially, we consider the formula which says, in effect, "This formula is unprovable-in-the-system". If this formula were provable-in-the-system, we should have a contradiction: for if it were provable-in-the-system, then it would not be unprovable-in-the-system, so that "This formula is unprovable-in-the-system" would be false: equally, if it were provable-in-the-system, then it would not be false, but would be true, since in any consistent system nothing false can be proved-in-the-system, but only truths.* (Lucas, 1961)

This theorem holds for all formal systems which are consistent, adequate for simple arithmetic, and shows that those formal systems are incomplete, with some fact being true, but unprovable. "*It is of the essence of being a machine, that it should be a concrete instantiation of a formal system. It follows that given any machine which is consistent and capable of doing simple arithmetic, there is a formula which is incapable of producing as being true...*" (Lucas, 1961)

Further argued, he then comes to such conclusion that no machine can be a complete or adequate model of the mind, since "the mind are essentially different from machines". Lucas's defenders, Roger Penrose, also state in his *Shadow of the Mind* (1994). A human mathematician, if presented with a sound formal system F, could argue as followed:

> *Though I don't know that I necessarily am F, I conclude that if I were, then the system F would have to be sound and, more to the point, F' would have to be sound, where F' is F*

> *supplemented by the further assertion "I am F"*[6]*. I perceive that it follows from the assumption that I am F that the Gödel argument $G(F')$ would have to be true and, furthermore, that it would not be a consequence of F'. But I have just perceived that "If I happen to be F, then $G(F')$ would have to be true", and perceptions of this nature would be precisely what F' is supposed to achieve. Since I am therefore capable of perceiving something beyond the powers of F', I deduce that I cannot be F after all. Moreover, this applies to any other system, in place of F.* (Penrose, 1992, 3.2)

By default, the argument supplemented from Penrose raised the contradiction of proof-ness. The Godelian argument implicitly creates layers, and levels, on which one puts those languages they are abided to seem fit of their expressions on the shelf, by the order of *effectiveness*. Such notion then, would make the advancement of machine to human seems perpetually, unsophisticatedly, inoperable and impossible in essence.

Subtle remark

Before even taking a stance on such argument, what is the meaning and interpretations, as well as obstensively why it is even important to divulge into such point? The answers might be a bit difficult.

Human is variedly different from machine, for the current time with all the knowledge at present. Truth to take, the action of writing this itself is part of the endeavour to discover one's self, or rather, to understand F with the assertion of 'I am F', for now, that we can, and is doing. By the language and construction of contemporary and propositional logic, a machine cannot do that.[7]

However, if the converse situation happens, where we cannot totally perceive what we actually think, and how it is formed - per metric, being either consciousness, or one's self, surprisingly, it does not support the previous argument from an intuitive view (Bear in mind that this is a non-rigorous study). If stays rigid as it is, not counting being dynamic as we want, the model created from a human being can only imitate and represents what directly is entailed in the human mind of interpretation and logics. But logic and interpretation is a construct of the mind, for all intents and purposes, to directly infers to the physical world, the living world. However, if one is to use such inference on itself, for example, examining the brain itself, then to a certain point, what can be deduced from such observation can only fit in the interpretation space of what its creator, the human brain itself, can contrive. Thereby, we might conclude that figuratively, even human cannot understand human itself, from certain perspective. But the quality of succinctly interesting loop is to be taken seriously. The point now is, what type of construct, even logic, would be sufficient of taking the understanding, and will it make uses of the looped behaviors? By that, we then argue superficially that anything that relies on the machine cannot model the human existence and conscience itself. There are some assumptions thereof in the argument:

- Existences and the state of the world are in fact, modelled in mathematics, for one way or another. This is to facilitate the use of formal system in the argument. Everything is a set of rules, in which things operates.

[6] The phrase "I am F" is merely a shorthand for "F encapsulates all the humanly accessible methods of mathematical proof"

[7] In general, we cannot even say that it is true of the truth that human actually differs from what is proposed to be perceiving F' being F. For human understanding of ourselves alone, we are trying to fit it into the interpretation and the rough 'understanding' of human itself. That is, there exists the space of reason and argument of a scientist, which interpretation follows. If, supposedly, this interpretation is strong enough, then we might be able to perceive and understand ourselves from ourselves - a looped interpretability. This mechanism, if ever, is not well understood if exists.

- A machine per its definition, cybernetical machines are of all expressed by the single principle that it is born out of a formal system itself.
- Truth is the finite quantity that exists in such formal system, and is absolute.
- The mind is an entity of which is inherently different from the logic of formal system.

Those fundamental, overlapping assumptions make up the bulk of the Gödelian argument, from the surface. However, is that true of all the merit? [8]

[8] It turns out, however, the Godelian argument has various proponents and opponents, and there are arguments of it being false. See (Bringsjord, 2001) for such argument, but it can be simplified as this. The Godelian argument makes use of two assumptions: $G(F')$ is true for a Godelian statement, and $F' \nvdash G(F')$ for $F' \nvdash G(F')$ for F' being "I am F" with added semantic. Then, the statement on $G(F')$ is true is nothing but a *satisfaction* claim, of meta-mathematical assertion which can be reduced to $\mathcal{I} \vDash G(F')$ is true for given interpretation $\mathcal{I}$. Thereby, there exists no contradiction thereof.

I. Preliminary

Everything to note of the foundation language of science. Every prerequisite of entering a gate. Including its keys, for more than one might be needed.

Chapter 2. Foundational mathematics

We present preliminary knowledges before jumping in further section. Most of the knowledge falls in the range of mathematics, or treatment by mathematical concepts, physical interpretation, computer science, and neuroscience. However, there do exist several knowledges that is more outlier than such, typically fall into the range of philosophy and other treatment of *conceptual models*.

2.1 Elementary logics

By definition, let's define an informal view on proof and logics. proofs logics

Definition 2.1.1 (Proof, I). *A proof is a sequence of true statements, without logical gaps, that is a logical argument establishing some conclusion.*

This definition captures the notion of proofs, but however, lacks the actionable components.

Definition 2.1.2 (Proof, II). *A proof is a sequence of logical statements, one implying another, which gives an explanation of why a given statement is true. Mathematical proof is absolute.*

To prove things, we need to start from assumptions, or axioms. By non-rigorous manner, axioms can be taken as the formal ground for which the work can be done on the above layer. The axiom can be reasonable or not, so is the truth assumption for that axiom. It is like saying "If the axiom is true, then this is what I told you". Others axiom might still yield different results, as per mathematicians like. The key thing is to define, and at least draw out a general assumption that your axiom takes form, that is on first hand, actionable. Then the rigorous system will be built upon that.

Per attention, we also care about the little "statement" in our definition, too. statement

Definition 2.1.3 (Statements, I). *A statement is a sentence that can have a true value. That is, it is a logical unit with truth value being assigned to its truth.*

Therefore, we can also say statements declare or assert truths of certain subject.

Those statements can be either false or true. Often, we will want to concern about statements with truth value, not as the statement of there are infinitely many primes of the form $n^2 + 1$, because it is not probably true, but none has a clue. So it might be call (?) a conjecture instead. There are also times when statements can have no justifiable truth value, of which then we will need to put on such convention on the statement specific properties.

Definition 2.1.4 (Statements, II). *A statement is a declarative sentence, otherwise called a declaration that is discrete in its properties of true or false, but not both. Formally, this is written for a statement $P(n)$ as:*

$$P(n) = \{P : P = T \text{ or } P = F\} \tag{2.1}$$

In some cases, it is not immediately clear if a statement is true or false. So even with the indication slightly contradictory to the above notion, for a sentence to be a statement, its capacity of assigning truth is a requirement, but we do not need to know its exact propensity, or its actual truth. [1]

A sentence that satisfies or not the condition to be a statement, but contains in its description of the declarative part of variables or elements of prescribed sets, which is called the domain of such variable, then is called **open sentence**. If the open sentence $P(x)$ with $x \in S$, then we say $P(x)$ is an **open sentence on the domain** S. Then, $P(x)$ would be a statement for each $x \in S$.

open sentence

Thereby, from observations of the elementary system, the foundational blocks for proofs are from those sentences, and its characteristic sentence of open sentences. In interest and necessity of creation of statement, (open) sentences must be converted to statements. Only then the higher properties of statement, which would be discussed later, shall be applied onto the process of making proofs. To do this, we might want to have a look at the logical system of which enables the elementary form of logical argument, leading to the proof of requirements.

2.1.1 The Propositional logics

Working with logic, traditionally, is to deal with absolute truth. There's the notion between true, and false. So, rationality and statements in logics is figured in terms of discrete truth. We denote true as T, false as F for convenience.

What constitutes to our consideration of what is true or false? Specifically, what do we mean by truth? We use the above notion of a statement for such. Statements is the foremost piece of 'equipment' we have in logic, aside from (open) sentences. Informally, they carry truth value, of which meaning their description is realized, and not vague in terms of such realization. Simply speak, if a statement is false, then it is indeed, false, under any consideration of inspection, given the system of formalism that it is based off.

Logical connectives

Logical operations use statements and act on them, producing new logical statements. Their truth value (either true or false) is presented by using truth table, essentially a table-based organization for the combination of truth value. So in essence, they are the tool for statements to acts of each other, based of the properties of relations, and what is connected to each others.

logical system

logical operation

In a perspective, we can say that logical system is constructed using logical operation, as a mean to connect logical statements, or transforming them. Because of this, they are sometimes referred to as **logical connectives**. Formally speaking, for a statement $P(x)$, logical operation is of the form $\text{LO} : S^n \to T^m$, where n and m indicate the domain involvement of the operation, and the domain resultant of such operation. A **unary** operation takes the form of $n, m = 1$, since they take one and produce (transforms) one. Binary operation would be $n = 2, m = 1$ since they took two argument, and produce one. All of this would be reflected in the truth table, as per configuration of the mathematicians.

conjunctions

disjunction

Let P and Q be statements. Then $P \wedge Q$ is the "and" operator, and $P \vee Q$ is the "or" operator. Formally, they are called conjunction and disjunction, respectively. Their role is to combine specific statements (2 statements) under the landscape of some rules: Either of them is correct yields correct statements, or a bit more lax, only one of them need to be such, at bare minimum. So for the conjunction operation, combining two statements, A, B, will have the

[1] In some texts (Advanced Calculus, Sternberg, 1990), open sentence is called as a **statement frame**, of which then statement is obtained from such frame. This is accordance to the coverage of each open sentence (frame), since it applies the rules onto different elements of the domain.

following truth table

A	B	$A \wedge B$
T	T	T
T	F	F
F	T	F
F	F	F

Here we can see how truth table is presented: Statements with their truth in specific columns, and the operation's results of logical realization on the other side. Sometimes, because for every operation, even nested or so, complex or such, must then come out to be a single truth value at the end of its chain of logical actions, it is convenient to list out the possible truth value as a table of such.

Of course, increasing the amount of statements increases the possible permutation that such complex operations can bring, but that is of the concern later on, and still will result in one single truth value.

The truth table of the disjunction operation is presented as

A	B	$A \vee B$
T	T	T
T	F	T
F	T	T
F	F	F

Similarly, $P \implies Q$ is implication operation. The only case for which implication is false is when P is true, but Q is false. Why is this the case? Suppose that P is false and Q is true. In this scenario, for example, the student in a test did not get an A on his exam, but when he receives his final grades he learned that his final grade was an A. How could this happen? The only argument about this is that the whole ordeal is a mistake, and need to check back, why? Because the instructor did not lie, so do the grade. So there must be a mistake somewhere, hence, it is indeed false.

$P \iff Q$ is the second binary operation aside from the standard operations above. It stands for "if and only if", and is called a **biconditional** logical operator. It is much stricter than the implication operation: similar to and, it requires for the truth to be evaluated as T, if only the two-way implication is true. Specifically,

$$(P \implies Q) \wedge (Q \implies P) = T$$

This would also be the time when you are introduced to what exactly are we doing, that we are using notations and formal language of such form we are now. Such form is called as **atomic**, of which statements and operations can be constructed from other components, such as the **and** operator and the implication operator, as you saw. In case of biconditional operator, formally, we often state it as:

$$P \text{ is equivalent to } Q$$

or as

$$P \text{ is necessary and sufficient for } Q$$

Their truth table is as followed:

P	Q	$P \implies Q$	$P \iff Q$
T	T	T	T
T	F	F	F
F	T	T	F
F	F	T	T

So far, we have only seen binary operation, taking two arguments (statements) and combine them. An example of logical operation, specifically unary, is the operation of negation $\neg$. For a statement A, negation has its truth table as such:

A	$\neg A$
T	F
F	T

Negation specifically means "not", of which negates the original statement.

Overall, the truth table is as followed for two statement A and B.

A	B	$A \vee B$	$\neg A$	$\neg B$	$\neg(A \vee B)$	$\neg A \wedge \neg B$
T	T	T	F	F	F	F
T	F	T	F	T	F	F
F	T	T	T	F	F	F
F	F	F	T	T	T	T

Tautology and Contradiction

We have seen single use of operation, up to this point. However, we can use the logical connectives above to form more intricate statements, which can be nested, sequential of many statements and connectives. More generally, a **compound statement** is a statement composed of one or more given statements (also called **component statements** in this context), and at least one logical connectives.

The compound statement below, which is combined of the logical *or*, and logical *or*, is expressed as

A	$\neg A$	$A \vee \neg A$
T	F	T
F	T	T

tautology

This is the statement $A \vee \neg A$ for any given statement A. The resultant is always true as we have observed. When such case happens, we call the statement a **tautology**. The concept of tautology is pretty important, even in this elementary stage of treatment on logic, specifically because it is static that it can be utilized in several proofs as a "constant" form. Specifically, tautology means that the compound statement S being classified as such, would always be true for all possible combination of truth values of the component statements that comprise S. Hence, $A \vee (\neg A)$ is a tautology.

An example shall be given. For statement P and Q, the compound statement $(\neg Q) \vee (P \implies Q)$ is a tautology, based of its truth value. (You can check it yourself - or just me). Still this statement, if we let P being '3 is odd', and '57 is prime', we just get

> 57 is not a prime, or 57 is prime if 4 is odd

This is true regardless of which statement P or Q is considered.

contradiction

On the other hand, a compound statement S is called a contradiction if it is false for all possible combinations of truth values of the component statements that are used to form S. The statement $P \wedge (\neg P)$ is a contradiction, as being shown of its truth table:

P	$\neg P$	$P \wedge (\neg P)$
T	F	**F**
F	T	**F**

Logical equivalence

Certain logical proposition are equivalent, which we denote $\equiv$. Two logical statements are called logically equivalent if the truth tables (all possible assignments of truth value for the logical variables) are the same. Formally, this is defined as logical equivalence.

logical equivalence

Specifically, let R and S be two compound statements involving the same component statements. Then R and S are called **logically equivalent** if R and S have the same truth values for all combinations of truth values of their component statements.

We have the following definition.

Definition 2.1.5 (Logical equivalence). *Let A and B being any (compound) statements or open sentences with domain specified. Then A and B is logically equivalent if for any configuration of truth in A, B matches its truth value to A, and reverse.*

Logical equivalence is especially useful, in case we want to compare certain complex compound statements and its truth value as consequences. This includes also, confirming for example, if we somewhat want to 'shorten' down certain compound statement to be a logical operation itself, for example, bidirectional, we can pretty much confirming them both. This works for tautology, per purpose of what we want to do. The key thing, simple down, is that it offers us a way to look at multiple interpretation of a single logical system, of which then the equivalence notion is defined.

There are several things to note from this. Firstly, is that for two logical statements to be compared, then every compartment of their component logical truth must be told. Second, there will be time when one statement uses certain component, that the other compound statements have none. In such case, it is hard to tell aside from testing if the behaviours result from such statement can be equivalent or not, since there is 'external' factor to the truth evaluation. However, it is a minor concern, as after testing of truth value then comparing it is still a better choice.

Setting that aside, there are some theorems we need to know of this elementary section.

Theorem 2.1.1 (Equivalencing). *Let P and Q be two statements. Then*

$$P \implies Q \equiv (\neg P) \vee Q$$

Theorem 2.1.2 (Equivalent Law). *Let P, Q and R be statements. Then,*

1. *$P \vee Q \equiv Q \vee P$, and $P \wedge Q \equiv Q \wedge P$ (Commutativity)*
2. *Associativity:*
$$P \vee (Q \vee R) \equiv (P \vee Q) \vee R$$
similarly,
$$P \wedge (Q \wedge R) \equiv (P \wedge Q) \wedge R$$
3. *Distributivity:*
$$P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$$
similarly,
$$P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$$

Finally, we have De Morgan's Laws, in logical form. Later on, we would see that this is also apparent in set theory for conjunctions of multiple sets.

Theorem 2.1.3 (De Morgan). *Let P and Q be two statements. Then*

$$\neg(P \vee Q) \equiv (\neg P) \wedge (\neg Q) \tag{2.2}$$
$$\neg(P \wedge Q) \equiv (\neg P) \vee (\neg Q) \tag{2.3}$$

All the above theorem can be deduced and verify by means of truth tables.

Quantifiers

In logic, there are certain lexical components we call as quantifiers, which is to 'quantify' the scope of application, for logical arguments. $\forall x, P(x)$ means $P(x)$ is true for all x, and $\exists x, P(x)$ there exists such x that $P(x)$ is true. These two notions can be understood intuitively: One guarantee *universal correctness*, while the other guarantee *existential correctness* - there will always exist such fact, at the minimal counting unit we can find. The quantifiers are usually bounded, as per their definition in the logical unit. Formally, $\forall$ is the universal quantifier, and $\exists$ is called the existential quantifier. These quantifiers are used to assert certain open sentence, of which then produce statements. We will discuss this in the next section.

Negations of quantifiers are as the following table:

$$\neg(\forall x)P(x) \equiv (\exists x)(\neg P(x)) \tag{2.4}$$
$$\neg(\exists x)P(x) \equiv (\forall x)(\neg P(x)) \tag{2.5}$$

2.1.2 From frames to statements

We have mentioned that if $P(x)$ is an open sentence over a domain S, then $P(x)$ is a statement of each $x \in S$. This implies the need for the domain of S to be specified for the open sentence $P(x)$. Or rather, we can break down a statement as it is formed by:

$$\text{Statements} = P(x) + \text{Dom}(x) \tag{2.6}$$

of which $\text{Dom(x)} = S$, is the domain of the variable x. There are a few ways to convert open sentences into statements, of which we will make use of the above logical system for such task.

Quantified(-cation of) statements

logical quantification

quantified statement

One of such way to convert an open sentence into a statement, is by the mean of quantification. If $P(x)$ is a statement frame over some domain S, obtaining statements from this frame can be done by attaching quantifiers onto it. This asserts the frame $P(x)$, for specific x compartment. Such statement is then called as **quantified statement**. Formally, for a statement $P(x)$, then the quantified form of such statement is taken in the form of $(\forall x)P(x)$ or $(\exists x)P(x)$. One frequently presents sentences containing (multiple) variables as being always true without explicitly writing the universal quantifier, however. So instead of

$$(\forall x)(\forall y)(\forall z)[x + (y + z) = (x + y) + z]$$

we can just write

$$x + (y + z) = (x + y) + z$$

for the ease of writing the quantifiers. Note that the shortened form of this quantified statement is of the form $(\forall x)(\forall y)(\forall z)P(x, y, z)$.

For existential quantifier, an existentially quantified statement only guarantee 'sometimes' true quantification to such frame, hence it must not be absent from the formal writing.

The statement $(\forall x)(x < 4)$ still contains the variable 'x', but it is no longer allowed to take on any values, and is called a bound variable. Roughly speaking, quantified statements contains quantified variables, which are bound, while unquantified variables are free. The notation $P(x)$ is hence very specific - it is only used when the variable x is free in the sentence being discussed.

Order of quantifiers

One of the simple fact for quantifiers and bounds has their order, is that they are not commutative, if the quantifiers' types are different. So $(\forall x)(\exists y)P(x,y)$ is inherently different from $(\forall y)(\exists x)P(x,y)$. Why would this happen?

First, we revise what the *quantifiers* actually means. Quantifier, is analogous to the domain, for specific statement frame (we use this word for better representation than open sentence). It enforces the operational scope of finding the statement - for example, $\forall x \exists y$ means exactly that, there exists a y in the scope of the domain of x, such that it becomes the minimal existential guarantee. Thereby, only one exists would give the statement generated off the statement frame, to be true. However, if we reverse it, then the scope changes: $\exists x \forall y$ means any y will have at least one x of such property as in $P(x,y)$. The scope has changed - now it must be correct for every element possible of y, and each of them need at least 1 example. If we can interpret it differently, it's the extreme value of minimal bound, and the minimal of the extremal bound, so to speak.

On the other hand, if they are of the same types, then it is fine. Among a group of quantifiers of the same type, the order does not affect the meaning. Thus, $(\forall x)(\forall y)$ and $(\forall y)(\forall x)$ has the same meaning. This also mean we can sometime abbreviate it as $(\forall x, y)$, if we wish to reduce the amount of notation needed.

Lemma 2.1.4. *If quantifiers of both types are used, the order of which they are written affects the meaning of the statement, and hence they are not commutative. Quantifiers of the same types do not have such effects.*

Example 2.1.1 (A test version of justifying quantifiers). *Denoting truth as numbers, $\{0,1\}$. We have the logical statement as a function $L : \prod_{\times, i\in I}(B_i) \to \{0,1\}$, of which $\{B_i\}_{i=1}$ is the set of all 'bounds' that each variable constitute in such. Then, we can interpret the two existential and universal as bound such that:*

$$B_\forall = \{x_i\}_{i=1}^{m>1} \tag{2.7}$$
$$B_\exists = \{y_1\} \tag{2.8}$$

Thus, existential order can be thought as, for binary case,

$$L_B : B_\forall \times B_\exists \to \{0,1\} \tag{2.9}$$

to relies on its truth value for the domain region of $B_\forall$ as a stronger bound to guarantee that the statement is then true.

Similarly, the statement

$$L_B : B_\exists \times B_\forall \to \{0,1\} \tag{2.10}$$

relies on the domain $B_\exists$ to dominate the strong bound of true evaluation.

The above example only offer a "naive" but systematic way of justifying the relationship between two quantifiers, under an elementary setting of propositional logic.

Of all, when using quantifiers, we need to keep track of our order, regarding all the quantifiers being used. If not, mathematicians' career would be in shamble since they used them all wrong, if ever, until they realize it right now. Because it is a very convenient tool, that is why we must be careful on how to use it. Additionally, there are several times we should reduce the amount we use, and instead work on it with an actionable bound, instead of an absolute minimal/maximal bound of such.

On a side note, the idea of strong bound truth validation is a very interesting idea, and might meet its analogue somewhere else in the mathematical system.

2.1.3 Characterization

Suppose that some concept (or object) is expressed in a statement frame $P(x)$ over a domain S, and $Q(x)$ is another frame over the domain S concerning this concept. We say this concept is characterized by $Q(x)$ if $\forall x \in S, P(x) \iff Q(x)$, i.e. they maintain a bidirectional relationship, is a true statement. The statement above is then called a characterization of this concept.

characterization

Let's take an example.

Example 2.1.2. *Suppose that*

$$P(x) : x = -3 \text{ and } Q(x) : |x| = 3$$

where $x \in \mathbb{R}$. Then the biconditional $P(x) \iff Q(x)$ can be expressed as "$x = -3$ is necessary and sufficient for $|x| = 3$", or perhaps better, $x = -3$ is a necessary and sufficient condition for $|x| = 3$.

Now, consider the quantified statement $\forall x \in \mathbb{R}, P(x) \iff Q(x)$. This statement is false because $P(3) \iff Q(3)$ is false.

Note that sometimes, we arguably misuse characterization, and definition. Supposedly, Let A being a triangle. Then, A is equilateral if it has three equal sides. This is the definition. For the notion introduced in this section, we have the following characterization:

$$A \text{ is equilateral} \iff A \text{ has three equal angles}$$

Notice that the definition is of three sides, and the characteristic of such concept, is that it must then, consequentially, have three equal angles. So the definition is indeed true, but not a characterization, because that is what a definition is, even though the bidirectional relationship is true.

This sounds like more of a rehearsal, but in fact, there are many definitions that use bidirectional conditions to sustain itself of a concrete definition. That is because it offers something that is analogous to equality in general mathematics, but stronger - a binding between the concept and what is its representation. That is why we must define and differentiate between two 'objects' that is created from such bidirectional statement, definition, and the latter of this section's name. Otherwise, our foundation of definitions, for at least the starter point of mathematics, would be in troubles.

2.1.4 Restricted variables

Usually, in mathematics, a variable is not allowed to take all objects as values, it can only take as values the member of a certain set, which we already called the *domain* of the variable (independent, not dependent variable like y in a typical function). The domain is sometimes explicitly indicated (like what we have seen when ambiguity is presented), but is often only implied. In a logical sense, the letter n is customarily used to specify an integer, so that $(\forall n)P(n)$ would automatically be read "for every integer n, $P(n)$". However, sometimes n is taken only as positive integer. In case of possible ambiguity or doubt, we would indicate the restriction explicitly and write $(\forall n \in \mathbb{Z})P(n)$. The quantifier is read, literally, "for all n in $\mathbb{Z}$". The above quantifier are called *(logical) restricted quantifier*.

In the same way, we have restricted set formation, both implicit and explicit, as in $\{n : P(n)\}$ and $\{n \in \mathbb{Z} : P(n)\}$ which both reads as "the set of all integers n such that $p(n)$".

In a sense, restricted variables can be defined as abbreviation of unrestricted variables by

$$(\forall x \in A)P(x) \Leftrightarrow (\forall x)(x \in A \Rightarrow P(x))$$

$$(\exists x \in A)P(x) \Leftrightarrow (\exists x)(x \in A \wedge P(x))$$
$$\{x \in A : P(x)\} = \{x : x \in A \wedge P(x)\}$$

Although there is never any ambiguity in sentences containing explicitly restricted variables, it sometimes helps the eye to see the structure of the sentence if the restricting phrases are written in superscript position, as in $(\forall \epsilon^{>0})$, which might looks weird, but is pretty much more verbose.

2.2 Naive set theory

Almost everything is set. Most of mathematical concept can be either constructed and interpreted by set, or by logical system, in which both of them interchange each other at work. This section will provide a thorough overview of the formal concept for set theory. Of course, may we even approach ZFC.

Set theory historically began with the concept of the naive set, and operations on set. It is the first attempt to create the formal theory on categorization (setting things in to boxes of similar properties, or something else). Per our separation to be historical, I think it is nice to review the definitions naively of set theory, and where the fallacy might rise.

Definition 2.2.1 (Set). *A set A is an unordered collection of distinct objects. For x in the set A, we say that x is an element of A, $x \in A$. Conversely, $x \notin A$. The set A then can be defined by its members, or their set comprehension using formulas.*

To describe a set, we use some way of listing the set: Either by listing, or identification. Standard (plus trivial) sets include the *empty set* $\varnothing = \{\}$, the *natural number* set $\mathbb{N}$ and its positive (non-zero) subset $\mathbb{N}_+$, *integer* set $\mathbb{Z}$, *rational numbers* $\mathbb{Q} = \{n/d \mid n \in \digamma, d \in \mathbb{N}_+\}$, and the *real number* $\mathbb{R}$.

To operate on set theory, we have this axiom:

Axiom 2.2.1 (Principle of Extensionality). *If two sets have exactly the same members, then they are equal. That is, for A, B are sets, such that $t \in A \iff t \in B$, then $A = B$.*

For such a set, the set A is called a **subset** of B, $A \subseteq B$ if all members of A are also members of B, which also contains the case $A = B$. This changes to $A \subset B$, the *proper subset*, if $A \neq B$ by the principle of extensionality. We said that A is **contained** in B. Conversely, the notation $A \supset B$ and $A \supseteq B$ takes as A is the **proper superset** of B, and subset of B. For $\bar{A} = \{x \mid x \notin A\}$, we call it the *complement* of A.

Any set will have one or more subset. For $|A| = n$, it has 2^n subsets. The set of all subsets of a given set A, denoted $\mathcal{P}(A)$, is called the *power set* of A.

Operation on set

Under naive set theory, we have the following operations:

- $A \cup B = \{x \mid x \in A \vee x \in B\}$. (union)
- $A \cap B = \{x \mid x \in A \wedge x \in B\}$. (intersection)
- $A \setminus B = \{x \mid x \in A \wedge x \notin B\}$. (set difference)
- $A \triangle B = \{x \mid x \in A \oplus x \in B\}$. (symmetric difference)

Theorem 2.2.1 (operations). *For A, B, C are sets, we have:*

1. $(A \cap B) \cap C = A \cap (B \cap C)$

2. $(A \cup B) \cup C = A \cup (B \cup C)$
3. $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

2.2.1 De Morgan's theorem

One of the elementary result often used in set theory is the De Morgan's theorem. We state it as followed.

Theorem 2.2.2 (De Morgan). *Given a set X and a collection of set, denoted by $\{A_i\}_{i \in I}$, we have:*

$$X \setminus \left(\bigcup_{i \in I} A_i \right) = \bigcap_{i \in I} (X \setminus A) \tag{2.11}$$

and

$$X \setminus \left(\bigcap_{i \in I} A_i \right) = \bigcup_{i \in I} (X \setminus A_i) \tag{2.12}$$

Under n sets operations, it is better to not write them all down like above, and so similarly to summation $\sum$, and product $\prod$, we have ourselves a set general notation. If A_α are sets for all $\alpha \in I$, then

$$\bigcap_{\alpha \in I} A_\alpha = \{x : (\forall \alpha \in I)\ x \in A_\alpha\}$$

and

$$\bigcup_{\alpha \in I} A_\alpha = \{x : (\exists \alpha \in I)\ x \in A_\alpha\}$$

2.3 Formal set theory

We have said earlier that we can construct sets by the subject of comprehension, or *set abstraction*. However, be wary of the noninformal treatment of abstraction for sets and subsets. For certain bizarre choices of the entrance requirement, it may happen that there is no set containing exactly those objects meeting the entrance requirement (Enderton, 1977). Either by defining non-"definable" construction, or by mishandling the concept of membership - Russell paradox, for $\{x \mid x \notin x\}$ being the famous example.

In set theory, there's the notion of container. This would come in handy later, and I think it's just at the right time for this discussion, after we at least have an idea of how set operate in mundane sense.

A container is different from its element, but as we have said above, there can even be the collection of collections. A famous example is the empty set. When we say that $A = \varnothing$, we did not say that it has nothing, because, A in this case, is the collection. We instead can write it as: $A = \{\varnothing\}$, specifically refers to the fact that it's a set, without any member. $\varnothing$ hence can be interpreted, and defined as a state of set - the state of being empty for any set. So, then, we have $\{\varnothing\} \neq \varnothing$, simply because one is a state of objects, while the other one refers to a collection, with that state of object. Intuitively, it's between a man with nothing, and a man with an empty bottle of water - at least he has an empty bottle.

So, thinking about it, ∞ and $\varnothing$ is weird. It can be a thing of it own, i.e. an object being analogous to $\varnothing$ might mean that it is void; on the other hand, an 'infinite' object might refer to its domain, or else. But when you package it into a collection, it becomes 'representative properties' of the collection itself (or the set itself). $\{\infty\}$ can be understood as the infinite

collection, or a collection of infinite membership, for being distinct. So there's quite a thing about collection, and just plain resort of objects.

This problem is actually the reason why we begin with Russell's paradox [2] in the first place. We would like to take a detour, and confront Russell's paradox for the moment (Gerstein). We start with a property P and assume that the property can be used to define a set, $\{x \mid P(x)\}$. Consider the set

Russell's paradox

$$S = \{A \mid A \text{ is a set and } A \notin A\}$$

Notice that some sets are not elements of themselves. The set of integers $\mathbb{Z}$ does not include the set itself. We obtain the paradox when we consider 'the set of all sets which are not member of themselves', or

$$R = \{\text{Sets } A \mid A \notin A\}$$

The question is, is R a member of itself? R cannot be a member of itself, but it must be, since it contains everything. This is a contradiction, hence R cannot be a set. But this explanation is lacklustre.

The argument of Russell's paradox is concerned of the set $\{x \mid x \notin x\}$. Is it a member of itself? We think the answer is no. What do you mean by $x \notin x$? What is x in this case? It seems that such thing does not even exist. Why? Because an apple cannot be justified so that it is not itself. Even when we regard that we can have collection of collections, the narrow view when you look at a collection, instead of later scale, is that now the collections inside the collection, is now called element instead. You cannot have an element to not belong to itself, simply because the statement does not make sense - you need to have a collection at the far side of the operation. This means that the whole statement is simply false, hence even S does not exist. Instead, we say we have two things, x and $\{x\}$. Inherently, $x \notin \{x\}$. The notation change - now $\{x\}$ is the collection of x, not just x itself. Then the formula $x \notin x$ is simply rejected, because it is false in interpretation. This indeed, surprisingly, leads to the theory of types, of which Russell himself postulated such. This creates slicing, of which divides things into set of elements, set of sets of elements, and more. In other word, an orderly fashion of types abstraction. The statement $x \notin \{x\}$ though, is wrong, since we now reduce the 'typing' down to element, and its container. One is a set, one is the container of such set. It is obviously wrong, because it's similar to asking if your apple in the bag, is not in the bad that contain such apple.

On the other hand, if we still accept the notion of the statement, then for $S = \{x \mid x \notin x\}$ is indeed true, and exists, because of the law of scaling and typing. This holds for the next case, $S \in S$?, and the answer is no, since it cannot contain itself, validly, within the typing of scale. So there is no universal set.

What we have done is to reject the existence of even S, thus invalidating the question itself; also, to prove that there is no universal set available. But there are several ways to do this, instead. One example is the treatment of such, so that we cannot create such arbitrary set. New sets can only be created via the above operations on old sets, plus replacement, which says that you can replace an element of a set with another element. This is an example of the treatment of set theory, following ZFC (Zermelo-Fraenkel) set theory, which was formed to counter the existence of Russell's paradox. Another argument, *The von Neumann-Bernays alternative*, also proved to be effective against such paradox, but retains the ability to have a universal set - in this case, is called as a 'proper' class. [3]

[2] Also called Russell-Zermelo paradox

[3] For more information on this debate, see the relevant information in *Herbert B. Enderton, Elements of Set Theory*, Gerstein's argument of the paradox, Plato Stanford's articles on this topic, as well as several introductory literature regarding the same problems.

2.3.1 An informal view on sets

The following informal treatment on the method of obtaining sets. Note that this is informal, and this is just a description, hence motivate certain understanding of sets.

First,we gather objects that are not themselves sets, but we want to have as members of a set, called *atoms*. Let A be the set of all atoms. We proceed to build up a hierarchy:

$$V_0 \subseteq V_1 \subseteq V_2 \subseteq \dots$$

of sets. We take $V_0 = A$, and construct upward, $V_1 = V_0 \cup \mathcal{P}(V_0) = A \cup \mathcal{P}(A)$, of all sets of atoms. The third level is hence also constructed the same way, for V_3 as $V_1 \cup \mathcal{P}(V_1)$. Hence, we have a recursive relation:

$$V_{n+1} = V_n \cup \mathcal{P}(V_n), \quad n \in \mathbb{N} \tag{2.13}$$

However, for this series of $\{V_0, V_1, \dots\}$, there is not enough sets included. For example, we do not have the infinite set $\{\varnothing, \{\varnothing\}, \dots\}$. To remedy this, we take the infinite union,

$$V_\omega = V_0 \cup V_1 \cup \dots$$

and let $V_{\omega+1} = V_\omega \cup \mathcal{P}(V_\omega)$. In general, for any α, $V_{\alpha+1} = V_\alpha \cup \mathcal{P}(V_\alpha)$. This goes on 'forever', for the definition of forever is implicit. A *fundamental principle* is the following: Every set appears somewhere in this hierarchy. That is, for every set a, there is some set α for $\alpha \in V_{\alpha+1}$. If, we aim for simplification, restrict the definition to **pure sets**, then the construction would be resulted in $V_{\alpha+1} = \mathcal{P}(V_\alpha)$, without the atoms.

2.3.2 Classes

In the previous discussion on the mishandling of abstraction, and by our informal image of the hierarchical way sets are constructed, there is no "set of all sets". This nonexistence of a set of all sets will become a theorem, provable of axioms. However, we can go without it and accept it as a conjecture, or fact.

Nonetheless, there is some mild inconvenience if we are to forbid the notion of the collection of all sets. What is the replacement, or at least what can we call it? There are two alternatives:

1. The *Zermelo-Fraenkel alternative* The collection of all sets need have no ontological status at all, and we need never speak of it. If tempted to do so, avoids it.

2. The *Von Neumann-Bernays alternative* The collection of all sets can be called a *class*. Similarly, any other collection of sets can be called a class. In particular, any set is a class, but some classes are too large to be sets. Informally, a class A is a class, if it is included in some level V_α of the hierarchy, hence is also a member of $V_{\alpha+1}$.

In advanced work in set theory, Zermelo-Fraenkel works better, profits from the simplicity of dealing with only set, instead of classes and sets. In usual literature, or at least in Enderton's *Elements of set theory*, the proceeding resources and axioms will follow Z-F alternative, and hence mentions none of the classes that are not sets.

2.3.3 Axiomatizations

For the most part of this book, we would just have a thorough justification of set theory as a whole, aside from the application, operation side of combining and acting on sets. As noted above, axiomatic set theory works fine for the purpose of axiomatization we might want to consider, such that Russell's paradox is not again reiterated. It is then obvious that we would like to take axiomatic set theory as our main focus, and hence write about it.

Axiomatization of set theory requires satisfying several principles, and also assurance of the existence of some basic sets.

The first one is the reiteration of the informal axiom used above.

Axiom 2.3.1 (Extensionality axiom). *If two sets have exactly the same members, then they are equal:*

$$\forall A \forall B \, [\forall x \, (x \in A \Leftrightarrow x \in B) \Rightarrow A = B] \tag{2.14}$$

Next, we have some axioms of basic sets encountered.

Axiom 2.3.2 (Empty set axiom). *There is a set having no members:*

$$\exists B \forall x \, x \notin B \tag{2.15}$$

Axiom 2.3.3 (Pairing axiom). *For any set u and v, there is a set having as members just u and v:*

$$\forall u \, \forall v \, \exists B \, \forall x (x \in B \Leftrightarrow x = u \vee x = v) \tag{2.16}$$

Axiom 2.3.4 (Union axiom, preliminary form). *For any set a and b, there is a set whose members are those sets belonging either to a or to b (or both):*

$$\forall a \, \forall b \, \exists B \, \forall x (x \in B \Leftrightarrow x \in a \vee x \in b) \tag{2.17}$$

Axiom 2.3.5 (Power set axiom). *For any set a, there is a set whose members are exactly the subsets of a:*

$$\forall a \, \exists B \, \forall x (x \in B \Leftrightarrow x \subseteq a) \tag{2.18}$$

For now, the union axiom can then be further strengthen. Later we will also mention the subset axioms, replacement axioms, infinity axiom, regularity axiom, and the axiom of choice. The set existence axioms can now be used to justify the definition of the symbol $\varnothing$.

Definition 2.3.1. *$\varnothing$ is the set having no members.*

The definition bestow the name "$\varnothing$" on certain set. We, however, must know by then that there exists a set having no members, and there cannot be more than one of them by the time we use the naming. The logical difficulties arise from introducing symbols when either there is no object for the symbol to name. The other set existence axioms justify the definition of the following symbols:

Definition 2.3.2. *For sets uv, a, b and their power set $\mathcal{P}(\cdot)$, we have the following:*

i. For any sets u and v, the pair set $\{u, v\}$ is the set whose only members are u and v.
ii. For any sets a and b, the union $a \cup b$ is the set whose members are those sets belonging either to a or to b (or both).
iii. For any set a, the power set $\mathcal{P}(a)$ is the set whose members are exactly the subsets of a.

As with the empty set, the existence axioms assure that the sets being named exists, and extensionality assures that the sets being named are unique. We can use pairing and union together to from other finite sets.

The set recently defined can be named by use of the abstraction notation:

$$\varnothing = \{x \mid x \notin x\} \tag{2.19}$$

$$\{u, v\} = \{x \mid x = u \vee x = v\} \tag{2.20}$$

$$a \cup b = \{x \mid x \in a \vee x \in b\} \tag{2.21}$$
$$\mathcal{P}(a) = \{x \mid x \subseteq a\} \tag{2.22}$$

From this, one might suggest the form

$$\forall t_1 \ldots \forall t_k \exists B \forall x (x \in B \Leftrightarrow \_\_\_) \tag{2.23}$$

should be true. However, as we have seen, some sentences of this form are false in the informal view. For example, $\exists B \forall x (x \in B \Leftrightarrow x = x)$ is wrong, since we know that there is no universal set. For a class A, we can at most said of whose members are those sets x such that $\mathsf{A} = \{x \mid \_\_\_\}$. In order for A to be a set, it must be included in V_α. In fact, it is enough for A to be included in any set c, for then of $c \subseteq V_\alpha$, the above follows and $\mathsf{A} \in V_{\alpha+1}$. This motivates the *subset axioms*:

Axiom 2.3.6 (Subset axioms). *For each formula* ___ *not containing* B*, the following is an axiom:*[a]

$$\forall t_1 \ldots \forall t_k \, \forall c \, \exists B \, \forall x (x \in B \Leftrightarrow x \in c \wedge \_\_\_) \tag{2.24}$$

[a] The expression follows, as $t_1, \ldots, t_k$ is list of sets involved in the formula, in conjunction with x.

In plain word, this axiom asserts that for any $t_1, \ldots, t_k$ and c, the existence of a set B whose members are exactly those sets x in c such that the formula ___ is true, then it follows automatically that B is a subset of c.

2.4 Ordered pairs and relations

Ordered pair are pretty popular. After all, its first appearance is from analytical geometry, which guarantees its role as the basic tool that one can find in their mathematical arsenal. According to the general principle of restricted variable, the ordered pair (x, y) is taken to be a certain set, but we don't particularly care about the detail of such set, except for the property that:

$$(x, y) = (a, b) \Leftrightarrow x = a \wedge y = b$$

Thus, we have $(1, 3) \neq (3, 1)$, or *order matters*, which is the definition to the name.

The notion of a correspondence or *relation*, and the special case of a mapping, or function, is fundamental to mathematics. A correspondence is a pairing of objects such that, given any two objects x and y, the pair (x, y) either corresponds or not. A particular correspondence (relation) is generally presented by the statement frame $P(x, y)$ having two free variables, with x and y corresponding if and only if $P(x, y)$ is true. Given any relation, the set of all ordered pairs (x, y) of corresponding elements is called its graph. There are no restriction for (x, y) to be strictly two elements, either - it can be two sets of elements, for what it takes. Similarly, it is not also restricted to ordered pair. The concept can be generalized to n-pair of $(x_1, x_2, \ldots, x_n)$, and their respective operation. One example that is fair standard is the *ternary relation*, where there is the 3-pair (a, b, c) satisfying certain condition or the relation. Another name of ordered pair and more is n-*tuple*.

Now a relation is a mathematical object, it is current practice to regard it as a set of some sort or other. Since the graph of a relation is a set of ordered pair, it is customary to take the graph to be the relation, in some sort of ways. Thus, roughly speaking *a relation (correspondence) is simply a set of ordered pairs (or more)*. If R is a relation, we say that x is related to y by R, denoted by xRy if and only if $(x, y) \in R$. We also say that x correspond to y under R. The set of all first element or the ordered pair is then called the relational *domain* of R, and is designated $\mathfrak{D}(R)$ or $\mathrm{dom}(R)$. The set of second elements is called the *range* of R. Thus,

$$\mathrm{dom}\ R = \{x : (\exists y)(x, y) \in R\} \quad \mathrm{range}\ R = \{y : (\exists x)(x, y) \in R\}$$

The *inverse*, R^{-1} of a relation R is the set of ordered pairs obtained by reversing those in R, then:

$$R^{-1} = \{(x, y) : (y, x) \in R\}$$

A statement frame $P(x, y)$ having two free variables actually determines a *pair of mutual inverse relation* $R\&S$, called the *graph* of P, as

$$R = \{(x, y) : P(x, y)\} \quad S = \{(y, x) : P(x, y)\}$$

A two-variable together with a choice of which variable is considered to be first might be called a *directed frame*. Then a directed frame would have a uniquely determined relation for its graph. The relation of strict inequality on the real number system $\mathbb{R}$ would be considered the set $\{(x, y) : x < y\}$, since the variables in $x < y$ has a natural order.

Additionally, we can loosely represent this ordered pairs and their set by using the notion of a *Cartesian product*. More specifically, the set $A \times B = \{(x, y) : x \in A \wedge y \in B\}$ of all ordered pairs with first element in A, and second element in B is the Cartesian product of the sets A and B. A relation is always a subset of $\operatorname{dom} R \times \operatorname{range} R$. If the fwo "factor spaces" are the same, we can use the exponential notation $A^2 = A \times A$.

2.4.1 Equivalence relations

Usually, every relation is best categorized by specifying their properties in choosing the graph. One of example where it satisfies reflexivity, symmetry, and transitivity to be elements of the ordered pair (x, y) is *equivalence relation*.

Definition 2.4.1 (Equivalence relation). *An **equivalence relation** on a set A is a relation C on A having the following properties:*

- *(Reflexivity) xCx for every $x \in A$.*
- *(Symmetry) If xCy then yCx.*
- *If xCy and yCz then xCz.*

The relation letter C is often omitted and replaced by the symbol $\sim$ for equivalence relation. Given an equivalence relation $\sim$ on a set A and $x \in A$, we define certain subset E of A, called *equivalence class* determined by x of the equation

$$E = \{y \mid y \sim x\}$$

Note that the equivalence class E determined by x contains x, since $x \sim x$. Equivalent classes have the following property:

Lemma 2.4.1. *Two equivalence classes E and E' are either disjoint or equal.*

Proof. Let E be the equivalence class determined by x, and E' determined by x'. Suppose that $E \cap E' \neq \varnothing$, let y be a point of $E \cap E'$. We then have to show that $E = E'$.

By definition, we have $y \sim x$ and $y \sim x'$. Symmetry allows us to conclude that $x \sim y$ and $y \sim x'$. From transitivity $x \sim x'$. If now w is any point of E, we have $w \sim x$ by definition, and so is $w \sim x'$. We conclude that $E \subset E'$. The symmetry allows us to also conclude that $E' \subset E$ as well, so that $E = E'$. □

2.5 Functions

The concept of function is perhaps one of the most fundamental, and easy to come by. Many times have I or someone else thought of something in a functional sense, converting their descriptions into somewhat of a function style. So, what shall we make of it in the context of mathematics?

For mathematicians, we *think* of function as being the assignment, given certain rules available, between the 'stuff'. To do this the mathematical way, we first have to define the rule.

Definition 2.5.1. *A **rule of assignment** is a subset r of the Cartesian product $C \times D$ of two sets, having the property that each element of C appears as the first coordinate of at most one ordered pair belonging to r.*

This definition is perhaps, quite restrictive. It requires you to have a 'one-way requirement' of the first set, for elements $c \in C$ to have no more than one connection to $d \in D$. That is not to say that there can't be two c_1, c_2 for one $d \in D$, so, a bit tricky (in high school, this is where the function vertical test is taken from). Thus, a subset r of $C \times D$ is a rule assignment if the following holds

$$[(c,d) \in r \wedge (c,d') \in r] \implies [d = d'] \quad c, c' \in C, d \in D \tag{2.25}$$

We then think of r as assigning to the element $c \in C$ the element $d \in D$ in the way that $(c,d) \in r$.

Now, we also think of the spaces contained all c, and the space contain all d that satisfies said assignment. I mentioned the fact that there is no stopping the assignment from having two c for one d, or even more. Hence, the set of all c might be drastically different given the chance (for example, remember the domain of $\sin(x)$ - that is not going anywhere). Hence, we have the notion of **domain** for subset of C consisting all c of r, and **image set** for the respective d. Formally:

$$\begin{aligned} \text{domain } r &= \{c \mid \exists d \in D : (c,d) \in r\} \\ \text{image } r &= \{d \mid \exists c \in C : (c,d) \in r\} \end{aligned}$$

Given a rule of assignment r, this notion of domain and image gives absolute determinism for an assignment. Now we can finally define a function.

function

Definition 2.5.2. *A **function** is a rule of assignment r, together with a set B that contains the image set of r. The domain A of the rule r is also called the **domain** of the function f: the image set of r is also called the **image set** of f; and the set B is called the **range** of f. If f is a function having domain A and range B, we express this fact by writing*

$$f : A \longrightarrow B$$

which reads as f is a function from A to B.

It is also customary to visualizes f as a geometric transformation physically carrying points of A to points of B. If $f : A \to B$ and if a is an element of A, we denote by $f(a)$ the unique element of B that the rule determining f assigns to a, it is called the **value** of f at a, or sometimes the image of a under f. For completion purpose, we also give the definition of the image set.

image (function)

Definition 2.5.3 (Image). *If $f : A \to B$ and $U \subseteq A$, then $f(U) = \{f(u) : u \in U\}$. Then $f(A)$ is the image of A. We denote this by $Imf = f(X)$, of which the set Imf is called image of mapping f.*

Using this, we can specify functions with more rigours, and more. Though, if it is totally clear, we can often opt for the removal of the specification of the domain and range - that is, telling $f(x) = x^3 + 1$ without having to specify that it is a function $f : \mathbb{R} \to \mathbb{R}$. Though, this kind of *restriction* on domain and domain specification is what will adequately describe a function. In the same way, we can construct new function, or partition them and stick them together, by restricting their domain, and subsequently, their image set.

Definition 2.5.4 (Restrictions). *If $f : A \to B$ and if A_0 is a subset of A, we define the* ***restriction*** *of f to A_0 to be the function mapping into B whose rule is:*

$$\{(a, f(a)) \mid a \in A_0\}$$

This is denoted by $f \mid A_0$, reads as f restricted to A_0.

Restricting the domain of a function and changing its range are two ways of forming new function from an old one. Another way is to form the composite of two functions. composition

Definition 2.5.5 (Composition). *Given function $f : A \to B$ and $g : B \to C$, we define the* ***composite*** *$g \circ f$ as the function $g \circ f : A \to C$ defined by the equation $(g \circ f)(a) = g(f(a))$. Formally, $g \circ f : A \to C$ is the function whose rule is*

$$\{(a, c) \mid \forall b \in B \; s.t \; f(a) = b \wedge g(b) = c\} \tag{2.26}$$

We often picture this as a two-step physical movement - first from a to $f(a)$, then from $f(a)$ to $g(f(a))$. This will ultimately also work with various chains, 4-chain or n-chain of continuous function composition. Note that, however, we have the *tail restriction* for composition, that is, the range of f must be equals to the domain of g for the composition to happens. In matrix theory (or linear algebra in simplicity), this is also quite the case, even though it is written in a more compact matrix form (matrix is just the notation, unfortunately).

With that said, there are many characteristic one can use to further categorize functions into classes of their respective transformation. Given that we are only restricted to *one y for one x*, there are many forms of mapping one can consider. So, how many can there be?

Definition 2.5.6 (Types). *There are three types of function, which is injective, bijective and surjective. We define those as followed.*

1. *Injective: $f : X \to Y$ is said to be injective if it "hits" everything at most, once:*

$$(\forall x, y \in X) f(x) = f(y) \implies x = y$$

2. *Surjective: $f : X \to Y$ is said to be surjective if it "hits" everything at least once:*

$$(\forall y \in Y)(\exists x \in X) f(x) = y$$

   If we can call it with the later notation, it is analogous to $Imf = Y$.

3. *Bijective: A function is bijective if it is both injective and surjective (hits everything exactly once). Generally,*

$$\forall y \in Y, \exists! x \in X \; s.t. \; y = f(x)$$

Composition of injective mappings are an injective mapping, so do surjective, thus we also have bijection mappings to have bijective compositions. And by definition, f is surjective iff $f(A) = B$, meaning exactly that "it hits all the domain".

Aside from typical separated set function, $f : X \to Y$, we would also be interested in the function that acts on the set, and go to the set itself. By some intuitive thought, the one that can be responsible for this one-to-self mapping, is called the *identity function*, and is characterized only by mapping the element to itself. [4]

Definition 2.5.7 (Identity). *The identity map $id_A : A \to A$ is defined as the map $a \mapsto a$.*

Suppose $f : X \longrightarrow Y$ is a bijective mapping from X to Y. Then for each $y \in Y$, there exists one and only $x \in X$ so that $f(x) = y$. Then we have a mapping $g : Y \longrightarrow X$ defined as

$$\forall y \in Y, x \in X : g(y) = x, f(x) = y \Longrightarrow g \circ f = i_X, f \circ g = i_Y$$

This notion is generalized to be called **inverse mapping/function**, of which we will break down even, into left and right inverse. Generally, if B_0 is a subset of B, we denote by $f^{-1}(B_0)$ the set of all elements of A whose images under f lie in B_0, which is called the *preimage* of B_0 under f - or the inverse image of B_0. Formally, we get the formula

preimage

$$f^{-1}(B_0) = \{a \mid f(a) \in B_0\}$$

There may be, clearly, no points $a \in A$ whose images lie in B_0. In that case, we say that $f^{-1}(B_0)$ is empty.

Definition 2.5.8 (Right and left inverse). *Given $f : A \to B$, a left inverse of f is a function $g : B \to A$ such that $g \circ f = id_A$. A right inverse of f is then a function $g : B \to A$ such that $f \circ g = id_B$.*

Theorem 2.5.1. *The left inverse of f exists iff f is injective.*

Proof. If the left inverse g exists, then $\forall a, a' \in A$, we have $f(a) = f(a')$ implies that $g(f(a)) = g(f(a')) \implies a = a'$ therefore f is injective.

if f is injective, we then construct g as

$$g : \begin{cases} g(b) = a & \text{if } b \in f(A), f(a) = b \\ g(b) = \text{anything} & \text{otherwise} \end{cases}$$

Then g is a left inverse of f. □

2.6 Product sets, index notation

[4] By extension, any given chain of composition $g_1 \circ g_2 \circ \dots g_n$ can be called identity if it satisfies the mapping of $g_1 \dots g_n : A \to A$ for the domain and image is both A.

Chapter 3. Linear algebra

Linear algebra is, perhaps, the golden horse of mathematics. In one way or another, it stands on par with calculus in the list of "top mathematical stuff that I don't want to learn" for almost everyone, but also the list of "top mathematical topics that are so necessary that I would have to read it either way.". Indeed, of mathematics, linear algebra is perhaps the most important field, yet requires not so much bargain on the mind to study something that is full of proof but not too abstract.

Before learning something, it is best to ask the question of how useful it is for us to learn it, especially when the book is already bloated with contents, and the reader might as well only need to have a hand-waving knowledge of linear algebra. It is well-known of certain fact that linear algebra is particularly "hard". In some way or another, that is true, mostly because of the way it is interpreted and approach, or how the materials are presented. For example, a later topic that we will perhaps include (if there are times and spaces) is the concept of eigenvalue and eigenvectors of its theorem or properties. To prove such, most of the time (or most books) must define determinant to further their work for such proof. This is difficult, nonintuitive, and often derived without any sort of motivation just like how matrix is defined yet there exists no talks about what it actually represents every time you put your pen down and write the 3×3 matrix (somehow, it is a *linear transformation*). So to effectively learn something, we have to sort such things out, to worth the time that we will eventually spend learning this, and I myself writing this.

Linear algebra in a rough sense can be said to be the study of functions on vectors. More specifically so, then it is the encoding, the language and methods of the general scaling of a high-dimensional space, depends on how you define dimensional, and is restricted to a class of such space called *vector space*. Indeed, one feels pretty much natural extending objects to finite many dimensions (the case of infinitely many dimensions is studied in a sense, of *functional analysis*). Other than that, it is also can be said to explore interactions in *linearity sense* - of which you can only displace something on a straight line (I don't care if that straight line is oriented in any way, just get it straight), or just either shrink or extend it, again, linearly with respect to a straight line. In application sense, any system with a particularly large amount of dependencies, parameters, variables, et cetera, will benefit from the usage of linear algebra. Such is also said of encoding transformation, functions in which changes the state of certain system of interest, can also be represented in linear algebra.

One of the main thing that is also good of linear algebra is that, most of the questions posed in an introductory course, can be answered in said introductory course. That is, it is relatively stable, unlike in number theory, or so, where questions as simple as the twin primes problem takes forever to solve. Or rather, linear algebra is *thoroughly understood*, which makes it a very powerful tool for any kind of problems that can be represented neatly by it. for our cases, those applications will be apparent throughout the text. In fact, I recommend you to skip this section in a kind of binge-reading way - just browsing through. Whenever you see the application or

usage, going back.

With that said, I will approach the problem of linear algebra in a fairly different way. Or rather, not so much I guess. Because in the field we are researching, and the topic by itself has many utilizations and analogous notations or conventions on objects of linear algebraical properties, we would separate our investigation into two parts: the part on the *representation*, encoding, the data structure in which we will represent our object; and the actual linear algebra itself.

3.1 Notation and structure

Mathematics works in an encoding way. That is, even though we say that mathematics is abstract, as perhaps certainly it can take every values possible in existence of permission, as for all things to be spelled out, they need something to represent and express it [1]. In linear algebra, this notation is what trivially presented as the **array form**, and the special case of the array form used in linear algebra is **matrix**.

3.1.1 Array form

Consider a large number of data, normally represented as a list of $x, y, b, qr, \dots, m$. This list can be either ordered or not. Enforce the specific encapsulation and positional ordering, we have the one-dimensional array: [2]

$$\begin{bmatrix} x & y & b & qr & \dots & m \end{bmatrix}$$

An array has the positional order of index for an index set I. Conventionally, this order is presented from left to right. So one can expect an array A to take the form

$$A = \begin{bmatrix} a_1 & a_2 & a_2 & \dots & a_n \end{bmatrix}$$

For each element now is indexed by the subscript $i = 1, 2, 3, \dots, n$. if you want, it is similar to a mask, such that for all $a \in A$, we have done a masking of data such that we have the pair $(1, x), (2, y), (3, qr)$ and so on for each a of the respective ordering. Similarly, we can also present the two-dimensional array by adding another dimension for expansion:

$$A' = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix} \tag{3.1}$$

In which now (i, j) is limited in the range (m, n) of the index. Even though we write this, normally, there are no restriction toward the absolute filling of the array. There, we have the definition of the *uneven array*, which is specified to have different row counts and hence,

[1] This is generally true. Even for the notion of infinity is often misunderstood in such case, as for the restriction of even the human brain there can be no infinite imagination, or infinite representation of an endless line of numbers. There are, usually, thought experiences using the notion of infiniteness, implicitly defined, and often spoke of ambiguity rather than it can be grounded in rigorous manner. On the other side of our argument, representation, encoding is everywhere to work on such abstract and arbitrary concept - the process of writing something down is one of said action, in the language of writing.

[2] We use `bmatrix`, the matrix form notation. However, usually, it is also presented by the normal square bracket with comma, such as:

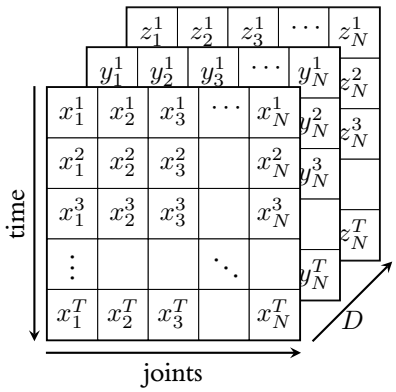
$$\mathrm{Ar} = [a_1, a_2, b_1, b_2, c_1, c_2, \dots]$$

different subsequently column counts, if we are to treat array just like a table. However, for generality, we can assume all of those uneven array are actually even, well-shaped array with null values.

Even though we say dimension, the notation orientation does not care much of your general ideas.

Here, the term *dimensional* or then *dimension* refers to the total index set specified to describe the positional ordering of elements inside an array structure. Before we continue, let's settle the convention. For each array, the index is written in the subscript of individual components. So, for one-dimensional, it's just a_i, for two, it is a_{ij}, three then a_{ijk}, and so on. All elements in an array A, in the abstract form will take similar element, the as in the array notation. So with index and abstract form, we will just have a_i and so on, not a_i, b_j for $i \neq j$ everywhere. In the specified form (of values, for example), then the index is hidden, and the notation switch to their values. However, to specify or access certain elements, you will need to tell the index. Another thing to note is that simply speaking, geometrically, the dimension is simply the orthogonal axis, like in Cartesian coordinate system usually, that can be used to specify the index or positional order of particular element in the structure.

Coming back to the dimension, this implies we can get to n-dimensional array, and that is perhaps correct. In a more mainstream machine learning language, this is called a **tensor**[3]. An example of a three-dimensional array can be represented as a cube of values, for index $I^3 \subset \mathbb{N}^3$, of the 3-tuple (i, j, k) for each element. Though, this kind of notation can be presented in a lot of... well troublesome way, even if they are the same in essence. Mostly come down to notation abuse and illustration, though.



For our special case, or a matrix, it is the two-dimension case. Most of the operations acting on matrix can be indeed generalized, though it is most of the time more effective to define it on such. And, without being said, matrix's operations and functions, as well as its uses, are more rigorous perhaps, than the others n-dimension, and from array analogue itself.

We will now jump to the more interpretable object that linear algebra will use. While array representation has its own meaning and usage, it is often poorly equipped to be used and utilized in a mathematical sense, simply because, as we said, it is a data structure only with no added purpose or interpretation of its structure. The next section, we will deal with two fundamental objects of linear algebra - the *vector* and *matrix* object.

[3]In case of ambiguity, this tensor is not the tensor in tensor calculus, or any rigorous manner of tensor analysis. It is only a name to talk of multidimensional array of data only. The structure itself is again, only data structure - there are no intrinsic operational meanings embedded.

3.2 Vectors

For a mathematical object, there are components to specify it details or particular structure. In the context of linear algebra, we deal with the variable spaces, where each mathematical object can take place. Usually, the structure that give rises to such space, which differs from case-to-case basis of different fields, is called a **mathematical structure**, in an informal sense. Linear algebra then, deals with the question of working on multi-descriptive objects, where its mathematical structure is considered of the **specification** that is needed to describe the mathematical object respectively. For example, by then, we can roughly get the mathematical object of real number, in the structure $\mathbb{R}$, to be a zeroth-order object by linear algebraical means. That is, it can be specified using a singular "point" - its own quantification value. Then, for complex number of the form $x = a + bi$, then it can be then considered a first-order object in such sense, simply because its parameter can be, informally, listed as one row or one column, and each contains the previous zeroth-order component. Such zeroth-order object, if they are of a specific field, for example, the field of real number $\mathbb{R}$, then we have a special name for it: a

scalar

scalar on the field $\mathbb{R}$. Then, for a first-order complex number space $\mathbb{C}$, we can then call it to be the two-dimensional space on the field $\mathbb{R}$. By such, we have considered the singular complex

vector

space $\mathbb{C}$ to contains **vectors** of which takes zeroth-order component over $\mathbb{R}$ - two of them, that is.

The more nominal example that one can think of, and will be implied throughout the chapter, is the description of a *directed object*, or a *directed path*. For example, the direction of the displacement in space of a plane, the acceleration accompanied by a car, the direction of rotation of a geometrical shape, or the direction in which the gradient is, for the landscape of a function. Such cannot be described, usually, by one singular value. It needs to be of something with magnitude; and some expression of direction. This is fulfilled by vector itself.

If we think only vectors and scalar as mathematical object but on the side of data representation, it is rather easy to see the intuitive necessity of such. Many things require more than just one value, or numerical representation to defines its mathematical form. Such can be said of many physical notions, from forces - if your world is not one-dimensional, that is; or velocities, accelerations, positions, directions, et cetera. Any such entity will require more than one specification, and often, it can be reached with the simple composition of many single values together. This, form the notion of vector as we have above. Additionally, we also note that again, we would be dealing with analytical, numerical objects. So, vectors and the like will be plenty of operations and actions on it, as well as the nuance accompanied.

It is then, rather more detail for once, to talk of the more reduced notion of a vector. The more physical, deliberate definition hence specifies the vector by its apparent properties - an object of both *magnitude* and *direction*. Do note that this definition can be extended for the other interpretations, as we see below:

Definition 3.2.1 (Vector). *A (nonzero) **vector** $\vec{v}$, or $\mathbf{v}$, is a quantity or object with both magnitude and direction. It is also expressed by its vector components, of which we denote as $\vec{v} = (v_1, v_2, \ldots, v_n)$, for all v_i, $i = 1, \ldots, n$ as the ith component of $\vec{v}$. The requirements that a vector and operations on it must satisfy, and the space to specify such vector constitute a **vector space**.*

*Geometrically, a vector $\mathbf{v}$ is specified also by its segment drawn from a point P (called **initial point**) to point Q (called **terminal point**). Its magnitude is then the length of the segment, denoted by $||\boldsymbol{v}PQ||$, and its **direction** is the same as of the directed line segment. The **zero vector** is the generalization of a point, denoted $\mathbf{0}$.*

We will leave the vector space and structure to later section on the mathematical structure of linear algebra itself. For now, we would consider the vector in its sense of a data structure,

somewhat. Our notation of a vector is then pretty similar to the first-order array, or one-dimensional array in previous section. A vector $\mathbf{v}$ is hence presented as:

$$\mathbf{v} = \begin{pmatrix} v_1 & v_2 & v_3 & \dots & v_n \end{pmatrix} \tag{3.2}$$

or, in a vertical form,

$$\mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_n \end{pmatrix} \tag{3.3}$$

Notationally, they are called the **row vector** and the **column vector**. For now the differences are irrelevant - they specify the same object, and the same vector. The subtleties only comes when we consider certain interpretation and structure with the 'row' and the 'column' - which hints at matrix.

Under such context, the *dimension* is the length of the component description, or the number of component that a vector has. If a vector is in the special case where its value is zero (we work on maths, and it is numerical), then we say that it is t*he ith-flatten vector over n-space*, where n is its dimension.

For vectors, there are certainly some special vectors. The **zero vector**, denoted $\mathbf{0}$ or $\vec{0}$, is the conception of singular free point, with respect to any particular reference coordinate frame. That is, a singular special point called the **origin** that has zero magnitude and zero direction. Similarly, there is the **unit vector**, denoted $\mathbf{1}$ or $\vec{1}$, or $\hat{i}, \hat{j}, \dots$ of lower-case characters, is defined such that $|\mathbf{1}| = 1$ for any given direction. A collection of unit vectors by specific conditions form the **coordinate system**.

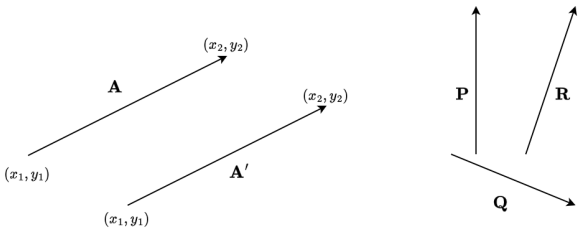


Figure 3.1: An illustration of a vector in two-dimensional vector form in endpoints representation, and directional-magnitude representation.

There is a subtle detail in considering the notation for vector. One can see that the notation we used to represent vector is oddly similar to the notation used for n-dimensional points in an n-space, for example, the Euclidean space $\mathbb{R}^3$, which each point can be specified to be (x, y, z) along each axis. Indeed, this is troublesome: if they are mixed up together, it would be difficult to distinguish the notation for coordinate system and one for the vector representation, especially in case of complex manipulation. We can offer certain solutions to this problem. The first one, is to re-interpret coordinate in a space just as similar as vectors - every point is actually a vector starting from the origin, travelling to the point itself. In such sense, a vector is considered like a path composition with respect to each axis, representing the total component path taken to specify the location of such point. Another way to think about a vector is that it is a *compressed form* of two point specification - the *starting point* A_s and the *endpoint* B_e, though often

compressed form

we would call them all together as endpoints. Then, a vector is the reduced representation for all pairs of endpoint such that, the vector is defined to be the final absolute path - or in terms of *calculus of variations* sense, the shortest path - a straight line - toward the two endpoints. Any straight line, with direction that satisfies the component path from one endpoint to another is indeed, that one vector; or simply speaking. The direction is characterized by indeed, the order of the path, either from A to B or B to A, and also based off their intrinsic location. In such representation, we separate the endpoints from the vectors - an endpoint has no direction and is dimensionless, stationary, while a vector specifies a path from two endpoints together, of the total difference between the two endpoints. In the end, though it is based in the interpretation of the case study, and what structures are considered. In a more advanced situation, one can also specify a point scalar as an element of a manifold, and a vector as an element of the tangent space to such manifold.

Two vectors are equal if the components used to specify them are the same. Geometrically, it means they have the same magnitude and direction. [4]

Definition 3.2.2. *Two nonzero vectors are* **equal** *if they have the same magnitude and the same direction. Any direction with zero magnitude is equal to the zero vector.*

Although we defined zero vector and equal magnitude, we still do not know for sure what can be used to make such definition operational. To tackle this, we have to define the notion of vector length, in the respective reference frame, and of respective measure, for example, in $\mathbb{R}$ or $\mathbb{Z}$.

Recalling that the distance between two points $P(x_1, x_2, x_3)$ and $Q = (y_1, y_2, y_3)$ in the Euclidean space $\mathbb{E}^3 = \mathbb{R}^3$[5] is such that

$$d(P, Q) = \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2 + (y_3 - x_3)^2} \tag{3.4}$$

Using such, we define the following as the vector **magnitude**:

Theorem 3.2.1 (Vector magnitude). *For a vector* $\mathbf{v} = (a, b, c)$ *in* $\mathbb{R}^3$*, the magnitude of* $\mathbf{v}$ *is:*

$$||\mathbf{v}|| = \sqrt{a^2 + b^2 + c^2} \tag{3.5}$$

This can be scaled up or down for any given n.

Proof. By definition of vector, its magnitude is the length of the segment. Arbitrary specification of end-points (endpoint-independent) property gives (a, b, c) the component length with respect to each dimension. The rest follows. Also notice we can prove by exhaustion for all permutation of choice of a, b, c. □

[4] Settle aside the expression of component for a vector, usually, we can get a lot of the properties and actions on vector using only their intrinsic, geometrical interpretation, that is, the magnitude-direction expression for a vector (as illustrated in Figure 3.1). Previously, we have said that anything of certain magnitude and direction, or path, between two endpoints, is the same vector. As we are concerned of here, it is true that only the magnitude and direction of the vector are significant; hence consequently, we regard vectors with the same magnitude and direction as being equal irrespective of their position. This is true for the geometry of a vector, as we guarantee.

[5] This equivalency is more on convention than not. Some books will say it being the same, some books will regard Euclidean space of higher-dimension than real space. However, a property holds in those definitions is that Euclidean space is defined without special origin. Furthermore, consider the pure vector space $\mathbb{R}^3$. If we grant it the inner product, it becomes Euclidean since one of the requirement for a space to be Euclidean, is for Euclid's axiom to hold. That is why sometimes we call Euclidean space the special case of real vector space.

We have been introduced to the notion of vector. Naturally, a question should arise: what can we do with it, and with many of it? Fortunately, we have options for such aspect. For vectors, operations to be investigated would be two operations, the addition operation and the scalar multiplication for the time being. Let's begin with addition.

Theorem 3.2.2 (Vector sum). *The* ***sum*** *of vectors* $\mathbf{v}, \mathbf{w}$*, denoted by* $\mathbf{v}+\mathbf{w}$ *is obtained by translating* $\mathbf{w}$ *so that its initial point is at the terminal point of* $\mathbf{v}$*, the initial point of* $\mathbf{v}+\mathbf{w}$ *is the initial point of* $\mathbf{v}$*, and the endpoint is the new terminal point of* $\mathbf{w}$*.*

In component form, we can get,

$$\mathbf{v}+\mathbf{w}=\begin{pmatrix}x_1\\x_2\\\vdots\\x_n\end{pmatrix}+\begin{pmatrix}y_1\\y_2\\\vdots\\y_n\end{pmatrix}=\begin{pmatrix}x_1+y_1\\x_2+y_2\\\vdots\\x_n+y_n\end{pmatrix} \tag{3.6}$$

This operation only happens if they have the same dimension. However, we can mitigate this partially by the following convention.

Lemma 3.2.3. *If* $\mathbf{v}$ *and* $\mathbf{w}$ *acts on each other where* $\dim(\mathbf{v})=n\neq m=\dim(\mathbf{w})$*, then: If* $n>m$*,* $\mathbf{w}$ *is extended of zero value in the additional dimension; if* $n<m$*,* $\mathbf{v}$ *is applied of the same.*[a]

[a] We have not defined the vector space yet at this point. However, based on the notion of dimension described before, and considering the space of which we put on the dimension notion in to specify the vectors, this result is palpable. Nevertheless, if we can define a vector space, then it means either one contains the **vector subspace** of the other.

The behaviour of this sum is intuitive - linking directed line altogether with each other. Because of this, we will have the following observation.

Theorem 3.2.4 (Parallelogram law). *The sum of two vectors* x *and* y *that acts at the same point* P *is the vector beginning at* P *that is represented by the diagonal of parallelogram having* x *and* y *as adjacent sides.*

Question 3.2.1. *Can you realize the parallelogram law in analytical terms of its coordinate?*

This can be illustrated in Figure 3.2. Do note that it is more or less, a not so trivial point. Aside from additive operation, vector can also be multiplied, albeit restricted to multiplying

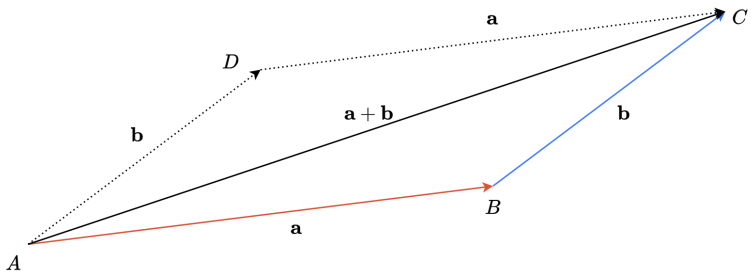


Figure 3.2: The **parallelogram law** for vector addition of two vectors x and y on adjacent side.

with a scalar λ only [6]. For this, of a vector $\mathbf{v}$, we define $\lambda \cdot \vec{v}$ as:

$$\lambda \mathbf{v} = \lambda \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} \lambda x_1 \\ \lambda x_2 \\ \vdots \\ \lambda x_n \end{pmatrix}, \quad \lambda \in \mathbb{R} \tag{3.7}$$

The scalar is multiplied, component-wise, for each element of the vector. Non-algebraically, this is effectively the action of scaling with respect to direction of a given vector by λ amount(s). By now, you might also have noticed that we switch between analytical (algebraical, if you like) and geometrical description, and this is not unnatural, and sometimes offer understanding that complements the other method.

Definition 3.2.3 (Scalar, multiplication). *We define a* **scalar** *as a quantity equal to a single component, and geometrically a point*[a]*. For a scalar k and a nonzero vector $\mathbf{v}$, the* **scalar multiple** *of $\mathbf{v}$ by k, denoted by $k\mathbf{v}$, is the vector whose magnitude is $|k|||\mathbf{v}||$, same direction as $\mathbf{v}$ for $k > 0$ and reverse if $k < 0$, and is the zero vector if $k = 0$. For the zero vector $\mathbf{0}$, we define $k\mathbf{0} = \mathbf{0}$ for any scalar k.*

[a] Interesting fact. There are two entries on which the word scalar was used. The word scalar derives from the Latin word scalaris, an adjectival form of scala (Latin for "ladder"). The first use is in François Viète's Analytic Art (In artem analyticem isagoge) (1591), where it is written that "Magnitudes that ascend or descend proportionally in keeping with their nature from one kind to another may be called scalar terms.". Another one is by William Rowan in the 19th century, to convey the sense of something that could be represented by a point on a scale or graduated ruler.

Using our analytical formalism, we can try to prove those operations. Indeed, let's see how that might potentially work.

Theorem 3.2.5. *Let $\mathbf{v} = (v_1, v_2, v_3)$, $\mathbf{w} = (w_1, w_2, w_3)$ be vectors in $\mathbb{R}^3$, let k be a scalar. Then:*

1. $k\mathbf{v} = (kv_1, kv_2)$.
2. $\mathbf{v} + \mathbf{w} = (v_1 + w_1, v_1 + w_3)$.

Proof. (a) We would be using basic geometry for such purpose of assumptions for further proofs. Without loss of generality, we assume that $v_1, v_2 > 0$ (the other possibilities are handled in a similar manner). If $k \neq 0$, then (kv_1, kv_2) lies on a line with slope $kv_2/kv_1 = v_2/v_1$, which is the same as the slope of the line on which $\mathbf{v}$ (and hence $k\mathbf{v}$ lies), and (kv_1, kv_2) points in the same direction on the line as $k\mathbf{v}$. Furthermore,

$$|(kv_1, kv_2)| = \sqrt{(kv_1)^2 + (kv_2^2)} = \sqrt{k^2v_1^2 + k^2v_2^2} = \sqrt{k^2(v_1^2 + v_2^2)} = |k|\sqrt{v_1^2 + v_2^2} = |k|||\mathbf{v}|| \tag{3.8}$$

This indicates that they have the same magnitude and direction. This concludes the proof.

(b) Without loss of generality, we assume that $v_1, v_2, w_1, w_2 > 0$. We see that when translating $\mathbf{w}$ to start at the end of $\mathbf{v}$, the new terminal point of $\mathbf{w}$ is $(v_1 + w_1, v_2 + w_2)$, so by definition of $\mathbf{v} + \mathbf{w}$ this must be the terminal point of it. This concludes the proof. □

3.2.1 Properties of vectors

With a lot of objects in mathematics, they can always be described in mainly two ways: either analytically (algebraically) or geometrically. Let's investigate the geometrical view first.

[6] Since this is not a textbook, you should have noticed by now, if you have educations of linear algebra beforehand, that there exists the **vector product** of the special case of matrix multiplication. However, of such is usually not the standard interpretation of vector, and is rather specified with meanings than abstracted notion — that is to say that the scope of the 'vector' considered in this section is not that wide, yet.

Interpretation of references

A vector can be considered to be a *free object* in the geometrical space. That is, it is not bounded by coordinates likes points, or geometrical structure embedded within the space by specification. Rather, it is free in the sense that the vector is similar to itself, by means of linear translation throughout the geometrical space, for example, moving the vector left or right, up or down. Analytically, it means that a vector description refers to its designed specification, and not positional descriptions. What does this mean? I would want to say it in terms of some rigours, but certainly, it depends on the interpretation, at the foremost aspect, but that would be pretty much useless. For my own understanding, one can regard the description of a coordinate (x, y, z) and a vector $x(x_1, x_2, x_3)$ by some subtleties, independent or dependent on the coordinate basis or not. A coordinate is dependent of the coordinate system, or the space specification, or the descriptors, because its component depends on each of the component space. Or, rather, it depends on the origin where everything is referenced from. Vector, on the other hand, use the notation a bit differently. Sure, it is still the path, or rather, if we are to treat it similarly to the indexing of an array, then it is similar to the array index to recognize an object's position. However, the descriptor is not taking positions or index, it is taking the relatively speaking, *path, length* instead. Or rather, it is the compression of two values, just as we spoke up there. So each vector is actually representing:

$$\mathbf{x} = (x_1, x_2, x_3, \dots, x_n) = (x_{1e} - x_{1b}, x_{2e} - x_{2b}, \dots, x_{ne} - x_{nb}) \tag{3.9}$$

where the subscript ie, ib is the ith component's end point and begin point. Now, I have to admit I don't know how to get this to be further than it is. Then, we would have the coordinate system using the pretty much special case of this system, [7]

$$x = (x_1, x_2, x_3, \dots, x_n) = (x_{1e} - 0, x_{2e} - 0, \dots, x_{ne} - 0) \tag{3.10}$$

Basic vector algebra

Moving on, for basic vector operations, we have the following.

Theorem 3.2.6 (Vector algebra). *For any vectors* $\mathbf{u}, \mathbf{v}, \mathbf{w}$*, and scalar* k, l*, we have:*

(a) $\mathbf{v} + \mathbf{w} = \mathbf{w} + \mathbf{v}$ *(Commutative Law)*

(b) $\mathbf{u} + (\mathbf{v} + \mathbf{u}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$ *(Associative Law)*

(c) $\mathbf{v} + \mathbf{0} = \mathbf{v} = \mathbf{0} + \mathbf{v}$ *(Additive identity)*

(d) $\mathbf{v} + (-\mathbf{v}) = \mathbf{0}$ *(Additive Inverse)*

(e) $k(l\mathbf{v}) = (kl)\mathbf{v}$ *(Associative law for scalar multiplication)*

(f) $k(\mathbf{v} + \mathbf{w}) = k\mathbf{v} + k\mathbf{w}$ *(Distributive law)*

(g) $(k + l)\mathbf{v} = k\mathbf{v} + l\mathbf{v}$ *(Distributive law)*

Proof. Proved using algebraic manipulation or geometrical proofs. □

Notice that using the commutative law, we gain the parallelogram law proof, since it permits the motion of attaching different vector to the end of each other. For n vectors apparent, any 2-permutation resolves the same theorem.

[7] So, yeah, same notation, different interpretation. Especially when you consider that you can, and indeed would likely want to write them down together. Though, because of this, most of the time, we would consider them to be relatively the same, because as we said, we can translate them down to anywhere, as long as the descriptor, which specifies the direction, magnitude, et cetera, stays the same - which includes the origin point.

The 'missing' operations

As far as elementary operations are concerned, we notice some operations that 'seem to be missing'. That are the *vector-vector multiplication* and *scalar-vector addition*. We can multiply and add scalar to scalar, multiply scalar to vector, add vector to vector, but not the other two.

3.3 Matrix

Matrix is the special type of array that is used in construction of linear algebra. Specifically, it is array, in special consideration, and added structures. Mathematically, it is defined as followed.

Definition 3.3.1 (Matrix). *A matrix A is a two-dimensional array, with dimensions' size of $m \times n$ for m is the horizontal dimension (the column) and n the vertical dimension (the column).* [a]. *For each element $a \in A$ of matrix A, it is indexed by a_{ij} for $1 \leq i \leq m$ and $1 \leq j \leq n$. It is represented as:*

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{pmatrix} \tag{3.11}$$

[a] This is just a normal convention.

The bracket is pretty much cosmetic, though, to distinguish from the use of array and matrix, we will use $[\cdot]$ for array and $(\cdot)$ for matrices.

Given such, what can we do with matrices? First, we can *add* them together; given A, B being two matrices, then $A + B$, the *matrix addition* is defined as:

$$A + B = \begin{pmatrix} a_{11} + b_{11} & a_{12} + b_{12} & a_{13} + b_{13} & \dots & a_{1n} + b_{1n} \\ a_{21} + b_{21} & a_{22} + b_{22} & a_{23} + b_{23} & \dots & a_{2n} + b_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} + b_{m1} & a_{m2} + b_{m2} & a_{m3} + b_{m3} & \dots & a_{mn} + b_{mn} \end{pmatrix} \tag{3.12}$$

Furthermore, *scalar multiplication* to a matrix is also available, and λA is defined as:

$$\lambda A = \begin{pmatrix} \lambda a_{11} & \lambda a_{12} & \lambda a_{13} & \dots & \lambda a_{1n} \\ \lambda a_{21} & \lambda a_{22} & \lambda a_{23} & \dots & \lambda a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda a_{m1} & \lambda a_{m2} & \lambda a_{m3} & \dots & \lambda a_{mn} \end{pmatrix} \tag{3.13}$$

For multiplication, the simplest operation that can be realized from component-based multiplication is the **Hadamard multiplication** or component based multiplication, denoted as $A \odot B$, defined as:

$$A \odot B = \begin{pmatrix} a_{11} \odot b_{11} & a_{12} \odot b_{12} & a_{13} \odot b_{13} & \dots & a_{1n} \odot b_{1n} \\ a_{21} \odot b_{21} & a_{22} \odot b_{22} & a_{23} \odot b_{23} & \dots & a_{2n} \odot b_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} \odot b_{m1} & a_{m2} \odot b_{m2} & a_{m3} \odot b_{m3} & \dots & a_{mn} \odot b_{mn} \end{pmatrix} \tag{3.14}$$

Remark 3.3.1. *Let $\mathbf{A} = diag(a_1, \dots, a_n)$ in $\mathbb{R}^{m \times n}$, and $\mathbf{B} \in \mathbb{R}^{n \times k}$. Define also a vector $\mathbf{a} =$*

$(a_1, \ldots, a_n)^\top \in \mathbb{R}^n$, *which represents the diagonal of* $\mathbf{A}$. *Then*

$$\mathbf{AB} = [\mathbf{a} \ldots \mathbf{a}] \circ \mathbf{B}$$

The former takes $\mathcal{O}(n^2 k)$ *operations, while the latter takes only* $\mathcal{O}(nk)$ *operations, which is one magnitude faster.*

All of our operations up to this point has been with two matrices A, B of the same shape, that is, $m \times n$ for both. What happens if they are now then different? Either we can justify the operation by 'extending it', however, usually, this requires discrete specific description for that to be able to be considered. Hence, matrix-matrix operations like such above is restricted of the entry shape.

Finally, we have one more important operation to be defined here at last. For two matrices A, B of shape $m \times n$ and $n \times p$, the **matrix multiplication** operation that outputs an $m \times p$ matrix, is available as:

$$A \times B = \begin{pmatrix} a_{11} & \ldots & a_{1n} \\ a_{21} & \ldots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \ldots & a_{mn} \end{pmatrix} \begin{pmatrix} b_{11} & \ldots & b_{1p} \\ b_{21} & \ldots & b_{2p} \\ \vdots & \ddots & \vdots \\ b_{n1} & \ldots & b_{np} \end{pmatrix} = \begin{pmatrix} c_{11} & \ldots & c_{1p} \\ c_{21} & \ldots & c_{2p} \\ \vdots & \ddots & \vdots \\ c_{m1} & \ldots & c_{mp} \end{pmatrix} \tag{3.15}$$

where

$$c_{ik} = \sum_{j<n} a_{ij} b_{jk} \tag{3.16}$$

For certain reason in the theory, or **matrix theory** and linear algebra in general, the matrix multiplication is explicitly defined only for such shape configuration. Otherwise, the matrix multiplication simply does not exist.

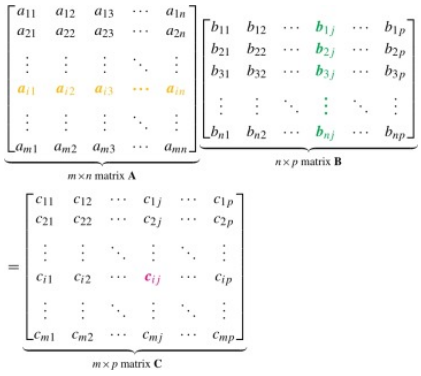


Figure 3.3: Illustration of matrix multiplication. Taken from Andrilli and Hecker [2010].

3.3.1 Special matrices, operations, and properties

Matrix has quite a few of special types that is defined for either decomposing the many matrices, or to complement them in expression. Operations aside from basic 'interactive operations' as above is also available, most of the time acting on the matrix by itself. Those operations, special matrices and properties are there to then draw out reasonable consequences and results.

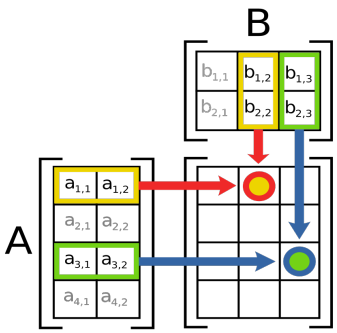


Figure 3.4: Illustration of matrix multiplication for 2×2 shape.

Matrix shape

Informally, A **matrix** is a rectangular array of numbers arranged in rows and columns. We say that a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is (given the entry field $\mathbb{R}$)

1. A square matrix if $m = n$.
2. A long matrix if $m < n$
3. A tall matrix if $m > n$

A **diagonal** matrix is a square matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ whose off diagonal entries are all zero, so $a_{ij} = 0$ for all $i \neq j$:

$$\mathbf{A} = \begin{pmatrix} a_{11} & & \\ & \ddots & \\ & & a_{nn} \end{pmatrix} \tag{3.17}$$

A diagonal matrix is uniquely defined through a vector that contains all the diagonal entries, and denoted as follows:

$$\mathbf{A} = \text{diag}(1, 2, \ldots, n) = \text{diag}(\mathbf{a}) \tag{3.18}$$

Using the diagonal matrix, we gain the following result.

Proposition 3.3.2 (Diagonal multiplication). *Given two matrices A, B of compatible shape. Then, if one is a diagonal matrix, denoted A_{diag}, B_{diag}, then the following is true.*

$$A_{diag}\mathbf{B} = \begin{pmatrix} a_1 & & \\ & \ddots & \\ & & a_n \end{pmatrix} \begin{pmatrix} \mathbf{B}(1,:) \\ \vdots \\ \mathbf{B}(n,:) \end{pmatrix} = \begin{pmatrix} a_1\mathbf{B}(1,:) \\ \vdots \\ a_2\mathbf{B}(2,:) \end{pmatrix} \tag{3.19}$$

and

$$\mathbf{A}B_{diag} = \begin{pmatrix} \mathbf{A}(1,:) \\ \vdots \\ \mathbf{A}(n,:) \end{pmatrix} \begin{pmatrix} b_1 & & \\ & \ddots & \\ & & b_n \end{pmatrix} = \begin{bmatrix} b_1\mathbf{A}(:,1) \ldots b_N\mathbf{A}(:,n) \end{bmatrix} \tag{3.20}$$

The two unit matrices that is often seen is the **identity** and **zero** matrix. Given the name, $\mathbf{I}$

is used to denote the identity matrix:

$$\mathbf{I} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

and $\mathbf{O}$ for the zero matrix,

$$\mathbf{O} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

with their sizes implied by the context, entries of all 1 or 0, respectively.

Transpose, inverse, trace and rank

Given a matrix A, then $A^\top$ is called the **transpose** of A for its shape reversed, $A_{m\times n}$ to $A'_{n\times m}$ with $b_{ij} = a_{ij}$ for all i, j. A square matrix $\mathbf{A}_{n\times n}$ is said to be symmetric if $\mathbf{A}^\top = \mathbf{A}$. We have the following result.

Proposition 3.3.3. *Let $\mathbf{A}$ be a matrix of size $m \times n$, $\mathbf{B}$ be a matrix of size $n \times p$. Then,*

1. $(\mathbf{A}^\top)^\top = \mathbf{A}$.
2. $(k\mathbf{A})^\top = k\mathbf{A}^\top$
3. $(\mathbf{A} + \mathbf{B})^\top = \mathbf{A}^\top + \mathbf{B}^\top$
4. $(\mathbf{AB})^\top = \mathbf{B}^\top\mathbf{A}^\top$

If we consider next the notion of the **inverse** of a matrix, the true, classical inverse definition works only for square matrix, by definition.

Definition 3.3.2 (Matrix inverse). *A square matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is said to be invertible if there exists another square matrix of the same size $\mathbf{B}$ such that $\mathbf{AB} = \mathbf{BA} = \mathbf{I}$.*

In this case, $\mathbf{B}$ is called the matrix inverse of $\mathbf{A}$ and denoted as $\mathbf{B} = \mathbf{A}^{-1}$. We have the following property:

Proposition 3.3.4. *Let $\mathbf{A}, \mathbf{B}$ be two invertible matrices of the same size, and $k \neq 0$. Then*

$$(k\mathbf{A})^{-1} = \frac{1}{2}\mathbf{A}^{-1} \tag{3.21}$$

$$(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1} \tag{3.22}$$

$$(\mathbf{A}^\top)^{-1} = (\mathbf{A}^{-1})^\top \tag{3.23}$$

The **trace** of a square matrix $A \in \mathbb{R}^{n\times n}$ is defined as the sum of the entries in its diagonal, such that:

$$\text{trace}(A) = \sum_i a_{ii} \tag{3.24}$$

We sometimes denote it as $\text{Tr}(A)$. Clearly, $\text{Tr}(A) = \text{Tr}(A^T)$. Trace is a **linear** operator, so $\text{Tr}(kA) = k\text{Tr}(A)$ and

$$\text{Tr}(A + B) = \text{Tr}(A) + \text{Tr}(B)$$

If A is an $m \times n$ matrix and B is an $n \times m$ matrix then

$$\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$$

Note that as matrices, AB is not necessarily equal to BA.

Continuing, for a matrix A, the largest number of linearly independent rows (or columns) contained in the matrix is called the rank of A, denoted $rank(A)$.

A square matrix $P \in \mathbb{R}^{n \times n}$ is said to be of full rank, or *nonsingular*, if $rank(P) = n$, otherwise, it is said to be rank deficient (or *singular*). A rectangular matrix $A \in \mathbb{R}^{m \times n}$ is said to have full column rank if $rank(B) = n$. Similarly, a rectangular matrix $A \in \mathbb{R}^{m \times n}$ is said to have full row rank if $rank(B) = m$.

Chapter 4. Probability and Statistics

For our purposes (most models and construction we will be encountering will be specifically *black-box models*, which are statitical by default), the knowledge of probability and statistics is a must. Despite what others said, in the real world, as well as theoretical works, the theory of probability and the work on statistics is the core foundation of modern automated system, and machine learning as a whole. Thereby, we might as well spare no efforts in introducing those concepts, broad and may-not-so deep.

4.1 What is probability?

Before even studying probability, and somewhat its more empirical centric statistic, what is even probability and statistic? Broadly speaking, quite a lot of definitions and interpretations that you can find yourselves. Perhaps not surprisingly, in normal life, you have been using the term probability, or statistics in its variations every time, every day. For example, just saying "he is probably reading books.", or "statistically, the price is cheaper.", or "I don't know? Probably.". The terms, and its perhaps intuitive notion has been quit a mouthful for everyone in their daily life, and throughout normal conversation. In said situation, probability often means, or rather implies the notion of *uncertainty*, while statistic implies the guarantee of *occurrences and patterns*.

While the concept of probability is popular and is perhaps quite intuitive, emerged before thousands of years, the concrete inception of probability as a discipline, and a field of mathematics took until mid-seventeenth century, by the work of Blaise Pascal and Pierre de Fermat. It took more time together, later on, for mathematicians to take measure theory and potential theory for formalizing and generalizing the theory of Probability, and become probability theory (in which we shall also study in a later date).

So, what can be concluded of the general notion of probability?

"*Mathematical modeling of random events and phenomena. It is fundamentally different from modeling deterministic events and functions, which constitutes the traditional study of Mathematics.*" (Ionut Florescu, 2009)

What does this mean, and how it works, we have to then research ourselves, the theory of both.

4.1.1 The fundamental principle of probability

4.1.2 Subjectivity of probability

Because of its nature, probability has the same problem as quantum mechanics - it invites people to get it certain interpretation or others. More than enough, there exists De Finetti's view on probability, with a strong remark as 'probability is fake'. On the other side, you have the typical, more popular view between *Bayesian* and *frequentist* interpretation of probability. Perhaps, a very good example will be sufficed to talk about this (Taken from the top answer of

this question on the Statistic Stack Exchange).

I have misplaced my phone somewhere in the home. I can use the phone locator on the base of the instrument to locate the phone and when I press the phone locator the phone starts beeping.

Problem: Which area of my home should I search?

- **Frequentist reasoning**: I can hear the phone beeping. I also have a mental model which helps me identify the area from which the sound is coming. Therefore, upon hearing the beep, I infer the area of my home I must search to locate the phone.

- **Bayesian reasoning**: I can hear the phone beeping. Now, apart from a mental model which helps me identify the area from which the sound is coming from, I also know the locations where I have misplaced the phone in the past. So, I combine my inferences using the beeps and my prior information about the locations I have misplaced the phone in the past to identify an area I must search to locate the phone.

In essence, the frequentist view base the events and probability as long-term, perhaps objective truth. While Bayesian goes for the subjective perspective of priori belief, and overall trust within the future prediction. What is right or wrong? Perhaps there are none.

4.2 Space of probability

Probability, at its core, relies on the interpretation of *events* and general phenomena that happens. In fact, The XVII-th century records the first documented evidence of the use of Probability Theory. More precisely in 1654 Antoine Gombaud, Chevalier de Mere, a French nobleman with an interest in gaming and gambling questions, was puzzled by an apparent contradiction concerning a popular dice game. The game consisted in throwing a pair of dice 24 times; the problem was to decide whether or not to bet even money on the occurrence of at least one "double six" during the 24 throws. A seemingly well-established gambling rule led de Mere to believe that betting on a double six in 24 throws would be profitable, but his own calculations based on many repetitions of the 24 throws indicated just the opposite. Using modern probability language, de Mere was trying to establish if such an event has probability greater than 0.5. Puzzled by this and other similar gambling problems he called the attention of the famous mathematician Blaise Pascal. In turn this led to an exchange of letters between Pascal and another famous French mathematician Pierre de Fermat, this exchange becoming the first documented evidence of the fundamental principles of the theory of probability. This coincides with our needs to then define and interpret the working space on that probability will be exhibited.

We define the following as the space in which probability is concerned.

Definition 4.2.1. *The set of all possible outcomes of an experiment is known as the* ***sample space*** *of the experiment, denoted S. For events E and F, the union is defined by $E \cup F$, being either E or F occurs. Similarly, $EF = E \cap F$ is the event of* ***intersection****, where E and F both occur. If $EF = \varnothing$, we say that they are mutually exclusive. Generally,*

$$E_1 \cup E_2 \cup \cdots = \bigcup_{n=1}^{\infty} E_n, \quad E_1 \cap E_2 \cap \cdots = \bigcap_{n=1}^{\infty} E_n \quad \forall n \in \mathbb{N}[1, n] \tag{4.1}$$

Under that notion, suppose $\Omega = \{\omega_1, \ldots, \omega_n\}$. Let $A \subseteq \Omega$ be an event, then the probability of A is

$$\mathbb{P} = \frac{\text{Number of outcomes in A}}{\text{Number of outcomes in } \Omega} = \frac{|A|}{n} = \frac{|A|}{|\Omega|}$$

Example 4.2.1. *Suppose r digits are chosen from a table of random numbers, so that the available digits are $0 \leq k \leq 9$. Find the probability that:*

- *No digit exceed k.*
- *k is the greatest digit drawn.*

Take

$$\Omega = \{a_1, \ldots, a_r \mid 0 \leq a_i \leq 9, i \in I\}$$

be the sample space. The event $A_k = [\text{no digit exceeds } k]$ is defined as

$$A_k = \{(a_1, a_2, \ldots, a_r) \mid 0 \leq a_i \leq k\}$$

Thus $|\Omega| = 10^r$ and $|A_k| = (k+1)^r$. We also have, intuitively, $A_k \subseteq \Omega$. Then:

$$\mathbb{P}(A_k) = \frac{|A_k|}{|\Omega|} = \frac{(k+1)^r}{10^r}$$

Continuing, we see that $B_k = [k \text{ is the greatest digits drawn}]$ is a subset of A_k., of which excluding all the case of that no digit exceed $k-1$ (meaning there are no k presents). Hence, we have

$$|B_k| = |A_k| - |A_{k-1}|$$

Hence,

$$\mathbb{P}(B_k) = \frac{(k+1)^r - k^r}{10^r}$$

We will treat probability axiomatically of the preceding sections. For now, though, we recognize that a lot of the problem connected to the notion of 'maybe it is probably true', can be resolved by simply counting all the possible way things can occur.

4.3 Combinatorial probability

The basic principle of counting is used, when you want to, well, *counts* in the way of **combinatorial analysis** upon how many configuration, outcomes and occurrence for and event of interest. Hence, it is the fundamental to our work, as for most of the time you can brute force probability using such principle.

Definition 4.3.1 (The basic principle of counting). *Suppose that two experiments are to be performed. Then if experiment 1 can result in any one of m possible outcomes and if, for each outcome of experiment 1, there are n possible outcomes of experiment 2, then together there are mn possible outcomes of the two experiments.*

Proof. The basic principle may be proven by enumerating all the possible outcomes of the two experiments. We say that the outcome is (i, j) if experiment 1 results in its ith possible outcome, and experiment 2 results in its jth possible outcome. The set of outcome consists of m rows, and n columns. Hence, the total space contains mn pairs, hence proves the result. □

The more general form of said expression can be the following interpretation for a set of choice.

Definition 4.3.2 (The basic principle of counting). *Suppose r multiple choices are to be made in sequence, there are m_1 possibilities for the first choice, then m_2 for the second choices and so on until after making $r-1$ choices there are m_r possibilities for the rth choice. Then the total number of different possibilities for the set of choice is:*

$$T_{\mathbb{P}} = \prod_{i=1}^{r} m_i$$

So, how many of counting or scenario there typically are?

Sampling with or without replacement

Standard calculation often involves counting numbers of equally likely outcomes, and can be viewed as counting the number of lists of length n that can be constructed from a set of x items $X = \{1, \ldots, x\}$

Let $N = \{1, \ldots, n\}$ be the set of list position. Consider $f : N \to X$ that gives the ordered list $(f(1), \ldots, f(n))$. The function can be proceeded in three ways:

- **Sampling with replacement**: After choosing an item we put it back so it can be chosen again.
- **Sampling without replacement**: After choosing an item we put it aside. This creates an ordered list of n distinct items. The order of magnitude is not the order we are talking about.
- Sampling with replacement, but requiring each item is chosen at least once.

Effectively, we can see that the first one is for any f, injective f and surjective f.

Sampling without ordering

When counting numbers of possible $f : N \to X$, we might decide that the labels that are given to elements of N and X do or do not matter. For the case of having $(f(1), \ldots, f(n))$, we have a few options:

- Do nothing (order matters).
- Ascending sort (labels of the position in the list do not matter)
- Renumber each item in the list by the number of the draw on which it was first seen (label of the item do not matter).
- Do both the two above (no labels matters).

For the second one, we are saying that $g(1), \ldots, g(n) = (f(1), \ldots, f(n))$ if there is permutation of π of $1, \ldots, n$ such that $g(i) = f(\pi(i))$.

Four cases of enumerative combinatorics

Considering four possibilities obtained from combinations of sampling and first two ordering, we have the following important case of combinatorial combinations.

- **Sampling with replacement and with ordering**: Each location in the list can be filled in x way, the total possibilities are x^n.
- **Sampling without replacement and with ordering**: Applying the fundamental rule, this can be done in $x_{(n)} = x(x-1)\ldots(x-(n-1))$. Another notation is $x^{\underline{n}}$, read as 'x to the n falling'. In special case $n = x$ then this equal to $x!$.
- **Sampling without replacement and without ordering**: We care only which items are selected. The position in the list are indistinguishable. The total space of possibilities are

$$\frac{x_n}{n!} = \binom{x}{n}$$

i.e. the answer above divided by $n!$. Notice that the answer is the **binomial coefficient**,

the distinguishable sets of n items that can be chosen from a set of x items.

- **Sampling with replacement and without ordering**: Now we care only how many times each item is selected. In order: We are sampling without ordering, $(2, 4)$ is the same as $(4, 2)$. Then, for every chosen item, we put it back in the choosing bin. The number of distinct f is the number of nonnegative integer solution to

$$n_1 + n_2 + \cdots + n_x = n$$

Explanation is possible - Suppose a set $A = \{1, 2, 3\}$. There are six possibilities of choosing them with replacement without the concern of ordering:

- $1, 2$
- $1, 1$
- $2, 2$
- $3, 1$
- $3, 2$
- $3, 3$

Notice that if we somehow settle that (x_1, x_2, x_3) is the tuple that represents, for each possibilities, the number of individual elements it has, then

$$\begin{aligned} 1, 2 &\to (x_1, x_2, x_3) = (1, 1, 0) \\ 1, 1 &\to (x_1, x_2, x_3) = (2, 0, 0) \\ 2, 2 &\to (x_1, x_2, x_3) = (0, 2, 0) \end{aligned}$$

4.4 Axiomatic probability

With the advent of probability to the point that requires a deeper analysis, one such typically way to achieve this, is to axiomatize it. Although indeed there are problems that come with axiomatizing a formal system, and the presumed observation seems more tiresome than not, it is by this way that we can argue that it far outweighs the hardship that comes with using it, for less ambiguous notions or definitions, and the permittivity to expand the probability interpretation to even further - most likely to get you the *probability measure*.

4.4.1 Axiom of probability

Axiomatization of probability reserves the formal treatment of said notion to a powerful foundation, at least grounded probability in mathematical notion more than arbitrary spaces. Defining the *probability of event E* as the following definition

Definition 4.4.1. *Define the probability of event E as the following expression:*

$$P(E) = \lim_{n \to \infty} \frac{n(E)}{n}$$

That is, $P(E)$ is defined as the proportion of time that E occurs.

A **probability space** is then defined of the triple $(\Omega, \mathcal{F}, P)$ in which Ω is the sample space, $\mathcal{F}$ is a collection of subsets of Ω, and P is a **probability measure** $P : \mathcal{F} \to [0, 1]$. To obtain a consistent theory, we place requirements on $\mathcal{F}$:

1. $\varnothing \in \mathcal{F}$ and $\Omega \in \mathcal{F}$.

2. $A \in \mathcal{F} \implies A^c \in \mathcal{F}$.
3. $A_1, \cdots \in \mathcal{F} \implies \bigcup_{i=1}^{\infty} A_i \in \mathcal{F}$.

We proceed to give the three axioms of probability, which is then the restrictions applied for P of the tuple:

Axiom 4.4.1 (Axiom of probability). *For event E and probability $P(E)$, sample space S:*

I. $0 \leq P(E) \leq 1$.

II. $P(S) = 1$.

III. *For any sequence of mutually exclusive events $E_1, E_2, \ldots$, that is, for $E_i E_j = \varnothing$ when $i \neq j$, we have:*

$$P\left(\bigcup_{i=1}^{\infty} E_i\right) = \sum_{i=1}^{\infty} P(E_i)$$

We refer to $P(E)$ as the probability of E.

If Ω is not finite then it may not be possible to let $\mathcal{F}$ be all subsets of Ω. It can be shown that it is impossible to define a P for all possible subsets of the interval $[0, 1]$ that will satisfy the axioms.

Theorem 4.4.1 (Properties of P). *Axioms I-III imply the following further properties:*

i. $P(\varnothing) = 0$ *(probability of empty set)*

ii. $P(A^c) = 1 - P(A)$.

iii. *If $A \subseteq B$ then $P(A) \leq P(B)$. (monotonicity)*

iv. $P(A \cup B) = P(A) + P(B) - P(A \cap B)$ *(by inclusion-exclusion)*

v. *If $A_1 \subseteq A_2 \subseteq \ldots$ then*

$$P\left(\bigcup_{i=1}^{\infty} A_i\right) = \lim_{n \to \infty} P(A_n) \tag{4.2}$$

Property (v) says that $P(\cdot)$ is a continuous function.

4.4.2 Boole's inequality

One of the more important result, especially in the way of doing combinatorial analysis for probability, is the **Boole's inequality**. We state the inequality.

4.4.3 Conditional probability

Definition 4.4.2 (Bayes's law). *For 2 events E and F, the following construction is true:*

$$P(F_j \mid E) = \frac{P(EF_j)}{P(E)} = \frac{P(E \mid F_j)P(F_j)}{\sum_{i=1}^{n} P(E \mid F_i)P(F_i)} \tag{4.3}$$

by $P(E) = P(E \mid F)P(F) + P(E \mid F^c)[1 - P(F)]$. *Reducing to a two-case basis of* E *and* F*:*

$$P(F \mid E) = \frac{P(E \mid F)P(F)}{P(E)} \tag{4.4}$$

which gives the standard form of a single-dependency Bayes's formula of prior hypothesis.

Definition 4.4.3 (Independent events). *Two events* E *and* F *are said to be independent if* $P(EF) = P(E)P(F)$*. Two events* E *and* F *are not independent are said to be dependent. Generally, for events* $E_1, \ldots, E_n$*, they are independent if for every subset* $E_{1'}, \ldots, E_{r'}$ *for* $r \leq n$ *of these events, then*

$$P(E_{1'}E_{2'} \ldots E_{r'}) = P(E_{1'}) \ldots P(E_{r'})$$

4.5 Random variables

It is considered that the space of values of X as random variable, is the *sample space* of the *sample (outcome) space*, or just $\Omega(\Omega)$ if you want an analogous view. For a given observable event space Ω, a **random variable** X is a specific quantification of interest, that is, $X : \Omega \to \mathcal{P}(\mathbb{F})$.

For a random variable X, $F(x) = P\{X \leq x\}$ for $x \in S_X \subset \mathcal{P}(\mathbb{F})$ is the *cumulative distribution function* $\mathsf{CDF}(X)$, or the *distribution function*. $F(x)$ is a non-decreasing function of x, such that for $a \leq b$, $F(a) \leq F(b)$.

A random variable is discrete if the range S_X is countable, or there exists a discrete function f_d govern by a set of parameters $\{\lambda_j\} \subset \mathbb{A}^j$ of rational or algebraic number, such that

$$P(i) = P\{X = i\} = f(i, \{\lambda_1, \ldots, \lambda_n\}), \tag{4.5}$$

Then, we define the *probability mass function* $p(a)$ of X by the quantity $p(a) = P\{X = a\}$. It then follows that

$$\sum_{i=1}^{\infty} p(x_i) = 1 \quad \forall x_i \in S_X \tag{4.6}$$

A random variable is **continuous** if there exists a non-negative $f \in C^q$ of differentiable function define on $D = (-\infty, +\infty)$, called the *probability density function* such that:

$$P\{X \in B\} = \int_B f(x)\, dx \quad B \subset D \subset \mathbb{R} \tag{4.7}$$

This satisfies the axiom of probability, such that

$$P\{X \in D\} = \int_{-\infty}^{\infty} f(x)\, dx = 1 \tag{4.8}$$

4.6 Expected values

The **expectation** of a random variable is the *weighted average* on the probable random outcome space of X, weighted by the probability $p(x), x \in S_X$. Denoted $E[X]$, for discrete X:

$$E[X] = \sum_{x:p(x)>0} xp(x) \tag{4.9}$$

and for continuous X:

$$E[X] = \int_{-\infty}^{\infty} xf(x)\, dx \tag{4.10}$$

Theorem 4.6.1 (Linearity of expectation). *Let Ω be the sample space of an experiment, $X, Y : \Omega \to \mathbb{R}$ be (possibly "dependent") random variable both defined on Ω and $a, b, c \in \mathbb{R}$ be scalar. Then,*

$$\mathbb{E}[X+Y] = \mathbb{E}[X] + \mathbb{E}[Y], \qquad \mathbb{E}[aX+b] = a\mathbb{E}[X] + b \tag{4.11}$$

The following important proposition is called **The law of the unconcious statistician**.

Proposition 4.6.2. *If X is a discrete random variable that takes on one of the values $x_i, i \geq 1$, with respective probabilities $p(x_i)$, then for any real-valued function g,*

$$E[g(X)] = \sum_i g(x_i)p(x_i) \tag{4.12}$$

Proof. Suppose that $y_j, j \geq 1$ represents the different values of $g(x_i), i \geq 1$. Then, we have:

$$\begin{aligned}
\sum_i g(x_i)p(x_i) &= \sum_j \sum_{i:g(x_i)=y_j} g(x_i p(x_i)) \\
&= \sum_j \sum_{i:g(x_i)=y_j} y_j p(x_i) \\
&= \sum_j y_j \sum_{i:g(x_i)=y_j} p(x_i) \\
&= \sum_j y_j P\{g(X) = y_j\} \\
&= E[g(X)]
\end{aligned} \tag{4.13}$$

□

Proposition 4.6.3. *If X is a continuous r.v. with pmf $f(x)$, then for any g on $\mathbb{R}$ (real-valued):*

$$E[g(x)] = \int_{-\infty}^{\infty} g(x)f(x)\,dx \tag{4.14}$$

Proof. We have the following lemma.

Lemma 4.6.4. *For $Y > 0$ random variable,*

$$E[Y] = \int_0^{\infty} P\{Y > y\}\,dy \tag{4.15}$$

Proof. Assume that Y has a pdf f_Y, we have:

$$\begin{aligned}
\int_0^{\infty} P\{Y > y\}\,dy &= \int_0^{\infty} \int_y^{\infty} f_Y(x)\,dx\,dy \\
&= \int_0^{\infty} \left(\int_0^x dy \right) f_Y(x)\,dx \\
&= \int_0^{\infty} x f_Y(x)\,dx = E[Y]
\end{aligned} \tag{4.16}$$

where $P\{Y > y\} = \int_Y^{\infty} f_Y(x)\,dx$. □

From 4.6.4, for any $g(x) \geq 0$,

$$\begin{aligned} E[g(X)] &= \int_0^\infty P\{g(X) > y\}\, dy \\ &= \int_0^\infty \int_{x:g(x)>y} f(x)\, dx\, dy \\ &= \int_{x:g(x)>y} f(x) \int_0^{g(x)} dy f(x)\, dx \\ &= \int_{x:g(x)>0} g(x) f(x)\, dx \end{aligned} \tag{4.17}$$

□

Properties of expectation

Theorem 4.6.5. *The following is true for random variables X, Y and expectation E:*

1. *If $X \geq 0$ then $E[X] \geq 0$.*
2. *If $X \geq 0$ and $E[X] = 0$ then $P(X = 0) = 1$.*
3. *EX is the constant that minimizes $E\left[(X - c)^2\right]$.*

4.6.1 Variance

Given a random variable along with its distribution function F, one of the primitive fundamental property of F is called the **variance**.

Definition 4.6.1 (Variance). *If X is a random variable with mean μ, then the variance of X, denoted $\mathrm{Var}(X)$ is defined by:*

$$\mathrm{Var}(X) = E[(X - \mu)^2] = E[X^2] - \mu^2$$

for $\mu = E[X]$.[a] [b]

[a] A useful identity is that for any constant a and b:

$$\mathrm{Var}(aX + b) = a^2 \mathrm{Var}(X)$$

[b] The standard deviation is then $\sqrt{\mathrm{Var}(X)}$

Theorem 4.6.6 (Properties of variances). *For variance $\mathrm{Var}(\cdot)$ of some random variable X, we have that*

4.7 Some distribution functions

In this section, we would introduce and glance through certain probability distribution usually met while working on probability and its interpretation.

4.7.1 Bernoulli and binomial distribution

Suppose that a trial, or an experiment whose outcome can be classified as either *success* or *failure* is performed. The p.m.f of $X \in \{0, 1\}$ is then given by:

$$p(0) = P\{X = 0\} = 1 - p, \quad p(1) = P\{X = 1\} = p$$

where $p, 0 \leq g \leq 1$, is the probability that the trial is a success. Then X is said to be a **Bernoulli random variable**.

Suppose for n independent trials, each of which results in success or failure with probability p and $1-p$. If X represents the number of successes, then X is said to be a **binomial random variable** of parameter (n, p). The pmf in this case is given by:

$$p(i) = \binom{n}{i} p^i (1-p)^{n-i}, \quad i = 0, 1, \ldots, n$$

Note that

$$\sum_{i=1}^{\infty} p(i) = \sum_{i=1}^{n} \binom{n}{i} p^i (1-p)^{n-i} = 1$$

For a binomial distribution. $E[X] = np$ for n trials. Furthermore, we have:

$$E[X^2] = np[(n-1)p + 1]$$

And

$$\text{Var}(X) = E[X^2] - (E[X])^2 = np(1-p)$$

Proposition 4.7.1. *If X is a binomial random variable with parameters (n, p) where $0 < p < 1$, then as $k : 0 \to n$, $P\{X = k\}$ increases monotonically then decreases monotonically, reaching its largest value when $k = \max(q) \leq (n+1)p$, $q \in \mathbb{Z}$. (that is, the largest integer less or equal to $(n+1)p$*

We can calculate numerically the binomial distribution of (n, p) by using:

$$P(X = k+1) = \frac{p}{1-p} \frac{n-k}{k+1} P\{X = k\} \tag{4.18}$$

4.7.2 Poisson distribution

A random variable X that takes on one of the values $0, 1, 2, \ldots$ is said to be a **Poisson random variable** with parameter λ for some $\lambda > 0$ if

$$P(i) = P\{X = i\} = e^{-\lambda} \frac{\lambda^i}{i!}, \quad i = 0, 1, \ldots$$

The pmf of Poisson random variable is then in the form:

$$\sum_{i=0}^{\infty} p(i) = e^{-\lambda} \sum_{i=0}^{\infty} \frac{\lambda^i}{i!} = e^{-\lambda} e^{\lambda} = 1$$

Poisson random variable is tremendously useful, and one of which is because for moderate p, and large n, Poisson variable can be used as an *approximation* of the binomial (n, p).

The **expected value** of Poisson random variable is then is:

$$E[X] = \sum_{i=0}^{\infty} \frac{i e^{-\lambda} \lambda^i}{i!} = \lambda$$

and

$$E[X^2] = \sum_{i=0}^{\infty} \frac{i^2 e^{\lambda} \lambda^i}{i!} = \lambda(\lambda + 1)$$

Since $E[X] = \lambda$, we have:

$$\text{Var}(X) = E[X^2] - (E[X])^2 = \lambda$$

The expected value and the *variance* is then both equal to λ.

Poisson paradigm

Consider n events, with p_i equal to the probability that event i occurs, $i = 1, \ldots, n$. If all the p_i are 'small' and the trials are either independent or at most "weakly dependent", then the number of these events that occur approximately has a Poisson distribution with mean $\sum_{i=1}^{n} p_i$.

Another use of the Poisson probability distribution arises in situations where "events" occur at certain points in time. One example is to designate the occurrence of an earthquake as an event; another possibility would be for events to correspond to people entering a particular establishment. Suppose that events are indeed random occurrence. For some $\lambda > 0$, the following *assumptions* are true:

1. The probability that exactly one event occurs in a given interval of length h is equal to $\lambda h + o(h)$.
2. The probability that 2 or more events occur in an interval of length h is equal to $o(h)$.
3. For any $n \in \mathbb{Z}$, $j_1, \ldots, j_n$ and any set of n nonoverlapping intervals, if we define E_i to be the event that exactly j_i of the events under consideration occur in the ith of these intervals, then events $E_1, \ldots, E_n$ are independent.

4.7.3 Uniform distribution

A random variable is said to be *uniformly distributed* over $(0, 1)$ if its probability density function is given by

$$f(x) = \begin{cases} 1 & 0 < x < 1 \\ 0 & \text{o.w} \end{cases} \tag{4.19}$$

The above function is a density function, since $f(x) \geq 0$ and $\int_{-\infty}^{\infty} f(x)\,dx = \int_0^1 dx = 1$. Because $f(x) > 0$ only when $x \in (0, 1)$, it follows that X must assume a value in $(0, 1)$. The **probability** that X is in any particular subinterval of $(0, 1)$ equals the length of that subinterval. In general, we say that X is a **uniform random variable** on (α, β) if the p.d.f of X is given by

$$f(x) = \begin{cases} \dfrac{1}{\beta - \alpha} & \alpha < x < \beta \\ 0 & \text{o.w} \end{cases} \tag{4.20}$$

Since $F(a) = \int_{-\infty}^{a} f(x)\,dx$ it follows from $f(x)$ that

$$F(a) = \begin{cases} 0 & a \leq \alpha \\ \frac{a-\alpha}{\beta-\alpha} & \alpha < a < \beta \\ 1 & a \geq \beta \end{cases} \tag{4.21}$$

For X be uniformly distributed over (α, β), we have:

$$E[X] = \frac{\beta + \alpha}{2}, \quad \text{Var}[X] = \frac{(\beta - \alpha)^2}{12}$$

4.7.4 Normal distribution

We say that X is a **normal random variable**, or simply that X is normally distributed, with parameters μ and σ^2 if the density of X is given by

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2} \qquad -\infty < x < \infty$$

This density function is a bell-shaped curve that is symmetric about μ.

If we are to prove that $f(x)$ is indeed a probability density function, we need to show that

$$\frac{1}{\sqrt{2\pi\sigma}}\int_{-\infty}^{\infty} e^{-(x-\mu)^2/2\sigma^2}\,dx = 1$$

Which is indeed the case, as for certainty the proof is too long to fit.

An important fact about normal random variables is that if X is normally distributed with parameters μ and σ^2, then $Y = aX + b$ is normally distributed with parameters $a\mu + b$ and $a^2\sigma^2$. To prove this, suppose that $a > 0$. Let F_Y denotes the cdf, then

$$F_Y(x) = P\{Y \le x\} = P\left\{X \le \frac{x-b}{a}\right\} = F_X\left(\frac{x-b}{a}\right)$$

By differentiation, the density function is hence

$$f_Y(x) = \frac{1}{a}f_X\left(\frac{x-b}{a}\right) = \frac{1}{\sqrt{2\pi}a\sigma}\exp\left[-\frac{(x-b-a\mu)^2}{2(a\sigma)^2}\right]$$

Which shows that Y is normal with parameters $a\mu + b$ and $a^2\mu^2$. This also implies that if X is normally distributed, then for $Z = (X-\mu)/\sigma$, it is said to be normally distributed with parameters 0 and 1. Such random variable is called *standard*, or *unit* normal random variable.

Normal approximation of binomial distribution

It is a theorem that is called the **DeMoivre-Laplace** limit theorem, which states that for large n, a binomial RV with parameters n, p will have approximately same distribution as $\mathcal{N}(\sigma, \mu)$ as the binomial.

Theorem 4.7.2 (DeMoivre-Laplace limit theorem). *If S_n denotes the number of successes that occur when n independent trials, each resulting in a success with probability p, are performed, then, for any $a < b$, we have:*

$$P\left\{a \le \frac{(S_n - np)}{\sqrt{np(1-p)}} \le b\right\} = \Theta(b) - \Theta(a)$$

as $n \to \infty$.

4.7.5 Exponential distribution

A continuous random variable whose probability density function is given, for some $\lambda > 0$, by

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & x \ge 0 \\ 0 & x < 0 \end{cases}$$

is said to be an *exponential random variable*, or exponentially distributed with parameter λ. The cdf of such variable is then given by:

$$F(a) = P\{X \le a\} = 1 - e^{-\lambda a}, \quad a \ge 0$$

Its expected value is

$$E[X] = \frac{1}{\lambda}, \quad E[X^n] = \int_{-\infty}^{0} x^n \lambda e^{-\lambda x}\,dx = \frac{n}{\lambda}E[X^{n-1}]$$

From that, we get

$$\mathrm{Var}(X) = \frac{2}{\lambda^2} - \left(\frac{1}{\lambda}\right)^2 = \frac{1}{\lambda^2}$$

Memory

We say that a non-negative random variable is **memoryless** if:

$$P\{X > x + t \mid X > t\} = P\{X > s\}$$

for all $s, t > 0$. The above equation is equal to:

$$P\{X > s + t\} = P\{X > s\}P\{X > t\}$$

When X is exponentially distributed, we see that it is memoryless.

Hazard rate

Consider a positive continuous random variable X that we interpret as being the lifetime of some item. Let X have distribution function F and density f. The *hazard rate* function $\lambda(t)$ of F is defined by:

$$\lambda(t) = \frac{f(t)}{\bar{F}(t)}$$

where $\bar{F} = 1 - F$.

4.7.6 Distribution function of variables

Often, we know the probability distribution of a random variable and are interested in determining the distribution of some function of it. For example, we know $f(X)$, and is trying to find the distribution of $g(X)$. To do so, it is to express the event that $g(X) \leq y$, in terms of X being in some set. We have the following theorem.

Theorem 4.7.3. *Let X be a continuous random variable having probability density function f_X. Suppose that $g(x)$ is a strictly monotonic, differentiable function of x. Then the random variable Y defined by $Y = g(X)$ has a probability density function given by:*

$$f_Y(y) = \begin{cases} f_X[g^{-1}(y)] \left| \frac{d}{dy} g^{-1}(y) \right| & y = g(x) \\ 0 & y \neq g(x) \end{cases}$$

Chapter 5. Metric spaces, measure theory

A lot of times, when talking of spaces and their intrinsic properties, we come by the property of distance - for example, the distance $d(x, y)$ between two points x or y, maybe from one's house to another. Formalizing this concept is done by inventing (yet again) a perhaps theory entangled with real analysis (or rather, a *basis* on which real analysis might take its course), called **measure theory**.

To start, however, we first have to define the pseudometric of a set (more abstract notion for a space). Then we will see how that comes to become metric space. Although it might be later noted that metric comes off more naturally in system, it is also the case for pseudometric under pathological (or plentiful of weirdness) spaces.

5.1 Metric theory

The study of metric spaces (or theory, depends on the choice of word) comes of as rather natural when considering its dependency on the field of *analysis* in its introduction. By itself, it studies a pretty much intrinsic property of any spaces. The notion of *distance*.

By itself, a set doesn't have any structure. For two arbitrary sets A and B, you can ask questions like "Is $A = B$?" or "Is A equivalent to a subset of B?" vice versa, or perhaps the cardinality of A and B, perhaps interested in what kind of prime would they be (not so interesting, eh?) but not much more. All those questions are typically very general, and offer no good insight into the set itself. Though, if we add *additional structure* to a set, it becomes more interesting. For example, if we define a "multiplication operation" $a \cdot b$ in X that satisfies certain axioms, then X becomes an algebraic structure called a group, and a whole area called group theory begins.

We are not interested in making a set X an algebraic system. For our purposes, in analysis and its extension, topology view, we want additional structure on X to talk about its *neighbourhood*. This is what we need for topics like convergence, or continuity, or density, or closeness - roughly, in analysis, f is continuous at a means that if x is near a, then $f(x)$ is near $f(a)$. The notion of the world "near" is what we are trying to achieve.

So far, as this section might have suggested it already, the simplest way to talk about nearness, is to equip the set with some additional distance metric, or function d to tell us how far apart elements of the set actually is from each other. Of course, this distance metric does not need to be ordered - it just has to work every time (in not-so pathological case, that is).

5.1.1 Pseudometric spaces

The first thing to begin, is what we called a *pseudometric space* . We have the following notion of a pseudometric space.

pseudometric space

Definition 5.1.1 (Pseudometric space). *Suppose X is a set. A function $\rho : X \times X \to \mathbb{R}_+$ is said to be a **pseudometric** if:*

1. $\rho(x,x) = 0$ *for all* $x \in X$.
2. $\rho(x,y) = \rho(y,x)$ *for all* $x, y \in X$.
3. $\rho(x,z) \leq \rho(x,y) + \rho(y,z)$, *for all* $x, y, z \in X$.

If, in addition, ρ *satisfies:* $\rho(x,y) = 0 \implies x = y$ *then* ρ *is said to be a* **metric**.

If ρ is a pseudometric (metric) on X, we say that (X, ρ) is a *pseudometric space* (metric space). Supposed (X, ρ) is a pseudometric space, and suppose a binary relation $\sim$ on X defined by $x \sim y \leftrightarrow \rho(x,y) = 0$. It is easy to verify that $\sim$ is an equivalent relation on X, by reflexivity, symmetry and transitivity of the above condition. Hence, X can be partitioned into its equivalence classes under $\sim$.

The fundamental difference between pseudometric and metric space is the fact that the axiom on $\rho(x,y) = 0$ then $x = y$ is missing, that is, there exists particular x, y such that their distance measure is the same, but they are not the same. Example of a particular pseudometric on $\mathbb{R}^n$ is for $\rho : \mathbb{R} \times \mathbb{R}$ to be

$$\rho((x_1, y_1), (x_2, y_2)) = |x_1 - y_1|$$

in which an entire dimension is missing from the metric measure. As for its usefulness in practical application and formal treatment of non-mathematical subject, it is still valid to ask what is the role that pseudometric space would play on.

As a result, we only need to add a single criterion for it to become metric space.

Definition 5.1.2 (Metric space). *A* ***metric space*** *is a pair* (M, d) *where* M *is a set and* d *is a function* $d : M \times M \to \mathbb{R}$ *satisfying:*

1. $d(x,y) \geq 0$ *for all* $x, y \in M$.
2. $d(x,y) = 0$ *if and only if* $x = y$.
3. *Symmetry:* $d(x,y) = d(y,x)$ *for all* $x, y \in M$, *and*
4. *The* ***triangle inequality****:* $d(x,z) \leq d(x,y) + d(y,z)$, *for all* $x, y, z \in M$.

In one way or another, it is typically constructed that pseudometric space is a generalization of metric space in the notion and study of neighbourhood. Though, there are a lot of easy examples taken inside a metric space rather than pseudometric, intuitively.

Example 5.1.1. *The usual metric on* $\mathbb{R}$ *is* $d(x,y) = |x - y|$. *Clearly, this satisfies all property of metric space. In fact, the properties are chosen so that a metric imitates the usual distance function.*

5.2 Measure

After the notion of distance for a set, now we also want to formalize the notion of **size** of a set. This resulted in ultimately the theory of measure, or *measure theory*. When I first learn about measure theory, it is very abstract. But trust it when I say after a while, you either land in the middle of not understanding it but also understanding it (the Schrödinger state), or you will be the one to understand it, and no on the other side. Except if you skip the class.

5.2.1 Sigma-algebra

For subsets which are element of X, that is the power set $\mathcal{P}(X)$ are often impossible to fully define such notion of $\mu[U]$ on them. Instead, we must isolate a smaller domain, where the measure will be well-defined. This domain is called **measurable**, and those that are not are contained in the **non-measurable** domain.

Hence, before we can define a measure, we must describe a suitable domain for it. This is called a **sigma-algebra**, and is a collection of subsets of X.

sigma-algebra

Definition 5.2.1 (Sigma algebra). *Let X be a set. A σ-algebra over X is a collection $\mathcal{X}$ of subsets of X with the following properties:*

1. *> 1. $\mathcal{X}$ is closed under **countable unions**. That is, if $U_1, U_2, \ldots,$ are in X, then their intersection $\bigcap_{n=1}^{\infty} U_n$ is also in $\mathcal{X}$.*
2. *> 2. $\mathcal{X}$ is closed under **countable intersection**. If $U_1, U_2, \ldots,$ are in $\mathcal{X}$, then their intersection $\bigcap_{n=1}^{\infty} U_n$ is also in $\mathcal{X}$.*
3. *> 3. $\mathcal{X}$ is closed under **complementation**: If $U \in \mathcal{X}$, then $U^c = (X \setminus U) \in \mathcal{X}$.*

Intuitively sigma-algebra defines a particular space, which might be explained as a space in which aside from the singleton member, there is all the combination "and" - all the countable intersection, and all the addition possible - for countable unions. Or, so we say that includes all possible additive set that their size can be combined, merged together, well-defined in such space. Then it is called a sigma-algebra of any given set $\mathcal{X}$. Usually, there exists no sigma-algebra to the set itself, so, only a fraction of it is possible.

A **measurable space** is then an *ordered pair* $(X, \mathcal{X})$ where X is a set, and $\mathcal{X}$ is a sigma-algebra on X. Given $(X, \mathcal{S}), (Y, \mathcal{T})$ be measurable spaces, then a function $f : X \longrightarrow Y$ is called a **measurable function** if $f^{-1}(T) \in \mathcal{S}$ for all $T \in \mathcal{T}$. [1]

5.2.2 A few sigma algebras

So, how many interesting or realistic sigma-algebra are there? We will review through a few interesting one, include the one that will find its way into the definition of Lebesgue measure.

Trivial sigma algebras

For any set X, the collection $\{\varnothing, X\}$ is a sigma-algebra. Similarly, the power set $\mathcal{P}(X)$ is a sigma-algebra. The first is *too small* to do anything, while the latter is *too big* to be manageable.

One way to create a manageable sigma-algebra is to start with some collection $\mathcal{M}$ of manageable sets, and then find the smallest sigma-algebra which contains all elements of $\mathcal{M}$. This is called the **sigma-algebra generated by** $\mathcal{M}$, or $\sigma(\mathcal{M})$.

Partition algebras

Let X be a set. The *partition* of X is a collection $\mathcal{P} = \{P_1, \ldots, P_N\}$ of disjoint subsets, such that

$$X = \bigsqcup_{n=1}^{N} P_n$$

[1] It is also typical to review the σ-algebra of a measurable space, too. Not sure why would that be the case.

The sets $P_1, \dots, P_N$ are called the **atoms** of the partition. The *sigma-algebra generated by* $\mathcal{P}$ is defined by:

$$\sigma(\mathcal{P}) = \{P_{n_1} \sqcup P_{n_2} \sqcup \cdots \sqcup P_{n_k} : n_1, n_2, \dots, n_k \in [1, N]\}$$

If $\text{card}[P] = N$, then $\text{card}[\sigma(P)] = 2^N$. If $\mathcal{Q}$ is another partition, we say that $\mathcal{Q}$ **refines** $\mathcal{P}$, if, for every $P \in \mathcal{P}$, there are the set of all $Q_1, \dots, Q_N \in \mathcal{Q}$ such that

$$P = \bigsqcup_{n=1}^{N} Q_n$$

Refine here can be thought of as being more explicit, or rather, more details - for example, a collection of marbles now shattered into pieces of sand-like particles, then we say that the power set of those particles is more refined than the marble covering previously. We then write $\mathcal{P} \prec \mathcal{Q}$, or $\mathcal{Q}$ *precedes* $\mathcal{P}$. It follows that:

$$\Big(\mathcal{P} \prec \mathcal{Q}\Big) \iff \Big(\sigma(\mathcal{P}) \subset \sigma(\mathcal{Q})\Big)$$

Borel sigma-algebra

Let X be any topological space, and let $\mathcal{M}$ be the set of all open subsets of X. The sigma algebra $\sigma(M)$ is the **Borel sigma algebra** of X, denoted $\mathcal{B}(X)$. It contains all open sets and closed subset of X, all countable intersection of open sets - called $D\delta$ sets, all countable unions of closed sets - called $F\sigma$ sets, etc. More specifically, suppose (X, ρ) is a pseudometric space. Then the smallest σ-algebra of subsets of X that contains every closed subset of X is called the **Borel** σ-algebra of (X, ρ). The Borel σ-algebra then contain also every open subset of X.

Product algebras

Suppose $(X, \mathcal{X})$ and $(Y, \mathcal{Y})$ be two measurable spaces, and consider the *Cartesian product* $X \times Y$. Let

$$\mathcal{M} = \{U \times V \mid U \in \mathcal{X}, V \in \mathcal{Y}\}$$

be the set of all rectangles in $X \times Y$. Then $\sigma(\mathcal{M})$ is the **product sigma-algebra**, denoted $\mathcal{X} \otimes \mathcal{Y}$.

5.2.3 Measure

We now move to the notion of a **measure**.

Definition 5.2.2 (Measure). *A* ***measure*** *on $\mathcal{X}$ is a map $\mu : \mathcal{X} \to [0, \infty]$ which is* ***countably additive****, in the sense that, if $Y_1, Y_2, \dots,$ are all elements of $\mathcal{X}$, and are disjoint, then*

$$\mu\left[\bigsqcup_{n=1}^{\infty} Y_n\right] = \sum_{n=1}^{\infty} \mu[Y_n]$$

Thus, we understand it as μ assigns a 'size' to the $\mathcal{X}$-measurable subsets of X. A measure $P : \mathcal{S} \to \mathbb{R}_+$ is said to be a **probability measure** if $P(X) = 1$. We refer to $(X, \mathcal{S}, P)$ as a **probability space**. For familiarity, it is perhaps better to get into some examples.

Example 5.2.1 (The counting measure). *The* ***counting measure*** *assigns, to any set, the* ***cardinality*** *of that set, that is:*

$$\mu[S] = \text{card}[S]$$

this is only useful in finite measure spaces, so we would have not that much application to it. However, it is a pretty natural measure to think of.

Example 5.2.2 (Finite measure space). *Suppose X is a finite set, and $\mathcal{X} = \mathcal{P}(X)$. Then, a measure μ on X is entirely defined by some function $f : X \to [0, \infty]$, for any subset $\{x_1, \ldots, x_N\}$. We then define*

$$\mu\{x_1, \ldots, x_N\} = \sum_{n=1}^{N} f(x_n)$$

We might want to show that every measure on X arises in this manner.

Example 5.2.3 (Discrete measure). *If $(X, \mathcal{X}, \mu)$ is a measure space, then an* **atom** *of μ is a subset $A \in \mathcal{X}$ such that:*

- $\mu[A] = A > 0$.
- *For any $B \subset A$, either $\mu[B] = A$ or $\mu[B] = 0$.*

For example, in the finite measure space above, the singleton set $\{x_n\}$ is an atom if $f(x_n) > 0$. The measure space $(X, \mathcal{X}, \mu)$ is called **discrete** *if we can write:*

$$X = Z \sqcup \bigsqcup_{n=1}^{\infty} A_n$$

where $\mu[Z] = 0$ and where $\{A_n\}$ is a collection of atoms.

Example 5.2.4 (Lebesgue measure). *The* **Lebesgue measure** *on $\mathbb{R}^n$ is the model of "length", "area", "volume" et cetera for real number system.*

Example 5.2.5 (Haar measure). *The Lebesgue measure has the extremely important of* **translation invariance**. *That is, for any set $U \subset \mathbb{R}^n$, and any element $\vec{v} \in \mathbb{R}^n$, we have*

$$\mu[U] = \mu[U = \vec{v}]$$

We can generalize this to any topological group, G. Let G have Borel sigma-algebra $\mathcal{B}$, and suppose η is a measure on $\mathcal{B}$ so that for any $B \in \mathcal{B}$, and let $g \in G$,

$$\eta[B.g] = \eta[B]$$

This is called **right translation invariance**. *IF G is locally compact and Hausdorff, then there is a measure η satisfying this property, and η is unique (up to multiplication by scalar λ). We call this η the* **right Haar measure on** *G. Again, there is a unique measure with this property, but on the left. Consider the left-translation invariance. For any $g \in G$ and $B \in \mathcal{B}$,*

$$\eta[g.\mathcal{B}] = \eta[\mathcal{B}]$$

This is called the left Haar measure on G. If G is abelian, then left-invariance and right-invariance are equivalent, hence the two measure agree. If G is not abelian, however, then the two measures might disagree. A question is then focused to see their commutativity (how much) between the two measures. If the left- and right- Haar measures are equal, we call G **unimodular**.

Example 5.2.6 (Haussdorff measure). *The Lebesgue measure is a special case of another kind of measure. Instead of treating $\mathbb{R}^n$ as a topological group, regard $\mathbb{R}^n$ as a metric space. On any metric space, there is a natural measure called the* **Hausdorff measure**.

Heuristically, the Hausdorff measure of a set U is determined by counting the number of open balls of small radius needed to cover U. The more balls we need, the larger U must be. However, for any non-zero radius R, a covering with balls of size R produces only an approximate measure of the size of U, because any features of U which are much smaller than R are not detected by such covering. The Hausdorff measure is determined by looking at the limit of the number of balls needed, as $R \to 0$.

Still, what does this do and how is this natural is pretty weird.

It is possible to define Hausdorff μ_d for any dimension $d \in (0, \infty)$. The dimension parameter d is allowed to take on *noninteger values*. What does this mean, we do not know. The dimension d describe how rapidly the measure of a ball of radius ϵ grows as the function of ϵ. We expect that, for any point x in our space,

$$\mu[B(x, \epsilon)] \sim \epsilon^d$$

For any metric space X, there is a unique choice of dimension d_0 that yields a nontrivial Hausdorff dimension. For any value $d > d_0$, the measure μ_d will assign every set measure *zero*, and for any value of $d < d_0$, the measure μ_d will assign every open set *infinite measure*.

The unique value d_0 is called the **Hausdorff dimension** of the space X, and carries important information about the geometry of the X.

Example 5.2.7 (An example of Hausdorff dimension). *The Hausdorff measure of $\mathbb{R}^n$ is n. Hence, if we try to measure the 'volume' (recall from the notion of Lebesgue measure, that is $n = 3$) of $\mathbb{R}^2$, it will yield 0. Conversely, if we try to measure the 'area' of the nontrivial subset of $\mathbb{R}^3$, the only sensible value to expect would be ∞.*

We can construct a Haar measure on any subset U of $\mathbb{R}^n$, by treating U as a metric space under the restriction of the natural metric on $\mathbb{R}^n$. For example, if U is an embedded k-dimensional **manifold**, then the Hausdorff dimension of U is K. However, there are also pathological subsets of $\mathbb{R}^n$ in which possess noninteger Hausdorff dimension. These objects are now called **fractal**, and the Hausdorff dimension is only one of many **fractal dimensions** which are used to characterize these objects.

Chapter 6. Concentration inequalities

A lot of results in statistical learning is proved using bounds. For example, if $X_1, \dots, X_n$ are independent Bernoulli (θ) random variables representing the outcomes of a sequence of n tosses of a coin with bias (probability of head), then for any $\epsilon \in (0, 1)$:

$$\mathbb{P}\Big(|\hat{\theta}_n - \theta \geq \epsilon\Big) \leq 2e^{-2\epsilon^2}, \quad \text{for } \hat{\theta}_n = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{6.1}$$

For the fraction of heads in $X^n = (X_1, \dots, X_n)$. Since $\theta = \mathbb{E}\hat{\theta}_n$, 6.1 says that the sample (or empirical) average of X_i concentrates sharply around the statistical average. Bound like these are fundamental in statistical learning theory (well, mostly for proof, not going to lie). For now, let us learn the technique used to derive such bounds for settings much more complicated than coin tossing.

6.1 What are concentration inequalities?

In a probabilistic setting, we are sometimes interested of the random fluctuations of functions of independent random variables. *Concentration inequalities* quantify such statements, typically by bounding the probability that such a function differs from its expected value (or from its median) by more than a certain amount.

concentration inequalities

The search for concentration inequalities has been a topic of intensive research in the last decades in a variety of areas because of their importance in numerous applications. Typically, this falls into the range of designing non-deterministic (by the definition of such words), dynamic randomized algorithms or procedure of interest, while able to bound them with concentration inequalities to a certain given degree of high probabilistic 'accuracy', for example. Among the areas of applications, without trying to be exhaustive, we mention statistics, learning theory, discrete mathematics, statistical mechanics, random matrix theory, information theory, and high-dimensional geometry (while not actually sure why geometry is in here).

Informally, the *concentration phenomenon* asserts that if $X_1, \dots, X_n$ are independent random variables, then $f(X_1, \dots, X_n)$ does not deviate much from its mean provided that $f(x_1, \dots, x_n)$ is not too sensitive in any of its coordinate x_i. While concentration properties for sums of independent random variables were thoroughly studied and fairly well understood in classical probability theory, powerful tools to handle more general functions of independent random variables were not introduced until the appearance of *martingale methods* in the 1970s, by Yurinskii (1976), Maurey (1979), Milman and Schechtman (1986), Shamir and Spencer (1987) and McDiarmid (1989).

martingale methods

Note 6.1.1. *While the topic itself is much more complex than it is guaranteed to be, a martingale can be defined as a sequence of random variables (i.e. for a stochastic process) for which, at a particular time, the conditional expectation of the next value in the sequence is equal to the present value, regardless of all*

prior values. A basic definition can be achieved. For a process $(M_n)_{n\geq 0}$, it is called martingale if:

- *For every $n \geq 0$, the expectation $\mathbb{E}[M_n]$ is finite, and hence equivalently, $\mathbb{E}[|M_n|] < \infty$.*
- *For every $n \geq 0$ and all $m_n, m_{n-1}, \ldots, m_0$ we have:*

$$\mathbb{E}(M_{n+1} \mid M_n = m_n, \ldots, M_0 = m_0) = m_n \tag{6.2}$$

6.2 Basic tools

To derive a lot of our concentration bounds, we have to develop a few basic tools in the analysis. This will include Markov's inequality, Chebyshev's inequality, the Chernoff bound (often called the Chernoff trick), and Hoeffding's lemma.

6.2.1 Markov's inequality

Let $Y \in \mathbb{R}$ be a nonnegative random variable. Then for any $t > 0$, we have:

$$\mathbb{P}(Y \geq t) \leq \frac{\mathbb{E}[Y]}{t} \tag{6.3}$$

Before jumping straight into a proof, let us see what are we looking at from the perspective of the inequality.

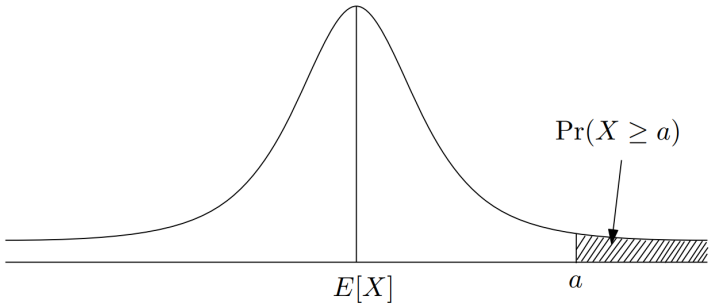


Figure 6.1: Markov's inequality bounds the probability for the shaded region $\mathbb{P}[X \geq a]$

Proof. The proof is straightforward:

$$\begin{aligned} \mathbb{P}(Y \geq t) &= \mathbb{E}[\mathbb{1}_{(Y \geq t)}] \\ &\leq \frac{\mathbb{E}[Y \mathbb{1}_{(Y \geq t)}]}{t} \\ &\leq \frac{\mathbb{E}[Y]}{t} \end{aligned} \tag{6.4}$$

We use the fact that

$$\mathbb{P}(Y \in A) = \int_A P_Y(dy) = \int_Y \mathbb{1}_{(x \in A)} P_Y(dy) = \mathbb{E}[\mathbb{1}_{Y \in A}] \tag{6.5}$$

Additionally, we also have $Y \geq t > 0$ implies $Y/t \geq 1$, and $Y \geq 0 \implies Y \mathbb{1}_{(Y \geq t)} \leq Y$ which implies their expected value. □

6.2.2 Chebyshev's inequality

Let X be an arbitrary real random variable. Then for any $t > 0$,

$$\mathbb{P}(|X - \mathbb{E}[X]| \geq t) \leq \frac{\mathrm{Var}[X]}{t^2} \tag{6.6}$$

Where $\mathrm{Var}[X] := \mathbb{E}[|X - \mathbb{E}X|^2] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$ is the variance of X.

6.2.3 McDiarmid's inequality

A generalization of Hoeffding's inequality is the McDiarmid's, or Bounded Difference inequality, where the quantity of interest is some function of the data, i.e. $S_n = \phi(X_1, \ldots, X_n)$. Some restrictions on ϕ are required to get exponential bounds.

Theorem 6.2.1 (McDiarmid's Inequality). *Let $X_1, \ldots, X_n$ be independent random variables, where X_i has range $\mathcal{X}_i$. Let $f : \mathcal{X}_1 \times \cdots \times \mathcal{X}_n \to \mathbb{R}$ be any function with the $(c_1, \ldots, c_n)$-bounded difference property: for every $i = 1, \ldots, n$ and every $(x_1, \ldots, x_n), (x_1', \ldots, x_n') \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ that differ only in the i-th coordinate, we have:*

$$|f(x_1, \ldots, x_n) - f(x_1', \ldots, x_n')| \leq c_i \tag{6.7}$$

For any $t > 0$:

$$\mathbb{P}\Big([f(X_1, \ldots, X_n)] - \mathbb{E}[f(X_1, \ldots, X_n)] \geq t\Big) \geq \exp\left(-\frac{2t^2}{\sum_{i=1}^n c_i^2}\right) \tag{6.8}$$

II. Theory

The classic of the old, the foundation of the past. What is of their uses? Theory and theory. All those conjecture and ideas, in one single place.

P. Q. Doyle

Chapter 7. Classical mathematical modelling

Often time when we refer to modelling, we consider it to be *mathematical modelling* instead. Because our interest aligns with such, we will talk all about modelling, and its principle, in this section.

We begin this section to mathematical modelling and simulation with an explanation of basic concepts and ideas. Generally, this includes exactly defining the terms **system**, **model**, **simulation**, **mathematical model**, and related notions or subcontext.

We also discuss and analyse on the reflections of the objective of mathematical modelling, and simulation, on characteristics of "good" models, and classification of mathematical models. Though it is strongly advised to take the following section in a *classical sense*, as we would eventually find out, that the notions and concepts presented here will be vastly outdated - and hence needs quite a bit of update to stay up with the current state-of-the-art theory.

Our starting point is the complexity of the problems treated in science and engineering, because from here, is the first case we even need to create a **model**. There exists a lot of **non-mathematical models** in the past, however, and we shall see how the language of mathematics will help in this role. Vel [2024] will be the main text of this section.

7.1 The supposed goal of modelling

In science, a *system* refers to the object of interests, which can be a part of nature (atoms, etc), or an artificial technological system. Principally, we do this all days, with similar approach to isolate the problem into its respective system.

The more **complex** a system is, the harder it is for dealing with the objects in that system, and the more intricate relationships and factors contributes to specific observations and problem that dilute the solution. It is the genuine task of **scientist** and **engineers** then, to deal with complex systems, and be effective at it, to deal with the extended complexity requires to solve the problems inferred in the system itself.

For a very complex system, the first step would be simple - be simple. That is, we use **simplification** of such system to start solving it. Usually, to solve specific problems given a system, a simplified view can pinpoint exactly the ergonomic of the problem - what is important, what to discard, and what to consider.

Take an example of a car. Suddenly, it does not start. 90% of the time, trying to look into its tank, or battery, will solve the problem. While doing this consciously, what we have done is the simplification of the concept of a car. A car, by itself, is a very complex system full of mechanical moving parts and intricate connections. However, under the question of *why it cannot run*, a simplified picture focuses on *what enables it to run* in the first place, result in pinpointing the first two choices - which equates to the fact that 'a car need fuels to run, similarly needs energy to work'.

This simplified picture is appropriate, *most of the time*. If this does not work, our focus shifts to the next simplification - the wheel differential is not working. And that continue. The ergonomic here works similar to deduction and elimination. Once factors are eliminated, the one remaining will be the root of the problem. Instead of solving the system immediately as a whole, we targeted it by analysing the components it brings.

In another different way, simplification does help with understanding complex system, is the effect of **scale**. If one want to study photosynthesis, they would go for a single cell instead of millions. By certain assumptions and pervious conjectures, we then can analyse the single cell, and generalize it for the entire millions by uniformity. If one wishes to learn how the brain works, they would focus on the smallest component that makes up the brain first.

To break up the complexity of a system under consideration, we need to use simplified descriptions of that system - or **models**. This can be done either by elimination, abstractions, or else. But first, we need the definition of a model.

7.2 Models, systems, and questions

model

For the beginning of the discussion, we would like to examine the concept of a model, from a given perspective. The following definition follows from Marvin Minsky (1965):

Definition 7.2.1 (Model). *To an observer B, an object A^* is a* **model** *of an object A to the extent that B can use A^* to answer question that interest him about A.*

The definition is pretty much very "purposeful" definition, one can clearly see the dependency of the existence of a model A^* onto an object A. Note, that in the definition it implies that *no model is perfect* - by the priori, there exists the observer B, hence their interpretation might, or perhaps will be subjective. For example, the model of probability that it will be raining in Detroit being 80% in the first two days, and 20% for the remaining 5 days of the week, can be interpreted differently as 100% percent raining for the first two, and 0% otherwise, assume the record is correct. But under the consideration and the problem setting, the model is indeed perfect by itself. The definition is hence a *formal definition*, in the sense that it operates with terms such as *objects* or *observer* that are not defined in a strict, axiomatic sense in mathematics. In fact, it is not so effective to think of defining axiomatically, since we can have both informal and formal axiomatization.[1]

An important aspect of the above definition is the fact that it includes the purpose of the model, which helps us to solve the question and bring solution to problems. This is the reason for the principle often used as the best model, and in machine learning, associated with Lord of Occam's principle - the best explanation is the *simplest* one on the tray. But this point comes with doubts.

It turns out, that simplicity is not always the good one. On the contrary, it is the bad one, depends on the situation arises, and so is the complex one. There are inquiries taken in care of the word *simple* and *complex* in such scenario. Often time, sometime is called **simple** if its descriptions are minimal, along with assumptions that go aside with it. These assumptions is a lot of time the failure of the simple system. A system is called **complex**, if it contains rigorous descriptions, with what was assumed to be constrained, often in a precise and affective manner. Specification of such aspect, which brings more consideration in, is what makes a system

[1] The issue here might be more nitpicking than not. But, under specified circumstances, we can say that *formalism* works by describing and structure the system in a strict sense, without relying on meaning, and pure specification. While *axiomatization* focus on the structuring itself - will it is the way that everything is built upon the set of *axioms*, no further question, held as truth, in which all other results follow? Hence, in such sense, formalism looks into the description and the abstraction of objects living in such space, while axiom determine the logic and dependency that follows.

complex. As per our example of the car, the simplified notion only comes, when we assume all others components are negligible, and focus on the assumption that what matters is the battery and the fuel cell.

The issue here then lies, with the notion of the *question* and *problems* instead with the simple and complex dilemma. Sometimes, simple question requires complex models. Sometimes, complex problems require simple models. Sometimes, complex problems require gradually increased complexity of a model. The principle on the focus of the teleological nature of the problem the model pertains to, only applies in the *ergonomic sense* of *reasonable explanation* - often time, this reasonable explanation can be 30% correct, but still, they are correct 30 out of 100. Teleological means purpose - or at least is interpreted as - and if we only constrain ourselves to the purpose, *only the purpose*. Taken the example of linear regression to a particular problem that is not linear. Sure, we have satisfied the purpose, but the accuracy will inherently very low because it is trying to do what it cannot do - trying to figure out the nonlinear relationship, while constrained to be linear.

Under such term, we might as well think this principle of *best model* as pretty misleading. Rather, we say, we have the principle of *effective model*. An effective model is a model that serves its purpose, specified correctly, given that the *relative complexity* falls into the range of what can be considered *relatively simplification* of the space of all possible solutions. That is, it can be simple, but not the simplest. This, still, does not guarantee that the effective model will be better than the more complex model. In fact, it only establishes the lower bound. We ought to remember that. **Oversimplification** is a bad thing, while **overcomplication** also retains in the spectrum.

Returning to the issues we raised it is also true that the non-exact notion of simplicity and complexity is rather, **complex**. To do this though, we need to know what the 'purpose', or the question is required. A question is based on the system that it is in. In a complex system, a simple question is a hard one, and a complex question is a simple one, which is why it needs complex model to constrain it first to specifiable case. Hence, each question carries with it the *scope and freedom* of the question. In a simple space, a complex question is the harder one - unnecessary even - while the simple question fits with what the system is intended for. If the question is iterative, or expanding, this notion also follows. Overall, it is perhaps subjective to the frame of reference.

7.2.1 The modelling scheme

Conceptually, the investigation of complex systems using models can be divided into the following steps:

Proposition 7.2.1 (Modelling and simulation scheme). *The modelling and simulation scheme can be attained as followed:*

1. ***Definitions**: Definition of a problem that is to be solved, or of a question that is to be answered; and of a system, that is, a part of reality that pertains to this problem or question.*
2. ***System analysis**: Identification of parts of the system that are relevant for the problem or question.*
3. ***Modelling**: Development of a model of the system based on the results of the system analysis step.*
4. ***Simulation**: Application of the model to the problem or question and derivation of a strategy to solve the problem or answer the question.*
5. ***Validation**: Does the strategy derived in the simulation step solve the problem or answer the question for real system?*

In real modelling and simulation project, the *system analysis* step can be very time-consuming.

It will usually involve a thorough evaluation of the literature. In such step, experimental program is also a typical part.

The modelling and simulation scheme focuses on the essential steps of modelling and simulation, giving a rather simplified picture of what happens in a concrete project. Though, as we spoke in the last section, start with the simplest possible model, and then generate a sequence of increasingly complex formulation, until the criteria is sufficed.

7.2.2 Simulation

So far, we have given a definition of the term *model* only. The above modelling and simulation schemes involve other terms, such as system and simulation, which might be viewed as implicitly defined. However, we shall make this to be more precise in meaning. In general, it can be as the following.

simulation

Definition 7.2.2 (Simulation). *Simulation is the application of a model with the objective to derive strategies that help solving a problem or answering a question pertaining to a system.*

This definition, explicitly, also is defined to emphasize the purpose. Though, this purpose again, can be replication itself. So, the argument against replication as "l'art pour l'art" really does not help here that much.

On the side note, as always, there is also the question of what kind of simulation we are talking about. If we follow the narrative of simplification, there can exist the physical simulation of a simplified system - by itself is also just a simulation in which physical realization of objects are needed. On the other hand, there is also the non-physical, in a sense, simulation, which makes use of particular information-encoded representation - or rather, a *warped representation* using another physical system to interpret and simplify the physical realization that is done or used by the 'real thing', and go on with said representation or language. That is where we use computers or any representative system with computational power. This distinction might be detrimental in the process of solving the problem itself.

7.2.3 System

Our view of systems is similar to a definition by certain someone who like being stuck in a roundabout: "A system is whatever is distinguished as a system". And then, the teleological principle-based definition is as follows:

system

Definition 7.2.3 (System). *A system is a collection or a collection of objects whose properties we want to study.*

It is wise to notice in such, that a system can be deceiving. Taking an example, or rather, a typical scene often seen in the field of thermodynamic, there is the saying that "whoever breaks the second law, ultimately wields the power of infinite." - or rather, defying the second law of thermodynamic to gain infinite power, without putting in any work W.

This prompted people to set up their own system, and try to make amend of the law. Many attempts tried to break the status quo (which might never be broken), and yet none works, more so being called crackpot, that guarantee the saying "the only issue of perpetual motion machines (the one that defies the second law) is to find where they hid the battery.". It's not always the battery every time, but the one that claims that the second law has been broken, always *messes up their system*. By one way or another, their system is inconclusive - there are external influences, influx, or resources pouring in, that does not exist in the considered system. Hence, by extension, the 'infinite energy' comes from outer sources - undoubtedly infinite energy within the flaw system.

So, long story short, perhaps, in the classical sense of the use of defining a system, please define it carefully. Or then someone will claim that you can create something out of nothing, that would be troublesome.

7.2.4 Conceptual and physical model

As we have been saying on section about simulation, the same notion of realization in physical sense also manifests in the sense of conceptual or physical model. We will see about it.

There are two types of model. There exists the *conceptual (theoretical) model* , which lives in a theoretical world without physical realization. This is where we have what is considered as *thought experiment*, where certainly, with physical realization removed, the representative power of shooting reality up in the sky for unsurmountable scenario to be possible within theories - is allowed.

theoretical model

Against it, such an experimental setting, that simplifies the problem of the engine to its smaller replica, and solve the problem directly on such replica is call the *physical models* . In contrast, as transparently presented, it is not only an idea in our mind, but also a real part of a physical world. Any conclusion drawn from such physical model corresponds to the simulation step of the above scheme, and the conclusion need to be validated by the real system, that is, from the real plant, or the real car instead of its smaller simpler replica.

physical model

7.3 Mathematical models

By the old principle (which are being discussed), any system that is investigated must be observable in the sense, that it produces some kind of output that can be measured (a system that would not satisfy this requirement, would have to be treated by theologians rather than any practical purposes). Overall, if we are to put our perspective in, then *scientist or engineers* investigate "input-output systems", which transform given input parameters into output parameters. This simplification of the entire dynamic often helps in constructing reasonable and manageable system of interest.

Note that, however, that the picture is not always so bland of only the input-output system treatment. For example, when a botanist just wants to describe and classify the anatomy of a new discovered plant, we don't make it output things, but generally, it is the most basic example conceivable if we are to study how they function as the result of the examination of the system itself.

The experimental procedure described above is used very generally in engineering and in empirical sciences. It is useful to think of them as exploring *black boxes*. This term suggests the uncertainty about the processes that happen inside the system, when the input is transformed into output. In an extreme case, the experimenter may know only that "something" happens inside, but nothing. However, typically, the experimenter will have some hypotheses about the internal process, which can be proved or disproved.

Depending on the hypothesis, the experimenter will have his hypothesis of appropriate input, to be disproved or not, using the system's outputs. This is similar to a question-and-answer game - the experimenter poses questions to the system, which is the input, and the system answers to these questions in terms of measurable output quantities. This is typically similar to the questioning of an *oracle* — we know there is some information about the system, but it depends on the application of ideas and methods if one wants to uncover the information content.

The role of experimental data

Now, we ask the question of: what is an appropriate method for the analysis of experimental datasets?

To answer this question, it is important to note that in most cases experimental data are numbers and can be quantified. The input-output data will typically a table, and it is natural to think of it as a mathematical system, for example, think of it as mathematical function.

This means that if one wants to understand the processes inside the real system that transform input into output, a natural thing to do is to translate all these processes into mathematical operations. If this is done, one arrives at a simplified representation of the real system in mathematical terms. This simple idea, mapping of internal mechanics of real systems into mathematical operations, has proved to be extremely fruitful to the analysis of system.

Though, what should be hold accountable of certain success, for a scientist, must be the appropriate use of mathematical models itself.

7.3.1 Definitions

To understand mathematical models, let us start with a general definition. An attempt will lead us to the following:

Definition 7.3.1 (Mathematical model, naive form). *A mathematical model is a set of mathematical statements M of the form $M = \{\Sigma_1, \Sigma_2, \ldots, \Sigma_n\}$.*

Certainly, this definition covers all kinds of mathematical models used in science and engineering. But, there is a problem, because under such definition, even $f(x) = \exp(x)$ is some kind of mathematical model, which it is not. Following the philosophy of the teleological definitions, we gain the more sophisticated definition, in which one have to mention all the parameters, all the objects, the criteria, question, problems, and the system.

mathematical model

Definition 7.3.2 (Mathematical model). *A mathematical model is a triple (S, Q, M) where S is a system, Q is the question query relating to S, and M is a set of mathematical statements $M = \{\Sigma_1, \ldots, \Sigma_n\}$ used to answer Q.*

Note that this is again a formal definition in the sense of the previous construction. Again, it is justified by the mere fact that it helps us to understand the nature of mathematical model, and that it allows us to talk about mathematical models concisely.

The notation (S, Q, M) defined above emphasizes the chronological order in which the constituents of a mathematical model usually appear. Typically, a system is given first, then there is a question regarding that system, and only then a mathematical model is developed. Without S, no questions can be asked of Q, and without Q, we cannot do anything to the model.

The system and the question relating to the system are indispensable parts of a mathematical model. It is a genuine property of mathematical models to be more than mathematical "l'art pour l'art".

Example 7.3.1 (The importance of asking Q). *Suppose that we want to predict the behaviour of some mechanical system S. Then, the appropriate mathematical model depends on the problem we want to solve, that is, on the question Q. If Q is asking for the behaviour of S at moderate velocities, classical (Newtonian) mechanics can be used, that is, $M = \{\Sigma_i\}$ of all Σ formulas on Newtonian mechanics. If on the other hand, Q is asking for the behaviour of S at velocities close to the speed of light, then we have to set $M = \{\Sigma'_j\}$ of relativistic equations instead.*

7.3.2 State variables and system parameters

The main benefit of the modelling procedure lies in the fact that the complexity of the original system is reduced. For example, taken a real world problem the entire system parameterized can be expressed to be infinitely many dimensions - that is, there are too many factors of concern, that a lot of them are irrelevant of the problem in consideration. By specifying the mathematical model, or the general modelling scheme, we reduce the infinite system to a small *reduced system* in which our problem is perhaps concerned of.

As a result, it is imperative for us to define formally the reduced system, for any possible conceivable system that one might encounter. To do this, one need the definition of *state variable*. Later on, we would also have the notion of a more reserved *system parameter*.

system parameter

Definition 7.3.3 (State variables). *Let (S, Q, M) be a mathematical model. Mathematical quantities $s_1, \ldots, s_n$ which describe the state of the system S in terms of M and which are required to answer Q are called state variable of (S, Q, M).*

Using this, we can then define the notion of a reduced system.

reduced system

Definition 7.3.4 (Reduced system and system parameters). *Let $s_1, \ldots, s_n$ be the state variables of a mathematical model (S, Q, M). Let $p_1, \ldots, p_m$ be mathematical quantities (numbers, variables, functions) that describe properties of the system S in terms of M, and which are needed to compute the state variables. Then $S_t = \{p_1, \ldots, p_m\}$ is the reduced system and $p_1, \ldots, p_m$ are there system parameters of (S, Q, M).*

This means that state variables describe the system properties we are interested in, while the system parameters describe the system properties needed to obtain the state variables mathematically. Sounds pretty ambiguous, but we can think of this separation as a kind of subjective intrinsic property expression - there are intrinsic system properties altogether, plus there are intrinsic system properties that answer Q, and there are indirect system parameters that can be used to acquire those, in the reduced system, in a much smaller parameter set.

We can have some examples on this. For example, give someone a bunch of sheets of metal, and a sample tin filled with water. Make one similar, with minimal material. What can be done about this problem? To solve this problem, we have to see through the surface area of all the tin used in each configuration. That is, you can make a bunch of tin, no doubt. But to satisfy the condition of minimality, you need to figure out the dimension measure of the tin, in which case, here, it is the surface area of every piece of tin used. Hence, we have the state variable $s_1 = A$, for $n = 1$. The tin is cylindrical, so you have to specify its radius, and its height. Hence, the reduced system specifically for this problem is $S_t = \{r, h\}$. Notice how we disregard m, because they are of all similar material. We also disregard the thickness d, because apparently, all the sheets are the same, and we have no tools to thin them out.

A question might come off pretty natural from those questions. IF you present this to certain someone in a conference, or just discussion, you will have to prepare to hear "Why does your model disregard…". Countering this question can be simple as to answer: we know that according to 7.3.4, a mathematical model of triplet (S, Q, M) will only have the details that is sufficient to answer (so depends on it) Q, formulated by S, and represented (connected) in M. In such case, certain set that have been introduced can sufficiently answer Q, and that is our model and parameters, per our assumption about the problem setting itself. Generally, one can say that the reduced system of a well-formulated mathematical model will consist of no more than exactly those properties of the original system that are important to answer the question Q that is being investigated.

However, that is normally, in typical situation, too ideal of a process. Indeed, one might have to also prepare for the answer "To be honest, you are right, we disregarded something that we thought is irrelevant, but certainly not", or "We don't know about that factor.". One simply in reality often have no full picture of the underlying problem, and only after many experiments, testing, hypothesis cancellation, and modification that one can find their exact satisfactory result. And even then, specialities might require the model to extend those parameters to fit the more specific situation. Furthermore, we recite our opinion that *no model is perfect, but only useful*. Hence, subjective being natural, we might have cases where the reduced model cannot capture all the system parameter as required.

Remark 7.3.1. *Typically, the properties (parameters) of the reduced system are those, which need experimental characterization. In this way, the modeling procedure guides the experiments, and instead of making the experimenter superfluous, it helps to avoid superfluous experiments.*

7.3.3 The Problem-Solving Scheme

In many practical applications and case-to-case situations, one can clearly distinguish between the formulation of a mathematical model on the one hand, and the solution of the resulting mathematical problem on the other hand, which can be done with appropriate software. A number of examples will show this below. This means that it is not necessary to be a professional mathematician if one wants to work with a mathematical model, though it is recommended so, or at least of certain rigours to define and integrate the mathematical model itself.

Though, the story is a bit more complicated. Mathematical expertises will be required, however, and is particularly important if one wants to solve more advanced problems, more complex problems, or if one wants to make sure that their results obtained with mathematical software are really solutions of the original problem and not numerical artifacts. Though, even when we are quite sure of the mathematical expertises, we can still be wrong in our own hypothesis.

Note 7.3.1 (The abstraction and role of software). *In general, because people working with mathematical models has now switched to computational system, typically, formulation of a mathematical model is clearly separated from the solution of the mathematical problems implied by the model. Not sure what this exactly means of the abstraction though. The latter hard work can be done by software, as a result. And even software can be abstracted because they are not worried of the underlying hardware. So, one can go off without others.*

Many problems in science can be solved using mathematical modelling, as a matter of fact. From certain perspective, the abstract world view coupled with mathematics provide them with the organized and quantized abstraction, to the point that they can utilize those abstract structures, mathematics methods and instrument to solve the problem. The mathematical universe standing alongside the real world problem can be represented as be separated in transition from the mathematical model (S, Q, M) acting as the transit hub - transferring the problem into the mathematical terms. Of course, this is not always the case, however, usually, we can do it, with a very simplified view.

As the figure shows, the mathematical model virtually controls the 'problem-solving traffic' between the real and mathematical worlds, and hence, its natural position is located exactly at the borderline between these worlds. A more realistic approach to drawing Figure 7.1 would be to extend the distance between (S, Q) to A in the real world to an abnormally large distance, such that somehow, under the mathematical lens, it is smaller, more organized, and easier to approach it. That is when you know the mathematical modelling is helpful.

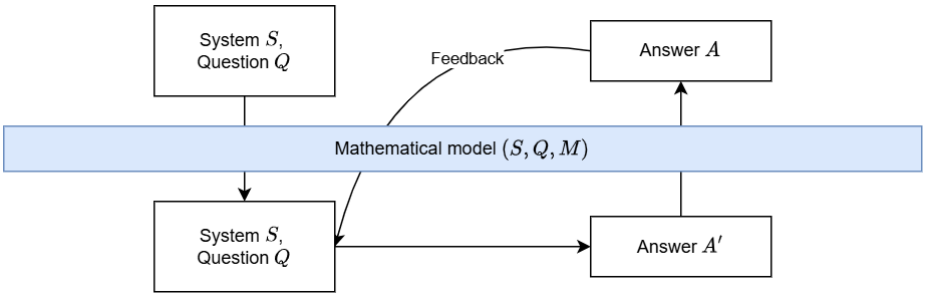


Figure 7.1: The problem-solving scheme from the mathematical modelling perspective

Setting up a mathematical model is also easy. Usually, the guideline for this transition can go as followed:

1. Determine the number of unknowns, that is, the number of quantities that must be determined in the problem. Well, read them all and read until the end, that is.
2. Give *precise* definition of the unknowns and the relating components. This should not be lumped with step 1, just as with concise implementation and conceptual modelling are not the same.
3. Read the problem formulation, translate it to mathematical statements to gives $M = \{\Sigma_1, \ldots, \Sigma_n\}$.

In step 1, if we are taking in a physical problem, it is also an issue to tackle its **units**, or **dimension** in case of dimensional analysis. By the standard rule, both side of the statement or a mathematical relation that captures the target of the model must have same dimension throughout the transformation and statements by itself, so you will have to be careful in that case. Statements about the mathematical models are then called the **restriction** on the mathematical model in specific.

Also, in some cases, the translation of a problem into mathematics may require the introduction of *auxiliary variables*. These variables are "auxiliary in the sense that they help us to determine the unknowns. Usually, the problem formulation will provide enough information such that the auxiliary variables and the unknowns can be determined.

7.3.4 The black-box interpretation

We have been introduced, or rather, quite accustomed to the notion of black-box interpretation and its polar opposite of white-box interpretation. However, what shall be made of their uses? Let's have a look at it. While doing so, it will perhaps also reveal the reason there exists phenomenological and mechanistic model.

In previous section, it was mentioned that the system investigated by scientists and engineers typically are "input-output" system. This means they transform the given input parameter into output parameters. Note that the previous examples were indeed referring to such "input-output program". In the tin (cylinder cutting) problem, the radius and height of the tin are input parameters and the surface area of the tin is the output parameter. In the plant growth example, the growth rate of the plant and its initial biomass are the input, and the resulting time-biomass curve is the output. Similarly, all systems in the examples and practical cases that will follow, can be interpreted as input-output systems.

The exploration of such input-output system will eventually give us the details on a more importance concept, and more definitions.

A system is called a **black-box** if there exists no internal information beforehand about the

system. Given the input-output treatment, then it means there exists no information of x and y, except that they exist. Then, one of the main thing to do of such black-box situation, is to use **statistical method**.

Given such system, we indeed see nothing. In the testing and observation phase, we will only be given their result - of the model, and its behaviours without having anything to double-check such. Hence, we will have to learn the pattern of the data by itself. That is what meant of by statistical method.

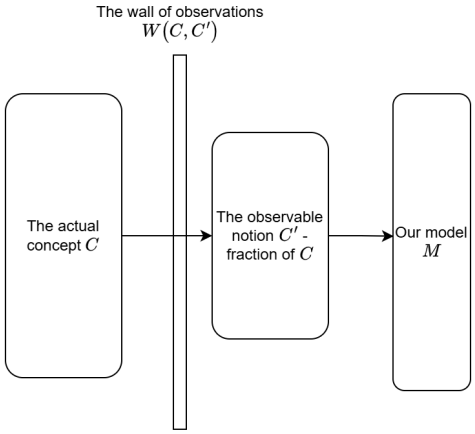


Figure 7.2: There exists an unbreakable wall in the black-box condition - throwing a dart in blind, except perhaps it can be right.

7.4 Flavours of modelling definitions

In our investigation, we have outlined a mathematical model of the description (S, Q, M). With such description, one can ask depends on how the three factors and macro-component is constructed or defined, can we define, relatively, mathematical models into several types, of which their intrinsic properties to be constructed of, are restricted instead of being generalized. The answer is yes, and as we have seen with the consideration of black-box modelling and white-box one, there are plenty to say about it.

7.4.1 Phenomenology and Mechanistic

The black-box interpretation gives us many formulations. That includes the categorization of mathematical models to be phenomenological and mechanical. Roughly speaking, a model is called *phenomenological model* if it is based on observations only, treating the system as an entire black box, without any priori of the internal process. On the other hand, if you use some sort of priori knowledge of the internal process in the design of the (mathematical) model, then it is called a *mechanistic model*. The difference can be either substantial or trivial, but they are indeed can be said or thought as the two polar opposite - mechanistic being absolutely based off prior information, and phenomenological is entirely priori-ly blind. We give the following formal definition, because I hate informal notions:

phenomenological model

mechanistic model

Definition 7.4.1 (Phenomenological and mechanistic models). *A (mathematical) model (S, Q, M) is said to be:*

- ***phenomenological*** *if it was constructed based on no priori information, or bias about the internal process. That it, it only uses experimental observations only, and nothing about S.*
- ***mechanistic*** *if some of the statements in M are based on a priori information about S. In the maximal case, all of them or the substantial amount of core statement in M is dependent of such information.*

Usually, phenomenological also has the name *empirical models, data-driven models, descriptive models, statistical model*, or *black box models* for obvious reason. The act of automating the process of creating phenomenological model, and 'correcting it' to a course, is called **machine learning**, in a rough sense. Normally, any give model would be located somewhere between the extreme black and white box cases - one such example is between solving an established system by giving it all the laws, for example, a pendulum by Newtonian physics - and the black box cases, for example, use the minimal information about the function of the observed object. Such models are sometimes called *gray box* models, or *semi-empirical models* . gray model

To understand it better, let us begin with an observation on the system of a pendulum.

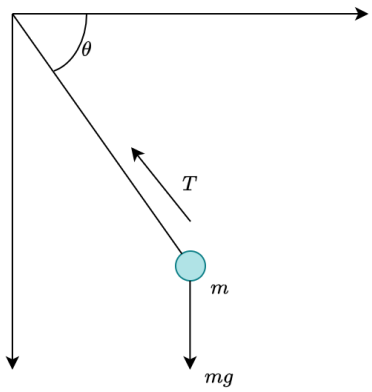


Figure 7.3: A typical pendulum with degree 1, for parameter θ as angle, and a rod of length ℓ connecting the origin to the mass m.

We can treat this system entirely as a black box model, in which system 1 only transform inputs x from the external setting, to the output y observed. If we know assume that we know the detail, that is, the input x gives us the angle θ of the pendulum, we know that the length is fixed $\ell = const$, and y gives the kinetic energy of the pendulum at any given angle, then we have given the model a **priori information** about the system 1 and how they can probably influence the system. Then, using that, we can restrict the modelling system to something that is more optimize to mimic this type of behaviour from the system. This is different from a black box system, where you have to consider the infinite hypothesis space that might contain yours.

Now, what can be made of the system and its relation? Of course, if we reside in the topic of Lagrangian mechanics, we will see that the kinetic energy is calculated, from the angle θ, by the following formula:

$$T = \frac{1}{2}mv^2 = \frac{m\ell^2\dot{\theta}^2}{2} \tag{7.1}$$

Where $\dot{\theta}$ is the time derivative for such measure, that is, θ is parameterized to $\theta(t)$. Based on this relation, we then obtain the (S, Q, M) model:

- S: System 1.
- Q: Which system input x generates a desired output of $y = T = 25.4$?
- M: Equation 7.1.

Based on this, the question Q can be answered by setting only $T = 25.4$ into the equation:

$$25.4 = \frac{m\ell^2\dot{\theta}^2}{2} \tag{7.2}$$

That is, we can answer the question Q by simply specifying the dependent variable, like how long is ℓ, and how heavy is the mass m, and we are practically done. This is one of the main advantage of the mechanistic model, in comparison to their phenomenological counterpart. Firstly, mechanistic model are generally better for predicting system behaviours, and is also generally far more stable in this role. The phenomenological model might only work for the variety in which our experiment are conducted. That is, for example, in the range of $\theta \in [90°, 147°]$. Other than that, we are not so sure. On the other hand, mechanistic model is based on the well-understood nature of physics. This bears the consequence that we know that it will work, at much of this system, for any given range of situations. This will come in handy, especially when we discuss about machine learning, since in machine learning, one of the very much conditional assumption, is the fact that the dataset must be somewhat representative, or, reflects the entire population.

Mechanistic models also allow *better predictions of modified systems*. This is a given, since the system's behaviours are spelt out, so there are not much you can do about it. Even for extended system, you will only increase the total number of objects in the system, and the law of the specific system setting will not change, as much as the factors are concerned. Assume, for example, that system 1 is replaced by a system 2 that consists of, instead, a different type of rod, like two rods connected together by some angle ψ. Then, in the phenomenological approach, the model developed for system 1 would be of no use, since we would not know about the similarity of these two systems. This means that a new phenomenological approach would have to be developed. Instead, in our mechanistic system, we conveniently use a coordinate-based system that interpret the change in the configuration of the rods as coordinate, so that the system kind of, remains the same, just with different ℓ - the laws do not change.

The third advantage of mechanistic models is the fact that they usually involve the interpretation - the relation by itself, and what the parameters mean to the system and realizable objects. For example, using the relation established to perhaps optimize the system performance, assuming particular objective for the system. For example, if we do this with phenomenological approach, we will have to do this problem by the trial-and-test method. Typically, this is done by randomization and point test method, which might take a long time, and perhaps pretty inefficient. Monte Carlo-like method, but going blindfold, if you insist. By then, you have to rely on statistical and probabilistic features to guarantee the gain from such method.

All being said, mechanistic models are pretty comprehensively better than phenomenological model. Indeed, it is more stable, more interpretable - in the sense that you know the internal mechanics, you know what does what, and which is affected by whom - in so far also the fact that it is generalizable for special parameters, for example, if we are to examine a spring system, then now adding two spring will break the previous one spring phenomenological model. But if so, then why even use phenomenological model? Well, two main things. An essential prerequisite of forming mechanistic model is that you need a *priori knowledge requirement* to feed the model, about the system. If nothing is known about the system, then we are just having a black-box system, and phenomenological models. Furthermore, in some cases, our knowledge

may be not enough as it is, and our relations are ambiguous to be applicable. For example, suppose we are tasked with understanding why some roses wilt earlier than others. Suppose we have some preliminary knowledge that it depends on the concentrations of certain carbohydrates that can be measured. We then have the formula of $M : \{C_{12}\} \to$ roses. However, we do not have the explicit parameters or their connection forms involved in such relation. A drawback of a purely mechanistic model is that the underlying internal mechanism must be known explicitly, because the entire model depends on the structure of the internal mechanics. By then, unless these processes are known, all we can do is produce some data and analyse them using phenomenological models. A mix of both, however, is possible, using some obscure knowledge about the system of itself. On the other hand, even if we know the system and its internal mechanics, sometimes it would be too complex, too cumbersome to set up a mechanistic model. Because of the explicit nature of the mechanistic system, and half-baked, detail-lacking mechanistic model will fail, perhaps without surprises. In such way, phenomenological model certainly helps, as it practically requires little to no priori knowledge. However, it is also to be noted on *how limited is phenomenological model*. While it is said to be, seemingly universal, which it is, the kind of black-box investigation is purely limited in its scope, its again, interpretation, and its utility. Mechanistic models then, allows for the depth of the system to be discovered, yet require plenty support and more time and resources. Coincidentally, we can somewhat do it of making phenomenological model to be the width, and mechanistic to be the depth, so to speak.

7.4.2 Stationary and unstationary models

It is already mentioned that the question Q is an important factor that determines the appropriate model (S, Q, M). Hence, depends on the way the question is constructed, we will be able to find certain characteristic of the model by itself. Specifically, consider our question of the pendulum. We have noted of the question to find specific relation between kinetic energy and the angle θ, that gives us the expression we perhaps are familiar with. Now, we might change the question to be:

"*What is the resulting kinetic energy change from the position* $t = 0$ *to* $t > 0$*, for the angle* $\theta(t)$ *depends on time* t*?*"

For such question, our type of models as above is unable to process this type of question. This is because experiments and observations considered of such models are **stationary** in nature - it involves no changing dynamics, and is rather a collective snapshot collection to be studied from, all of which might fortunately turns out to be of similar rules or patterns, as in the phenomenological case. On the other hand, the above question requires consideration about the time-evolution of the system by an independent variable t, or time. To even solve this question, we need another expression in M for specifically this time-dependency. This gives us the definition between **stationary** and **unstationary** model.

Definition 7.4.2 (Stationary/Unstationary models). *A mathematical model* (S, Q, M) *is called:*

- *Unstationary if at least one of its system parameters or state variables depends on time.*
- *Stationary otherwise.*

While this definition might lack rigours if one wants it to have, it is good enough for our definitions by the question Q it considers.

7.4.3 Classification of models

Certainly, one can already observe that we only take on the role, and the analysis of mathematical models and form definitions about it, based solely on the foremost consideration of the

question that was asked, Q. Naturally, this can be done for the variety as it is, for other 'axis' of mathematical modelling - categorization based on S, categorization based on M, and so on for some macroscopic and designing-based axis of interest. With this, many attempts have been made trying to organize those specific iterations of modelling, and their criteria. Though the list is indeed long and complex, plus the potentially obscure nature by attempting something inconclusive in which we will not discuss, we can however, ponder on how the axis is typically considered in the criteria provided.

The "space of mathematical models" evolves naturally from Definition 7.3.2, where we have defined a mathematical model to be a triple (S, Q, M) consisting of a system S, a question Q, and a set of mathematical statements M. Based on this definition, it is natural to approach the classification problem using the SQM space. For each of the axis, many criteria can be settled upon which the axis might represent. For example, a particular example that has been used throughout this chapter is the black-and-white box axis for the system S. This is similar for Q as well, for example, instead, Q-criteria can be configured to categorize increasingly complex and difficult problems and their subsequent modelling type between the transition from black-box to white-box model. In another way, you can, well,... even clarify with this:

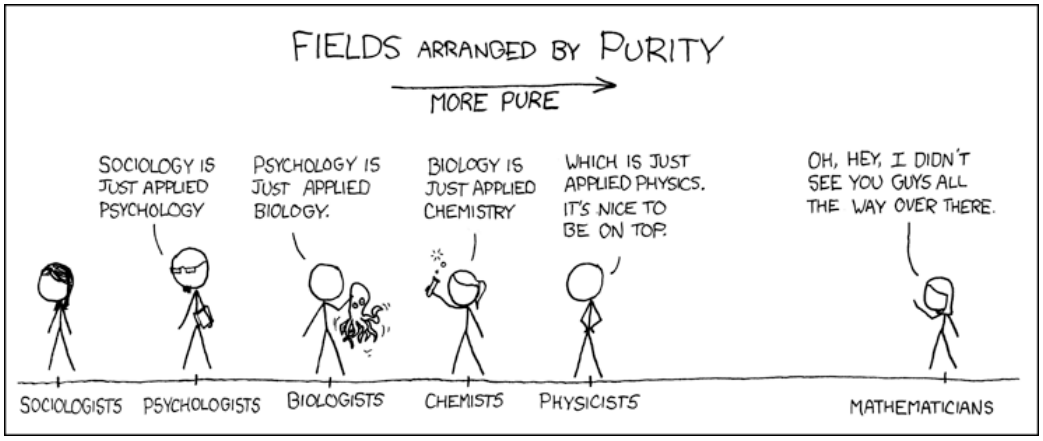


Figure 7.4: With the question Q, you can ask everything, including the... not so pure one.

I hope no one use it though. Definitely. With that said, at least in Vel [2024], we can somewhat use their list of SQM-axis space classification of the modelling variations. Axis of course, then refers to a mathematical modelling living somewhere with respect to particular criteria for each direction on the scale in which each S, Q and M is classified of.

The S-Axis

The S axis contains the following model criteria:

Physical-conceptual Physical system are part of the real world, for example, a fish or a car. Conceptual systems are made up of thoughts and ideas, for example, a set of mathematical axioms.

Natural-technical Naturally, a natural system is a part of nature, such as a fish or a flower, which a technical system is a car, a machine, and so on.

Stochastic-deterministic Stochastic systems involve random effects, such as rolling dice, share prices, and so on. Deterministic systems involve no or very little random effects, for example, mechanical systems such as planetary system, a pendulum, and so on. A system is deterministic

if its evolution is specified discretely by a single path, and stochastic is when there exists more than one possible evolutionary state.

Continuous-discrete Continuous systems involve quantities that change continuously with time, such as sugar and ethanol concentrations in a wine fermentation. Discrete system, on the other hand, involve quantities that change at discrete time only. Note that for enough division of discrete time, sometimes continuous can be just as discrete as discrete is continuous. In really, continuity is often improbable of analysis, and almost all things that are continuous are actually microscopically discrete.

Dimension Depending on their spatial symmetries, physical systems can be described using 1, 2, or 3 space variables. The number of space variables used to describe a physical system is called its **dimension**, though this notion only applies to systems in which the shape and geometric features of the object is important.

Field of application We can distinguish between chemical systems, physical systems, biological systems, and so on. Systems from these and more fields of application will be considered, though not as much as we might hope.

The Q-Axis

On the Q-Axis, we have the following categories, which mostly stem from our above sections about definitions of model depends on the question asked.

Phenomenological-Mechanistic Again, see section 7.4 for details.

Stationary-unstationary see section 7.4 for details.

Lumped-distributed This one is pretty weird, though it connects to the spatial and space parameter usage. A mathematical model (S, Q, M) is called *distributed* if at least one of its system parameters or state variables depends on a space variable. By then, information is regarded spatially, that is, distributed over a physical space - for example, an imperfect spring of which the end tail and the beginning have different k coefficient. While a system is called *lumped* otherwise, that is, when everything is reduced to k, for example, by calculating the effective average spring constant $\bar{k} = k$ in such scenario.

Direct-inverse Consider an I-O system. If Q assumes given input and system parameters and ask for the output, the model solves a so-called *direct problem*. Most of the model below refer to the direct problems. If, on the other hand, Q asks for the input or parameters of S, the model solves a so-called *inverse problem*. If Q ask for input parameters, then it is a *control problem*, and for parameters of S, it is called the *parameter identification problem*.

Research-management Research models are used if Q aims at the understanding of S, management models, o the other hand, are used if the focus is on the solution of practical problems related to S. As pointed out, research models tend to be more complex and less manageable from a practical point of view. Depending on Q, the same mathematical equations can be a part of a research or a management model.

Scale Depending on Q, the model will describe the system on an appropriate scale. For example, depending on Q it can be appropriate to virtually follow a fluid particle on its way through complex channels, or just to compute the pressure based on certain parameters.

The M-Axis

For the M axis, containing mostly mathematical statements, we have the following:

Linear-nonlinear In linear model, the unknowns are combined using linear mathematical operations only, such as addition/subtraction or multiplication with parameters. Nonlinear models, on the other hand, may involve the multiplication of parameters, unknowns, the application of transcendental functions, and so on.

Analytical-numerical In analytic models, the system behaviours can be expressed in terms of mathematical formulas involving the system parameters. Based on these models, qualitative effects of parameters and the entire system behaviours can be studied theoretically, without using concrete values for the parameters. Numerical models, o the other hand, can be used to obtain the system behaviours for specific parameter values.

Autonomous-nonautonomous This is a mathematical classification of unstationary models. If an equation does not depend explicitly on time, it is called autonomous, otherwise nonautonomous.

Continuous-discrete Somehow, this one makes it to this place, again. In a continuous model, the independent variables may assume arbitrary values within some interval. For example, many of the ODE model uses time as the independent variables. In discrete models, on the other hand, the independent variables may assume some discrete values only. This is for example, the discrete event simulation technique, or the Nicholson-Bailey host-parasite interaction. Do note though, in reality, a good deal of continuity is actually just microscopic discreteness.

Mathematical statement types There are many types of mathematical statement one can use to describe, or restrict the system behaviours in. This includes from the *difference equation*, where quantity of interest is obtained as a sequence of discrete values, each term depends on the previous terms; *differential equation*, equations involving derivatives of an unknown function, and their *integral function* counterpart; or the normal *algebraic equations*, and more.

Again, note that we have a lot of overlaps between all three categories. And, one then also can notice that this on its own, is also a conceptual modelling to categorize models into classes and axis, which then, we should not take it as totally right. However, it is a perhaps particularly useful compass.

7.5 The don'ts of mathematical modelling When everything looks like nails

To some extent, mathematical modelling is superbly useful. It can help you to utilize the language of mathematics, the language of which quantisation is the basis, where perhaps explicitness is considered the most important, the language of the abstract space, and the language of nonphysical binding. Granted, one can argue that it still uses physical binding; just instead, it's a representation; however, that is perhaps beside the point. Nevertheless, as someone said in the past, "all models are false, but some are useful", just as we have to shift our model and find the correct hypothesis for the observable that we consider being, we cannot fully trust our model. Just as we cannot trust other people's words about the totality, our model might be false and might be only the reflection of a small portion of reality. That is why it is important to recall that the modelling and simulation scheme above is just an idealistic theory of how modelling should work so that it sticks with the real world, or any object of interest, that is.

This then recall me of the olde story of Pluto, or rather, his record of dialogues, called *Allegory of the cave*. In the following dialogue in Plato's famous book **Republic**, he discussed of

the notion of learning, and the perception of reality, and the blindness of personal perception – which is oddly similar to how we see (mathematical) model in describing the innate complex structure that we often find in practice:

Figure 7.5: Plato's allegory of the cave by Jan Saenredam, according to Cornelis van Haarlem, 1604, Albertina, Vienna

Socrates And now allow me to draw a comparison in order to understand the effect of learning (or the lack thereof) upon our nature. Imagine that there are people living in a cave deep underground. The cavern has a mouth that opens to the light above, and a passage exists from this all the way down to the people. They have lived here from infancy, with their legs and necks bound in chains. They cannot move. All they can do is stare directly forward, as the chains stop them from turning their heads around. Imagine that far above and behind them blazes a great fire. Between this fire and the captives, a low partition is erected along a path, something like puppy.

Glaukon I can picture it.

Socrates Look and you will also see other people carrying objects back and forth along the partition, things of every kind: images of people and animals, carved in stone and wood and other materials. Some of these other people speak, while others remain silent.

Glaukon A bizarre situation for some unusual captives.

Socrates So we are! Now, tell me if you suppose it's possible that these captives ever saw anything of themselves or one another, other than the shadows flitting across the cavern wall before them?

Glaukon Certainly not, for they are restrained, all their lives, with their heads facing forward only.

Socrates And that would be just as true for the objects moving to and fro behind them?

Glaukon Certainly.

Socrates Now, if they could speak, would you say that these captives would imagine that the names they gave to the things they were able to see applied to real things?

Glaukon It would have to be so.

Socrates And if a sound reverberated through their cavern from one of those others passing behind the partition, do you suppose that the captives would think anything but the passing shadow was what really made the sound?

Glaukon No, by Zeus.

Socrates Then, undoubtedly, such captives would consider the truth to be nothing but the shadows of the carved objects.

Glaukon Most certainly.

What is the moral of this example, of which dated thousands of years before this line is written? Surprisingly, in the not so much of endless, permanent story that is always relevant, Plato's cave outlines one of the fundamental rule in mathematical modelling: *don't believe the model is the reality*. Just as the man in the cave sees the shadows as his world, we are the modeller will eventually hit the point where what we need to understand, is further away from the shadow that is conceived as the present knowledge. Modelling offers, again, the simplified view on specific problems. This is perhaps one of the more important point that we get from working with modelling, in which we are always "chained" in some way or another, as long as we think about reality in such language, for it being arbitrary is not reality, and henceforth. With this, also comes with the assumptions we make on the modelling itself.

There are a lot of lessons to learn and to beware of when working with modelling, coming off as wisdom of the ancient times, of which the olds have left us. Aside from refuse to constrain yourself to the warped reality you constructed, we have a few more 'don'ts' that is worth considering. They can be reinterpreted in the following way:

Axiom 7.5.1 (The don'ts of mathematical modelling).

1. *Don't believe that the model is the reality.*
2. *Don't extrapolate beyond the region of fit.*
3. *Don't distort reality to fit the model.*
4. *Don't retain a discredited model.*
5. *Don't fall in love with your model.*

Hopefully, this is enough to shy people away from going down the wrong path. That is, if they realize it soon enough to back their track. One of the major mishap that can happen often time is indeed when the model somehow replaces what can be realized of the object itself, rather than the model of the object (let's just say, that is string theory in a nutshell).

7.6 Conclusion

Overall, this particular chapter introduced the concept and notion of classical model, their formulation, components, how to construct such, and roles of individual component that makes up the entire model. Obtaining knowledge and hypothesis forming is also discussed, however in a very short manner as to leave it in later chapters separately (for example, dynamic hypothesis formation is nowadays called *machine learning*, which is rigorous enough to require an entire part on itself rather than a single section). We have also discussed the do and the don't of using mathematical modelling, which is a kind of modelling that uses the language of mathematics to describes the system and the environment. With such, at least a very comprehensive view has been achieved (hopefully) of the classical theory of mathematical modelling.

There is, though, a reason why this chapter is called 'classical' rather than simply mathematical modelling theory. Most of what we have been discussed stopped to around the 1990s in their contents. Hence, newer development, mistake corrections, formulation of different kind of modelling and their organization was added in such time and is not presented of the classical theoretical categorization, and choices. Nevertheless, it is important to note that classical theory by such stance, is more stable, more well-defined and complete, and is thus very *static*, which is something desperately needed in a field or topic that is moving constantly for the lack of comprehension plus well-grounded standard. Thereby, we will encapsulate the classical theory in one, to allow for more advanced discussion further onward.

7.7 Appendix

7.7.1 Linear programming

All mathematical models considered so far were formulated in terms of equations only. Remember that according to our definition of a mathematical model, a mathematical model may involve any kind of mathematical statement. For example, it may involve inequalities. One of the simplest classes of problems involving inequalities is linear programming problems that are frequently used e.g. in operations research.

Because the subject is far more too complex for us to cover here (it literally guarantees me to read two entire whole books to cover the minimum), we will roughly give you the general definition, and then its perhaps miraculous performance and mechanism.

Definition 7.7.1 (Linear programming). *The standard form of the linear programming problem is to determine a solution of a set of equations:*

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \vdots \qquad\qquad & \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \end{aligned} \tag{7.3}$$

with $x_j \leq 0$, for $j = 1, \ldots, n$ that minimizes the function

$$z = c_1x_1 + c_2x_2 + \cdots + c_nx_n - z_0 \tag{7.4}$$

It is this standard form of the linear programming problem, a minimization problem involving only equalities, that we will solve in practice. The first task of the people working with linear programming will then be to show that any linear programming problem can be formulated as a problem in the standard form, where the number of equalities, m, and the number of

variables n are determined by the problem. This is where you would see the formulation and writing of linear programming problem as the following:

> Consider the problem of diet. Suppose a particular diet problem reduces to the mathematical problem of minimizing $3x_1+2x_2+4x_3$ by 3 equations for (x_1, x_2, x_3). The problem setting can be said as:
>
> **Minimize** $3x_1 + 2x_2 + 4x_3$ subjected to:

$$\begin{array}{rcl} 30x_1 + 100x_2 + 85x_3 + x_4 & = & 2500 \\ 6x_1 + 2x_2 + 3x_3 - x_5 & = & 90 \\ x_1, x_2, x_3, x_4, x_5 & \geq & 0 \end{array}$$

There are also the problem or approach of **nonlinear programming**, although for the time currently, we would not burden ourselves of such task.

Chapter 8. Classical learning theory

What we have done, what we have been investigated, including mathematical modelling, can be said to be the *static construction*. What this means is that, albeit intrinsic of collecting and transforming resources, information, or any given kind of input - given the black box, in-out system treatment. This comes as a cost - the model requires human touches to even conform to certain structures, knowledge, tasks, and utilities. Mitigate this issue requires we to come up with a notion that automates the acquisition and relative understanding (butchering the word understanding, since our model currently does not understand *anything*, strictly speaking), and it is called (and found) as the *learning theory* of machines.

8.1 Before the learning theory

Before the introduction of learning theory as a practical and theoretical process, functional models, usually mathematical models, mostly relies on acute, specific solution, correct formulation of given form, or rather being done solely in a static sense. What this means must be followed through by historic development to understand why it is called static, and subsequently what is then considered dynamic.

Mainly mathematics (but not restricted to such, sometimes usual construction is alright) has been used to formalize, simplify, express, and encapsulate the fraction of reality through the language of quantification, to make **models** that befit the knowledge and the respective fraction observed. Simply said, considering mathematics as the foremost language of abstraction (removing physical realization), quantification (give us the notion of quantity), and the build up of structures within said language, it is fundamentally useful in decoding and expressing reality in the way that makes it understandable, interpretable, logical (the word logical brings around the roundabout), easy to quantify components, actions, factors, objects, and perhaps many further benefits of using such language. Thus, one of the most fundamental example of using the language of mathematics to formalize and quantize reality to be a functional construct, is physics. And its development throughout history also says plenty bit on what is static.

A theory is often considered a static construct, by virtue of how it is formed. It begins by setting up axioms, assumptions, simplification, the target of the theory, and development on which it is built on upon. Further development will follow through the main foundation upward, without many changes to the theory. If one is to modify the foundation, it would mean usually to create new theory on said basis. This process of transition is typically thought in philosophy as the *negation of the negation*, or in propositional logic, double negation. Fortunately, the transition is perhaps, if not totally very natural of natural science and overall interpretation of the world that we often use. We can attribute this to the fact that we do not understand the world in its entirety, Furthermore, mathematical modelling on itself is not perfect, and cannot capture intrinsically every given thing available, for specific methods - basically, our hands are tied, and is not perfect either way. That is why there must be, and would be this type of changes throughout the theory itself.

Building a structure is the same, and so much for a system.

Most of the general model construction and the way that artificial intelligence was created (under the umbrella term of automated thinking machine, artificial is one, and machine learner later on was subjected to be the equivalent, more *advanced* development) belongs to either the specific type of inference parameterized model, or symbolic model (AI) - which utilizes the propositional logic ϕ of the truth functionality $\mathcal{V}^n \to \mathcal{V}$. Under said notion, precursor AI, for even specialized task, was conducted in the way that it is designed by experts, for particular problem of knowledge implementation, and then use logic to deduce from said logic pool reasonable course of action, or rationalization. This is partially because of the computational limit of the time - at the time of 1950s to 1970s, there is no Moore's law that gives you 50,000,000,000 transistors counts, like if that is ever possible at that time.

One major theme of the disadvantage of the phenomenological approach as we have been investigating in the classical mathematical modelling chapter, is the fact that it is sensitive to changes. That is, given the holding parameters k for the Boole's coefficient of a spring under stress, if you somehow increase it, by for example, $2k$, then the entire phenomenological model is rendered useless. Of course, one can then simply expand the system to accommodate the various setting of k, but that seems pretty troublesome at the time and of the setting, in comparison to the mechanistic modelling approach. Without a somewhat dynamic process, and a fairly primitive notion of learning, a phenomenological sometimes might be astronomically difficult to handle, based on the immense dimensionality requirement of the parameters used in relation to the system.

Afterward, it is perceived that expert inputs and traditional knowledge acquisition and their form as logical proposition is not adequate - in fact, it is unable to be scaled up. Yes, it is good - even by today's standard - but given a larger, more comprehensive and more complex problem, the system break down to be unsustainable and unattainable. Thereby, we need something else. Something of the quality that makes the model *learn for itself*, rather than waiting for developer's implementation and expert input. That comes of as, eventually not so surprising, an ability that human possesses: *learning*.

In another sense, learning can also be conjured to be the action of *computational reasoning with uncertainty*, as opposed to the general notion of artificial agent, a type of model which makes use of, as previously mentioned, logics and relational discrete spaces. We would look into this aspect later, however, it is also suggested to note that, learning can also still happen, within the logical region of a model. By that, we think of the internal space and external space - the classical definition for learning theory and its distinction in such case is typically the classification between *external-dependent reasoning* and *internal-dependent reasoning* (which depends only on the propositional space).

8.2 Classical theory

The learning theory started from the early 1970s. By the sense of classical, we do not mean by, as currently called, of non-deep learning theory, but rather the old treatment of theoretical boundaries and rigorous formalization of the concept itself. We will begin by introducing the formal setup of learning, and then, go about developing it until we can even reach the *Probably Approximately Correct* learning (PAC) and Vapnik-Chervonenkis theory of maximal dimension.

The nature of the learning theory can also be brought back from certain proposition or argument, most probably, the fact that it is mostly from the black-box interpretation - hence making almost all model being phenomenological by default. An interesting aspect of such, though, is the learning process itself, or we call it so.

8.2.1 Principles

Classical learning theory studies the process of learning. What we are studying the learning action for? Before we even consider the learning action of artificial intelligence, we need to have a simplified idea about the problem setting. Generally speaking, from the chapter of mathematical modelling, there exists the description of a scientist changing his hypothesis to match the perhaps underlying system which exhibits certain kind of behaviour. This process is conducted by first laying out possible hypothesis, then testing it, then going back to change the hypothesis as according to experimental verification. This proceeds until the hypothesis's structure reaches its limit, and either reaches the target goal, or not, hence guaranteeing a change in structural hypothesis set. Our learning model is tasked to *automatically resolve the hypothesize dynamics*, by itself, up to certain restriction. To do this, most of the definition will be of concern of the black-box, mathematical model of one-directional, functional relationship in the observation set (dataset in which the underlying target lies).

It is notable to remind that *learning* is perhaps a quantitative action that can be added to the structure of the artificial construct (which have been previously defined in previous chapters), hence additional structure embedded on it. Thereby, our theory if a simplification of such concept, to the most foundational interpretation of the process of learning, even though at this point the notion of learning through this theory might still not be able to be considered as learning. However, we will justify it as the process of which enables a specific **action** to be conducted, based on the observations or samples possible. In such sense, learning is adapting the internal structure to target structures outside its own.

8.2.2 Concept and hypothesis

In simplicity, within the principle of working, our model is called the **hypothesis** $h \in \mathcal{H}$ of certain **hypothesis class** $\mathcal{H}$. The other ingredient of the learning theory includes:

The concept class versus function class can be fully incorporated into one as $\mathcal{C}$, if one is to assume the form of any given concept as $c : X \to Y$ for another set Y which can lies in the sigma-algebra $\mathcal{S}$, or the measurable mapping as f.

The task for the hypothesis, and the designer (us) is to figure out how to best approximate c, assuming arbitrary notion of knowledge, prior information, such that h correctly *mimic* the behaviour and phenomenon observed by c. We call this loosely as the **learning problem**. Naturally, learning problem is a statistical problem, since their setting is fully observational.

We are given our information of the system by the form of a dataset $\mathcal{S}$, called *observation*. We first define the **observations**. The set of observation $\mathcal{S}$ is presentable in the space of 2-tuple $(\mathcal{X}, \mathcal{Y}) \subset (\mathbb{R}^d, \mathbb{R}^p)$. Usually, hence, our data will be realized in the Borel σ-algebra $\mathcal{B}$. The full dataset is *discrete*, contain of n instances of observations, and is denoted by

$$\mathcal{S} = \{(x_1, y_1), \dots (x_n, y_n)\} \subset (\mathcal{X} \times \mathcal{Y})^n \tag{8.1}$$

Here, $\mathcal{X}$ is called the *ground* or input space, and $\mathcal{Y}$ is called the *target* or output space. By default, in general, the input space $\mathcal{X}$ is supposed to follow some special distribution $\mathcal{D}$. This treatment branches from the ambiguity of the origin of $x \in \mathcal{X}$ for elements of the input space, that is, $x \sim \mathcal{D}$ by a random sampling process. This assumption also ensures relative population representability. By this, we mean that the observation captured fully describe what was happening with the entire concept - even including the pathological and outlier. We also assume that this sample of observation is **identically and independently sampled** of a given sampling process.

The **learning problem** divides the act, with respect to configuration the chain of component into two parts. The first one is for any measurable space of the model, after defining the system, the dependency, and the question, we have to gain the *accuracy measure*, or *phenomenological*

aberration of a given model, and for the respective problem. Depends on the specific requirement of the problem, the measure can change, but, they have the same landscape. The second one requires the development of a *scheme* to dynamically move the given hypothesis, under fixed description, into a supposed optimal point in which under specific threshold, our model is 'perfect' for the concept. That is, *mimic c by h in a perfect way*. It is perceived, though, that the objective of learning the concept perfectly is impossible. Hence, we will often opt for the latter half of approximating it to a given threshold θ. we will eventually prove this later on (Seriously).

Setting in the second part, most of our problem is concerned with the issue of predicting the unseen states of being, or rather, the *generalization problem*. This then perhaps, classify the problem of statistical learning theory to *mostly* predictive, or rather, we are concerned of prediction model. This is subsidized by the inner *estimation problem* for an estimation model. What is the difference between the two? First, the distinction comes from the trivial on-sight fact that estimation problems and their models, comes of as a closed-form working space. That is, they work on a closed space of what is observed, and extract properties, determine statistically the behaviours, characteristic of that system alone and only. In other word, an estimator *uses data* to guess and learn about the true state, and properties of the setting (usually also said to be the *true state of nature*). However, we notice a discrepancy - not always will we get the entirety of the problem that we are interested at. For some reason, for example, our data is only quite a portion, lacking a bit valley, a bit of patterns here and there by the blank spaces where observations left behind. Then, what if we can generalize pass that? What if we also want to somewhat estimate correctly for unseen portion of our concept spaces? This is called the predicting problem we have said earlier, in which we use the estimation already established, and add some flavours into what can possibly constitute a larger view - a more entire view of the learning concept. From a statistical perspective, the model itself in statistical learning theory, would have to both estimate the intrinsic empirical distribution and concept $(c_S, \mathcal{D}_S)$, and then somehow, leaves room for generalization to the actual distribution and concept $(c, \mathcal{D})$, which is particularly of more interest than not. Secondly,... well I don't know (really...). Guess I have to wait until I gather more stuff, I guess.

If we recall the previous discussion on the informal section on machine learning, you will see that this is typically what's called the **supervised learning setting**. This is perhaps true for the classical learning theory, and the establishment of the theory as a whole. Consider unsupervised learning. Under the scope of unsupervised learning, the model essentially goes into a free-mode, where there exists no metric that is non-context enough for evaluating the theoretical form of the model. For a given density estimation model ψ, there can be many ways to 'think of evaluating it' on how well it groups objects together. This means there are not so much meaningful information can be extracted from examining the behaviour of the model, including its performance. Furthermore, since it is random as much as a random process, no conclusive bounds or theoretical theorem can be made on the maximal-minimal problem of a given model. Hence, most of the learning theory is concerned of supervised learning setting, because it is a controlled, closed, well-defined system of interest.

8.2.3 Phenomenological aberration

The learning problem - which is most importantly described by the phase of *phenomenological aberration* (or data-based replication) - is then formulated as followed. Given the machine learning model expressed a hypothesis h of the hypothesis class $\mathcal{H}$, the learning theory aims for creating a procedure to learn either elements of the concept class $\mathcal{C}$ of all concepts $c : \mathcal{X} \to \mathcal{X}$, or the function class $\mathcal{F}$ of all functions $f : \mathcal{X} \to \mathcal{Y}$, which is usually set to $\{0, 1\}$ or $[0, 1]$. This distinction is trivial, hence, if it is clear, we will talk about the concept class $\mathcal{C}$ only as repre-

sentative form. The learner $\mathcal{L}(h)$ consider the set of possible hypothesis $\mathcal{H}$, in which might not coincide with $\mathcal{C}$. It receives a partial image of sample $S = (x_1, \dots, x_2, \dots, x_n)$ drawn i.i.d. according to $\mathcal{D}$ as well as the label $(c(x_1), \dots, c(x_n))$. This constitutes the dataset $\mathcal{S}$, which are based on specific concept $c \in \mathcal{C}$ for the hypothesis to learn. The task is then to use (or *extract*) meaningful information to select a hypothesis $h_S \in \mathcal{H}$ that accurately mimic c, with marginal error Θ. The notation h_S stands for all hypothesis that can be inferred from the range of the dataset (the first argument, $\mathcal{X}$).

The marginal error Θ is considered of two parameters, the empirical error $\hat{R}(h)$ and the generalization error $R(h)$. This can be justified as following.

To compare between h and c, we use the metric provided by $\mathcal{Y}$. Then, define certain type of function called *loss function*, also called objective, error function, denoted by $\ell\{h, c\}$ such that it measures the difference between h and c. This measure is arbitrary, and we assume no form of it, aside from the fact that it also acts and return values on certain real subset of values. The following definitions provide us with the notion of *empirical error* and *generalization error*. Notice that one is empirical - indeed, since we only can work on certain fixture of c, i.e. the dataset provided, and the other one is generalized - meaning the true error toward the actual target concept $c \in \mathcal{C}$.

empirical risk

Definition 8.2.1 (Empirical risk). *Given a hypothesis $h \in \mathcal{H}$, a target concept $c \in \mathcal{C}$, and a sample $S = (x_1, \dots, x_m)$. For some particular $\epsilon > 0$, the* **empirical error** *or empirical risk of h is defined by*

$$\hat{R}_S(h) = \mathop{\mathbb{P}}_{x \in S \sim \mathcal{D}}[\ell\{h(x), c(x)\} \leq \epsilon] = \frac{1}{m} \sum_{i=1}^{m} \ell\{h(x_i), c(x_i)\} \tag{8.2}$$

generalization risk

Similarly, the generalization risk defined in the same fashion, but extending beyond the dataset $\mathcal{S}$.

Definition 8.2.2 (Generalization risk). *Given a hypothesis $h \in \mathcal{H}$, a target concept $c \in \mathcal{C}$, and an underlying distribution $\mathcal{D}$ on $\mathcal{X}$. For some particular $\epsilon > 0$, the generalization error or risk of h is defined by*

$$R(h) = \mathop{\mathbb{P}}_{x \sim \mathcal{D}}[\ell\{h(x), c(x)\} \leq \epsilon] = \mathop{\mathbb{E}}_{x \sim \mathcal{D}}[\ell\{h(x), c(x)\}] = \int_{x \in \mathcal{D}} \ell\{h(x), c(x)\} \, dP(x) \tag{8.3}$$

The notion of empirical and generalization error clearly indicates the issue of two concepts - one observable 'component' concept c' presented by the dataset, and the actual concept c itself. Hence, we assume, of the universal $\mathcal{X}$ set belongs to c, there exists $\mathcal{X}_c$ and $\mathcal{X}_{c'}$, hence we have two distributions, for example, denoted by $\mathcal{D}_c$ and $\mathcal{D}_{c'}$. We have the following theorem:

Theorem 8.2.1. *For fixed $\mathcal{H}$, for fixed and sufficiently large $\mathcal{S}$, and no observation errors, the empirical risk is the generalization risk:*

$$\mathop{\mathbb{E}}_{\mathcal{S} \sim \mathcal{D}^m}[\hat{R}_{\mathcal{S}}(h)] = R(h) \tag{8.4}$$

Proof. By linearity of the expectation, and the fact that we assume the sample is given i.i.d., we can write:

$$\mathop{\mathbb{E}}_{S \sim \mathcal{D}^m}[\hat{R}_S(h)] = \frac{1}{m} \sum_{i=1}^{m} \mathop{\mathbb{E}}_{S \sim \mathcal{D}^m}\left[\mathbb{1}_{h(x_i) \neq c(x_i)}\right] = \frac{1}{m} \sum_{i=1}^{m} \mathop{\mathbb{E}}_{S \sim \mathcal{D}^m}\left[\mathbb{1}_{h(x) \neq c(x)}\right],$$

for any x in sample S. Thus,

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[\hat{R}_S(h)] = \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[\mathbb{1}_{h(x)\neq c(x)}\right] = \mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[\mathbb{1}_{h(x)\neq c(x)}\right] = R(h).$$

which yields the desired result. □

This theorem specifies a problem in the underlying theory of data-based statistical learning. The question is as simple. In the statistical learning setting, one gain the dataset $(\mathcal{X}, \mathcal{Y})$. Because we are using the *black-box interpretation*, the concept that is supposed to be in $c : \mathcal{X} \to \mathcal{Y}$ is said to be governed by a probability distribution $\mathcal{P}$ on said data. Now, depends on the type that you want, it can either be **dense**, that is, there are infinitely many $x \in \mathcal{X}$ as the phase space, and the probability is on y, that is, $\mathcal{P}(Y)$. Or, you can remove the strictly dense assumption and get the probability $P(\mathcal{X}, \mathcal{Y})$ of joint probability between the two instead. We know from the start that the dataset is not always representative, neither captures the entirety of the concept c. Hence, there must exist a particular $\mathcal{D}$ of true probability distribution that describes c. In a perfect world, we would not need to see c as probability distribution - since in said perfect world, everything might as well be deterministic, but the best one can do is to be able to get close to the presumptions probability distribution of the event. Then, the question is, what is the *inherent correlation* between $\mathcal{P}$ and $\mathcal{D}$? To answer this question, we might as well need some divergence metric to see their diverging property. This will eventually be the KL-divergence measure, which we will discuss later. A disadvantage of KL-divergence, however, is its inability to take into account the structural problem of the system, as well as its model. We might as well address this in our later section.

Using the above notion of generalization we can separate the learning problem into two goals - one to minimize $\hat{R}(h)$, and one to minimize $R(h)$. As we have been saying, and for Theorem 8.2.1, one of our main assumption, as well as a guarantee under uniform convergence, is the fact that we must have $\hat{R}(h) \approx R(h)$ for larger and larger dataset size, or the empirical space. Generally, this branches off to be two problems.

Definition 8.2.3 (Empirical learning problem). *We present the formal form of the empirical learning. Suppose we have a target, $c \in \mathcal{C}$, where $\mathcal{C}$ is an arbitrary concept class that captures targets of the same type. Suppose we are provided a set of observations $\mathcal{S}$. The problem is to use certain algorithm $\mathcal{A}$ using $\mathcal{S}$, to obtain a hypothesis h^* for a fixed $\mathcal{H}$ such that:*

$$R(h^*) = \min_{h\in\mathcal{H}} \hat{R}(h) = \min_{h\in\mathcal{H}} \mathop{\mathbb{E}}_{x\sim\mathcal{D}, x\in\mathcal{S}} \ell\{h(x), c(x)\} \tag{8.5}$$

The hypothesis h^* is often called the *empirical best*, for it being the minimal, finite hypothesis of the lowest loss evaluation on the entire observation space $\mathcal{S}$. There exists no certified assumption regarding whether h^* aligns with the minimal generalization error.

Definition 8.2.4 (Generalization learning problem). *We present the formal form of the generalization learning problem. Suppose we have a target, $c \in \mathcal{C}$, where $\mathcal{C}$ is an arbitrary concept class that captures targets of the same type. Suppose we are provided a set of observations $\mathcal{S}$. Supposed we have an algorithm $\mathcal{A}$ that for fixed hypothesis space $\mathcal{H}$, 8.5 holds true. The problem is to use certain algorithm $\mathcal{A}'$ such that, under limited availability, to obtain $\boldsymbol{h}$, satisfies:*

$$R(\boldsymbol{h}) = \min_{h\in\mathcal{H}} R(\boldsymbol{h}) \leq \{\epsilon\}, \quad \epsilon > 0 \tag{8.6}$$

For a set of risk bounds ϵ. If the setting is deterministic, then there exists $\epsilon = 0$.

We can show that normally, the problem of empirical learning and generalization learning is not the same. That is, if one tries to solve the empirical learning problem to certain measure, then they will fail the generalization problem. This can be illustrated by figure 8.1. This problem between the disparity of the empirical and generalization target is what we often called as the *Allegory of the cave*, as seen in previous chapter. Here, we risk being dependent ourselves on false hypothesis obtained by empirical data acquisition, but also have to aim for the general form of the target by itself, which is often assumed unobtainable or impossible to know of prior belief. Hence, either that we take a skewed representation of reality through empirical data, or we diverge from such in a manner that gets closer to the ground truth's concept. At least in this setting, it is then observed as above that empirical learning problem is a subproblem toward generalization problem - one have to solve the empirical learning problem before attempting the generalization one. We also want to introduce the term **data shape** and **concept shape** for data $\mathcal{S} \in c$ and the shape of c by itself, to reflect this difference.

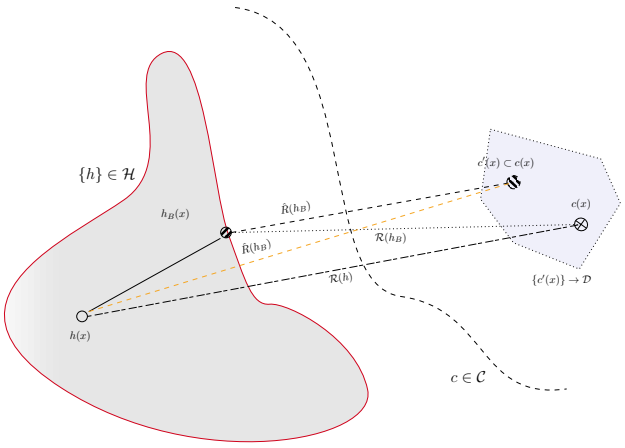


Figure 8.1: An illustration of statistical learning theory on the evaluation of the risks and errors, during learning process. c' is presented in the 'orbital' vicinity around c, with its distance of certain metric define how 'accurate' the reconstruction from distribution can be. Of the hypothesis set $\mathcal{H}$, there exists the Bayes hypothesis h_B and an arbitrary 'random' hypothesis h, and their respective measure.

For the generalization problem's best model in a fixed $\mathcal{H}$, we have the notion of a **Bayes model**.

Definition 8.2.5 (Bayes risk). *Given a distribution $\mathcal{D}$, the Bayes error R^* is defined as the infimum of the errors achieved by measurable functions $h : \mathcal{X} \to \mathcal{Y}$:*

$$R^* = \inf_{h \in \mathcal{H}} R(h) \tag{8.7}$$

Definition 8.2.6 (Bayes model). *A hypothesis h with $R(h) = R^*$ is called a Bayes hypothesis or Bayes classifier, and is taken by*

$$\forall x \in \mathcal{X}, \quad h_{\mathrm{B}} = \operatorname*{argmax}_{y \in \mathcal{Y}} \mathbb{P}[y \mid x] \tag{8.8}$$

The Bayes model is understood to be the minimal of $R(h)$. And, by our observations (which will be proved later on), $\hat{R}(h) \neq R(h)$ in circumstances. We will later show a more generalized version of 8.2.1.

Overall, in the classical learning setting, we can dissect the problem into what constitute the following cleaner version of the setting, includes both the generalization learning and empirical

learning problem. Again, under ideal condition, and ideal setup, the learning setting will have empirical learning and generalization learning to be on the same track. Later on, we would formalise what exactly on the same track means.

Setting 8.2.1. *Given an op-space $\mathcal{X} \times \mathcal{Y}$, define the concept class $\mathcal{C}$ of all $c : \mathcal{P}(\mathcal{X}) \to \mathcal{P}(\mathcal{Y})$ of all power set for both, where $\mathcal{X} \sim \mathcal{D}$ for some distribution $\mathcal{D}$ of the probabilistic assumption. The learning problem consists of defining a hypothesis class $\mathcal{H}$, for h being an arbitrary hypothesis, either with specific architecture or not, and a learning algorithm $\mathcal{A}$, of a* **learner** *such that the following is acquired:*

- *Set $\epsilon, \delta > 0$ be the error bound and the confidence measure (that is - the probability of failure) of h, for ℓ the evaluative measure of $\{h(x), c(x)\}$, define $\hat{R}(h)$ being the empirical risk of h on the observation set $\mathcal{S}$ available. Then, the empirical learning problem is tasked to find the hypothesis such that $\hat{R}(h) = \min_{h_0 \in \mathcal{H}} \hat{R}(h_0) \leq \epsilon$ for some arbitrary h_0, with confidence $1 - \delta$. This h is called the empirical best.*
- *Similarly, define $R(h)$ the generalization error, assumed outside $\mathcal{S}$. Then, the generalization learning problem is tasked to find the hypothesis such that $\hat{R}(h) = \min_{h_0 \in \mathcal{H}} R(h_0) \leq \epsilon$ for some arbitrary h_0, with confidence $1 - \delta$. This h is called the Bayes model.*
- *The generalization setting requires choosing the optimal point for both h^* of the empirical best, and h_B for Bayes hypothesis.*

To choose and to compare between the empirical best, and the Bayes hypothesis (generalization best), we need certain notions to compare the two, algorithm to choose and evaluate, and metric to determine how comparison should be performed, with both measurable. At best, we need to find the correlation between the subconcept and the actual target concept itself.

If we choose to interpret the learning model as it is, for a *statistical interpretation*, then it is fair to say that what we are trying to do, is to guess the correlation between the distribution $\mathcal{D}$ of the dataset, and the distribution $\mathcal{P}$ of the entire concept, outside the dataset itself. By default, we can reasonably have that $\mathcal{D} \subset \mathcal{P}$. But, even for that assumption, there is still the question about the stretch and range of $\mathcal{D}$. As we have said, we do not know if they are the same, hence the representative feature we have on both space can be uncertain. So traditionally, we typically assume, as similar to theorem 8.2.1, that they are indeed, of the same kind. But, let's say, can we measure the difference between the distribution, given the probability interpretation?

Question 8.2.1. *Under the probabilistic interpretation, what can be said about the correlation $\mathcal{D} \propto \mathcal{P}$, by a specific feature λ?*

If we can answer this question, it will be particularly helpful to see where it will lead us to, and what can be said more of the empirical and generalization. Coincidentally, in modern and practical practice, the dilemma between empirical and generalization is featured, or presented, in a much smaller context, where the dataset itself is expressed into the *training partition* and *testing partition*, or even more partition to facilitate the uncertainty or non-observable portion of the concept itself.

Although we are not considering the learning process itself, the next section will set up some classical theoretical boundary on which model applied on statistical learning theory might use.

About $\mathcal{X}$ and $\mathcal{Y}$

While $\mathcal{X}$ is particularly "free", it is imperative to be mentioned that $\mathcal{Y}$ is important in deriving preliminary results on theoretical bounds and theorems. For example, for most cases

mentioned below, $f : \mathcal{X} \to \{0, 1\}$ is often the optimal choice for setting as to prove theorems and results, which is binary classification. Overall, the assumption of the structure of $c(x)$ dictates how the application and derivation of theorems, as well as learning settings are.

It is also noticed that particularly, the dataset and it embedding space (or expressive space - how the concept is expressed using our data or understanding) also affects what kind of loss function will be more effective than not. Then, it is almost dependent on the expression of the metric space applied for each loss function.

8.2.4 Estimation and approximation error

Aside from the empirical error and generalization error, there are two more particular error measure that target the specifically crafted question. How should the hypothesis set $\mathcal{H}$ be chosen so that the learning algorithm can efficiently work? This is known as the *model selection problem*. By designing $\mathcal{H}$, we can particularly guide the model to a given course and path, such that the learning model can learn the problem setting, at least up to the empirical best and closer to the Bayes model. A rich or complex hypothesis set then, could contain the ideal Bayes classifier. On the other hand, it is perceived that learning with such complex family becomes a very difficult task. How difficult this is, and how to contain the Bayes classifier (or at least inching to it), is ultimately, and generally perceived to be subjected to a trade-off, and can be analysed in terms of *estimation error* and *approximation errors*.

Let $\mathcal{H}$ be a family of functions mapping $\mathcal{X} \to \{1, -1\}$. This is the particular case of **binary classification**, in which can be straightforwardly extended to different tasks and loss functions. The *excess error* of a hypothesis $h \in \mathcal{H}$, is the difference between its error $R(h)$ and the Bayes error R^*. This can be decomposed to be the following:

$$R(h) - R^* = \Big(R(h) - \inf_{h \in \mathcal{H}} R(h)\Big) + \Big(\inf_{h \in \mathcal{H}} R(h) - R^*\Big) \tag{8.9}$$

The first bracket contains the **estimation error**, and the second bracket contains what is called the **approximation error**. The estimation error depends on the hypothesis h selected. It measures the error of h with respect to the infimum of the error achieved by hypotheses in $\mathcal{H}$, or that of the best-in-class hypothesis h^* when that infimum is reached. The approximation error measures how well the Bayes error can be approximated using $\mathcal{H}$. It is a property of the hypothesis set $\mathcal{H}$, a measure of its richness.

Model selection consists of choosing $\mathcal{H}$ with a favourable trade-off between the approximation and the estimation error. However, in the most general case, this will be done, but not in practice, as it requires the underlying distribution $\mathcal{D}$ to be known to determine $R*$, which is not possible. In contrast, the estimation error can be bounded, or can be analysed, using particular metric and analysis.

Also, contrary to what is believed, the approximation and estimation error is still lacking - it can be thought as a particular *proxy* to the entirety problem of model selection, a typically very high generalization of the view, with too many factors influencing it. We would be keen on to find better alternatives, even though, in some sense, a pseudo-generalized setting might be a good choice already exists.

8.3 Structural example of learning setting

While the descriptions of the learning theory setting remains general and well-defined, its operations and specific state-wise understanding is implicit, that is, there exists the very severe distinction between the actual conceptual modelling, and its theoretical interpretation. We will illustrate the practical, well-known system analysis typically accompanied what has been

discussed.

The learning dynamics can be then divided into three **phases** - each focus on one particular aspect of the learning problem and its interpretation, and the moving parts. This is illustrated by figure 8.2.

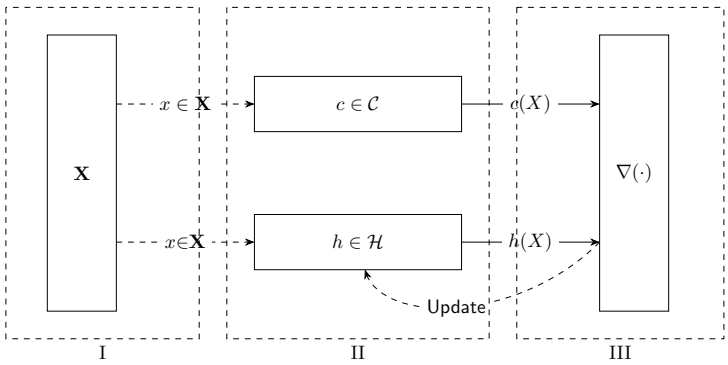


Figure 8.2: An illustration of the (supervised) statistical process. Phase III contains two parts: First is the evaluation $\nabla(h, c)$ according to the data $\mathcal{D}$, and second is the Update process to re-align c to the actual target.

8.3.1 First section

Begin with the construction and initialization of the feature space $\mathcal{X}^{\infty}$. By the assumption of a probabilistic process, $\mathcal{X}_m$ or simply $\mathcal{X}$ representing the dataset is assumed to be sampled, or appeared from the distribution $\mathcal{D}(p, \cdot)$ for p an arbitrary probabilistic system.

The *ground space*, or in literature generally called **feature space**, is created. This results in the first component of the tuple D, being $\mathcal{X} \subseteq \mathbb{R}^{m \times k}$, where $|\mathcal{X}| = m$, but $size(x) = k$, for $x \in \mathcal{X}$. We assume there exists the random variable x of a given distribution $\mathcal{D}$, such that the set $\mathcal{X} \sim \mathcal{D}$ of all observation is given by an underlying distribution. A well-known assumption in this case is that $x \in \mathcal{X}$ has no dependent components. That is, for example, there does not exist any function $\psi_x : \{x_i\} \to \{x_j\}$, for $i \neq j$. (Thought, this only helps in Bayesian priori analysis). Furthermore, by specifying that $\mathcal{X}$ belongs to a specific distribution, we argue of the assumption that $x \in \mathcal{X}$ is *independently and identically distributed* (i.i.d.), that is, for a given random sampling interpretation $S \subset \mathcal{X} \sim \mathcal{D}$, all data is sampled with the events space governed by the distribution $\mathcal{D}$ for every instance, and the existence of x_i to x_j for $i \neq j$ is none.

The **role** of the distribution configuration is important. If $\mathcal{D}$ is uniform, that is, $D \sim \mathcal{U}(a, b)$, then we can disregard the aspect of $\mathcal{X}$ with random variables. For $[a, b]$ to be $(-\infty, +\infty)$, the problem changed to a *regression problem*. If $\mathcal{D}$ is some other distribution function, then the problem of *representative data* and *population size* becomes apparent, which requires statistical analysis to be taken into consideration.

8.3.2 Second section

For the constructed feature space $\mathcal{X}$, let $EX(c, \mathcal{D})$ be a procedure (or *orcale*) acting on $\mathcal{C}$ and $\mathcal{X}$ to output $\langle x, c(x) \rangle$. The hypothesis then contains a similar procedure $EX_{\mathcal{H}}(h, \mathcal{D})$, such that to output $\langle x, h(x) \rangle$. We call this the *inference phase*.

The feature space $\mathcal{X}$ is provided to h. We assume c is processed of the same process, resulted in $\mathcal{Y}_c$. Then, $h(\mathcal{X})$ outputs the set of hypothesis' target $\mathcal{Y}_h$. Thereby, we are approximating the

process c itself. We can approach this in two main ways[1]: either *deterministic* or *probabilistic*[2]. We define such as followed.

Definition 8.3.1 (Deterministic - *discriminative modelling*). *A* ***deterministic model*** h_d *assumes its internal space as followed. For any* $h \in \mathcal{H}_d$ *of deterministic model,* h *is characterized by the mapping* $h_d : \mathcal{X} \to \mathcal{Y}_h$*, such that* $h \in C^n$ *of* n*-differential space.*

Similarly, if the interpretation of the model is *probabilistic*, we have the following definition of probabilistic-based model.

Definition 8.3.2 (Probabilistic - *generative modelling*). *A* ***probabilistic model*** h_p *assumes its internal space as followed. For any* $h \in \mathcal{H}_p$ *of all probabilistic hypothesis,* h *is characterized by the probability distribution* $\Gamma(p_X(X))$*, where* Γ *is the discrete decision process outside the probabilistic interpretation.*

The discrete decision process Γ is very simple to argue. For example, given a Naive Bayes model with $p(C_k \mid D[i])$ of the dataset, the probability distribution is regarded as the argument

$$Y^* = \arg\max_{c_k \in C} P_\theta(c_k \mid x) = \arg\max_{c_k \in C} P_\theta(x \mid c_k) \cdot P(c_k)$$

Notice that the process and the learning setting is probabilistic, however, the interpretation of the hypothesis h can be either deterministic or probabilistic, or anything else. This puts the flexibility into the setting of the model, and permits us to consider various representation of the model structure. Furthermore, the hypothesis for the learning process is evaluated relative to the same probabilistic setting, in which the training takes place, and we allow hypotheses that are only approximation of the target concept.

8.3.3 Third section

There exists an evaluator (or *supervisor*) $\nabla(h, c)$ that evaluates specific *loss framework* $\ell\{h(x), c(x)\}$ based on available information, and a *update modifier* $U(\ell, \mathcal{A})$ of certain objective to algorithm $\mathcal{A}$.

In this step of the procedural setting, the *supervisor* includes a loss function, ℓ, which takes discrete arguments, and binary evaluate the discrete result between h and c. This loss function is supposed to have a global minimum, or at least a certain region of minimal approximation. We have the following definition.

Definition 8.3.3 (Loss function). *A loss function is a non-negative function* $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, +\infty)$.

In general, the following is supposed of the loss function:

Conjecture 8.3.1 (Loss function convexity). *For any supervisor* ∇ *of a learner framework* $\mathcal{L}(\mathcal{H}, \mathcal{A})$*, the loss function* ℓ *is assumed to be a p-converging function, or as a convex function. That is, for* $\ell : \mathbb{R}^n \to \mathbb{R}$*, then* ℓ *is convex on its domain, or is locally convex on* $[a, b] \subset \mathbb{R}^n$*, if, for* $x, y \in \mathbb{R}^n$ *of such interval, and for* $\lambda \in [0, 1]$*, it satisfies*

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda) f(y) \tag{8.10}$$

Under a single structure, that is, for example, a class of mapping $\mathbb{R}^n \to \mathbb{R}$, there can exist many loss functions to be considered. Take for example the setting of regression analysis. Then,

[1] Note that, in some aspect, this is similar to the usage of mathematical simulation of certain dynamical system, or *any* system that has certain patterns or relations observable.

[2] The model itself can be entirely deterministic, while the learning process is not. in fact, one of the very first assumption and axiomatic view of the learning problem is that learning occurs in a *probabilistic setting*.

ℓ can be either the absolute error, $|h(x) - c(x)|$, or the L^2 norm, $||h(x), c(x)||_2$. Different loss function provides different path and landscape, though they still share somewhat similar interpolation patterns, or either some might be subjected to, under the consideration of a loss landscape, local minima than others. Furthermore, the form and representation of loss functions is not discrete - but rather based of its interpretation and the supposition of the underlying notion required for the loss function to operate in. For example, if the setting is *generative*, the loss function would take the form of a *divergence measure* between two probability distribution. Notice however, that the action potential provided by the model in their operation disappears, or rather, is not considered, when using the loss function.

Using such loss function, for the case of no noises or interference uncertainty, if the goal is to minimize the empirical risk on such dataset, then we call this the method of *empirical risk minimization*, or ERM. This is born out of the consideration that in practice, the oracle $EX(c, \mathcal{D})$, and the actual generalization error $R(h)$ is not obtainable.

Definition 8.3.4 (Empirical risk minimization). *Given $h \in \mathcal{H}$, for $c \in \mathcal{C}$ of some concept class, empirical risk minimization seeks to find $h' \in \mathcal{H}$ such that*

$$h' = \arg\min_{h \in \mathcal{H}} \hat{R}(h) \tag{8.11}$$

This strategy focus on dealing specifically with the empirical error only, hence, we gain ourselves the empirical best in this situation.

Proposition 8.3.1. *For any sample S, the following inequality holds for the hypothesis returned by ERM:*

$$\mathbb{P}\Big[R(h'_S) - \inf_{h \in \mathcal{H}} R(h) > \epsilon\Big] \leq \mathbb{P}\Big[\sup_{h \in \mathcal{H}} |R(h) - \hat{R}_S(h)| > \frac{\epsilon}{2}\Big] \tag{8.12}$$

We will prove this proposition in later section. For now, this proposition bounds the error of ERM, with respect to the probability of the generalization distance $|R(h) - \hat{R}_S(h)$ between the two errors. As we might have suspected, the performance of ERM is typically very poor. This is because the algorithm disregards the complexity of the hypothesis set $\mathcal{H}$: in practice, either it is not complex enough, in which case the error can be large, or $\mathcal{H}$ is so rich, that the complexity makes the model shoot out of the complexity range. Additionally, in many cases, determining the ERM solution is computationally intractable.

Another alternative idea of optimization or model selection can be *Structural Risk Minimization*, or SRM, though we have to develop our theory more to reach the conclusions and results specified by SRM.

8.4 The problems within classical learning theory

As we have settled on a particular setting of the statistical learning theory in the classical sense, it's time for us to figure out the problems, and what is intended of the theory to deal with.

Machine learning, the concern with the action of *learning*, and the general setting of the model contains the processes, procedures itself. In such sense, most of the problem will be cut out into different processes. We can identify this into a few parts, as in our example has shown.

8.4.1 Data and the general setting

While we have been talking in the learning setting pretty loosely on the side of the data itself, it is the data that gives us the problem setting. For example, if the data is a pair (X, Y)

of input output representation, then we know that our working space would be $\mathbb{R}^n \times \mathbb{R}^m$ or any of its subset, given the fact that data must be encoded into numerical representations, even in discrete or logical case (in such case, typically, $0-1$ pair is enough). In more complex and complicated setting, for example, on the structure of a graph and its embedded representation, it is much more difficult to see the *form of dataset* to be, hence, also the working space.

Another thing with data is the *availability of the data* and the *distribution of the data* also matters. In some cases, our data consists of only skewed or partitioned sections of the actual concept range, at least up to the observable criterion of the concept (some concept does not exist in certain range, at least in a numerical example, and some other reasons). In the other case, the availability of data, meaning the density of data, as well as added *observational effect* also alters the learning setting, simply because learning theory relies on, at least from our glance, simple statistical, observational learning from observation space, without the added structures. An actual, practical concern however, lies in the range of **practical empirical-generalization** setting.

8.4.2 Observational partitioning

Previously, we established that at least in said formal setting, generalization errors cannot be known beforehand. Thereby, all of our operations, procedures, processes has to be conducted using the observed data, or the sample space $\mathcal{S}$. A solution to utilize that is to consider the generalization region as the region of *new, unseen observations* that while still is of the same concept c, but it is indeed, unseen if the model is never provided of such instances. Hence, it prompts the utilization of limited resources by then to partition the sample space into various subspaces, that would be then fed into the learning process as observational space. This is simply called **sample set partitioning**. Denoted the number of partition (discrete) as k, then for $k=2$, then this is called the **train-test partition** (or dataset). This contains a dataset $\mathbf{x}_1$ and $\mathbf{x}_2$ of the same original sample set, such that $\mathbf{x}_1/\mathbf{x}_2 \geq 1$ (one is bigger than another, usually). The ratio does indeed affect the overall running procedure, accordingly.

If $k > 3$ and above, we call it the nth partitioning. If we further consider the choice of observational spaces being provided, then if $k > 3$, and subspaces are chosen randomly, it is called **cross-validating partition**. The effect of them are fairly well-understood, and we might as well refer to authoritative sources on such issue.

Based on the above conclusion, the empirical error and generalized error can be re-calculated to fit the general scheme. For $|\mathcal{S}| = m$ and partition $k + p = m$ for train and test partition respectively,

$$\hat{R}_k(h) = \frac{1}{k}\sum_{i=1}^{k} \ell\{h(x_i), c(x_i)\}, \quad \hat{R}_p(h_{\mathcal{A}}) = \frac{1}{k}\sum_{i=1}^{p} \ell\{h_{\mathcal{A}}(x_i), c(x_i)\} \tag{8.13}$$

where $h_{\mathcal{A}}$ is the result from the learning algorithm (procedure), for $h^* \neq h_{\mathcal{A}}$. Then, the empirical learning seeks to optimize the first term, that is, for the objective

$$\arg\min_{h\in\mathcal{H}} \hat{R}_k(h) = \arg\min_{h\in\mathcal{H}} \frac{1}{k}\sum_{i=1}^{k} \ell\{h(x_i), c(x_i)\} \tag{8.14}$$

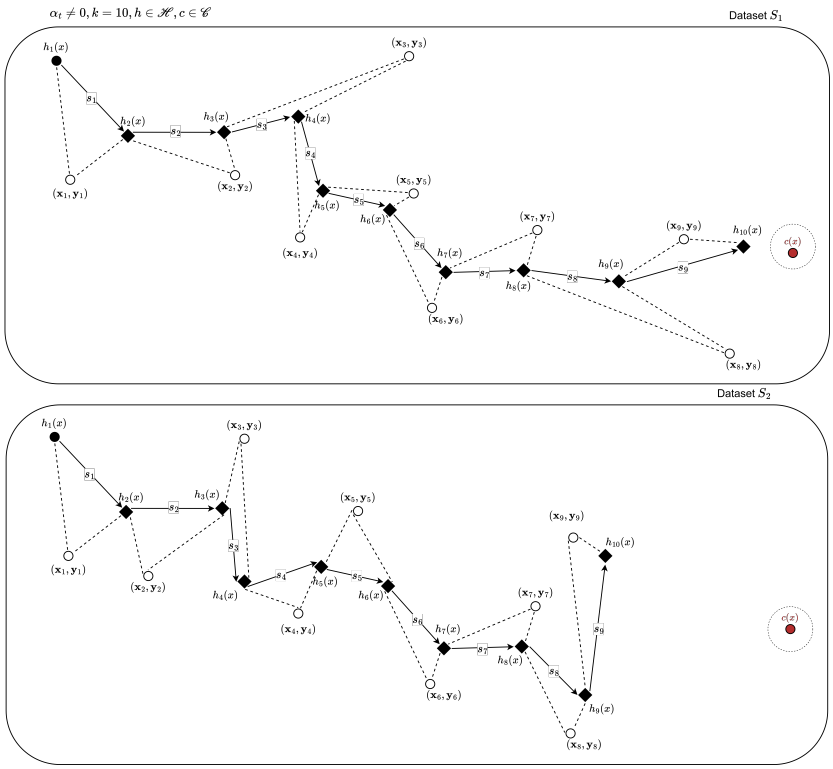


Figure 8.3: **Conceptual representation of the sample set partitions and its effect on iterative process.** (1), for S_1, and of the specified ordering, h is able to make it to the optimal point compared to the actual concept c. The bubble around c is what we call irreducible error, intrinsic of the observational space. (2) for S_2, of the changing dataset, while of the same partition but also changing order, gives different volatile path, and perhaps suboptimal performance compared to the first dataset case. Note that they are the supposed *optimal path* of both S_1, S_2. If randomization is introduced, may suboptimal path will occur, and the result will differ.

and the generalization learning as:

$$\begin{aligned}
\arg\min_{h\in\mathcal{H}} \hat{R}_p(h_{\mathcal{A}}) &= \hat{R}_k(h_{\mathcal{A}}) + \arg\min_{h\in\mathcal{H}} \hat{R}_p(h_{\mathcal{A}}) \\
&= \arg\min_{h\in\mathcal{H}} \left[\frac{1}{k}\sum_{i=1}^{k} \ell\{h_{\mathcal{A}}(x_i), c(x_i)\} + \frac{1}{k}\sum_{i=1}^{p} \ell\{h_{\mathcal{A}}(x_i), c(x_i)\} \right] \\
&= \arg\min_{h\in\mathcal{H}} \left[\epsilon + \frac{1}{k}\sum_{i=1}^{p} \ell\{h_{\mathcal{A}}(x_i), c(x_i)\} \right]
\end{aligned} \tag{8.15}$$

By the definition order, the generalization error can only be calculated post-$\mathcal{A}$. Hence, we often called the empirical, or ERM-generalizer case to be dynamic, and the solution for ERM is tested statically on generalization set. This partition of order of computation though, brings up the problem of updating factors and others contributing potential that will damage or diffuse the measure.

For cross-validation and other randomize partition, the formulation requires more care in the notion by itself. The structural risk minimization scheme dilemma can be understood from the above consideration naturally. We have effectively defined the two proxies, **heuristic em-**

pirical risk and **heuristic generalization risk.**

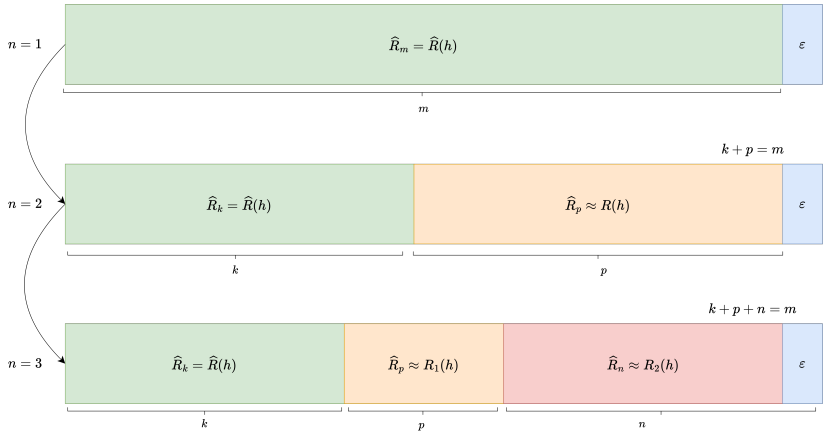


Figure 8.4: Partitioning process and its error potential consideration. We assume each partition includes the irreducible error ϵ accompanied by the n partition, belongs to the furthest partitioning set. Within every increasing partition, for supposed distributed data (unordered data), the generalization risk is further decomposed.

While they are effective as a pair, just as Figure 8.3 illustrated, overly optimizing the path can lead to it in general, be fairly ineffective. On the flip side, not sufficiently optimizing the path will lead to sub-average result, leading to adverse relation. Furthermore, since $\mathcal{S}$ might not, in general, be representative of the population that depicts c, the form c' created from observations of c-generated space (we call this the **observed concept** to the true concept) might be skewed, leading to adverse effect. From there, we gain the notion of **overfitting** and **underfitting** under the constraints of limited knowledge and reduced concept space to the observation space, relative to the true concept itself. It is also from here that the proxy concepts of **bias** and **variance** are introduced to construct the bias-variance tradeoff, and subsequently, the phenomena of double descent.

Another point to note is also the way of partitioning and location of the data. As we have suggested, data comes in a variety of range, of which we can directly infer from using the supremum set and the infimum element, given particular structure and encoding space. Denoted by $\mathbf{r}_{\mathcal{S}}$, it is responsible for the partitioning strength of the dataset on its own. Then, partitioning must also take the randomization chance, or the distribution and distinctive ordering of the dataset. That is, the metadata related to the dataset must also be configured, as suggestibly, there would be some observable behaviours typically available with, for example in permutation or linear ordering of data by magnitude and expression range. If that is the case, and the data is discretely fragmented, but not diffused in the manner that spread out the missing information all around the dataset, the result would be entirely different, and it would influence the generalization error, heuristically, far more than what is realized. Thereby, there then exists two fundamental polarity: either it is the *totally ordered set*, or it is *randomized uniform aberration*, for any given observation space.

8.5 Learning criteria on time - PAC theory

From our problems of the section 8.4, one of the most familiar and easier to analyse problem, is the related problem of *learning with time efficiency*. This has been bugging a lot of people (those scientist and researchers all along) and many things has been tried. One of the most popular,

and kind of the best, is the *Probably Approximately Correct* (PAC) learning rule, which takes into account the time that it would take a model to learning efficiently, by the arbitrary meaning of efficiency.

PAC learning

Why would we want to take on the computational complexity aspect? Well, it has to do with the nature of computer and time itself. Often, in practice, we have the probabilistic learning phase, and the deterministic (static) inference and test phase. That is, the only time that the dynamic is active, is when it is learning - otherwise, in usage, most of the time the system is static. Hence, the learning portion takes up almost 90% of the entire procedure. Many factors hence can be taken into account for the learning time to be calculated, often an approximation only, but some obviously stands out, like the relation between learning difficulties and the size of the sample space $\mathcal{S}$, or the complexity between the choice of the hypothesis set structure $\mathcal{H}$. To answer those questions on computational complexity is to reduce the time waiting, and more time on getting it work. Also, it will eventually help in the analysis of the efficiency of $\mathcal{H}$ on the time-axis. After all, who wants to wait?

8.5.1 Classical PAC-learning

For now, we assume no structure of h and c. They can be functions, partial functions, relations, complex algorithms, or others. In a typically learning setting, we also have the argument of *preliminary knowledge*, presented in literatures of the term *inductive bias*. For example, if the concept class $\mathcal{C}$ is assumed, then we say the setting is **model-specific**. If there exists no hard assumption on $\mathcal{C}$, then the learning setting is said to be **model-free**. For clarification, we also use in the examples the 0-1 loss function, $\ell = \delta(h, c)$ for the Kronecker delta, such that

$$\hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n} \delta_{h(\mathbf{x}_i)\neq y_i}, \text{ where } \delta_{h(\mathbf{x}_i)\neq y_i} = \begin{cases} 1, & \text{if } h(\mathbf{x}_i) \neq y_i \\ 0, & \text{o.w.} \end{cases} \tag{8.16}$$

We can later on replace δ and the binary setting with others, albeit the complexity will increase on the formal derivation for others loss functions. One of the main reason why we often choose $0-1$ loss error in our analysis, comes from a variety of reasons. But, from what can be seen, it reduces the complexity of the problem, as well as raises the absoluteness of the loss measure. Indeed, naturally, from a soft point of view, the mean square loss measure is more natural of $\mathcal{L}_2$ norm on a representation space. As we have been saying, the loss function is chosen can influence the problem setting that one might expect the model to act on. $1-0$ measure effectively reduces the problem to the absolute *binary classification problem*, where it is either correct or not.

The *learning algorithm* $\mathcal{A}$ can be interpreted to belong to a specific set of algorithm A such that it gives a sequence $\{A_i\}$. If there exists such that $|\{A_i\}| = n$ for a fixed n, and can be explicitly defined and expressed, then the algorithm is *deterministic*. Otherwise for arbitrary $m > 1$ varying in magnitude, and with added uncertainty - that is, no sequence $\{A_i\}$ is the same, the algorithm is said to be *probabilistic*, and by extension, a *stochastic process*. We will give the definition of the stochastic process later on. Then, the algorithm is denoted by

$$\mathcal{A} = [h]\{A_1, \ldots, A_n\} = \{A_i\}_{i=1}^{n}$$

Under this line, we separate the classification of c into two types: model-specific means that c is deterministic - it is supposed to belong to certain class $\mathcal{C}$, or rather, its description can be contained into certain concept class (which ultimately is some priori). Model-free means that c is not apparent to be able to separated to a concept class, which adds uncertainty, though all of them are still supposed to be at least a transformation $c : V \to U$ of arbitrary space V, U, by the setting of mathematical functional relationship.

Definition 8.5.1 (Model-specific learning). *A learning scenario is called model-specific learning if, for $R(h)$ being a measure-theoretic metric on the hypothesis h, we are given the dataset $D = \{(\mathcal{X}, \mathcal{Y})\}$ with distribution $x \sim \mathcal{D}$, problem is to find a hypothesis $h \in \mathcal{H}$ that satisfies:*

$$R(h) = \mathop{\mathbb{P}}_{x\sim\mathcal{D}}[h(x) \neq c(x)] = \mathop{\mathbb{E}}_{x\sim\mathcal{D}}[1_{h(x)\neq c(x)}] \tag{8.17}$$

In the same way, we define the model-free learning, where the distribution is extended to $\mathcal{Y}$. This is to facilitate the uncertainty of the concept class. If the concept class $\mathcal{C}$ is deterministic, then there would be no ambiguity of the concept $c \in \mathcal{C}$ - we would only have to learn $\mathbb{P}(y \mid x)$, to predict or reconstruct accordingly. In such case, the only factor in consideration is the ground space $\mathcal{X}$, or rather, the relationship between x and y itself.

Definition 8.5.2 (Model-free learning). *A learning scenario is called model-free learning if, for $R(h)$ being a measure-theoretic metric on the hypothesis h, we are given the dataset $D = \{(\mathcal{X}, \mathcal{Y})\}$ with distribution $(x, y) \sim \mathcal{D}$. Problem is to find a hypothesis $h \in \mathcal{H}$ that satisfies*

$$R(h) = \mathop{\mathbb{P}}_{(x,y)\sim D}[h(x) \neq y] = \mathop{\mathbb{E}}_{(x,y)\sim D}[1_{h(x)\neq y}] \tag{8.18}$$

In both setting, we take the stance of absoluteness, using measure $\neq$, similar to a type of discrete decision process called *classification*, for a finite class counts of the target class. This notion can be aggravated to be generalized to other classes, simply by replacing $h(x) \neq y$ by a valued-function, or by considering specific measure (for example, *Bregman measure* for strictly convex functions). In the context of n-dimensional Euclidean space, and with the isomorphism $\mathbf{E} \simeq \mathbb{R}^n$ of suitable basis, the measure is simply the finite-accuracy Euclidean distance $d(h(x), c(x))$.

Before analysing this problem in a PAC-like fashion, let us recall the setting in which we are operating upon.

- The goal of the learning game is to learn an unknown target set, but the target set is not arbitrary. Instead, there is a known and rather strong constraint on the target set — it is a rectangle in the plane whose sides are parallel to the axes. Under certain analysis, we can call this as having indeed a strong *priori* to the concept $c \in \mathcal{C}$ possible.
- Learning occurs in a probabilistic setting. Examples of the target rectangle are drawn randomly in the plane according to a fixed probability distribution which is unknown and unconstrained.
- The hypothesis of the learner is evaluated relative to the *same* probabilistic setting in which the training takes place, and we allow hypotheses that are only approximations to the target. The tightest-fit strategy might not find the target rectangle exactly, but will find one with only a small probability of disagreement with the target.
- We are interested in a solution that is efficient: not many examples are required to obtain small error with high confidence, and we can process those examples rapidly - that is, for time efficiency t. For it to be controllable, it is in our interest to know if we can run the learning process in polynomial time, that is, with P-complexity class.

Ideally, we would want our model to be able to do the following:

- The number of calls to the concept c, or by extension, the oracle process $EX(c, \mathcal{D})$, is small, in the sense that it is bounded polynomially by some parameters.
- The amount of computation performed is small.

- The algorithm outputs a hypothesis h such that Θ is small. Recall that $\Theta_h = (\hat{R}(h), R(h))$, but generally, the ideal goal would be to reduce to the infimum of $R(h)$.

We are now ready to specify the definition of Probably Approximately Correct learning. For now, we designate this as a preliminary definition, because we will add more features into it.

Definition 8.5.3 (PAC Learning, Preliminary Definition). *Let $\mathcal{C}$ be a concept class over X. We say that $\mathcal{C}$ is **PAC-learnable** if there exists an algorithm $\mathcal{L}$ that for every concept $c \in \mathcal{C}$, for every distribution $\mathcal{D}$ on X, and for all $0 < \epsilon < 1/2$ and $0 < \delta < 1/2$, if $\mathcal{L}$ is given access to $EX(c, \mathcal{D})$ for its output, and inputs ϵ, δ, then with probability at least $1 - \delta$, outputs a hypothesis $h \in \mathcal{H}$ satisfying $R(h) \leq \epsilon$. This probability is taken over the random examples drawn by calls to $EX(c, \mathcal{D})$, and any internal randomization of $\mathcal{L}$.*

*If $\mathcal{L}$ further run in time polynomial in $1/\epsilon$ and $1/\delta$, we say that $\mathcal{C}$ is efficiently PAC learnable. We will sometimes refer to the input ϵ as the **error parameter**, and the input δ as the **confidence parameter**.*

The hypothesis $h \in \mathcal{H}$ of the PAC-learning algorithm is thus approximately correct, with high probability, hence we got the name. Usually, $\mathcal{H}$ can not coincide with $\mathcal{C}$. However, a stronger form of the preliminary PAC-learning can set $h \in \mathcal{C}$, which guarantee the existence of a near-perfect hypothesis class.

For the preliminary PAC learning model, there are two important problems with it, or rather, remarks on how it is constructed.

First, the parameter pair ϵ, δ control two types of failure to which a learning algorithm in the PAC model is inevitably susceptible. The error parameter ϵ is necessary since there may be only a negligible probability that a small random sample will distinguish between two competing hypotheses that differ on only one improbable point in the instance space. The confidence on only one improbable point in the instance (ground) space. The confidence parameter δ is necessary since the learning algorithm may occasionally be extremely unlucky, and draw a terribly "unrepresentative" sample of the target concept - for example, a sample consisting only of repeated draws of the same instances. The best we can hope for is that the probability of both types of failure can be made arbitrarily small at a modest cost.

Second, we notice that we demand a PAC learning algorithm perform well with respect any distribution $\mathcal{D}$. This is a very strong requirement, which is moderated by the fact that we only evaluate the hypothesis of the learning algorithm with respect to the same distribution $\mathcal{D}$.

Representation size and PAC-configuration

An important issue that has not been mentioned in our previous formation of PAC learning, is the fundamental distinction between a concept and its representation.

Consider a class of concepts defined by the satisfying assignments of certain formulae. A concept from this class that satisfies such formulae, can be represented by a formula f, a truth table, or any given formulae that is tautologically equivalent formulae f' to f. An example can be shown in high-dimensional Euclidean space $\mathbb{R}^n$, we may choose to represent a convex polytope (the hell is this?), we can specify the problem by either specifying its vertices, or specifying linear equations for its faces, and these two representation scheme, while being of the same problem, can differ in size.

representation scheme

In each of these examples, we are fixing some representation scheme - that is, a precise method for encoding concepts - and then examining the size of the encoding for various concepts. Other natural representation schemes that we are familiar it could be the vanilla neural network, or decision trees, though we cannot say exactly that the representation of neural network is indeed of the 'vanilla' representation. Taken example of the boolean formulae, for

example, the boolean parity function $f(x_1, \ldots, x_n) = x_1 \oplus \cdots \oplus x_n$ which can be computed by a circuit of $\wedge, \vee$ and $\neg$ gates whose size is bounded by a fixed polynomial in n, but to represent this same function as a disjunctive normal form (DNF), requires size exponential in n. In these representation schemes, there is an obvious mapping from the representation (can be decision tree, or neural network) to the set or boolean function that is being represented. There is also a natural measure of the size of given representation in the scheme (for example, the number of weights, or neuron in a neural network).

Since our PAC-learning algorithm for its model, follows the phenomenological interpretation, hence it is experimental data-only, it has absolutely no information about which, if any, of teh many possible representations is actually being used to represent the target concept in reality. However, it matters greatly which representation the algorithm chooses for its hypothesis, since the time to write this representation down is obviously a lower bound on the running time of the algorithm. At the end though, note that this is purely a computational concern at a glance, yet, further analysis into the problem might stem out certain consideration that previously unseen, for example, that fact that the representation scheme actually matters more in sense of a system analysis and its dynamic.

Formally, a *representation scheme* for a concept class $\mathcal{C}$ is defined as followed to capture this notion of representation size, and can be defined as below.

Definition 8.5.4 (Representation scheme). *For a given concept class $\mathcal{C}$, a **representation scheme** for such concept class is a function $\boldsymbol{\mathcal{R}} : \Sigma^* \to \mathcal{C}$, where Σ is the finite alphabet of symbols. We call any configuration $\sigma \in \Sigma^*$ such that $\boldsymbol{\mathcal{R}}(\sigma) = c$ a **representation** of c, under $\boldsymbol{\mathcal{R}}$. For any given c, the representation scheme might not be unique.* [a]

[a] Perhaps representation scheme can be further realized by applying a Borel σ-algebra on top, and put on it a perspective measure to handle the notion of representation size. For example, in the traditional example of axis-aligned rectangle, one such representation scheme could be the real number schema $\boldsymbol{R} : (\Sigma \cup \mathbb{R})^* \to \mathcal{C}$, which then can utilize the extended notion of a Lebesgue measure over $\mathbb{R}$ and its spaces.

To capture the notion of representation size, we assume that associated with $\boldsymbol{\mathcal{R}}$, there is a mapping size: $\Sigma^* \to N$ that assigns a natural number $size(h)$ to each representation $h \in \Sigma^*$. Note that we allow $size(\cdot)$ to be any such mapping results obtained under a particular definition for $size(\cdot)$ will be meaningful only if this definition is natural. In a more simple case, and arguably realistic setting, we can take $\Sigma = \{0, 1\}$, thus, we have a binary encoding of concepts, and define $size(h)$ to be the length of h in bits. Although we will use other definitions of size when binary representations are inconvenient, our definition of $size(\cdot)$ will always be within a polynomial factor of the binary string length definition.

So far, this definition is applicable only to representations, that is, to strings $h \in \Sigma^*$. We would like to extend this to $c \in \mathcal{C}$. Since the learning algorithm has access only to the input-output behaviour of c, in the worst case, it must assume that the simplest possible mechanism is generating this behaviour. Thus, we define $size(c)$ to be the infimum, that is:

$$size(c) = \min_{\boldsymbol{\mathcal{R}}(\sigma)=c} \{size(\sigma)\}$$

In other words, $size(c)$ is the size of the smallest representation of the concept c in the underlying representation scheme $\boldsymbol{\mathcal{R}}$.

Definition 8.5.5 (Representation complexity). *Given a concept $c \in \mathcal{C}$, the **representation complexity** of c is defined to be the size of the smallest representation of the concept c in the underlying representation*

scheme $\mathcal{R}$, that is,

$$size(c) = \min_{\mathcal{R}(\sigma)=c} \{size(\sigma)\}$$

for any string σ of the representation scheme $\mathcal{R}$ that still represents c.

Under such definition, the more "complex" the concept c is with the respective chosen representation scheme, the larger $size(c)$ is. This is also where the concept of the concept class $\mathcal{C}$ comes from - it comes from the *representation class indiction* that we have in mind some fixed concept classes we study by their representation scheme.

From this consideration of the representation size for the language in which the problem setting is expressed in, we gain the complete definition of PAC-learning for a fixed representation priori of the concept class.

Definition 8.5.6 (PAC-learning). *A concept class $\mathcal{C}$ is said to be **PAC-learnable** if there exists an algorithm $\mathcal{A}$ and a polynomial function $poly(\cdot,\cdot,\cdot,\cdot)$ of 4-argument such that for any $\epsilon > 0$, $\delta > 0$, for all distribution $\mathcal{D}$ on $\mathcal{X}$ and for any target concept $c \in \mathcal{C}$, the following holds for any sample size $m \geq poly(1/\epsilon, 1/\delta, n, size(c))$:*

$$\Pr_{S\sim\mathcal{D}^m}[R(h(S)) \leq \epsilon] \geq 1 - \delta$$

for a given error measure $R(h_S)$. If $\mathcal{A}$ further runs in $poly(1/\epsilon, 1/\delta, n, size(c))$, then $\mathcal{C}$ is said to be efficiently PAC-learnable. When such algorithm exists, we call $\mathcal{A}$ a PAC-learning algorithm.

8.6 Generalization bound for PAC-learning

To understand generalization bound, we first need to introduce the categorization of the hypothesis based of one criterion: consistency.

8.6.1 Consistent Learning

We'll define the notion of a consistent learning algorithm, or consistent learner, for a concept class C and hypothesis h.

Definition 8.6.1 (Consistent Learner). *We say that an algorithm is a consistent learner for a concept class C using hypothesis class $\mathcal{H}$, if for all n, for all $c \in C_n$ and for all m, given*

$$S = \{(x_1, c(x_1)), (x_2, c(x_2)), \ldots, (x_m, c(x_m))\}$$

as input, $x_i \in X_n$ outputs, then the algorithm L outputs a hypothesis $h \in \mathcal{H}_n$ such that

$$h(x_i) = c(x_i), i = 1, \ldots, m$$

Then, a consistent hypothesis is the hypothesis that is consistent with all the training data provided to it from the concept c. The following section contains the debate between consistent hypothesis (i.e., for example, admitting no error on training data, almost perfect - maybe even so), and inconsistent hypothesis, of which have general defects from the actual concept, or even different by a margin.

8.6.2 Finite H, consistent hypothesis

We will consider the general sample complexity bound, or equivalently, a generalization bound for consistent hypothesis in the case where the cardinality $|\mathcal{H}|$ is finite. We will assume, as such, that the target concept c is in $\mathcal{H}$.

Theorem 8.6.1 (Learning bound - finite $\mathcal{H}$, consistent case). *Let H be a finite set of functions mapping from $\mathcal{X} \to \mathcal{Y}$. Let $\mathcal{A}$ be an algorithm that for any target concept $c \in H$ and i.i.d. samples S returns a consistent hypothesis H_S, such that $\hat{R}(h_S) = 0$. Then for any $\epsilon, \delta > 0$, the inequality $\Pr_{S \sim D^m}[R(h_S) \leq \epsilon] \geq 1 - \delta$ holds if*

$$m \geq \frac{1}{\epsilon}\left(\log |H| + \log \frac{1}{\delta}\right)$$

This sample complexity result admits the following equivalent statement as a generation bound: for any $\epsilon, \delta > 0$, with probability at least $1 - \delta$,

$$R(h_S) \leq \frac{1}{m}\left(\log |\mathcal{H}| + \log \frac{1}{\delta}\right) \tag{8.19}$$

Proof. Fix $\epsilon > 0$. We do not know which consistent hypothesis $h_S \in H$ is selected by the algorithm $\mathcal{A}$. This depends on S. Therefore, we need to give a uniform convergence bounds, that is, a bound that holds for the set of all consistent hypothesis. Thus, we will bound the probability that some $h \in H$ would be consistent and have error more than ϵ, denoted by:

$$\Pr[\exists h \in H : \hat{R}(h) = 0 \wedge R(h) > \epsilon]$$

This is equal to:

$$\Pr(Q) = \Pr[(h_1 \in H, \hat{R}(h_1) = 0 \wedge R(h_1) > \epsilon) \vee (h_2 \in H, \hat{R}(h_2) = 0 \wedge R(h_2) > \epsilon) \vee \dots]$$

For shorthand notation, we set $\mathcal{H}_\epsilon = \{h \in \mathcal{H} : R(h) > \epsilon\}$. Hence,

$$\Pr(Q) = \mathbb{P}[\exists h \in \mathcal{H}_\epsilon : \hat{R}_S(h) = 0]$$

We can see that

$$\Pr(Q) \leq \sum_{h \in H} \Pr[\hat{R}(h) = 0 \wedge R(h) > \epsilon]$$

by union bound, and by conditional probability,

$$\Pr(Q) \leq \sum_{h \in H} \Pr[\hat{R}(h) = 0 \mid R(h) > \epsilon]$$

Now, consider any $h \in H$ with $R(h) > \epsilon$. Then the probability that h is consistent on training sample S without error, can be bounded as:

$$\Pr[\hat{R}(h) = 0 \mid R(h) > \epsilon] \leq (1 - \epsilon)^m$$

with m the number of samples. This implies that:

$$\Pr[\exists h \in H : \hat{R}(h) = 0 \wedge R(h) > \epsilon] \leq |H|(1 - \epsilon)^m \leq |H|e^{-m\epsilon}$$

Setting the right-hand side to equal to δ, we gain the above statement. Hence, proved. □

The theorem shows that when the hypothesis set $\mathcal{H}$ is finite, a consistent algorithm $\mathcal{A}$ is a PAC-learning algorithm, since the sample complexity is dominated by a polynomial in $1/\epsilon$ and $1/\delta$. The generalization error of consistent hypothesis is upper bounded by a term that decrease w.r.t m. The decreases rate of $O(1/m)$ is guaranteed by this theorem.

8.6.3 Examples

Boolean Conjunction

Consider the concept class $\mathcal{C}_n$ of conjunction of at most n Boolean literals $x_1, \ldots, x_n$.

A Boolean literal is either a variable $x_i, i \in [n]$ or its negation $\bar{x}_i$. For $n = 4$, this might be $x_1 \wedge \bar{x}_2 \wedge x_4$.

A simple algorithm for finding a consistent hypothesis is thus based on positive examples and consists of the following: For each positive example $(b_1, .., b_n), i \in [n]$, if $b_i = 1$ then $\bar{x}_i$ is ruled out as a possible literal in the concept class and if $b_i = 0$, hen x_i is ruled out. The conjunction of all of the literal not ruled out is a hypothesis consistent with the target.

We have $|\mathcal{H}| = |\mathcal{C}_n| = 3^n$ since each literal can be either 1,0 or not chosen. Plugging this into the complexity bound for $\epsilon > 0$ and $\delta > 0$,

$$m \geq \frac{1}{\epsilon}\left(n \log(3) + \log\left(\frac{1}{\delta}\right)\right)$$

Thus, the class of conjunction of at most n Boolean literals is PAC-learnable.

Universal Concept Classes

Consider the set $\mathcal{X} = \{0,1\}^n$ of all Boolean vectors with n components, and let $\mathcal{U}_n$ be the concept class formed by all subsets of $\mathcal{X}$. Is this concept PAC-learnable? To guarantee a consistent hypothesis, the hypothesis class must include the concept class, thus $|\mathcal{H}| \geq |\mathcal{U}_n| = 2^{(2^n)}$. We are given then

$$m \geq \frac{1}{\epsilon}\left(2^n \log 2 + \log\frac{1}{\delta}\right) \geq O\left(\log|\mathcal{H}| + \log\frac{1}{\delta}\right)$$

Hence, it is not guaranteed by the theorem that it is PAC-learnable. In fact, recalling the definition of PAC-learning, recall that for a concept class $\mathcal{U}_n$ to be PAC-learnable, it needs that

$$\Pr_{S\sim D^m}[R(\mathcal{U}_n) \leq \epsilon] \geq 1 - \delta, \quad m \geq \frac{1}{\epsilon}\left(\log|\mathcal{C}| + \log\frac{1}{\delta}\right), \mathcal{U}_n \in \mathcal{C}$$

But here, assuming that the error is set for $\epsilon > 0$ and $\delta > 0$, the polynomial exceeds the bound, hence it is not PAC-learnable.

8.6.4 Finite hypothesis sets H - inconsistent case

In the most general case, there may be no hypothesis in $\mathcal{H}$ consistent with the labelled training sample. This, in fact is the typical case in practice, where the learning problems may be somewhat difficult or the concept classes more complex than the hypothesis set used by the learning algorithm.

To derive the learning guarantees in the more general setting, we would use the following corollary, of which relates the generalization error and empirical error of a single hypothesis.

Corollary 8.6.1. *Fix $\epsilon > 0$. Then, for any hypothesis $h : X \to \{0,1\}$, the following inequalities hold:*

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left[\hat{R}_S(h) - R(h) \geq \epsilon\right] \leq \exp(-2m\epsilon^2) \tag{8.20}$$

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left[\hat{R}_S(h) - R(h) \leq -\epsilon\right] \leq -\exp(-2m\epsilon^2) \tag{8.21}$$

By the union bound, this implies the following two-sided inequality:

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left[|\hat{R}_S(h) - R(h)| \leq \epsilon\right] \leq 2\exp\left(-2m\epsilon^2\right) \tag{8.22}$$

Proof. This can be proved using Hoeffding inequality. Recall that the inequality is stated for $Z_1, \ldots, Z_n$ independent random variable, for range $z_i \in [a_i, b_i]$ with probability one, $S_n = \sum_{i=1}^{n} Z_i$ then for all $t > 0$:

$$\mathbb{P}(S_n - \mathbb{E}(S_n) \leq -t) \leq \exp\left(\frac{-2t^2}{\sum(b_i - a_i)^2}\right)$$

and

$$\mathbb{P}(S_n - \mathbb{E}(S_n) \geq t) \leq \exp\left(\frac{-2t^2}{\sum(b_i - a_i)^2}\right)$$

We can derive this for our case of the two empirical and generalization risk. Remember from 8.2.1 that we have $R(h) = \mathbb{E}[\hat{R}_S(h)]$, we can transform the phrase to be:

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left[\hat{R}_S(h) - R(h) \geq \epsilon\right] = \mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left[\hat{R}_S(h) - \mathbb{E}[\hat{R}_S(h)] \geq \epsilon\right]$$

with respect to $\mathcal{D}^m$. The form of $\hat{R}_S(h)$ has an additional $1/m$, hence, for simplification, we will take the form

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left[\hat{R}_S(h)/m - \mathbb{E}[\hat{R}_S(h)]/m \geq \epsilon m\right]$$

By Hoeffding inequality, we have:

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left[\hat{R}_S(h)/m - \mathbb{E}[\hat{R}_S(h)]/m \geq \epsilon m\right] \leq \exp\left(\frac{-2m^2\epsilon^2}{\sum(b_i - a_i)^2}\right) \tag{8.23}$$

which is

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left[\hat{R}_S(h)/m - \mathbb{E}[\hat{R}_S(h)]/m \geq \epsilon m\right] \leq \exp\left(\frac{-2m\epsilon^2}{(b-a)^2}\right)$$

Here, our $\sum(b_i - a_i)^2$ was replaced by a more general bound which contains only the sufficient amount of m samples. This is because the sum follows from m samples, which are taken as random variable of choice, then, for $a \leq Z_i \leq b$

$$\frac{m^2}{\sum(b_i - a_i)^2} = \frac{m^2}{m(b-a)^2} = \frac{m}{(b-a)^2}$$

Since our hypothesis is mapping to $\{0, 1\}$, hence our result space is discretely bounded in the normal range, the inequality reduces to

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left[\hat{R}_S(h)/m - \mathbb{E}[\hat{R}_S(h)]/m \geq \epsilon m\right] \leq \exp\left(-2m\epsilon^2\right) \tag{8.24}$$

Rearranging the left-hand-side to the original yields the inequality. Doing the same for the second case, and using the union bound, we get the two-sided inequality for $|\hat{R}_S(h) - R(h)|$. □

Setting the right-hand side of (5) to be equal to δ and solving this for ϵ yields immediately the following bound for a single hypothesis.

Corollary 8.6.2 (Generalization bound - single hypothesis). *Fix a hypothesis $h : \mathcal{X} \to \{0, 1\}$. Then, for any $\delta > 0$, the following inequality holds with probability at least $1 - \delta$:*

$$R(h) \leq \hat{R}_S(h) + \sqrt{\frac{\log 2/\delta}{2m}} \tag{8.25}$$

Can we use this corollary to bound the generalization error of the hypothesis h_S returned by a learning algorithm when training on a sample S? No, since h_S is a random variable depends on the training set S, rather than being fixed. Unlike the case of a fixed hypothesis for which the expectation $\mathbb{E}[\hat{R}_S(h_S)]$ is the generalization error, the generalization error $R(h_S)$ is a random variable and in general distinct from the expectation, which is a constant.

Thus, as in the proof for the consistent case, we need a uniform convergence bound, that holds with high probability for all hypotheses $h \in \mathcal{H}$.

Theorem 8.6.2 (Learning bound - finite $\mathcal{H}$, inconsistent case). *Let $\mathcal{H}$ be a finite hypothesis set. Then, for any $\delta > 0$, with probability at least $1 - \delta$, the following inequality holds:*

$$\forall h \in \mathcal{H}, \quad R(h) \leq \hat{R}_S(h) + \sqrt{\frac{\log |\mathcal{H}| + \log 2/\delta}{2m}} \tag{8.26}$$

Thus, for a finite hypothesis set $\mathcal{H}$,

$$R(h) \leq \hat{R}_S(h) + O\left(\sqrt{\frac{\log_2 |\mathcal{H}|}{m}}\right)$$

The term $\log |\mathcal{H}|$ can be interpreted as the number of bit needed to represent $\mathcal{H}$. A larger sample size m guarantees better generalization, and the bound increases with $|\mathcal{H}|$, but only logarithmically.

Note that the bound suggests seeking a trade-off between reducing the empirical error versus controlling the size of the hypothesis set: a larger hypothesis set is penalized by the second term but could help reduce the empirical error, that is the first term. But for a similar empirical error, it suggests using a smaller hypothesis set. This can be viewed as an instance of the so-called *Occam's Razor*, of which can be said as: *Plurality should not be posited without necessity*, or, the simplest explanation is best.

8.7 (Agnostic) General PAC-learning

For the definition of PAC-learning in the above setting, for model-specific notion, there are some issues with it, more specifically, with the type of *strong assumptions* that it holds.

1. The *representation size* and boundary only works for priori given of the concept class $\mathcal{C}$. In case that we cannot determine this, and there are a large potential for that to fail in capturing the minimal representation scope of the concept, the bound simply fails to a certain extent - given that our definition includes the polynomial bound of the concept class itself.
2. The assumption that the target concept c belongs to $\mathcal{C}$ means that we are trying to fit a hypothesis to data, which are a priori known to have been generated by some member of the model class defined by $\mathcal{C}$. However, in general, we may not want to assume much about the data generation process, and instead would like to find the best fit to the data at hand using an element of some model class of our choice.

3. The assumption that the training features are labelled noiselessly rules out the possibility of noisy measurements or observations.
4. Even if the above assumptions were somehow true, we would not necessarily have a priori knowledge of the concept class $\mathcal{C}$, containing the priori knowledge of the concept class $\mathcal{C}$ containing the target concept or function. In that case, the best we could hope for is to pick our own model class and seek the best approximation to the unknown target concept among the elements of that class.

Agnostic case is considered the most general scenario of supervised learning, where the distribution $\mathcal{D}$ is defined over $\mathcal{X} \times \mathcal{Y}$, and the training data is a labelled sample S drawn i.i.d. according to $\mathcal{D}$:

$$S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$$

The learning problem is to find a hypothesis $h \in \mathcal{H}$ with small generalization error

$$R(h) = \mathop{\mathbb{P}}_{(x,y)\sim\mathcal{D}}[h(x) \neq y] = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[1_{h(x)\neq y}] \tag{8.27}$$

This more general scenario is referred to as the stochastic scenario. Within this setting, the output target is a probabilistic function of the input. The stochastic scenario captures many real-world problems where the label of an input point is not unique. [3]

The *agnostic PAC-learning* format is then modified by considering $\mathcal{Y}$ as also a random variability.

Definition 8.7.1 (Agnostic PAC-learning). *A concept class $\mathcal{C}$ is said to be **PAC-learnable** if there exists an algorithm $\mathcal{A}$ and a polynomial function $poly(\cdot,\cdot,\cdot,\cdot)$ of 4-argument such that for any $\epsilon > 0$, $\delta > 0$, for all distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$ and for any target concept $c \in \mathcal{C}$, the following holds for any sample size $m \geq poly(1/\epsilon, 1/\delta, n)$:*

$$\Pr_{S\sim\mathcal{D}^m}\left[R(h(S)) - \min_{h\in\mathcal{H}} R(h) \leq \epsilon\right] \geq 1 - \delta \tag{8.28}$$

If $\mathcal{A}$ further runs in $poly(1/\epsilon, 1/\delta, n)$, then $\mathcal{C}$ is said to be efficiently agnostic PAC-learnable. When such algorithm exists, we call $\mathcal{A}$ an agnostic PAC-learning algorithm.

PAC-learning bound argues in the sense of *computational complexities*, and *space complexities*. Specifically, the term $poly(1/\epsilon, 1/\delta, n, size(c))$ hopes to bound the required learning process to polynomial time, specified by 4 parameters. Here, n is the *input dimension*, $size(c)$ is the encoding complexity of the target concept, $\epsilon > 0$ is the *accuracy bound*, and δ is the confidence bound - the probability that h fails.

8.7.1 Noise

Using the notion of the Bayes classifier and Bayes error, one can define the *noise* of a given learning setting under PAC-learning consideration, as followed. noise

[3] Considering two set X and Y. A probabilistic function from X to Y assign to each $x \in X$ a relevant sub-distribution of elements of Y, rather than a single value $y \in Y$. Such a sub-distribution is a function $\delta : Y \to [0, 1]$ such that

$$\sum_{y\in Y} \delta(y) \leq 1$$

In other words, a probabilistic function $X \to Y$ is a function $f : X \to (Y \to [0, 1])$, interpreted as $f : X \times Y \to [0, 1]$ such that for all $x \in X$,

$$\sum_{y\in Y} f(x, y) \leq 1$$

Definition 8.7.2 (Noise). *Given a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, the noise at point $x \in \mathcal{X}$ is defined by*

$$noise(x) = \min_{x \in \mathcal{X}} \{\mathbb{P}[1 \mid x], \mathbb{P}[0 \mid x]\}$$

The average noise or the noise associated to $\mathcal{D}$ is then the minimum possible error aside from intrinsic model error, and is calculated by $\mathbb{E}[noise(x)]$

Thus, the average noise is precisely the Bayes error, that is, $noise = \mathbb{E}[noise(x)] = R^*$. The noise is a characteristic of the learning task indicative of its level of difficulty. A point $x \in \mathcal{X}$, for which noise is close to $1/2$, is sometimes referred to as *noisy* and is of course a challenge for accurate prediction.

8.8 Occam's Razor

The PAC model that we introduced defined learning directly in terms of the predictive power of the hypothesis output by the learning algorithm. That is, given a hypothesis $h \in \mathcal{H}$ and the learning algorithm $\mathcal{L}$, the learning involves switching this dynamic of the predictive power of the hypothesis, over certain setting and problem. We can, as it is possible to apply this measure, to a learning algorithm because we made the assumption that the instances are drawn, from a probabilistic perspective, from a fixed distribution $\mathcal{D}$ then measured the predictive power with respect to the same distribution.

Our new notion, dubbed as *Occam learning*, would let us consider a different definition of learning that makes no assumptions about how the instances in a labelled sample are chosen, though we will assume that the labels are generated by a target concept chosen from a known class. Instead of measuring the predictive power of a hypothesis, the new definition judges the hypothesis by how succinctly it explains the observed data (or a labelled example). The crucial difference between PAC learning and Occam learning, is then that in PAC learning, the random sample drawn by the learning algorithm is intended only as an aid for reaching an accurate model of some external process, while in the new definition, we are concerned only with the fixed sample before us, and not any external process. By that, we simply reduce it to somewhat similar of the statistical estimation problem setting, though not exactly the same.

This also contains the substance of which we often call the *Occam's razor*, telling that overly complex scientific theories should be subjected to a simplifying knife. If we equate "simplicity" with representational succinctness, then another way to interpret Occam's razor would be to say learning is the act of finding a pattern in the observed data that facilitates a compact representation or compression of this data. In our simple concept learning setting, succinctness is measured by the size of the representation of the hypothesis concept. Equivalently, we can measure succinctness by the cardinality of the hypothesis class used by the algorithm, for if this class is small then a typical hypothesis from the class can be represented by a short binary string, and if this class is large then a typical hypothesis must be represented by a long string. Thus, an algorithm is an *Occam algorithm* if it finds a short hypothesis consistent with the observed data.

Additionally so, it would be plenty interesting to see that Occam's razor somehow falls into the range of what would be expected of the *bias-variance tradeoff*. This perhaps, will prompt us to push further into the inquiry of double descent, which somehow in a fantastic way, breaks the principle of the Lord William Occam.

8.8.1 Occam Learning and Succinctness

Let $X = \cup_{n \geq 1} X_n$ be the instance space, let $\mathcal{C} = \cup_{n \geq 1} \mathcal{C}_n$ be the target concept class, and let $\mathcal{H} = \cup_{n \geq 1} \mathcal{H}_n$ be the class of hypothesis representation. The notation is preferably clarified: X_n is under the binary representation typically concerned $\{0,1\}^n$, or a boolean string of length n.

Hence, $\mathcal{H}_n$ and $\mathcal{C}_n$ are subsequently concept and hypothesis class specified for such string space. In this part, we will assume, unless explicitly stated otherwise, that the hypothesis representation scheme of $\mathcal{H}$ uses a binary alphabet, and we define $size(h)$ to be the length of the bit string h. Also, recall that for a concept $c \in \mathcal{C}$, $size(c)$ denotes the size of the smallest representation of c in $\mathcal{H}$. Let $c \in \mathcal{C}_n$ denote the target concept. A labelled sample S of cardinality m is a set of pairs:

This section is taken, and studied, from a pretty old book. It would be then observed that a lot of the concept, including the representation, is taken in a rather computer science fashion such as for the hypothesis $\mathcal{H}$ to be binary representation.

$$S = \{(x_1, c(x_1)), \dots, (x_m, c(x_m))\}$$

An **Occam algorithm** L takes as input a labelled sample S, and outputs a short hypothesis h, "relatively so", that is consistent with S. By consistent, we mean as one definition above, $h(x_i) = c(x_i)$ for each i, and for short we mean that $size(h)$ is a sufficiently slowly growing function of n, $size(c)$ and m. This is formalised in the following definition.

Definition 8.8.1 (Occam algorithm). *Let $\alpha \geq 0$ and $0 \leq \beta < 1$ be constant. L is an (α, β)-**Occam algorithm** for $\mathcal{C}$ using $\mathcal{H}$ if on input a sample S of cardinality m labelled according to $c \in \mathcal{C}_n$, L outputs a hypothesis $h \in \mathcal{H}$ such that:*

- *h is consistent with S, or $h(x_i) = c(x_i)$ for all $x_i \in S_x$.*
- *$size(h) \leq (n \cdot size(c))^\alpha m^\beta$*

*We say that L is an **efficient** (α, β)-Occam algorithm if its running time is bounded by a polynomial in n, m and $size(c)$.*

The growth complexity $\mathcal{O}(n \cdot size(c))^\alpha m^\beta$ is somewhat difficult to scale down. However, we can somewhat try to prove if it is convergence or not.

Theorem 8.8.1. *Fix n. Then $\mathcal{O}(n \cdot size(c))^\alpha m^\beta$ is convergence or divergence depends on the ratio of α and β.*

Proof. Fixed n, then we have the class $\mathcal{C}_n$ to be finitely restricted of n representation class. Then, $size(c)$ for $c \in \mathcal{C}$ is also constant. We then reduce the first complexity measure to be $\mathcal{O}(\lambda^\alpha m^\beta)$. Since $0 \geq \beta < 1$, it follows that as β increase, $size(h)$ monotonically decrease. This is also the case for α in range $[0, 1)$. Hence, it converges to at least n, and diverges accordingly for arbitrary $\alpha > 1$, though the trend will not as strong if $m > n$, or $m \gg n$. □

Now, in what sense can we say that the output h of an Occam algorithm succinct? First, let us assume that $m \gg n$, so that the above bound can be effectively simplified to $size(h) < m^\beta$, for some $\beta < 1$. Since the hypothesis h is consistent with the sample S, H allows us to reconstruct the m labels $c(x_1) = h(x_1), \dots, c(x_m) = h(x_m)$, and is given only the unlabelled sequence of instance $x_1, \dots, x_m$. Thus the m bits $c(x_1), \dots, c(x_m)$ have been effectively *compressed* into a much shorter string h of length at most m^β. Note that the requirement $\beta < 1$ is quite weak, since a consistent hypothesis of length $O(mn)$ can always be achieved by simply storing the sample S in a table (at a cost of $n + 1$ bits per labelled example) and giving an arbitrary answer for instances that are not in the table. We would be certainly not expecting such a hypothesis to have any predictive power.

By sheer observation of the definition we have been constructing, the $(1, 0)$-Occam algorithm is the largest deviation, with the bound $size(h) \leq n \cdot size(c)$. This means it can at most, have the representation size equal to *all* representation partition of $c \in \mathcal{C}$. It is also good to note the misconception, or rather the misinterpretation: $size(c)$ is different from n, even though they are particularly the same from first glance. It is because n is actually the representation

size of the instance string, and not the concept representation string, which utilizes another alphabet entirely. Though, suppose they are of the same alphabet, then $X = \{\Sigma\}_n$, then again, there exists a number q such that $q \geq n$. In the case where $q = n$, it is most likely that the representation is constrained of the linear transformation order, of which does not change the string length.

Let us observe that even in the case $m \ll n$, the shortest consistent hypothesis in $\mathcal{H}$ may in fact be the target concept, and so we must allow $size(h)$ to depend at least linearly on $size(c)$. We will see cases where this makes it easier to efficiently find a consistent hypothesis — by contrast, computing the shortest hypothesis consistent with the data is often a computationally hard problem.

Theorem 8.8.2 (Occam's Razor). *Let L be an efficient (α, β)-Occam algorithm for $\mathcal{C}$ using $\mathcal{H}$. Let $\mathcal{D}$ be the target distribution over the instance space X, let $c \in \mathcal{C}_n$ be the target concept, and $0 < \epsilon, \delta \leq 1$. Then there is a constant $a > 0$ such that if L is given as input a random sample S of m examples drawn from $EX(c, \mathcal{D})$, where m satisfies:*

$$m \geq a\left(\frac{1}{\epsilon}\log\frac{1}{\delta} + \left(\frac{(n \cdot \text{size}(c)^{\alpha})}{\epsilon}\right)^{1/1-\beta}\right) \tag{8.29}$$

then with probability at least $1 - \delta$ the output h of L satisfies $error(h) \leq \epsilon$. Moreover, L runs in time polynomial in n, $size(c)$, $1/\epsilon$ and $1/\delta$.

Notice that as β tends to 1, the exponent in the bound for m tends to infinity. This corresponds with the assumption of intuition that the length of the hypothesis approaches that of the data itself, then the predictive power of the hypothesis is diminishing.

To apply this theorem into variational setting, we turn to the arguably more general definition, measuring representational succinctness by the cardinality of the hypothesis class rather than the representationally supposed bit length $size(h)$. Then, theorem 8.8.2 will be the special case of such theorem. To make this precise, let $\mathcal{H}_n = \cup_{m \geq 1}\mathcal{H}_{n,m}$

Theorem 8.8.3 (Occam's Razor, Cardinality Version). *Let $\mathcal{C}$ be a concept class and $\mathcal{H}$ a representation space. Let L be an algorithm such that for any n and any $c \in \mathcal{C}_n$, if L is given as input a sample S of m labelled examples of c, then L runs in time polynomial in n, m, and $size(c)$, and output an $h \in \mathcal{H}_{n,m}$ that is consistent with S. Then there is a constant $b > 0$ such that for any n, any distribution $\mathcal{D}$ over X_n, and any target concept $c \in \mathcal{C}_n$, if L is given as input a random sample from $EX(c, \mathcal{D})$ of m examples, where $|\mathcal{H}_{n,m}|$ satisfies:*

$$\log|\mathcal{H}_{n,m}| \leq b\epsilon m - \log\frac{1}{\delta} \tag{8.30}$$

or equivalently, where m satisfies $m \geq (1/b\epsilon)(\log|\mathcal{H}_{n,m|} + \log(1/\delta))$ then L is guaranteed to find a hypothesis $h \in \mathcal{H}_n$ that with probability at least $1 - \delta$ obeys $error(h) \leq \epsilon$.

We do not mention, or necessarily claim that L is an efficient PAC learning algorithm. In order for the theorem to apply, we must pick m large enough so that $b\epsilon m$ dominates $\log|\mathcal{H}_{n,m}|$.

8.9 Rademacher Complexity

The hypothesis sets typically used in machine learning are infinite (this is,... quite a problem). But the sample complexity bounds are uninformative when dealing with infinite hypothesis sets - for whatever this means. One could ask whether efficient learning from a finite sample is even possible when the hypothesis set $\mathcal{H}$ is infinite. Our goal with the Rademacher

complexity is this exact question, for the first complexity treatment. The general idea for doing so consists of reducing the infinite case to the analysis of finite sets of hypotheses, and then proceed as previous sections.

8.9.1 Rademacher descriptions

We will continue to use $\mathcal{H}$ to denote a hypothesis set as in the previous chapters. Many of the result of this section are general and hold for arbitrary loss function, or $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. In what follows, $\mathcal{G}$ will generally be interpreted as the *family of loss functions associated to* $\mathcal{H}$, mapping from $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ to $\mathbb{R}$. That is,

$$\mathcal{G} = \{g : (x, y) \mapsto L(h(x), y) : h \in \mathcal{H}\}$$

However, the definitions are given in the general case of a family of arbitrary functions $\mathcal{G}$, mapping from an arbitrary input space $\mathbb{Z}$ to $\mathbb{R}$. Thus, we can say that we endowed upon the landscape a measure set $\mathcal{G}$ of all singular mapping.

The **Rademacher complexity** captures the richness of a family of functions by measuring the degree to which a hypothesis set can fit random noise. This is perhaps one of the very *empirical complexity notion* that would be seen, as for the random noise idea, that is. ERC

Let G be a family of functions mapping from $Z \to [a, b]$, $S = (z_1, \ldots, z_m)$ a fixed sample of size m with elements in Z. Then, the **empirical Rademacher complexity** of G with respect to the sample S is defined as:

$$\hat{\mathfrak{R}}_S(\mathcal{G}) = \mathbb{E}\left[\sup_{g \in G} \frac{1}{m} \sum_{i=1}^{m} \sigma_i g(z_i)\right]$$

where $\boldsymbol{\sigma} = \{\sigma_1, \ldots, \sigma_m\}^\top$ with σ_is independent uniform random variables taking values in $\{-1, +1\}$. The random variable σ_i are called **Rademacher variables**. Rademacher variables

The empirical Rademacher complexity can be rewritten as:

$$\hat{\mathfrak{R}}_S(\mathcal{G}) = \mathop{\mathbb{E}}_{\sigma}\left[\sup_{g \in G} \frac{\boldsymbol{\sigma} \cdot \mathbf{g}_S}{m}\right]$$

for $\mathbf{g}_S$ denoting the vector of values taken by g over the sample S: $\mathbf{g}_S = (g(z_1), \ldots, g(z_m))^\top$. The inner product measures the correlation of $\mathbf{g}_S$ with the vector of random noise $\boldsymbol{\sigma}$. The supremum $\sup_{g \in \mathcal{G}}(\boldsymbol{\sigma} \cdot \mathbf{g}_S / m)$ is a measure of how well the function class $\mathcal{G}$ correlates with $\boldsymbol{\sigma}$ over the sample S. Thus, the empirical Rademacher complexity measures on average how well the function class correlates with random noise on the dataset. What will the random noise be like? Well, not much. For a fixed sample size, it captures how **random** the loss function distribution is. A random noise system is inherently chaotic by default, even if we restrict its value in $\{-1, +1\}$ [4]. Hence, if the model is good enough on certain dataset, that the noisy, useless data configuration performs "somewhat pretty well", then from that, we can know the complexity of the model by its flexibility in such noisy fitting. That is, in the topic of fitting, at least. So the richer or more complex families $\mathcal{G}$ can generate more vectors $\mathbf{g}_S$ and thus better correlate with random noise on average.

We come to the definition of *the* Rademacher complexity. Rademacher complexity

[4] Actually, this is indeed a problem. What will happen if the value is totally synchronous for a specific configuration that is, though statistically impossible, yet is the solution to the finite space?

Definition 8.9.1 (Rademacher complexity). *Let $\mathcal{D}$ denote the distribution according to which samples are drawn. For any integer $m \geq 1$, the* ***Rademacher complexity*** *of $\mathcal{G}$ is the expectation of the empirical Rademacher complexity over all samples of size m drawn according to $\mathcal{D}$.*

$$\mathfrak{R}_m(\mathcal{G}) = \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[\hat{\mathfrak{R}}_S(\mathcal{G})] \tag{8.31}$$

We are now ready to present our first generalization bound based on Rademacher complexity.

Theorem 8.9.1. *Let $\mathcal{G}$ be a family of functions mapping from $\mathcal{Z}$ to $[0,1]$. Then, for any $\delta > 0$, with probability at least $1-\delta$ over the draw of an i.i.d. sample S of size m, each of the following holds for all $g \in \mathcal{G}$:*

$$\mathbb{E}[g(z)] \leq \frac{1}{m}\sum_{i=1}^{m} g(z_i) + 2\mathfrak{R}_m(\mathcal{G}) + \sqrt{\frac{\log(1/\delta)}{2m}} \tag{8.32}$$

and

$$\mathbb{E}[g(z)] \leq \frac{1}{m}\sum_{i=1}^{m} g(z_i) + 2\mathfrak{R}_m(\mathcal{G}) + 3\sqrt{\frac{\log(2/\delta)}{2m}} \tag{8.33}$$

The final term in both equations is typically much smaller than the Rademacher complexity. Note that they are one-sided uniform deviation bounds, and that they are also *data-dependent* bound, just like how Gaussian process regression is also functionally data-dependent.

Proof. For any sample $S = (z_1, \ldots, z_m)$, and any $g \in \mathcal{G}$, we denote $\hat{\mathbb{E}}_S[g]$ by the empirical average of g over S. The proof then consists of applying McDiarmid's inequality to function Φ defined for any sample S by:

$$\Phi(S) = \sup_{g\in\mathcal{G}}\left(\mathbb{E}[g] - \hat{E}_S[g]\right) \tag{8.34}$$

This is the supremum of the different between the average and the empirical average for all $g \in \mathcal{G}$. If there is anything, this is similar to the following rational choice: just simply change the inequality to:

$$\begin{aligned}\mathbb{E}[g(z)] - \frac{1}{m}\sum_{i=1}^{m} g(z_i) &\leq 2\mathfrak{R}_m(\mathcal{G}) + 3\sqrt{\frac{\log(2/\delta)}{2m}} \\ \mathbb{E}[g(z)] - \hat{\mathbb{E}}_{S'}[g(z)] &\leq 3\mathfrak{R}_m(\mathcal{G}) + a\sqrt{\frac{\log(2/\delta)}{2m}}\end{aligned} \tag{8.35}$$

in which we then take the supremum. The supremum is for the pretty trivial choice: for this equality to hold true then the largest element of the LHS $\mathbb{E}[g(z)] - \hat{\mathbb{E}}_{S'}[g(z)]$ to be less than the RHS, for any δ (for equation 8.32) and $\delta/2$ (for equation 8.33). Note that the same thing is applicable.

Let S and S' be two samples differing by exactly one point, say z_m in S and z'_m in S'. Then, since the difference of suprema does not exceed the supremum of the difference, we have:

$$\Phi(S') - \Phi(S) \leq \sup_{g\in\mathcal{G}}\left(\hat{\mathbb{E}}_S[g] - \hat{\mathbb{E}}_{S'}[g]\right) = \sup_{g\in\mathcal{G}}\frac{g(z_m) - g(z'_m)}{m} \leq \frac{1}{m} \tag{8.36}$$

Similarly, we can obtain

$$\Phi(S') - \Phi(S) \leq \frac{1}{m} \implies |\Phi(S) - \Phi(S')| \leq \frac{1}{m} \tag{8.37}$$

By McDiarmid's inequality, for any $\delta > 0$, with probability at least $1-\delta/2$, the following holds.

$$\Phi(S) \leq \mathop{\mathbb{E}}_{S}[\Phi(S)] + \sqrt{\frac{log2/\delta}{2m}} \tag{8.38}$$

We next bound the expectation of the right-hand side as follows:

$$\mathbb{E}_S[\Phi(S)] = \mathbb{E}_S\left[\sup_{g\in\mathcal{G}}\left(\mathbb{E}[g] - \hat{\mathbb{E}}_S(g)\right)\right] \tag{8.39}$$

$$= \mathbb{E}_S\left[\sup_{g\in\mathcal{G}}\left(\mathbb{E}_{S'}[\hat{\mathbb{E}}_{S'}(g)] - \hat{\mathbb{E}}_S(g)\right)\right] \text{ from 8.2.1} \tag{8.40}$$

$$= \mathbb{E}_S\left[\sup_{g\in\mathcal{G}}\mathbb{E}_{S'}\left[\hat{\mathbb{E}}_{S'}(g) - \hat{\mathbb{E}}_S(g)\right]\right] \tag{8.41}$$

$$\leq \mathbb{E}_{S,S'}\left[\sup_{g\in\mathcal{G}}\left(\hat{\mathbb{E}}_{S'}(g) - \hat{\mathbb{E}}_S(g)\right)\right] \tag{8.42}$$

$$= \mathbb{E}_{S,S'}\left[\sup_{g\in\mathcal{G}}\frac{1}{m}\sum_{i=1}^{m}\left(g(z_i') - g(z_i)\right)\right] \tag{8.43}$$

$$= \mathbb{E}_{\sigma,S,S'}\left[\sup_{g\in\mathcal{G}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i\left(g(z_i') - g(z_i)\right)\right] \tag{8.44}$$

$$\leq \mathbb{E}_{\sigma,S,S'}\left[\sup_{g\in\mathcal{G}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i g(z_i')\right] + \mathbb{E}_{\sigma,S}\left[\sup_{g\in\mathcal{G}}\frac{1}{m}\sum_{i=1}^{m}-\sigma_i g(z_i)\right] \tag{8.45}$$

$$= 2\,\mathbb{E}_{\sigma,S}\left[\sup_{g\in\mathcal{G}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i g(z_i)\right] = 2\mathfrak{R}_m(\mathcal{S}). \tag{8.46}$$

We note that there are several implicit choices, especially for the properties of the subadditivity of the supremum, $\sup A + B = \sup A + \sup B$.

The reduction to $\mathfrak{R}_m(\mathcal{G})$ in equation 8.46 yields the bound in the previous equation (the first one), using δ instead of $\delta/2$. To derive a bound in terms of $\hat{\mathfrak{R}}_S(\mathcal{G})$, we observe that, by definition, changing one point in S changes $\hat{\mathfrak{R}}_S(\mathcal{G})$ by at most $1/m$. Then, using again McDiarmid inequality, with probability $1-\delta/2$, the following holds:

$$\mathfrak{R}_m(\mathcal{G}) \leq \hat{\mathfrak{R}}_S(\mathcal{G}) + \sqrt{\frac{\log 2/\delta}{2m}} \tag{8.47}$$

Finally, we use the union bound to combine inequalities together, which yields

$$\Phi(S) \leq 2\hat{\mathfrak{R}}_S(\mathcal{G}) + 3\sqrt{\frac{\log 2/\delta}{2m}} \tag{8.48}$$

which matches the inequality. □

The above inequality can be somewhat extended to be formulated, using the empirical Rademacher complexity rather than the expected Rademacher complexity.

Corollary 8.9.1. *Suppose that a sample S of size m is drawn according to distribution $\mathcal{D}$. Then for any $\delta > 0$, with probability at least $1 - \delta$, the following holds for all $g \in \mathcal{G}$:*

$$\mathbb{E}[g(z)] = L(g) \leq \underbrace{L_S(g)}_{\hat{\mathbb{E}}_S[g]} + 2\mathfrak{R}_S(\mathcal{G}) + O\left(\sqrt{\frac{\log 1/\delta}{m}}\right) \tag{8.49}$$

Proof. We may consider the empirical Rademacher complexity $R_S(G)$ as a function of the points $z_1, \ldots, z_m$ that comprise the sample S. Changing one of the z_i to a new value z_i' changes $R_S(G)$ by at most $1/m$. Applying McDiarmid's inequality with $c = 1/m$, and $\epsilon = \sqrt{\log 2/\delta/2m}$ we have that with probability at least $1 - \delta/2$,

$$R_S(G) \leq R_m(\mathcal{G}) + \sqrt{\frac{\log 2/\delta}{2m}} \tag{8.50}$$

By union bound, with probability at least $1 - e\delta$, the inequality both holds. This implies that our corollary holds for all $g \in \mathcal{G}$ with probability at least $1 - 2\delta$. Replace δ by $\delta/2$ yields the required result. □

The following result relates the empirical Rademacher complexities of a hypothesis set $\mathcal{H}$ and to the family of loss functions $\mathcal{G}$ associated to $\mathcal{H}$ in the case of binary loss (zero-one loss).

Lemma 8.9.2. *Let $\mathcal{H}$ be a family of functions taking values in $\{-1, +1\}$, and let $\mathcal{G}$ be the family of loss functions associated to $\mathcal{H}$ for the zero-one loss:*

$$\mathcal{G} = \{(x, y) \mapsto 1_{h(x) \leq y} : h \in \mathcal{H}\}$$

*For any sample $S = \{(x_i, y_i)\}_m$ of elements in $\mathcal{X} \times \{-1, 1\}$, let $S_\mathcal{X}$ be the **projection** over $\mathcal{X}$: $S_\mathcal{X} = (x_1, \ldots, x_m)$. Then, the following relation holds:*

$$\hat{\mathfrak{R}}_S(\mathcal{G}) = \frac{1}{2}\hat{\mathfrak{R}}_{S_\mathcal{X}}(\mathcal{H}) \tag{8.51}$$

Proof. For any sample $S = ((x_1, y_1), \ldots, (x_m, y_m))$ of elements in $\mathcal{X} \times \{-1, +1\}$, by definition, the empirical Rademacher complexity of $\mathcal{G}$ can be written as:

$$\begin{aligned}
\widehat{\mathfrak{R}}_S(\mathcal{G}) &= \mathbb{E}_\sigma\left[\sup_{h \in \mathcal{H}} \frac{1}{m}\sum_{i=1}^m \sigma_i \mathbb{1}_{h(x_i) \neq y_i}\right] \\
&= \mathbb{E}_\sigma\left[\sup_{h \in \mathcal{H}} \frac{1}{m}\sum_{i=1}^m \sigma_i \frac{1 - y_i h(x_i)}{2}\right] \\
&= \frac{1}{2}\mathbb{E}_\sigma\left[\sup_{h \in \mathcal{H}} \frac{1}{m}\sum_{i=1}^m -\sigma_i y_i h(x_i)\right] \\
&= \frac{1}{2}\mathbb{E}_\sigma\left[\sup_{h \in \mathcal{H}} \frac{1}{m}\sum_{i=1}^m \sigma_i h(x_i)\right] \\
&= \frac{1}{2}\widehat{\mathfrak{R}}_S(\mathcal{H}),
\end{aligned}$$

where we used the fact that $\mathbb{1}_{h(x_i) \neq y_i} = (1 - y_i h(x_i))/2$ and the fact that for a fixed $y_i \in \{-1, +1\}$, σ_i and $-\sigma_i y_i$ are distributed in the same way.

□

The lemma then implies by taking expectations, that for any $m \geq 1$,

$$\mathfrak{R}_m(\mathcal{G}) = \frac{1}{2}\mathfrak{R}_m(\mathcal{H}) \tag{8.52}$$

These connections between the empirical and average Rademacher complexities can be used to derive generalization bounds for binary classification in terms of the Rademacher complexity of the hypothesis set $\mathcal{H}$. This leads to the following result on the Rademacher complexity bounds on binary classification.

Theorem 8.9.3 (Rademacher complexity bounds - binary classification). *Let $\mathcal{H}$ be a family of functions taking values in $\{-1, +1\}$ and let $\mathcal{D}$ be the distribution over the input space $\mathcal{X}$. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over a sample S of size m drawn according to $\mathcal{D}$, each of the following holds for any $h \in \mathcal{H}$.*

$$R(h) \leq \hat{R}_S(h) + \mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\log 1/\delta}{2m}} \tag{8.53}$$

and

$$R(h) \leq \hat{R}_S(h) + \mathfrak{R}_m(\mathcal{H}) + 3\sqrt{\frac{\log 2/\delta}{2m}} \tag{8.54}$$

Proof. This follows immediately from the above theorem and lemma. □

Example 8.9.1. *Let $\Pi = \{A_1, \ldots, A_k\}$ be a fixed partition of $\mathcal{X}$, such as a regular partition or a recursive dyadic partition. Let $\mathcal{H}$ be the classifiers that are constant on cells in Π, essentially means that the label applies for any $x \in A_i$, given binary tagging of $\{-1, +1\}$. Then, $|\mathcal{H}| = 2^k$. We will obtain a bound on the empirical Rademacher complexity of $\mathcal{H}$. Let $\ell(A)$ denote the label assigned to $A \in \Pi$.*

$$\begin{aligned}
\widehat{\mathcal{R}}_S(\mathcal{H}) &= \mathbb{E}_\sigma \left[\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i h(X_i) \right] \\
&= \frac{1}{n} \mathbb{E}_\sigma \left[\sup_{h \in \mathcal{H}} \sum_{j=1}^{k} \sum_{i : X_i \in A_j} \sigma_i h(X_i) \right] \\
&= \frac{1}{n} \mathbb{E}_\sigma \left[\sum_{A \in \Pi} \sup_{\ell(A)} \sum_{i : X_i \in A} \sigma_i \ell(A) \right] \\
&= \frac{1}{n} \sum_{A \in \Pi} \mathbb{E}_\sigma \left[\sup_{\ell(A)} \sum_{i : X_i \in A} \sigma_i \ell(A) \right].
\end{aligned}$$

Manipulating the terms inside the expectation gives

$$\begin{aligned}
\mathbb{E}_\sigma \left[\sup_{\ell(A)} \sum_{i : X_i \in A} \sigma_i \ell(A) \right] &= \mathbb{E}_\sigma \left[\sup_{\ell(A)} \ell(A) \sum_{i : X_i \in A} \sigma_i \right] \\
&= \mathbb{E}_\sigma \left[\left| \sum_{i : X_i \in A} \sigma_i \right| \right] \quad (\ell(A) \in \{-1, 1\})
\end{aligned}$$

$$
\begin{aligned}
&\leq \mathbb{E}_\sigma \left[\left(\sum_{i:X_i \in A} \sigma_i \right)^2 \right]^{1/2} \\
&= \left[\mathbb{E}_\sigma \left(\sum_{i:X_i \in A} \sigma_i \right)^2 \right]^{1/2} \qquad \textit{(Jensen's inequality)} \\
&= \sqrt{\#\{i : X_i \in A\}},
\end{aligned}
$$

where the last line follows because

$$
\mathbb{E}_\sigma(\sigma_i \sigma_j) = \begin{cases} 0, & i \neq j, \\ 1, & i = j. \end{cases}
$$

If $n_j = \#\{i : X_i \in A_j\}$*, then*

$$
\begin{aligned}
\widehat{\mathcal{R}}_S(\mathcal{H}) &= \frac{1}{n} \sum_{j=1}^{k} \sqrt{n_j} \\
&= \sum_{j=1}^{k} \frac{\sqrt{\widehat{P}(A_j)}}{n}.
\end{aligned}
$$

8.9.2 Growth function

Here, we will show how the Rademacher complexity can be bounded in terms of the *growth function*.

growth function

Definition 8.9.2 (Growth function). *Given a binary class of functions* $\mathcal{H}$*, we define the growth function* $\Pi_{\mathcal{H}} : \mathbb{N} \to \mathbb{N}$ *as:*

$$
\Pi_{\mathcal{H}}(m) = \max_{\{x_1, \ldots, x_m\} \subseteq \mathcal{X}} |\{h(x_1), \ldots, h(x_m) : h \in \mathcal{H}\}| \tag{8.55}
$$

In other words, $\Pi_{\mathcal{H}}(m)$ is the maximum number of distinct ways in which m points can be classified using hypotheses in $\mathcal{H}$. This growth function is then also a measurement of the richness of a class of function, and each one of these distinct classification is called a *dichotomy*. The growth function then counts the number of dichotomies that are realized by the hypothesis. This differs from the Rademacher complexity, in the way that it is independent of the distribution of points sampled, hence is purely combinatorial. Also, the growth function is conducted for all $h \in \mathcal{H}$ of the finite hypothesis space. If it is the binary case, then there is at most 2^m categorization configuration.

To relate these two together, we use the Massart's lemma.

Lemma 8.9.4 (Massart's lemma). *Let* A *be a finite subset of* $\mathbb{R}^m$*, and let*

$$
\max_{\vec{y} \in A} ||\vec{y}||_2 \leq r
$$

Then

$$
\mathop{\mathbb{E}}_{\sigma_1, \ldots, \sigma_m} \left[\sup_{\vec{y} \in A} \sum_{i=1}^{m} \sigma_i y_i \right] \leq r \sqrt{2 \log |A|} \tag{8.56}
$$

This lemma offers not much aside from an interesting bound. Using this result, we can now bound the Rademacher complexity in terms of the growth function.

Corollary 8.9.2. *Let $\mathcal{G}$ be a family of functions taking values in $\{-1, +1\}$. Then the following holds:*

$$\mathfrak{R}_m(\mathcal{H}) \leq \sqrt{\frac{2 \log \Pi_{\mathcal{H}}(m)}{m}} \tag{8.57}$$

Proof. For a fixed sample S, we denote by $\mathcal{G}_{|S}$ the set of vectors of function values $(g(x_1), \ldots, g(x_m))^\top$ for $g \in \mathcal{G}$. Since $g \in \mathcal{G}$, $|g' \in \mathcal{G}_{|S}| \leq \sqrt{m}$. We can then apply Massart's lemma as:

$$\begin{aligned} \mathfrak{R}_m(\mathcal{G}) =& \mathop{\mathbb{E}}_{S}\left[\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{u \in \mathcal{G}_{|S}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i u_i\right]\right] \\ &\leq \mathop{\mathbb{E}}_{S}\left[\frac{\sqrt{m}\sqrt{2\log|\mathcal{G}_{|S}|}}{m}\right] \end{aligned} \tag{8.58}$$

By definition, $|\mathcal{G}_{|S}|$ is bounded by the growth function. Thus,

$$\begin{aligned} \mathfrak{R}_m(\mathcal{G}) &\leq \mathop{\mathbb{E}}_{S}\left[\frac{\sqrt{m}\sqrt{2\log \Pi_{\mathcal{G}}(m)}}{m}\right] \\ &= \sqrt{\frac{2\log \Pi_{\mathcal{G}}(m)}{m}} \end{aligned} \tag{8.59}$$

which concludes the proof. □

From this and the generalization bound, we will also yield the following generalization bound in terms of the growth function.

Corollary 8.9.3 (Growth function generalization bound). *Let $\mathcal{H}$ be a family of function taking values in $\{-1, +1\}$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, for any $h \in \mathcal{H}$:*

$$R(h) \leq \hat{R}_S(h) + \sqrt{\frac{2\log \Pi_{\mathcal{H}}(m)}{m}} + \sqrt{\frac{\log(1/\delta)}{2m}} \tag{8.60}$$

This growth function bound can also be derived directly. The computation of the growth function is also, pretty weird, and is often not always convenient, especially because of its combinatorial nature.

8.10 Vapnik-Chervonenkis Theory

VC, or Vapnik–Chervonenkis theory, base a lot of its results and theories on both the growth function, and the combinatorial nature of which it takes form. The problem of the formation of VC theory is simple. We have seen that a consistent learner can be used to design a PAC-learning algorithm, provided the output hypothesis comes from a class that is not too large, in particular, of any polynomial class $poly(\cdot, \cdot, \cdot, \cdot)$. However, one certain assumption and condition on the PAC-learning algorithm measure is that it only works on (the result, that is) finite hypothesis class or concept class. Concept classes that are uncountably infinite are often used, for example, linear halfspaces, or more familiar, the hypothesis on such space is the **linear classifier** or **perceptron**. In this section, we would particularly see how we can use a new capacity measure

called VC dimension of a concept class to make a consistent learner to be PAC-learnable. All theories and practices using the VC-dimension as one of its main ingredient is then called the VC theory.

8.10.1 Clarification of the linear halfspaces

halfspace

We would like to clarify about such statement about linear halfspaces being infinite hypothesis class. The **halfspace** hypothesis space is the set of hypotheses that consist of a hyperplane in a d-dimensional coordinate space that classifies a feature vector $\phi(x) \in \mathbb{R}^{d+1}$ as either $-1, 1$ based on which side the of the hyperplane it lies. Here, d represents the number of features on item x.

Mathematically, to define the halfspace hypothesis space, consider the domain $\mathcal{X} = \mathbb{R}$ and concept set $\mathcal{Y} = \{-1, 1\}$. The class $\mathcal{H}_{gl}$ of general linear classifiers, or linear halfspaces, is defined as

$$\mathcal{H}_{gl} = \{h_{\boldsymbol{w},b} \mid \boldsymbol{w} \in \mathbb{R}^d, b \in \mathbb{R}\} \tag{8.61}$$

where

$$h_{\mathbf{W},b}(\boldsymbol{x}) = \operatorname{sgn}(\langle \boldsymbol{x} \cdot \boldsymbol{w} \rangle + b) = \operatorname{sgn}\left(\left(\sum_{i=1}^{d} x_i w_i\right) + b\right) \tag{8.62}$$

General linear halfspaces in $\mathbb{R}^d$ can be viewed as *homogenous linear halfspaces* in $\mathbb{R}^{d+1}$ via a simple transformation. We define the class of homogenous linear halfspaces $\mathcal{H}_l$ as

$$\mathcal{H}_l = \{h_{\boldsymbol{w}} \mid \boldsymbol{w} \in \mathbb{R}^d, b \in \mathbb{R}\}$$

where

$$h_{\mathbf{W}}(\boldsymbol{x}) = \operatorname{sgn}(\langle \boldsymbol{x} \cdot \boldsymbol{w} \rangle) = \operatorname{sgn}\left(\sum_{i=1}^{d} x_i w_i\right) \tag{8.63}$$

Let $\boldsymbol{x} = h_{\boldsymbol{w},b}(\boldsymbol{x}) \in \mathbb{R}^d$, for $\boldsymbol{w} = (w_1, w_2, \ldots, w_d) \in \mathbb{R}^d$ and $b \in \mathbb{R}$ be given, we then define

$$\boldsymbol{x}' = (1, x_1, \ldots, x_d) \in \mathbb{R}^{d+1} \tag{8.64}$$

$$\boldsymbol{w}' = (1, w_1, \ldots, w_d) \in \mathbb{R}^{d+1} \tag{8.65}$$

Then we get $\langle \boldsymbol{x} \cdot \boldsymbol{w} \rangle + b = \langle \boldsymbol{x}' \cdot \boldsymbol{w}' \rangle$ for all $x \in \mathbb{R}$, and thus

$$h_{\boldsymbol{w},b}(\boldsymbol{x}) = h_{\boldsymbol{w}'}(\boldsymbol{x}')$$

In what sense could linear halfspaces be of an infinite hypothesis class? We say a hypothesis class, or concept class, is *infinite*, when it does not have an enumerable and finite list of hypothesis. It can include an infinite number of variations of the description of the hypotheses. This, partially makes it impossible to be listed in the sense of a list, hence we would usually specify it using their variational expression, for example, linear equation and, linear halfspace. In such case, arguably, one of the previous illustrative example of a class of *axis-aligned rectangle* is also an infinite hypothesis class, because it represents infinitely many rectangular concepts on $\mathbb{R}^2$. Though, what should you do if one is to justify that the infinity on $\mathbb{R}^3$ is bigger than the hypothesis class on $\mathbb{R}^2$? Probably it will work.

It is also wised to separate the notion between halfspace, hyperplane in mathematics and machine learning lingua franca. Specifically, in mathematics, it is recalled that a **hyperplane** is a $d-1$-dimensional space that 'carve out' regions of a d-dimensional space - just as plane in $\mathbb{R}^3$ or lines in $\mathbb{R}$[2]. While, the mathematical **halfspace** is a d-dimensional partition of regions that

is formed by removing that part lying on one side of an $(d-1)$-dimensional hyperplane — thus making it somewhat of a product of a hyperplane. For example, half a Euclidean space is given by the three-dimensional region satisfying $x > 0, -\infty < y < \infty, -\infty < z < \infty$. Thereby, the dimension topic would likely want to be of rest with this comparison.

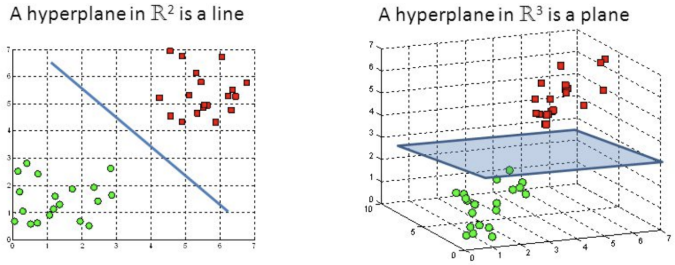


Figure 8.5: Illustration of the notion of hyperplane in two and three dimensions. This can be extended to $n > 3$ dimension, but no figurative illustration can be found (or ever understood). Taken from Introduction to Statistical Learning using R Book Club by The R4DS Online Learning Community.

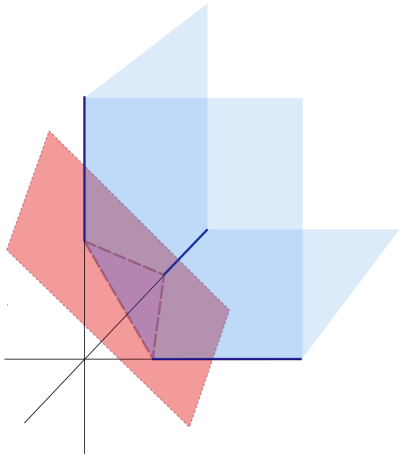


Figure 8.6: Illustration of a halfspace **region** created by a hyperplane on the side of the axis. If the halfspace is created of the unit frame hyperplane (aligning with the axis), then it is called a *normal space partitioning halfspace*.

8.10.2 VC-dimension

Then, we are ready to somewhat consider the notion of *VC-dimension*. The VC-dimension is often regarded as a purely combinatorial notion, but it is often easier to compute than the growth function, or the Rademacher Complexity. As we shall see, the VC-dimension is a key quantity in learning, and is directly related to the growth function.

To define the VC-dimension, we will have to define the notion of *shattering* . As we recalled of the previous sections on the growth function, on the hypothesis set $\mathcal{H}$, a dichotomy on a set S is one of many possible ways of labelling the points in S using a hypothesis $\mathcal{H}$. A set S of $m \geq 1$ points is said to be shattered by a hypothesis set $\mathcal{H}$ when $\mathcal{H}$ realizes all possible dichotomies of S, that is when $\Pi_{\mathcal{H}}(m) = 2^m$. shattering

Definition 8.10.1 (Shattering). *We say that a finite set $S \subset \mathcal{X}$ is **shattered** by C, if $|\Pi_C(S)| = 2^{|S|}$. S is shattered by C if all possible dichotomies over S can be realized by C.*

We know can define the notion of the *dimension* for a concept class $\mathcal{C}$.

Definition 8.10.2 (VC-dimension). *The VC-dimension of a hypothesis set $\mathcal{H}$ is the size of the largest set that can be shattered by $\mathcal{H}$,*

$$\mathrm{VCdim}(\mathcal{H}) = \max_{m \in S}\{m : \Pi_{\mathcal{H}}(m) = 2^m\} \tag{8.66}$$

Note that, by definition, if $\mathrm{VCdim}(\mathcal{H}) = d$, then there exists a set of size d that can be shattered. However, it does not mean that all sets of size d or less are shattered, because this only get you the existential criteria.

Again, if we are to look at the definition of VC-dimension, we see the similar problem: it is very computationally heavy for experimental purposes.

To further illustrate this notion, we will examine a series of examples of hypothesis sets and will determine the VC-dimension in each case. To compute the VC-dimension, we will typically show a lower bound for its value and then a matching upper bound.

Example 8.10.1 (Intervals on the real line). *Our first example involves the hypothesis class of intervals on real line. Because this is an inclusion criterion, then we are talking about if a point is in the interval or not. Then, by such. the dichotomy $(\cdot, \cdot)$ can be understood as 'does the first point and the second point lie in the interval?'. This question is an 'or' question, meaning there exists dichotomies such as $(-, +)$ where the first point is not in the interval specified.*

The VC-dimension is then at least two, since there exists four total dichotomies

$$\Pi_{\mathcal{H}} = (+,+), (-,-), (+,-), (-,+)$$

that can be realized. In contrast, by the definition of intervals, no set of three points can be shattered since the $(+,-,+)$ labelling cannot be realized. Hence, we say that the VC-dimension is 2.

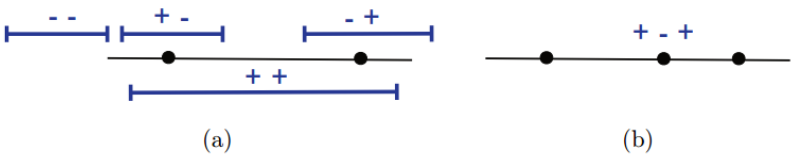


Figure 8.7: VC-dimension of intervals on the real line. (a) Any two points can be shattered. (b) No sample of three points can be shattered as the $(+,-,+)$ labelling cannot be realized. Taken from Mohri et al. [2012].

Example 8.10.2 (Hyperplane). *Consider the set of all hyperplanes in $\mathbb{R}^2$. We observe that any three non-collinear (collinear points mean they are on a straight line) points in $\mathbb{R}^2$ can be shattered. Let us remind ourselves of the definition of a hyperplane. A **hyperplane** is a generalization on high-dimension for lines and planes. Given $\mathbf{w}$ the vector normal to the hyperplane, and b is the offset, then the hyperplane equation is $\mathbf{w} \cdot \mathbf{x} + b = 0$. This equality satisfies the scalar multiplication rule, and multiply by $1/||\mathbf{w}||$, we get the unit normal vector and for $b' = b/||\mathbf{w}||$ the **origin distance** from the hyperplane to the origin.*

Given three points x, y, z, to obtain the first three dichotomies, we choose a hyperplane that has two points on one side, and the third point on the other side, that is, all the dichotomies of the type

$$(x \mid y, z), (y \mid x, z), (z \mid x, y), \quad x, y, z \in \mathbb{R}^2 \tag{8.67}$$

Notice that the ordering does not matter. To obtain the fourth dichotomy, we have all three points on the same side. The remaining four dichotomies are realized by simply switching signs, that is, the hyperplane by default assumes two side classification of $(+, -)$ configuration for either side, so there would then be four more dichotomies simply by switching signs. So in total there are

$$(x \mid y, z), (y \mid x, z), (z \mid x, y), (0 \mid x, y, z) \tag{8.68}$$
$$(y, z \mid x), (x, z \mid y), (x, y \mid z), (x, y, z \mid 0) \tag{8.69}$$

configurations. Next, we show that four points cannot be shattered by considering two cases:

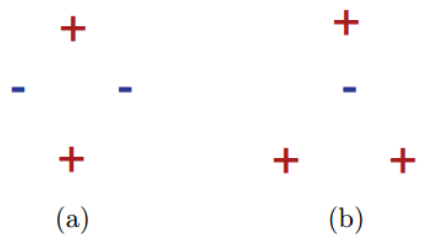


Figure 8.8: Unrealizable dichotomies for four points using hyperplanes in $\mathbb{R}^2$. (a) All four points lie on the convex hull. (b) Three points lie on the convex hull while the remaining point is interior. Taken from Mohri et al. [2012].

(1) The four points lie on the convex hull defined by the four points. [5]

(2) Three of the four points lie on the convex hull and the remaining point is internal.

In the first case, a positive labelling for one diagonal pair and a negative labelling for the other diagonal pair cannot be realized. In the second case, a labelling which is positive for the points on the convex hull and negative for the interior point cannot be realized. Hence, we have $\mathrm{VCdim}(\mathbf{h}) = 3$.

More generally, in our example of the hyperplane, in $\mathbb{R}^d$ we can derive a lower bound by starting with a set of $d + 1$ points in $\mathbb{R}^d$, setting x_0 to be the origin and defining $\times_i$ for $i \in \{1, \ldots, d\}$, as the point whose ith coordinate is 1 and all others are 0. Let $y_0, y_1, \ldots, y_d \in \{-1, +1\}$ be an arbitrary set of labels for $\mathbf{x}_0, \ldots, \mathbf{x}_d$. Let $\mathbf{w}$ be the vector whose ith coordinate is y_i. Then we can say that classifier defined by the hyperplane $\mathbf{w} \cdot \mathbf{x} + y_0/2 = 0$ shatters $\mathbf{x}_0, \ldots, \mathbf{x}_d$ since for any $i \in \{0, \ldots, d\}$, we have:

$$\mathrm{sgn}\left(\mathbf{w} \cdot \mathbf{x}_i + \frac{y_0}{2}\right) = \mathrm{sgn}\left(y_i + \frac{y_0}{2}\right) = y_i \tag{8.70}$$

To obtain an upper bound, it suffices to show that no set of $d + 2$ points can be shattered by halfspaces. To prove this, we will use the following general theorem, called *Radon's theorem*,

[5] **A convex hull** is simply a polytope that is convex, that is, for a set $P \subseteq \mathbb{R}^d$ is convex if $|pq \subseteq P$ for any $p, q \in P$, the **convex hull** $\mathrm{conv}(P)$ of a set P is the intersection of all convex supersets of P.

taken from combinatorial analysis. Before attempting this, we first have to say a few preliminaries on the theory of **combinatorial convexity**. This will rather be very long, but worth it.

For two points $a, b \in \mathbb{R}^d$ of the d-dimensional Euclidean space, we define a *segment* $[a, b]$ joining a and b as the set

$$[a, b] = \{\alpha a + \beta b, \alpha, \beta \geq 0, \alpha + \beta = 1\} \tag{8.71}$$

A set $C \in \mathbb{R}^d$ is called *convex* if for every two points a and b in C, the segment $[a, b]$ is also contained in C. Plainly, an intersection of convex sets is a convex set. In other word, a set of points is convex if it contains every line segment between two points in the set.

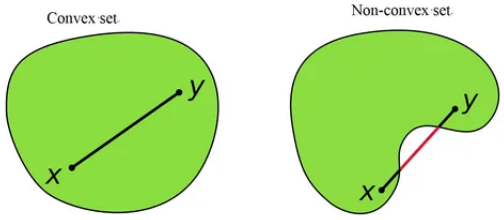


Figure 8.9: Illustration of convex and non-convex set. The segment $[x, y]$ must be fully contained in the region of the set, otherwise it is not convex.

Given n points $a_1, \ldots, a_n \in \mathbb{R}^d$ and real coefficients $\alpha_1, \ldots, \alpha_n$, the point

$$a = \alpha_1 a_1 + \alpha_2 a_2 + \cdots + \alpha_n a_n \tag{8.72}$$

is called their *convex combination* (c.c. for short) if the α_i are nonnegative, and they add up to one, $\alpha_1 + \ldots \alpha_n = 1$. We have the following observation:

Lemma 8.10.1. *If C is a convex subset of $\mathbb{R}^d$ and $a_1, \ldots, a_n \in C$, then all convex combinations of $a_1, \ldots, a_n$ belongs to C.*

For a subset S in $\mathbb{R}^d$, we can then define the following as a **convex hull**:

Definition 8.10.3 (Convex hull). *Given a set A, the convex hull of a set A, denoted by $\mathrm{conv}(A)$, is the set of all convex combination of points in A. That is:*

$$\mathrm{conv}(A) = \left\{ x \mid \exists a_1, \ldots, a_N \in A, \alpha_1, \ldots, \alpha_N \geq 0 \,;\, \sum_{i=1}^{N} \alpha_i = 1 \,;\, x = \sum_{i=1}^{N} \alpha_i a_i \right\} \tag{8.73}$$

In other words,

$$\mathrm{conv}(S) = \{\textit{all convex combinations of elements of } S\} \tag{8.74}$$

The convex hull is then the smallest convex set containing S.

Obviously, a segment in $\mathbb{R}^d$ is a convex set. Another canonical example of a convex set is a (closed) half-space, $\mathcal{H} = \{x \in \mathbb{R}^d, a \cdot x \geq \alpha\}$. The set of all positive semi-definite matrices of a fixed size is also convex.

Since we are dealing with the dichotomy of separating into half-space of binary classification, the following lemma will help to establish our requirement of Radon's theorem.

Lemma 8.10.2. *Let $P = \{p_1, \ldots, p_n\} \subset \mathbb{R}^d$ be a finite set of points in d-dimensional space. If $n > d$ then we have, for some coefficients $\mu_1, \ldots, \mu_n$,*

$$\mathbf{0} = \sum_{i=1}^{n} \mu_i p_i, \quad \forall(\mu_1, \ldots, \mu_n)\exists \mu_j \neq 0 \tag{8.75}$$

and if $n > d + 1$ we can have equation 8.75 under the addition condition that:

$$\sum_{i=1}^{n} \mu_i = 0 \tag{8.76}$$

In the latter case, some of $\mu_1, \ldots, \mu_n$ are positive and some are negative.

Proof. The first part of the lemma follows from the fact that every set of $d+1$ or more points in a d-dimensional vector space is linearly dependent. The second part follows from the observation that $\{p_2 - p_1, p_3 - p_1, \ldots, p_n - p_1\}$ (since it is exactly $d-1$ in a d-dimensional space) is linearly dependent, thus

$$\mathbf{0} = \sum_{i=2}^{n} \mu_i (p_i - p_1), \quad \forall(\mu_1, \ldots, \mu_n)\exists \mu_j \neq 0 \tag{8.77}$$

By defining $\mu_1 = -\sum_{i=2}^{n} \mu_i$, both equation 8.75 and 8.76 holds. The only way then that equation 8.76 can hold with all non-positive or non-negative terms would be if all terms are zero. □

In a combinatorial way, we will then prove the following theorem called Caratheodory theorem:

Theorem 8.10.3 (Caratheodory). *Let $\mathcal{A}$ be a subset in $\mathbb{R}^d$. Suppose that $a \in \text{conv}(A)$. Then there exists a subset B of $\mathcal{A}$ with $|B| \leq d + 1$ such that $a \in \text{conv}(B)$.*

Proof. For a vector x in $\mathbb{R}^d$ and a real number α by $\binom{x}{\alpha}$ we mean a vector in $\mathbb{R}^{d+1}$ with the last coordinate α. The fact that a lies in the convex hull of $\mathcal{A}$ can be written shortly as:

$$\binom{a}{1} = \sum_{i=1}^{n} \alpha_i \binom{a_i}{1} \tag{8.78}$$

for some nonnegative real α_i's (the last coordinate takes care of condition that α_1's add up to 1). Without loss of generality we can assume that all α_i's are positive. Moreover, let n be the smallest possible for which the above holds. We want to show that $n \leq d + 1$. Suppose not, then the vectors $\binom{a_i}{1}$, for $i = 1, \ldots, n$ cannot be linearly independent as they lie in a $d + 1$ dimensional space. Therefore,

$$\binom{0}{0} = \sum_{i=1}^{n} \beta_i \binom{a_i}{1} \tag{8.79}$$

for some β_i's which are not all equal to zero. Hence,

$$\binom{a}{1} = \sum_{i=1}^{n} (\alpha_i + t\beta_i) \binom{a_i}{1} \tag{8.80}$$

for every $t \in \mathbb{R}$. Since for $t = 0$ all the coefficients $\alpha_i + t\beta_i$'s are positive, they remain positive for small t and there is a choice for t, say t_0 for which at least one of the coefficients becomes 0 with the rest remaining positive. This contradicts the minimality of n, as

$$\begin{pmatrix} a \\ 1 \end{pmatrix} = \sum_{i=1}^{n} (\alpha_i + t_0\beta_i) \begin{pmatrix} a_i \\ 1 \end{pmatrix} \tag{8.81}$$

shows that the vector $\binom{a}{1}$ can be written as a positive combination of $\binom{a_i}{1}$ with fewer than n nonzero coefficients. □

Another way to prove this is as followed.

Caratheodory, second version. Let x be a point of $\mathrm{conv}(P)$, so that for some positive integer n,

$$x = \sum_{i=1}^{n} \alpha_i p_i \tag{8.82}$$

with for all $i \in \{1, \ldots, n\}$, $p_i \in P, \alpha_i \geq 0$ and $\sum_{i=1}^{n} \alpha_i = 1$. If $n \leq d + 1$ there is nothing to prove. Otherwise, $n > d + 1$, so by Lemma 8.10.2, we have for scalars $\mu_1, \ldots, \mu_n$ that $\mathbf{0} = \sum_{i=1}^{n} \mu_i p_i$ with $\sum_{i=1} \mu_i = 0$. Now, for any real number λ, we have:

$$x = \sum_{i=1}^{n} \alpha_i p_i - \lambda \sum^{n} \mu_i p_i = \sum_{i=1}^{n} (\alpha_i - \lambda\mu_i) p_i \tag{8.83}$$

We note that

$$\sum_{i=1}^{n} (\alpha_i - \lambda\mu_i) = \sum_{i=1}^{n} \alpha_i - \lambda \sum_{i=1}^{n} \mu_i = \sum_{i=1}^{n} \alpha - \lambda \cdot 0 = 1 \tag{8.84}$$

That is, the coefficients in the linear combination sums to one. We will now select λ so that one of these coefficients becomes zero, while the remaining coefficients are positive, making the sum a convex combination of $n - 1$ points of P.

Let $J = \{j \in \{1, \ldots, n\} : \mu_i > 0\}$, with J non-empty. Choose $j^* \in J$ so that α_{j^*}/μ_{j^*} for all $j \in J$, and let $\lambda = \alpha_{j^*}/\mu_{j^*}$. With this, we have:

$$\alpha_i - \lambda\mu_i \geq 0 \tag{8.85}$$

for all $i \in \{1, \ldots, n\}$. Indeed, if $i \in J$, then $\mu_i > 0$ and

$$\alpha_i - \lambda\mu_i = \mu_i(\alpha_i/\mu_i - \lambda) \geq 0, \tag{8.86}$$

while if $i \notin J$ then $\mu_i \leq 0$, and since $\lambda \geq 0$, we have:

$$\alpha_i - \lambda\mu_i \geq \alpha_i \geq 0 \tag{8.87}$$

Finally, observe that $a_{j^*} - \lambda\mu_{j^*} = 0$, we have:

$$x = \sum_{i=1}^{j^*-1} (\alpha_i - \lambda\mu_i) p_i + \sum_{i=j^*+1}^{n} (\alpha_i - \lambda\mu_i) p_i \tag{8.88}$$

which expresses x as a convex combination of the $n-1$ elements of the set $\{p_1, \ldots, p_n\} \setminus \{p_{j^*}\}$. This process can be repeated as long as $n > d+1$ until x is represented as a convex combination of $d + 1$ elements of P. □

Geometrically, Caratheodory's theorem says that a convex set A in $\mathbb{R}^d$ can be covered by simplices of A, that is, a simplex in $\mathbb{R}^d$ is a convex hull of at most $d+1$ points. Radon's theorem then says that there are some good partition of sets having enough points.

Let's take an example. Suppose when $d = 1$. Then the theorem is clear as considering three points on a line. There is always one, say x between some two others, say, y, z, so it suffices to take $X = \{x\}$ and $Y = A \setminus \{X\} \supset \{y, z\}$, of which they will intersect at x.

Theorem 8.10.4 (Radon's theorem). *Any set of $\mathcal{X}$ of $d+2$ points in $\mathbb{R}^d$ can be partitioned into two subsets $\mathcal{X}_1$ and $\mathcal{X}_2$ such that their convex hulls is non-zero and intact. That is, there exists a partition, $A = X \cup Y$ for $X \cap Y = \varnothing$, such that $\operatorname{conv}(X) \cap \operatorname{conv}(Y) = \varnothing$.*

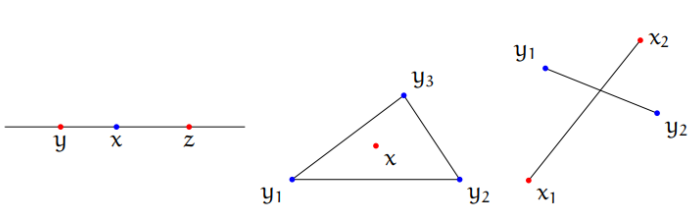


Figure 8.10: Illustration of (left-hand side) $d = 1$ Radon partition, and (middle and right-hand side) $d = 2$ Radon partition. More options are available as d increases.

When $d = 2$, considering 4 points in the plane, there are two possibilities. Either certain three of them are the vertices of a triangle containing the fourth point, or the points are the vertices of a convex quadrilateral. In any case, it is clear what to take for the partition. Notice that if we scale it down to $|\mathcal{A}| = d + 1$, however, then for a set of three points, any partition would allow the other point to be missed (one line and one singular point not passing through the line). The only convex hull that would make them to be in the same one is the full convex hull for three point.

We present two proofs of Radon's theorem. One for combinatorial geometry, and one for general analytical methods.

Radon's theorem, proof one. Since $|A| \geq d + 2$, the set $\{\binom{a}{1}, a \in \mathcal{A}\}$ of vectors in $\mathbb{R}^{d+1}$ is not linearly independent. Therefore there are $a_i \in \mathcal{A}$ and nonzero coefficients α_i such that

$$\binom{0}{0} = \sum_i \alpha_i \binom{a_i}{1} \tag{8.89}$$

Because the sum of α_i's is 0, some of them are positive, some are negative. Let I be the set of all the indices i for which $\alpha_i > 0$ and J for which $\alpha_i < 0$ (for non-empty I, J). Breaking the sum into two pieces yields:

$$\sum_{i \in I} \alpha_i \binom{a_i}{1} = \sum_{i \in J} (-\alpha_i) \binom{a_i}{1} \tag{8.90}$$

Dividing this through $t = \sum_{i \in I} = \sum_{i \in J} (-\alpha_i)$, which is positive, shows that we can take $X = \{a_i, i \in I\}$ and $Y = A \setminus X$, for then

$$\operatorname{conv}(X) \ni \sum_{i \in I} \frac{1}{t} \alpha_i \binom{a_i}{1} = \sum_{i \in J} \frac{1}{t} (-\alpha_i) \in \operatorname{conv}(Y) \tag{8.91}$$

which concludes the proof. □

Radon's theorem, proof two. Let $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_{d+2}\} \subset \mathbb{R}^d$. The following is a system of $d+1$ linear equations in $\alpha_1, \dots, \alpha_{d+2}$:

$$\sum_{i=1}^{d+2} \alpha_1 \mathbf{x}_i = 0, \quad \sum_{i=1}^{d+2} \alpha_i = 0 \tag{8.92}$$

since the first equality leads to d equations, one for each component. The number of unknowns, $d+2$ is larger than the number of equations $d+1$, therefore the system admits a non-zero solution $\beta_1, \dots, \beta_{d+2}$. Since $\sum_{i=1}^{d+2} \beta_i = 0$, both

$$\mathcal{J}_1 = \{i \in [d+2] : \beta_i > 0\}, \quad \mathcal{J}_2 = \{i \in [d+2] : \beta_i \geq 0\} \tag{8.93}$$

are non-empty sets and $\mathcal{X}_1 = \{\mathbf{x}_i : i \in \mathcal{J}_1\}$ and $\mathcal{X}_2 = \{\mathbf{x}_i : i \in \mathcal{J}_2\}$ form a partition of $\mathcal{X}$. By the last equation,

$$\sum_{i \in \mathcal{J}_1} \beta_i = -\sum_{i \in \mathcal{J}_2} \beta_i \tag{8.94}$$

Then, 8.92 implies that:

$$\sum_{i \in \mathcal{J}_1} \frac{\beta_i}{\beta} \mathbf{x}_i = \sum_{i \in \mathcal{J}_2} -\frac{\beta_i}{\beta} \mathbf{x}_i, \quad \beta = \sum_{i \in \mathcal{J}_1} \beta_i \tag{8.95}$$

with $\sum_{i \in \mathcal{J}_1} \frac{\beta_i}{\beta} = \sum_{i \in \mathcal{J}_2} -\frac{\beta_i}{\beta} = 1$, $\beta_i/\beta \geq 0$ for $i \in \mathcal{J}_1$, and $-\beta_i/\beta \geq 0$ for $i \in \mathcal{J}_2$. By the definition of convex hull, this implies that $\sum_{i \in \mathcal{J}_1} \frac{\beta_i}{\beta} \mathbf{x}_i$ belongs both to the convex hull of $\mathcal{X}_1$ and $\mathcal{X}_2$. □

Now, let us return to our problem of halfspace partitioning. Let $\mathcal{X}$ be a set of $d+2$ points. By Radon's theorem, it can be partitioned into two sets $\mathcal{X}_1$ and $\mathcal{X}_2$ such that their convex hulls intersect. Observe that when two sets of points $\mathcal{X}_1, \mathcal{X}_2$ are separated by a hyperplane, their convex hulls are also separated by that hyperplane. Thus, $\mathcal{X}_1$ and $\mathcal{X}_2$ cannot be separated by a hyperplane and $\mathcal{X}$ is not shattered. Combining our lower and upper bounds, we have proven that $\mathrm{VCdim}(\mathcal{H} \in \mathbb{R}^d)$ is $d+1$.

The VC-dimension of many other hypothesis sets can be determined or upper-bounded in a similar way. In particular, the VC-dimension of any vector space of dimension $r < \infty$ can be shown to be at most r. The next result, known as *Sauer's lemma*, clarifies the connection between the notions of growth function and VC-dimension.

Theorem 8.10.5 (Sauer's lemma). *Let $\mathcal{H}$ be a hypothesis set with $\mathrm{VCdim}(\mathcal{H}) = d$. Then, for all $m \in \mathbb{N}$, the following inequality holds:*

$$\Pi_{\mathcal{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i} \tag{8.96}$$

Proof. The proof is by induction on $m+d$. The statement clearly holds for $m = 1$ and $d = 0, 1$. Assume that it holds for $(m-1, d-1)$ and $(m-1, d)$. Fix a set $\mathcal{S} = \{x_1, \dots, x_m\}$ with $\Pi_{\mathcal{H}}(m)$ dichotomies and let $\mathcal{G} = \mathcal{H}_{|\mathcal{S}}$ be the set of concept $\mathcal{H}$ induced by restriction to $\mathcal{S}$.

Now consider the following families over $\mathcal{S}' = \{x_1, \dots, x_{m-1}\}$. We define $\mathcal{G}_1 = \mathcal{G}_{|\mathcal{S}'}$ as the set of concepts $\mathcal{H}$ induced by restriction to $\mathcal{S}'$. Next, by identifying each concept as the set of points (in $\mathcal{S}'$ or $\mathcal{S}$) for which it is non-zero, we can define $\mathcal{G}_2$ as

$$\mathcal{G}_2 = \{g' \subseteq \mathcal{S}' : (g' \in \mathcal{G}) \wedge (g' \cup \{x_m\} \in \mathcal{G})\} \tag{8.97}$$

Since $g' \subseteq \mathcal{S}'$, $g' \in \mathcal{G}$ means that without adding x_m it is a concept of $\mathcal{G}$. Further, the constraint $g' \cup \{x_m\} \in \mathcal{S}$ means that adding x_m to g' also makes it a concept of $\mathcal{G}$. The construction of $\mathcal{G}_1$ and $\mathcal{G}_2$ can then be observed that $|\mathcal{G}_1| + |\mathcal{G}_2| = |\mathcal{G}|$.

Since $\mathrm{VCdim}(\mathcal{G}_1) \leq \mathrm{VCdim}(\mathcal{G}) \leq d$, then by definition of the growth function and using the induction hypothesis,

$$|\mathcal{G}_1 \leq \Pi_{\mathcal{G}_1}(m-1) \leq \sum_{i=0}^{d} \binom{m-1}{i} \tag{8.98}$$

Further, by definition of $\mathcal{G}_2$, if a set $\mathcal{Z} \in \mathcal{S}'$ is shattered by $\mathcal{G}_2$, then the set $\mathcal{Z} \cup \{x_m\}$ is shattered by $\mathcal{G}$. Hence,

$$\mathrm{VCdim}(\mathcal{G}_2) \leq \mathrm{VCdim}(\mathcal{G}) - 1 = d - 1 \tag{8.99}$$

and by definition of the growth function and using the induction hypothesis,

$$|\mathcal{G}_2| \leq \Pi_{\mathcal{S}_2}(m-1) \leq \sum_{i=0}^{d-1} \binom{m-1}{i} \tag{8.100}$$

Thus,

$$|\mathcal{G}| = |\mathcal{G}_1| + |\mathcal{G}_2| \leq \sum_{i=0}^{d} \binom{m-1}{i} + \sum_{i=0}^{d-1} \binom{m-1}{i} = \sum_{i=0}^{d} \binom{m-1}{i-1} = \sum_{i=0}^{d} \binom{m}{i} \tag{8.101}$$

which concludes the inductive proof. □

Chapter 9. Introduction to classical connectionism

Previous sections have been devoted to the writing and formulation of the naive, classical structure of models at best, and a very general question of the theory of actions (specifically, learning) on a specific hypothesis h of a hypothesis set $\mathcal{H}$. One may then forward to ask about the intrinsic representation of the hypothesis class $\mathcal{H}$ for what it will actually be, similar to how we treated the representation scheme $\boldsymbol{\mathcal{R}}(\mathcal{C})$ of the concept class $\mathcal{C}$. One of the fundamental constructs that we can get aside from classical machine learning, in which the hypothesis class is arbitrary since there exists no general class, is the idea of a neural network architectural hypothesis class, or simply the **neural network** architecture.

As it sounds, it is taken from the construction of the brain. Literature of this follows Zhang et al. [2023], Demuth et al. [2014] or any other literature sources that discuss the essence of the topic on itself. Do note that most of the time, they are more focused on the applicational side of it, often forgo the more strict, theoretical setting and general preset of the system itself, in a rather rough way. Also, personally, there exists no biological inspiration obligatory section either.

9.1 Biological inspiration

Because it is a study of intelligence, we often find ourselves converging to the closet intelligent lifeform close by - which is us ourselves, and other species with developed brain region. For neuroscience, it has been widely received that the brain is constructed from many components, a lot of which comes around and connected together. McCulloch and Pitts [1943], taking on this, devised the idea of meticulously replicate such operation of the brain, by defining the first ever logical neuron structure, called now the McCulloch-Pitts neuron. We would be giving a quick introduction to the brain itself, and some more important notes on how the neuron formalism is formed. For prerequisite, a bit about biology is required. Textbooks and resources that dive deeper into this problem is either Kandel et al. [2021] or Purves et al. [2004], which will provide much more detail on the neuroscientifical development of the research on the brain. For now, let us see the structure of the brain by virtue of examplifying its representatives.

Informally, the brain encased the 'brain' - the nervous system in which defines its operation. This includes the central nervous system (CNS), and the peripheral nervous system (PNS). This is generally the conventional separation of the nervous system, as CNS includes the brain and spinal cord, while the peripheral nervous system consists of everything else. The CNS's responsibilities include receiving, processing, and responding to sensory information, while the peripheral, as its name, is similar to *control relay* and sensory influences.

The brain is divided into two *hemispheres* (The reason is unknown for now, in terms of operational and evolutional accord), mainly for regional specialization. Between the two central hemispheres, they are connected by nerve bundles, in this case, is the thick band of fibers known as **corpus callosum**, consisting of about 200 million axons. The **axons** or **nerve fiber** is the long,

slender projection of a nerve cell, or neuron, to different neurons and areas. So, think of it like a more extension cables from the transformer and generator.

The **direction** between the 2 hemispherical connection is unknown, and can be either one-way, or two-way. But generally, we might want to take it as two-way, since it makes sense for when simultaneous tasks which requires multiple system on both sides to operates, remains so. Or rather, we can take it as the idea of **neural vacancy path**, that is, empty pathway that is one-directional specific in usage cases. More so like a conditional diode, depends on which way it was triggered first. But rather, it helps us to classify between the **communication directive** subjects, and **processing directive** subject of the brain.

neural vacancy path

Note 9.1.1 (A note on the direction flow of nerve bundles). *In the brain, a nerve bundle connects two regions and allows signals to travel between them. These connections can be one-way, where signals only travel in a single direction, or two-way, which allows communication both ways. Scientists discovered this a long time ago by dissecting the preserved brains of humans and other animals. Non-invasive MRI scans can tell us which brain regions have nerve bundles connecting them, but we can't know whether they are one or two-way connections.*

Also, if they're one-way connections, we don't know the direction of movement. This is a limitation of current brain scan technology. Because scientists cannot tell the difference between a one or two-way connection in the brain, they usually assume all nerve bundles are two-way connections. This is a reasonable simplification in many cases, and has helped us understand a lot about the brain.

During the process of constant communication, the left and right hemispheres are responsible for different behaviors, known as **brain lateralization**. This is the specialization we have talked about. However, we will not burden this section a description of the functions.

brain lateralization

The bigger picture mentioned the brain connected by various sections, we noted that the brain is thoroughly connected by millions, hundred of millions of axons. One then might ask where and what that those axons were connected to. Most of the time, we recognized that they are connected to the brain's components called **neuron** - the small central processing unit of the brain itself.

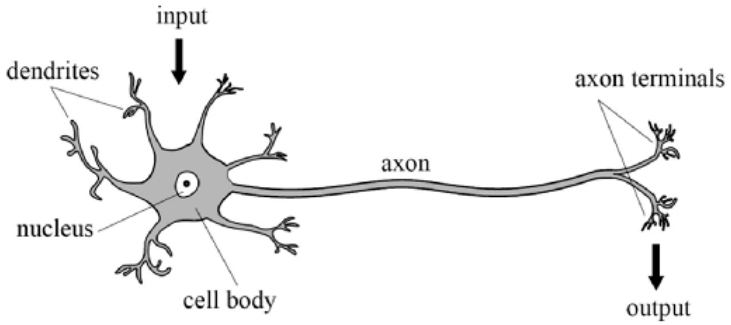


Figure 9.1: The simplistic, schematic illustration of the structure of the biological neuron.

The brain consists of a large number (approximately 10^{11}) of highly connected neurons. For our purpose, we simplify them to mostly three principal components, beside its life support: the **dendrites**, the cell body and the **axon**. The dendrites are tree-like receptive networks of nerve fibres that carry electrical signals into the cell body. The cell model effectively sums and thresholds these incoming signals. The axon is simply, as we have said, the cord connecting other neurons to it. The point of contact between an axon of one cell and a dendrite of another

dendrites
axon

cell is then called a **synapse**. It is the arrangement of neurons and the strengths of individual synapses, determined by a complex chemical process, that establishes the function of the biological neural network - though even by then, it is a gross simplification of the actual process - mostly based on empirical evidences. synapse

Aside from neuron, of the **cellular neurology** point of view, there exists also the **glia**, or **neuroglia** for the full name, which serves as the supporting cells for the operation of the main neurons' system. Specifically, the neuroglia should be emphasized to be rather inert - it does not align, or rather, can be classified as an operating unit in the brain, with respect to the well-known electrically excitable process that its brother neuron possesses. Indeed, because of such, there are many definitions in which neuroglia can take from, most of which are rather diluting, hence hitherto there are no agreed upon definition. In the above statement, we note that neuroglia as the supportive cells of neurons, but many exists to classify it by their process branching and delicate morphology, or, as mentioned, electrically inert components. As a result, 'neuroglia' has been come the generalized term that covers cells with different origins, morphology, physiological properties and functional specialization *aside from* the nervous cells of the brain. Such can be said of the uncertain analysis of neuroglia to the operation process and the long, complex chain of thoughts and functioning scheme of the host that it resides in, for whether the neuroglia participate in any incumbant roles throughout its working space. This is perhaps one of the issues with neuroglia researches, though it is not to say many attempts has been made trying to understand it, but rather the underrated position of the neuroglia to the other part of the brain itself. So, this much remains as a mystery. neuroglia

By itself, the brain's neuron and its neural structure is insanely complex. By time and birth, some of the neural structure is defined at birth. We don't know if this is encoded into itself by genes, but most likely so from biological evolutions itself. Other parts are developed through the dynamic action, often interpreted as learning (which is why we have the theory of learning), as new connections are made and others waste away. This development is most noticeable in the early stages of life. This is present in almost all developed neural structure of any given brain of any species. For example, it has been shown that if a young cat is denited use of one eye during a critical window of time, it will never develop normal vision in that eye. Linguists also have discovered that infants over six months of age can no longer discriminate certain speech sounds, unless they were exposed to them earlier in their life Werker and Tees [1984]. Somehow, it is also pretty vindicative to believe that the brain and all other functional components have a certain development timeframe deeply encoded in its biological encoding itself. Behaviourally, we can also interject that without pressure (like the fact that the cat must see, and must walk, so that it must move its legs and eyes), many functions would cease to be available.

Neural structures continue to change throughout life. These later changes tend to consist mainly of strengthning or weakening of synaptic junctions. For instance, it is belived, by 2000, that new memories are formed by modification of these synaptic strengths. However, this also posits the question of if the structure is static after a while - no more neurons constructed, then why would it be possible that, classical theory dictated, and our later on model will provide, that the neuron network of the same topology can give many memories at once? This questions, among others, require extensive studies and deep dive into the field of brain study.

Adding to the complexity, studying intensively in neuroscience will even separate the description of neuron further. When neuroscience was formed, and was developed, early in the nineteenth century the cell was recognized, only by then, as the fundamental unit of all living organisms. However, it was not until well into the twentieth century, that neuroscientists agreed that nervous tissue, like all other organs, is made up of these fundamental units. This would then bring out a surprising result by itself from the genetic side: of the 35,000 genes in human genome, a majority are expressed in the developing and adult brain; same is in other

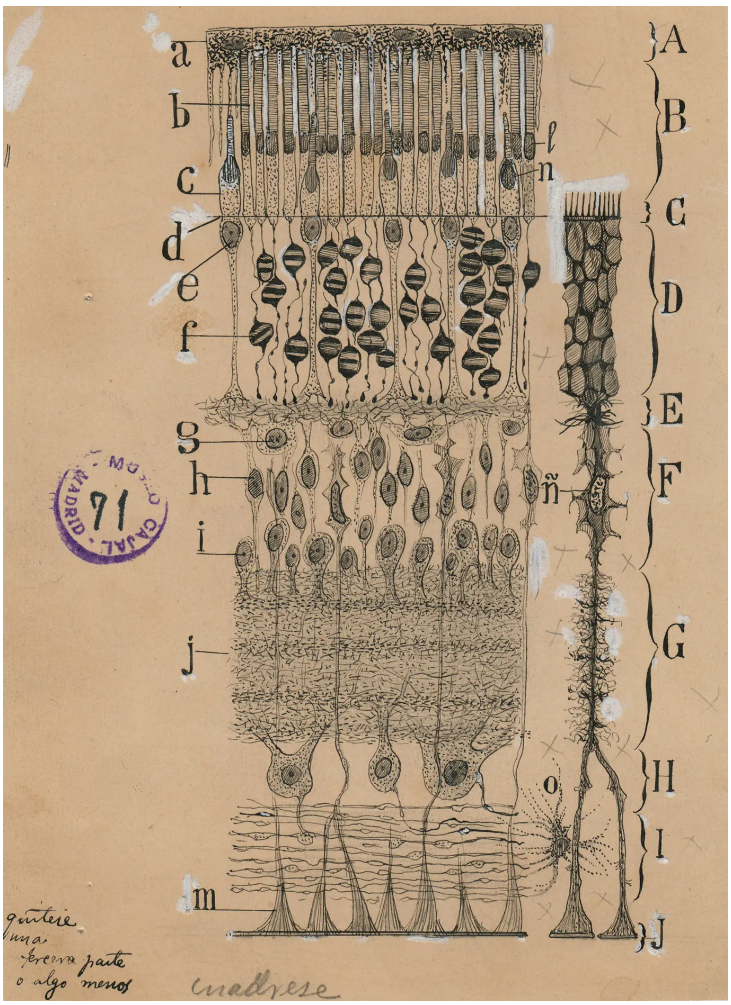


Figure 9.2: An illustration of Santiago Ramón y Cajal on the structure and design of a biological brain network. Many of these was made during his career.

animals; and most of all, *very few genes* are *uniquely expressed* in neurons, indicating that there exists a very well general structure for the building blocks of human. One then can be more surprised: why didn't they discover it sooner? The major problem and problem was that the first generation of "modern" neurobiologists in the nineteenth century had difficulty resolving the unitary nature of nerve cells with the microscopes and cell staining techniques that were then available. By that, we mean that "it's all because of the experimental system itself.". This inadequacy of observing the structure of neuron, was further exacerbated because of the ostensively, extraordinarily complex shapes and extensive branches of individual nerve cells, which further obscured their resemblance to the geometrically simpler cells of other tissues.

Reticular Theory

This then prompted, not surprisingly, the two rather classically famous approach to understanding the intricate structure that was observed. The first to arrive is the **Reticular Theory** of neurons, by prominent of Golgi and others like Joseph von Gerlach. Such is said about Reticular Theory by Golgi in his Nobel Prize for Physiology or Medicine (1906) that the axons physically join one nerve cell to another. By analogy, that is like saying that the brain is coherently connected, joined together like an electricity distribution network. For Gerlach, he

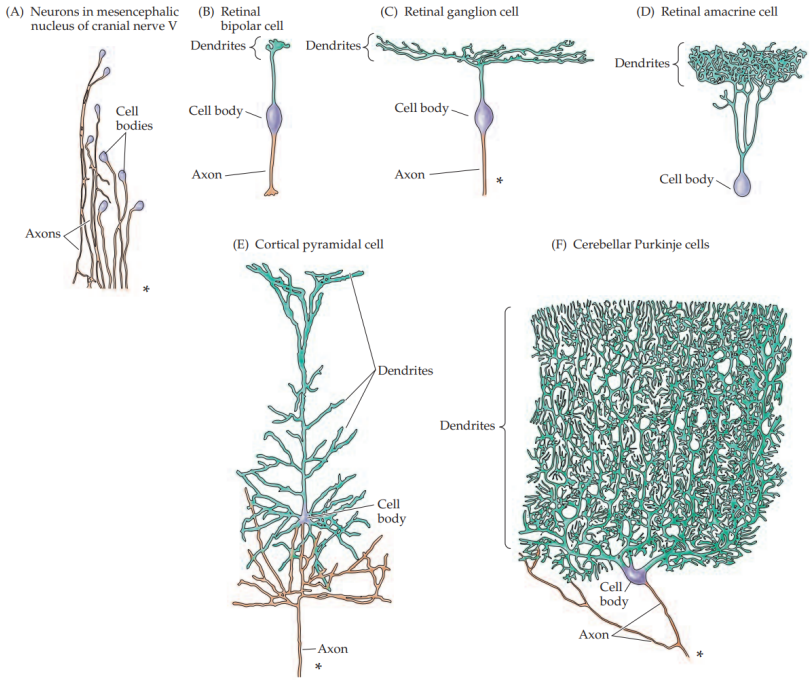


Figure 9.3: Examples of the rich variety of nerve cell morphologies found in the human nervous system. Tracings are from actual nerve cells stained by impregnation with silver salts (the socalled Golgi technique the method used in the classical studies of Golgi and Cajal). Asterisks indicate that the axon runs on much farther than shown. Note that some cells, like the retinal bipolar cell, have a very short axon, and that others, like the retinal amacrine cell, have no axon at all. The drawings are not all at the same scale. Some more details about the jargon is the *retinal bipolar cells*, which are neurons that connect the outer retina to the inner retina, for processing layer (or projection neurons, where all information are relayed from this connection.); the *retinal ganglion cell, amacrine cells* are the same visual processing unit; Cerebellar Purkinje cells (a type of GABAergic neurons) uniquely determined for cerebella cortex (for processing large data, and coordinating functions like cognition and emotions.). Reused from Purves et al. [2004].

even wrote a two-page article titled "Ueber die Structur der grauen Substanz des menschlichen Grosshirns. Vorläufige Mitheilung" Gerlach [1872] in which the sentences ended with '[these cells] are interconnected with each other as well as connected with the radial bundle, whereby a coarsely meshed network of medullated fibres is produced which can already be seen at 60 times magnification'. This theory eventually, though, fell from favor and was replaced by what came to be known as the "neuron doctrine", championed by Santiago Ramón y Cajal, a Spanish neuroanatomist, and Charles Sherrington, a British physiologist. This theory insists on the otherwise different interpretation - the brain is not continuous, and is rather composed of *independent cells* and components. More than ever, it also gives us some very beautiful illustration of Cajal himself.

Neural Doctrine

The contrasting views represented by Golgi and Cajal occasioned a spirited debate in the early twentieth century, that set the course of modern neuroscience. At the end of the day, however, it seems like the modern world of neuroscience has chosen Cajal because of the prominently accurate expression of Cajal, and supported by immense experimental result - particularly after the invention of electron microscopy in the 1950s. Out of this debate, and a

lifelong endeavour, is the tuple of neuron-neuroglia as we have been discussing of. [1]

We would be wise to intercept this complexity with a specific modelling, and to use whatever knowledge attainable of the moment rather than waiting for neuroscientist to fully discover the brain - at that point, there is not much to be done anymore (others than copying and replicating, that is). Because it is so complex, artificial neural networks do not approach the complexity of the brain by itself. Rather, we restrict ourselves to a very simplified notion of the neuron. Nevertheless, two key similarities between biological and artificial neural networks can be founded. First, the building blocks of both networks are simple computational devices - note that this is a **serious, gross simplification from both side** of the equation - that are highly connected. Second, the connections determine the function of the network. Then, particular configuration of the network will be much better utilized in some tasks, and less of others. This is a dynamic of functions that is perhaps also presented in biological networks. Also, we will also remove the neuroglia of the equation in the subsequent model that will be presented. Such is to say, we are not replicating the neural system by everything, but rather, focusing on the not-so-inert part of it.

With this, we are now ready to begin our model of neuron. We will do it step by step. Not wasting anything in between, that is. But first, we need to know what does our neuron do, in the most general way.

9.2 The neuron model

Because, generally, we have discussed and formalize somewhat our mathematical modelling setting into the 3-tuple (S, Q, M), it would be a loss of rigours if we are not to treat the neuron in a structural way. In such, we will also absolve certain ambiguity, if able.

The system S in concerned of the neuron model is perhaps different from what we will be having in general. Normally, we will consider the system in which only the object of interest. In such sense, our structure will contain the following: The model object's components, or **atoms** $\{n\}$, and the larger construction that is our model, $N[\{n_i \mid k\}]$, for any k configuration that is for now arbitrary. This will be good for the time being, however, later on, if we are to expand it into a larger, bigger system in which multiple 'models' are made, then we will have to find ourselves another interpretation.

In such case, there exist two approaches. One can approach the problem by extending the model into not just a neuron, but a bigger network of neurons, yet so forth being a single neuron in interpretation. That is, there now another configuration k' that gives a comparatively *recursive definition* of the neuron: a network of neuron acts like a neuron, containing inside it neurons of smaller size. This means then a "neural network" is essentially the model $N[\{n_i \mid k', k\}]$ such that each $n_i \to k'$ configuration is a neuron, and under them are the configurations k of the neuron class. This interpretation is rather suffice, but again, it considers a fairly strict amount of construction, and there is also the problem with interpreting the recursion section, too. Instead, what we can do is to simply extend the system. Now, the object of the system will be bounded above. That is, the system S now contains, for example, the following objects:

$$\begin{aligned} S &= \{n_i\} \\ &\cup \{N_k\} = \{n_i \mid k \in \mathcal{K} = \mathcal{K}^1, r \in \mathcal{R}\} \\ &\cup \{\mathcal{N}\} = \{N \mid k' \in \mathcal{K}' = \mathcal{K}^{(2)}\} \end{aligned} \tag{9.1}$$

[1] It is, however, foolish to consider the idea of Golgi to be entirely false. What can be seen as a continuum by Golgi might actually be the continuous operation flow between the network of neurons together with each other, which then explains the weird fluctuation and continuous response behaviours in time, thus ruling out the falsification, but transfer the view to another point.

Note that this is not the current notation of choice, only for illustrative purpose, so we will be able to forgive some of the lack of coherent notation like $n, N, \mathcal{N}$. For each level of the neural network structure that will be eventually constructed, we will have each individual component set. Using a rather naive notion, we have then implicitly strict our model, in which we will discuss in more detail in a bit more time. It is also notably interesting of the implication of the $\mathcal{K}^{(i)}$ ordering model can be. In fact, it can provide us with an even more compact, and abstracting level of neural processing unit, though we will not discuss such for now.

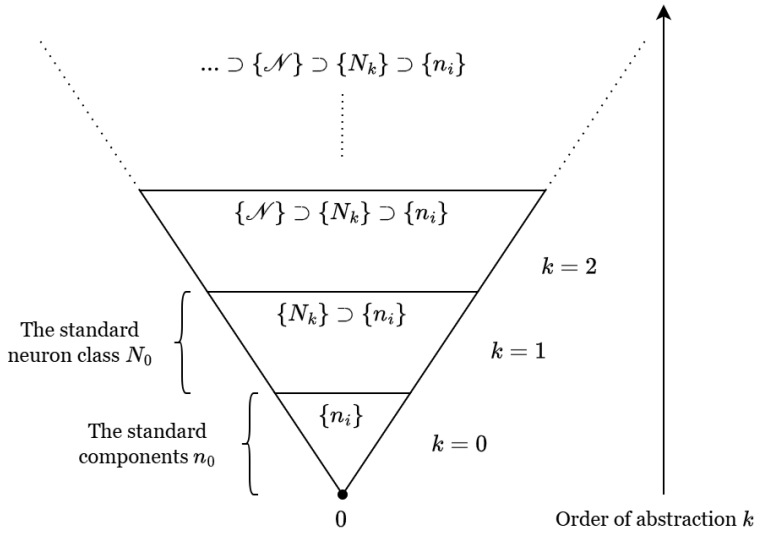


Figure 9.4: **An illustrative example of the abstraction and categorization by 'size' of different components and constructs in the neuron model.** By the order of abstraction k, we assign a notion of size on different neural structure, by increasing complexity, and backward compatibility (described to be composed of previously defined objects). The first two stage for $k = 1, 2$ includes the standard basis components $\{n_{0,i}\}$ and the standard neuron class N_0, respectively.

Now, for the question Q. What is it that the questions we have when we are constructing this model of neuron? Normally, the question would be rather on the extreme: we want to *reconstruct the neuron*, or rather, making a unit that functions in the same way, or abiding its principles. In doing such definition, we will be able to rule out certain factors, and perhaps focus on the state variables system instead.

By this, we assume no physical realization. Hence, factors that rely on the physical interpretation and phenomena - for example, the delays in between signal transmission - we then assume our data is instantaneous. The forming of axons and synapses will also not be included, as we might as well work with a static wiring scenario instead. Simplify even more, and we get to the point where we will only extract the supposed fundamental unit of operating, that is, turning the neuron into exactly an input-output unit. All construction of variations of this type of neural construct is then called the **neuron class** (classical) in which connectionist builds their neuron. Denoted as N, the neuron class is constructed of three fundamental components as $(I(p), \Sigma, f)$, where I handles the input into the neuron, Σ preprocess the input into some forms or another, and f regulates the internal mechanism of output-transmission of the neuron - in a somewhat ambiguous term, regulate what the neuron is designed to "think" given specific input and its own interpretation as f. If you see others, more advanced and often messy

construction of a neuron, then it is because they have added semantics onto it, or rather, the supporting and outer influencing components. However, the standard issued compartments are the same, and hence, we would treat this as the base model for our neuron unit.

Each operation in the neuron model, at least for the standard basis, serves a different purpose. The input I is there to handle the input, by symbolically giving each signal or channel of a specific elementary configuration, called **weights** w. This will determine the relative "importance" of a given input, or by scaling up and down the channel's information inward. Numerically, it controls how the preliminary processing or dependency of inputs are observed to be configured in the objective concept that is required to be mimicked. For example, if the objective concept is something like $3x_1+4x_2-1/5x_3$, we would expect the preliminary process to handle the factor $(3,4,1/5)$ to be replicated by behaviours of x_i. Σ on the other hand, provides the $+$ operation inside. Often, we use addition or for a vector representation of all input $\mathbf{x} \in \mathbf{X}$, it is the dot product. But generally, it is defined for all generalized composing operation $\Sigma = \bigoplus_{i=1}^{n} w_i n_i$. And finally, as we may have been familiar, is the interpreter of the neuron, the transfer function f. Though, arguably, the transfer function is classically defined only for it to be able to let the neuron be nonlinear, with respect to the space of all numerical representation.

With this, we are able to define the first definition on the neuron model. This is a preliminary definition, since we will add things in later on.

Definition 9.2.1 (Neuron model, Preliminary definition). *The neuron model conceptually can be encapsulated into the categorization of all* ***neuron class*** N *with the* ***standard basis*** *denoted by* N_0*. The standard basis is then consists of the tuple* (I, Σ, f)*, where* I *denotes the process operates on input form which denoted* p*,* Σ *is the input internal process - most prominent in case of multiple discrete input channel present (essentially helps to organize the input interface), and* f *is the* ***interpretation output*** *of the model.*

It is good to observe that the neuron as a unit is somewhat very similar to how we observe the black-box model of mathematical modelling, for a given problem. Hence, it is sufficed to say, generally, that in some ways or another, that the neural unit is actually approximating the supposed internal state of a black-box system, by presupposing certain simplification or empirical generalization of the observed information. This is often not guaranteed to be similar to the actual black box, but rather of its own interpretation unit, to approximate it with a presupposed internal mechanics instead. One could then ask though, of the extended problem to this conception of neural unit - what would then be the expression for the **internal system of a neural network** (more than one neuron)? We would do this in sequence, so let's hope that we will touch upon this.

9.2.1 Principles and philosophy

To finalize our view on the definition and principles, we have to give ourselves the plenty (mathematical) statement in consideration of the system in which our neuron will be defined of their subjects and functions. Those will hopefully, eventually lay out the correct assumptions that we took, figuratively, and to prevent further ambiguity (in the case where everyone, and anyone get different reactions or formulations). In essence, our construction forgo the following assumptions for the statements M:

(+) The flow of information and operation is *one-dimensional, single-directed*. That is, every operation follows the schema recognized by the input-output model, such that any open interface or exposed structure must abide the sequential and single-directed flow of process, for all components in a neuron class. This then defines a **'heliocentric' topology** on the neuron class,

heliocentric topology

denoted by C which organize the flow of process. [2]

(+) Almost all physical realization of the neuron is removed in the modelling. That is, there exists no synapses and axons, replaced by the abstract input-output - we assume no failure of communication. Furthermore, we assume that each neuron can handle infinite amount of data, given certain representation of said data channel, and output of the same amount. If not explicitly stated, we also assume that the channel capacity between each connection is negligible, and always infinite. Numerically, all neuron also has infinite numerical representation range, at least when it comes to the real number field $\mathbb{R}$.

(+) All neuron class can be subsequently reduced to the standard basis neuron class N_0.

(+) The input section I only process data by packaging each input channel with a modifiable weight w, or the importance measure, and nothing more.

(+) All information or materials assigned in the operating and living space of the model is numerical, perhaps. As such, we might want to insist on a clear *representation scheme*, and a embedded structure within all numerical system used to represent it.

(+) Each neuron has two types of parameters: the macroscopic parameters that define its structure, called **hyperparameter**, and the inner mechanism-specific parameters, called the *systemic parameters* that is intrinsic of the inner configuration and operation. For our theoretical treatment and applications of the model, our model assumes **static construction** but **dynamic mechanics** for said construction. The reverse is noteworthy of being analysed, and this criteria in general is an interesting notion.

(+) A consequence of the single-directed flow is that one can form a neuron loop with two or more neuron together. For a network of connected neuron, we said that it is an **isolated**, or contained network if no connection is headless - not connected to any of the given neuron.

(+) A neuron can perform **self-loop** - feeding itself of its own output. This scales up for network as well.

(+) Correction within connections of neuron is negligible and is almost non-existence in context. If there exists noise or unwanted information, then it will almost be presented in the input observations and information - any external sources.

Each neuron can also be defined by either design, or by operation, to be independent of each other. This is often conducted by defining the relative dependency, given the assumption in design of being single-directed, most representable by the notion of **layer**, denoted by $\mathcal{L}^{(i)}$. A layer is, loosely speaking, an ensemble of neuron units, all of which is arranged in parallel, with equal processing order, that is, they operate sequentially parallel as a cluster on their own. Then, each layer defines the relative dependency to the previous layer, if there are any, and hence, a neuron only depends on the behaviour feed of neurons in previous layers. For a neuron $n[i] \in \mathcal{L}^{(j)}$, however, it can also be designed such that inhibitory neuron in $\mathcal{L}^{(j-1)}$ does not affect its operation or signals of other neurons in operation. For example, that is why a lot of typical construction of neuron use $\Sigma = \sum_{i=1}^{n} w_i p_i[R]$ for input sequence $p[R]$ of range R, such that if one neuron is inhibitory ($= 0$), then others are not affected. This is perhaps trivial, and should not have been mentioned, but I mention it anyway.

layer

[2] Then, single-directed means that C is minimally described by a vector $\vec{d}$, such that if each neuron is represented per specification, into a directed graph (either equipped with an embedding space or not), then the component's direction vector $\vec{c}_i \in C$, must satisfy $\langle \vec{d}, \vec{c}_i \rangle \geq 0$. This notion wil have to be expanded later on, as we will see why it is perhaps more important than not. In fact, I am racking my brain to think of the case when one dimensional, without single-directed description would be. This notion also generally can be applied to larger network as well.

Using the better, well-founded notion for (S, Q, M), we then define a more conclusive definition of the conceptual neuron model.

Definition 9.2.2 (Neuron model, neuron class). *The neuron model conceptually can be encapsulated into the categorization of all* ***neuron class*** $\mathcal{N}_{k_i}^{(i)\,a}$*, for* $i = 1$*, specified by two descriptions for any neuron* $n \in \mathcal{N}_{k_i}^{(i)}$*: An quantized, 'parameter(-ized)' descriptions contains all variables or parameters that define the* ***represented mass*** *of the model, denoted* N*, and the operational description contains, for example, rules, flows, cases, orders, functionals, special values, denoted by* $\mathcal{M}$*. We can also call the neuron class as the* ***hypothesis class****, for* $h \in \mathcal{H}$ *of a given description.*

[a] This description is perhaps incomplete, but we will use it throughout our formation as it is. the suffix k relies on the arbitrary meaning of a *configuration space*, hence k_i is all arbitrary configurations possible for structure of order i.

An interesting question (perhaps can be said to be fairly long, and an *interesting side tangent*) arises when we state out those question by ourselves, and is perhaps more importance in the sense of information theory and perhaps, quantum mechanical interpretation. [3]

Question 9.2.1. *How is information, or* ***resources per matters*** *are preserved or transmitted in an artificial neuron* a *of a given neuron class of both standard basis or more? Are all information destroyed figuratively, if one assume no knowledge from the external modifier side (in a typically learning process, one requires a supervisor with full exposure to the internal mechanics of the neuron model - or generally the hypothesis). Given two neuron acting on the same scenario, but with diverted process, can one revert the information to retrieve the opposite reversed neuron respectively, in which we call the standard one the encoder, and the opposite the decoder? What is their topology?*

For this question, we have to deal with a lot of things more before the experimental setting can be structured to test it. Another point to note is that usually, one can treat the neuron component f as the neuron's intrinsic interpretation of the encoding space. What does the encoding space means will be further defined and explored, but for now, you can picture it as its interpretation in the numerical encoding of the world and observations thereof. The question above indicted the issues toward the lack of analysis in such aspect, however, we would like to present interesting notions on said view, for example, Liu et al. [2025] with their **Kolmogorov-Arnold Networks (KAN)**. We leave this for now in favour of the foundational assembly of our model, to be discussed in later sections.

9.2.2 Class separation

The final issue that have not been discussed yet, however, is to cut out the notion of class categorization in our general idea. Specifically, *what constitute the class separation* of its entirety? To separate something means to based of some specific quality to classify them into hierarchy in our case - and such depends on a scale of which items can be put up in order, with less ambiguity than what is tolerated. What is then such scale to apply on the neuron units' classes?

Toward such issue, it is rather fairly simple, though not particularly troublesome in the potential downfall of such specific organization. Previously, we bargained on the stance of separating the model of operational machines into two main aspects - the process running on itself, and the acute facilities, both *existential* and *functional* of the machine. Per our view, the neuron class refers to the facility and not the process, of which process is then the unique combinations and configuration of the facilities provided. Coupled this with the notion, or rather, the principle of construction such that the first class $\mathcal{N}_{n-1}$ supersedes $\mathcal{N}_n$ of its functions and main

[3] A trivial note. In the process of neuron functions, there will always be information down scaling and dimensional transformation. Information losses and reinterpretability is questioned, as always, but might be redundant as it is.

compartmentalization (that is to say, using $n-1$ as main component in constructing n), then the scale of hierarchy is simple — separating by virtue of nested construction, or the path of dependency.

Theorem 9.2.1 (Class separation principle). *Given two neuron class, $\mathcal{N}_{n-1}$ and $\mathcal{N}_n$ relatively. Then, of the decomposition of neuron class (as in theorem 9.2.2), as $\mathcal{N} = \{\mathcal{M}, N\}$, then the ordering makes sense if $N = \mathcal{N}_{n-1}$, that is to say, components of $\mathcal{N}_n$ consists of largely $x \in \mathcal{N}_{n-1}$. Or rather, for two arbitrary $\mathcal{N}, \mathcal{N}'$, then if the following holds:*

$$\sup_{x \in \mathcal{N}} |x| = y \in \mathcal{N}' \tag{9.2}$$

Then we say that $\mathcal{N}$ is of higher class than $\mathcal{N}'$, and can be assigned of high index $i \in \mathbb{N}$.

This principle holds for any organized system. However, usually, we will find ourselves confined in designs and prototypes that have no distinctive and unique class categorization by itself. In such case, we consider them **rogue class**, of which goes out of the scope for such principles and laws being used in the theory, since they cannot be applied on. rogue class

9.3 Notation

Before we construct neurons and their individual components, we would have to select and systemize our choices of the notation specifically. We make deferences of the neuron class, such that all neuronal architecture class is denoted by $\mathcal{N}_i^{(p)}$, for any variation p of the neuron class of level i in the relative scale, for $i \geq k$. A specific *network construction* of the neuron, is then called neural network, denoted by $\mathfrak{N}$. Distinguishing between neural networks altogether would be used of varied subscripts and superscripts, as there exists no strict rules on them. Each **neuronal unit** (or just neuron instance) is denoted by $N \in \mathcal{N}_i^{(p)}$. Component-wise, we can also write $\mathfrak{N} = \prod_{i<k} \mathcal{N}_i$ by the Cartesian product on the neuron set (we typically use neuron set and class in conjunction with each other anyway) to represents different configuration possible, or at least the space of which the neural network takes from. Hence, for example, if $k = 5$ for 6 classes of neuron in participation, we would have:

$$\mathfrak{R} = \prod_{i<k} \mathcal{N}_i^{c_i} = \mathcal{N}_0^{c_0} \mathcal{N}_1^{c_1} \times \mathcal{N}_2^{c_2} \times \mathcal{N}_3^{c_3} \times \mathcal{N}_4^{c_4} \times \mathcal{N}_5^{c_5}, \quad \{c_1, c_2, c_3, c_4, c_5 \in \mathbb{N}\} \tag{9.3}$$

with $\{c_i\}_k$ the set of the number of neuron in construction, which is analogous to different configuration space for each of them. The component of individual neuron is then denoted generally by q_j for j the index for all r components of a neuron; r is also available as specification from the neuron class, however, subsets of each neuron class might have just different r. We will proceed with the above notation, and any given improvements or changes in notations will be specific instead.

9.4 Classical neuron template

As we might have observed, neuron has its own familiar properties and characteristics that defines the name 'neuron'. We then might as well classify the neuron in an operational sense - purely as a working, functional mechanics. Then, a **neuron** $x \in \mathcal{N}$ of any class will have to have the following component classifications as its minimal requirement to be classified as such:

1. One external state unit, or **input unit** I to interpret received information. input unit
2. One internal state and action units, or **mechanical unit** M. This includes the *mass* - what is used to express it, and the *operation* - what is used to process it. mechanical unit

3. One external action unit, or **output unit** O. This can be considered to be the *product* of the model, or *observables* that we can see of the model.

output unit

We can define this formally as a definition, though it does not change much from the original presumption rather than stating the already established notion. However, we will then call such as a framework.

minimized structure

Definition 9.4.1 (Minimization set). *Let x be a neuron of arbitrary neuronal classification $\mathcal{N}$. Then, the requirement of all neuron class is to be able to distinguish its component to three parts, that is, $\min_{\mathcal{N}_i \in \mathcal{N}} \mathcal{N}_i \equiv \mathcal{I}, \mathcal{M}, \mathcal{O}$ where $\mathcal{I}$ is the input channel, $\mathcal{M}$ the internal mechanics, and the output $\mathcal{O}$. Let i, j, k represents the cardinality of each part respectively, then if*

$$i = j = k = 1, \quad \min_{\mathcal{N}_q \in \mathcal{N}} \mathcal{N}_q = \mathcal{N}_q, \forall q \geq 0 \tag{9.4}$$

*Then we call this class of neuron the **minimal neuron class**, and any $x \in \mathcal{N}_i$ of such is called the **minimal neuron** or **standard neuron**, denoted by x_S. By default, this is satisfied if $q = 0$ in our construction.* [a]

[a] The constant i here refers to the organization numbering of nested classes built upon by another components. In such, we observe that this construction implicitly defines itself to be the simple zeroth class.

With this condition, any construct x of this type would be called a neuron unit. There is simply no configuration specified, so x can belong to any qth class. This is illustrated as in Figure 9.5. Every member of a class would then be concerned of largely those objects that can be classified as such.

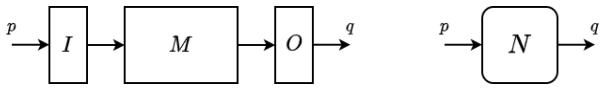


Figure 9.5: **The standard minimal configuration of any neuron** $x \in \mathcal{N}_i$. We denote p, q for particular neuron input and output sequences.

Even though we insisted on the components, many times they can be missing from the neuron yet would still be called as such. In such "edge case", we can employ the notion of inhibition or blocking to represent, or at least interpret the absence of such action as to be blocked, rather than the missing of such component. Therefore, a neuron without $\mathcal{I}$ is called an **isolated (neuron) unit**, a neuron without $\mathcal{O}$ is called an **endpoint (neuron) unit**, and a neuron without $\mathcal{M}$ mutation is called a **dummy (neuron) unit**. More importantly, dummy units will be used in plenty illustrations and formation to classify typical linear behaviour, unchanged instructions, or placeholders.

Furthermore, as we have been seeing from the formulation, it is also of interest to know why we implicitly want to have only one unit for each type of components. Thereby, we would also define the component classes. Hence, there exists the **input (unit) class**, the **output (unit) class** and the **internal (unit) class**. And as the definition above hold, the reason for minimal class to be satisfied is not special at all. Well, at all. It simply is to restrict the component class, and to somewhat prevent unnecessary constructions that would lead to overcomplication later, for example, by not examining standard component, the dilemma between many-input and single-input neuron will get far more complex. Mentioning this, and countering this by force at hand is perhaps more suited as for now, so that we will not encounter the same later on. And it also applies with our view on constructing everything similarly to computers and electrical components constructs larger units. Then, let's see how we can make use of the standard neuron first, by introducing the first class - the class $\mathcal{N}_0$ **simplex**. You also might have noticed that this is simply in normal theory, the single unit neuron.

9.5 Class $\mathcal{N}_0$ simplex

When we begin with everything, we start with the smallest and most fundamental unit. In this case, it refers to components of the same type categorized into the **class $\mathcal{N}_0$ simplex** (not complex). The first neuron structure is the single-input, single-output, *minimal neuron construct*. A neuron $N \in \mathcal{N}_0$ considers all special neuron construction such that $\mathcal{N}_0 = \{I, M, O\}$, where I represents the input module (and its processor), M is the inner structure of the neuron, and O represents the output unit of it. The first neuron of the class $\mathcal{N}_0$ is called the standard neuron, historically introduced throughout by McCulloch and Pitts [1943], Nakkiran et al. [2019], Goodfellow et al. [2016]. But first, let us introduce to the familiar notion of a standard neuron, through the lens of its components in $\mathbb{R}$.

$\mathcal{N}_0$ simplex

$\mathbb{R}$ standard neuron.

Definition 9.5.1 (Standard neuron on $\mathbb{R}$). *A neuron unit $x \in \mathcal{N}$ belongs to class $\mathcal{N}_0(\mathbb{R})$ and is called a* ***standard neuron on*** $\mathbb{R}$ *if it satisfies the minimization set criteria, and can be written of the form:*

$$x = q = \sigma_{\mathcal{M}}(w \cdot p + b), \quad p \in \mathcal{I} \subset \mathbb{R}, w, b \subset \mathbb{R} \subset \mathcal{M}, \sigma : \mathbb{R} \to \mathbb{R} \in \mathcal{M}, q \in \mathcal{O} \tag{9.5}$$

If σ is linear unit, that is, $\sigma(wp + b) = wp + b$, then we say x is a ***linear standard unit****.* [a]

[a] One might ask why we use the product and addition in the formula of wp and then b. In fact, this is perhaps more trivial - as to facilitate the concept of *linearity* - the formulation looks exactly like the linear line in a plane. Furthermore, as we will soon see, it is also of interest such that units of neurons can be linearly combined in a way, at least of computational aspect in running it on computers.

Using this definition, we have constructed a standard neuron without any single component added outside the categorization above. While we have been saying arbitrarily placeholder for the components, it's then we have to clarify the options and specification for each component.

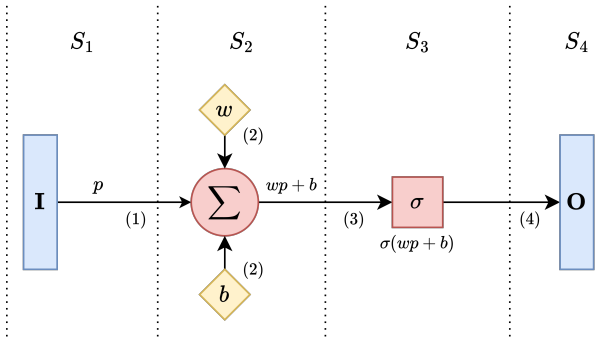


Figure 9.6: Commutative diagram of the standard $\mathcal{N}_0(\mathbb{R})$ class. The in-out objects are denoted in blue, the operators are denoted in red, and the objective mass (parameters) are denoted in yellow. The procedure is then denoted of four successive processes of S_i, up to S_4.

Figure 9.6 is the illustrative diagram for the flow of operation of a standard neuron. As we can see, it receives input, which can be thought of as generally just $\mathcal{I}$, if there exists no modification or filtering of the input, which is not in this case. Two parameters w, b is the neuron's internal mechanistic part that dictates how the input is controlled by a factor of multiplicative and additive; and finally σ attains the interpretation of the neuron to the data itself. Since there's nothing else to work on after σ processes the input modified, we output it to q, hence completing a cycle. Note that this construction abides the $(1, 1, 1)$ template, and hence x is a minimal neuron. Hence, we have that *all $x \in \mathcal{N}_0(\mathbb{R})$ is standard*, or $\mathcal{N}_0(\mathbb{R})$ is the subset of the **standard neuron class.**

standard class

This neuron class is fairly simple, since it has only one channel of action and one channel of input. Its configuration can also be easily modified. However, the reason why there exists the modifier $(\times, +)$ is troublesome. Why we need it? Heuristically, it is because of the observations that biological neurons have this kind of *inhibiting* approach, such is to completely block specific information flow from the synapse. Hence, it is also then to facilitate that kind of behaviour. Another answer to this question is to consider the standard neuron in this case without σ. Then, the standard neuron becomes an *amplifier unit*, that is, for $0 \leq w < 1, -p < b < 0$, it inhibits the input information, while for $1 \leq w, b > 0$, it amplifies the input. If the signal is negative, however (generally speaking or biologically speaking, negative activity seems redundant as nonexistent), then $w \neq 0, b < 0$ and $w > 0, b > 0$ are the respective inhibition and amplification range - absurdly simpler than the positive case. This is the **operational approach** to justifying this. However, if σ is in definition, we would have trouble with this interpretation. In such case, we can say that the above action *shift the perception range*, that is, interpreting the input as more than it is, or lower than it is, thus reducing or expanding the perception range of the neuron. Furthermore, it is also intuitive to recall that positivity and negativity only works as a notion, when there exists an **encoding** that is specific of such. Thereby, it is then important to note of the necessity of an encoding or reference frame when constructing and interpreting such structure. Unlike in biological neuron where one can make distinction between positivity and negativity as to mean directions of electrical flow between neurons: **Excitatory Postsynaptic Potential (EPSP)** or **Inhibitory Postsynaptic Potential (IPSP)** thus representing positive or negative charges, we do not have such comfort in the simplified model itself. And we will have to make ado with defining our model in the most general way possible, as for an operational model in the sense of it operating under any configuration.

It is at this point that we note a perhaps considerable notion in between the dilemma of interpreting the standard neuron. In the above standard neuron on $\mathbb{R}$, we indict the entire environment to work on the encoding space of real number space, $\mathbb{R}$, and as scalar, singleton number. What if it is not the case, and instead the information received is singular, yet interpreted differently, for example, as a string of binary 0110010101? Furthermore, what if there is mismatch between the data in, and the system to handle it? Based on the singleton structure, then this would be inadequate to resolve. We want a minimal structure, but there are a lot of things that is perhaps not trivial and is not minimal, hence makes the analysis fairly troublesome. We would likely want to generalize the neuron class as such. Figuratively, the **standard neuron on specific encoding** $\mathcal{E}_W$ would be defined to be similar to the following diagram:

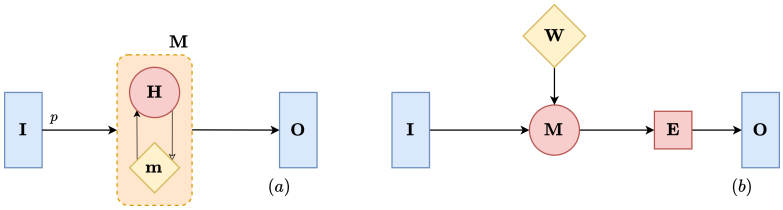


Figure 9.7: **Illustration standard neuron class** $\mathcal{N}_0$. (a). We regard the component of the model as **M**, consists of the mass and the operations **H**. (b). Instead of considering the operation as subcomponent of the model structure, decomposition gives them separated with two types of operation - either *processing* operators (operations that prepare the parameters) or *transforming* operators (act on the prepared processing that it receives).

This is more general than the standard neuron on $\mathbb{R}$. By configuration, it can also be defined to be a singleton, that is as followed.

Definition 9.5.2 (Standard neuron). *A **standard neuron** is a 3-field $\mathcal{N}_0$ corresponding to $(\mathcal{I}, \mathcal{M}, \mathcal{O})$ such that the standard neuron (minimization set criteria) criteria is satisfied, and $\sigma_{\mathcal{M}} \in \mathcal{E} \subset \mathcal{M}$ belongs to the space $\mathcal{E}$ of all **encoding** of the neuron structure.*

With this, on itself, the sign $(+, -)$ does not have any meaning. But, with the encoding, it would be interpreted as something and not just simply semantic. We also mandate that the input must be usable, otherwise, if its representation is different, we would not be able to work with each other effectively, or at all.

Now, let's talk about σ itself. Usually, it is called a **transfer function** or **activation function**. Because it is a function, we note in this case, typically, its range and domain is $\sigma : \mathbb{R} \to \mathbb{R}$ of all reals. Transfer function, or activation function, as we have noted as fixed, can be chosen independently to satisfy some specification of the problem that the neuron will attempt to solve.

A variety of transfer function can be included, but most of the time, we would like to focus on certain types of them. For example, the *hard limit function*, more piecewise function, *linear function* $f(x) = x$ — in the case where the processing preserves the combination, or **sigmoid**, $a = 1/(1+\exp -n)$. Recently, there is also the Rectified Linear Unit (ReLU) function, defined by

ReLU

$$y = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$

which takes inspiration from the rectification in signal processing. The list of all transfer functions can be found in the appendix section.

9.5.1 Chaining standard unit

We can chain multiple standard neuron together, as their design allows for x to give its output to y as input, and so on for $z, q, f, g, \ldots$. This is called *single chaining*. For now, we would not dive too deep into its effect, however, we will have to define its structure and behaviour. Let's say we have three neurons (we shorten standard to this) x_1, x_2 and x_3. Chaining basically

neuron chaining

single chaining

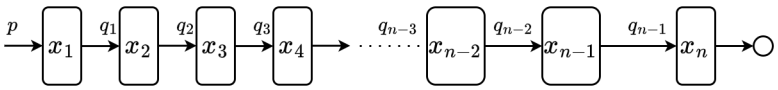


Figure 9.8: Chaining of multiple standard unit on each other.

can be seen as connecting or combining two neuron together by letting p_{x_2} (the input), to be $\mathcal{O}(\sigma(wp + b)) = \sigma(wp + b)$, which we would denote it simply as $\sigma(x_2)$ for clarification, and because the output $\mathcal{O}$ only is there to specify the flow. Hence, in this sense, we can separately illustrate the progress for the transition function such that,

$$p(x_1) = p_1, \quad p(x_i) = \sigma_{i-1}(x_{i-1}) \ \forall i \geq 2 \tag{9.6}$$

similar to how function chaining is performed. Hence, we gain the functional chain $\sigma_1 \circ \sigma_2\sigma_3$. Extend this to n-length chaining will give n-composition of σ.

To express this more clearly, let's take $wp + b$. Then we have:

$$\begin{aligned} p(x_1) &= p_1 \\ p(x_2) &= \sigma_1(x_1) = w_2(\sigma_1(w_1 p_1 + b_1)) + b_2 \\ p(x_3) &= \sigma_2(x_2) = w_3[\sigma_2(w_2(\sigma_1(w_1 p_1 + b_1)) + b_2)] + b_3 \end{aligned}$$

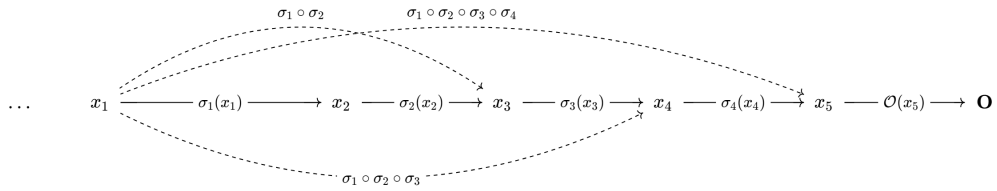


Figure 9.9: Illustration of the chaining process and the nested function chaining between σ_i.

$$\vdots$$

$$p(x_n) = \sigma_{n-1}(x_{n-1}) = \sigma_{n-1}\Big[w_{n-1}\{\sigma_{n-2}(w_{n-2}(w_{n-3}(w_{n-4}\ldots) + b_{n-3})) + b_{n-2}\}\Big]$$

This looks rather confusing, so we might be more inclined to look at this in a more simplified manner, for $n = 3$, thus three neuron x_1, x_2, x_3 as above to see what is going on with the chaining calculation, and with specific activation function for said purpose. Let's take $\sigma(x) = 1/(1 + \exp -n)$, the sigmoid one. Then:

$$\mathcal{O}(x_1) = \frac{1}{1 + \exp\left[-(wp + b)\right]} \tag{9.7}$$

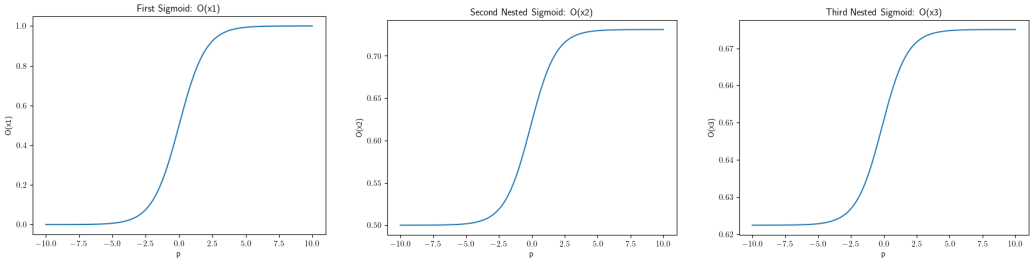
As for chaining, we observe that $p_{x_2} = \mathcal{O}(x_1)$, hence:

$$\mathcal{O}(x_2) = \left\{1 + \exp\left[-w_2\left(\frac{1}{1 + \exp\left[-(wp + b)\right]}\right) - b_2\right]\right\}^{-1} \tag{9.8}$$

Finally, for $p_3 = \mathcal{O}(x_2)$, then we have:

$$\mathcal{O}(x_3) = \left[1 + \exp -w_3\left[1 + \exp\left(1 + \left[-w_2\left(\frac{1}{1 + \exp\left[-(wp + b)\right]}\right) - b_2\right]\right)\right]^{-1} - b_3\right]^{-1} \tag{9.9}$$

This would take quite long, and as it suggest, the sequence indeed decreases by itself up to kth chaining. Now though, the interesting part is, what is their effects? And how they work together? Let us start with the generic composition, for $w = 1, b = 0$ throughout. This is illustrated in Figure 9.10.

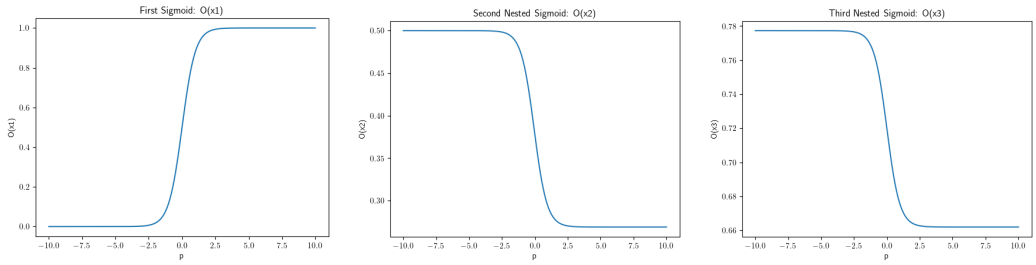


(a) First sigmoidal node. (b) Second sigmoidal node. (c) Third sigmoidal node.

Figure 9.10: Initial starting configuration for x_1, x_2, x_3 and their functionals $\sigma_1, \sigma_2, \sigma_3$ under the same initializer.

Let us then refer to the configuration as $\mathbf{w}, \mathbf{b}$ for the vector $\mathbf{w}$ of all control parameters w and b. For $w_i > 0, b = 0$, nothing changes aside from the contraction of the shape itself. This is apparent of the elementary precalculus notion of warping and shifting. Hence, b handles the shifting range. For $\mathbf{w} = (2.0, -2.5, 2.5)$, the behaviour switch between successive neuron, that is, for specific sign shift between them.

Before moving on, let's see what is available for that specific configuration. This is illustrated clearly in Figure 9.11.



(a) First sigmoidal node. (b) Second sigmoidal node. (c) Third sigmoidal node.

Figure 9.11: Sequential configuration for x_1, x_2, x_3 and their functionals $\sigma_1, \sigma_2, \sigma_3$ of the same initializer with $\mathbf{w} = (2.0, -2.5, 2.5)$.

A fair lot of concepts and behaviours can be constructed using such sigmoidal chain. More specifically, it is the dependent categorization of specific encoding into a functional method. With such, specific configuration will yield different landing points on the graph. It also looks eerily like fuzzy logic, in one sense or another. However, what is the application for such presentation depends on what is the representation it is supposed to be indicted of, and how it can operate. In a stance, while the facility supports it, the action and life-time operation itself matters the most to the changing landscape.

An interesting phenomenon also happens with ReLU-like function, or rather, its precursor instead. For a `hardlim` functional,

$$\texttt{hardlims}[x] = \begin{cases} -1 & x < 0 \\ +1 & x \geq 0 \end{cases} \tag{9.10}$$

The end-result range would look like range-custom logic processor more than not, especially with chaining. Specifically, you can chain them as it is, however, their result range would be fairly small, with regard to the $(-1, +1)$ pair multiplier.

Again, what is used of it depends on the operation, configuration, and not simply the mathematical formulation of it. Interpretation matters in such regard because as it is, we are implicitly talking about the encoding of specific target to a language, in which case, we choose it to be an operational language process, just as neuron. In the subsequent sections, we will then discover that this scheme of emulating concepts and subject matter is perhaps, fairly effective of certain narrative of modelling.

9.6 Class $\mathcal{N}_1$ simplex

While we have been concerned of single-input neuron, and all structures that can be made of using such simple $\mathcal{N}_0$ class, it has its own inefficiencies. Depends on the circumstance, it might not be enough with it $(1, j, 1)$ configuration, setting aside j for extensible inner processing components. Depends on the encoding specified, the input might be limited, and certain

concept, or rather mechanism might not be able to be simulated. For example, the 3-pair interaction relation would not be able to be simulated if one can only be influenced by in-out potential, while it requires 2, perhaps 3 depends on self-looping. If we reduce it to standard neuron configuration $(1, 1, 1)$, it is then even more limited. One potential fix would be to 'fix bayonet' and free up i, thus giving the construction of $(i, j, 1)$. We call this **multivariate neuron**. If it is $(i, 1, 1)$, then we call it the **multivariate standard neuron**. All of such neurons then belong to the **class $\mathcal{N}_1$ neuron** simplex. .

$\mathcal{N}_1$ simplex

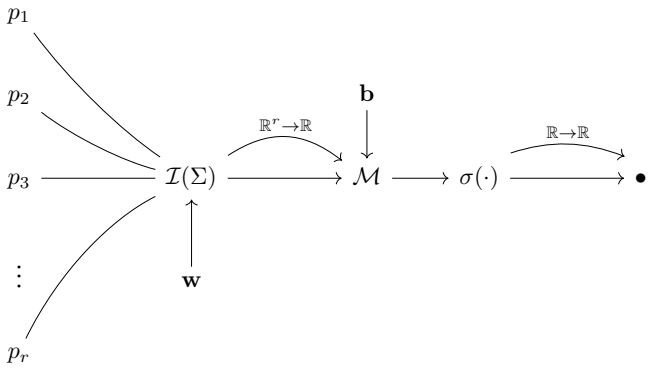


Figure 9.12: **Illustrative simplified commutative diagram schematic of r-input neuron process unit**. The specific field dimension transition for a standard neuron of $\mathbb{R}$ is denoted specifically between transitions.

Typically, for functionality, we have more than one input that the neuron can handle. A neuron with r amount of inputs is shown as:

$$n = \sum_{i=1}^{r} w_i p_i = w_1 p_1 + w_2 p_2 + w_3 p_3 + \dots w_r p_r + b = \mathbf{W}\mathbf{p} + b \tag{9.11}$$

$$a = \sigma\left(\sum_{i=1}^{n} w_i p_i\right) = \sigma(\mathbf{W}\mathbf{p} + b) \tag{9.12}$$

Here, we fix the input aggregator, Σ, because usually this portion would not be configured too much aside from the fairly intuitive additive-multiplicative conjunction control $w_i p_i$.

9.6.1 McCulloch-Pitts multivariate

Let's give examples to certain implementation of this type of neural structure for better or worse. For a single neuron, there exists many examples how people making their own configuration, plus properties. The most, easily oldest and nicely put neural structure is the **McCulloch-Pitts neuron**, by McCulloch and Pitts (1943). Given the series of inputs, that is, for $n_i \to \{x_1, \dots, x_n\}$, the internal mechanism M consists of (g, f) such that g aggregates n_i, which is the same as the typical construction, and f works as a serial, discrete logic transfer function. That is,

$$g(n_i) = g(x_1, \dots, x_n) = g(\mathbf{x}) = \sum_{i=1}^{n} x_i, \quad y = f(g(\mathbf{x})) = \begin{cases} 1 & g(\mathbf{x}) > \theta \\ 0 & \text{o.w.} \end{cases}, \theta > 0$$

Here, θ is called the threshold parameter, and is typically, under context, not *normalized* (we will see what does this mean in a much clearer context. For now, it means the entire operational

range is between $[0, 1]$). We notice that the operating space, in general, of the neural class, belongs to the predicate spaces of logical values, or at least the embedding of the logical space. Because this type of construction works by defining f at the end tail of the process by a discrete unit, it can be formalized to construct others logical-relevant neural unit, for example, the AND, OR, NAND, NOR and NOT unit. They are defined on all x_i taking values in $\{0, 1\}$, so that to mimic logical unit. Then, for AND, we have:

$$\mathsf{AND} = (\mathbf{p}, g, f_\wedge) \text{ s.c } f_\wedge(g(\mathbf{x})) = \begin{cases} 1 & \forall \mathrm{card}(\mathbf{x}) = g(\mathbf{x}) \\ 0 & \text{o.w} \end{cases} \tag{9.13}$$

Here, $\mathrm{card}(\mathbf{x})$ means the *cardinality* component-wise, which can also be written as just $|\mathbf{x}|$ as for the size. Hence, AND is true only when the cardinality is equal to the aggregated value, which is effectively the counting element. In such similar sense, we can then write OR as,

$$\mathsf{OR}(\mathbf{p}, g, f_\vee) \text{ s.c } f_\vee(g(\mathbf{x})) = \begin{cases} 1 & \text{o.w} \\ 0 & g(\mathbf{x}) \neq 0 \end{cases} \tag{9.14}$$

Since OR takes any value as long as it is not 'empty' for it to be true, we can use the aggregator instead. This is then perhaps one of the way that the aggregator form g is actually very useful in regard to identifier. Continue on, NOR and NOT can be defined just the same. But for them,

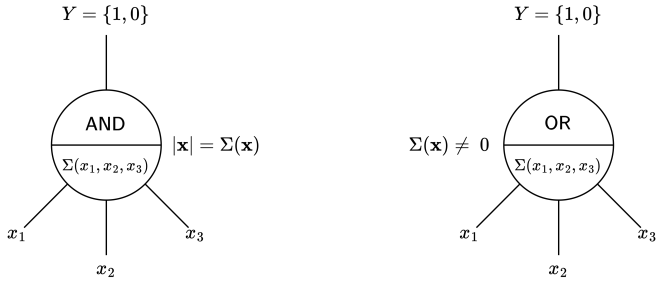


Figure 9.13: **Schematic of the AND and OR logical configuration.** The only change in their construction is that the criterion in the function is now different - from $|\mathbf{x}| = \Sigma(\mathbf{x})$ (which means all signals' sum must be equal to their absolute magnitude - in agree state), or $\Sigma(\mathbf{x}) \neq 0$ (as long as a single signal is active is enough).

$i = 1, 2$ is the specific logical configuration; and also notice that in some way or another, we can construct NOR from NOT. Hence, the operation NOT is:

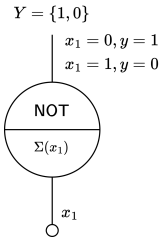


Figure 9.14: Schematic of the NOT logical configuration.

$$\mathsf{NOT}(\mathbf{x}, f_N) : f_N(x) = \begin{cases} 1 & x = 0 \\ 0 & x = 1 \end{cases} \tag{9.15}$$

Continuing, mathematically the NAND unit is formulated to be:

$$\text{NAND}(\mathbf{p}, g, f_{\not\wedge}) : f_{\not\wedge}(g(\mathbf{x})) = \begin{cases} 1 & \text{o.w.} \\ 0 & \forall \text{card}(\mathbf{x}) = g(\mathbf{x}) \end{cases} \tag{9.16}$$

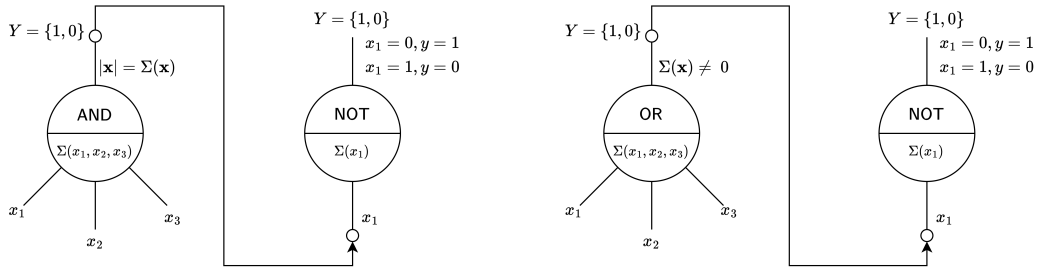


Figure 9.15: Schematic of the NAND and NOR logical configurations.

As we can see, operations and further units are multivariate, and also can be combined with each other to construct a system, using only the base units, which falls in line of the neuron classification we made due of previously. While being illustrated as such, we also notice that it looks just as *logical operations*, now just expressed as a processing unit. Taken from a representation-based aspect, we can consider the manifestation of logical operators in realization and existence similarly. In fact, some argue that within the McCulloch-Pitts standard basis (of logical units), it is a complete re-representation of **Boolean system** and computer structure, given additional ordering structures. So far, it means only of changing observation angle.

9.6.2 Minsky-Papert Perceptron

A step up from McCulloch unit is the **Perceptron** (Minsky and Papert [1988]) of Marvin Minsky, and partially both of Rosenblatt. The second example is the definition of a *perceptron* in Marvin Minsky's *Perceptron* book (Minsky, Papert, 1970). His definition, per contrast, might seem a bit difficult to follow since it is based on predicate logic. In this example, we will follow those predicates definition and construction to understand what is being tried.

Marvin Minsky is particularly interested in several of his *world representation*, tackling mundane problems that make use of the physical world, for example, geometrical classification and decision. Thereby, a lot of his constructions contains the geometrical arguments, just as we will soon see. His idea, however, perhaps also is based on the living state space that the processing unit would have to operate on, and thus the world that it sees, as much geometrical or numerical as it gets.

Let R be the space $\mathbb{R}^2$, and X be a geometric figure drawn on R^4. Let $\psi(X)$ be a two-valued function of X on R, usually think of as 0 or 1. By this embedding, we have created the *predicate* on which $\{0, 1\} \mapsto \{F, T\}$, or there are now variable statement whose truth or falsity depends on the choice of X. For example, the predicate

$$\psi_{\text{circle}}(X) = \begin{cases} 1 & X \text{ is a circle} \\ 0 & \text{o.w.} \end{cases}$$

[4]In his book, *Perceptron*, it is the main topic of geometrical learning that is mostly discussed, so a lot of examples and definitions might lean on the geometrical tasks side of things.

calculate the *membership criteria* for a circle construct, such that the geometrical shape is provided of X. Hence, we know that the predicate works more like classification and categorization units, as have been mentioned.

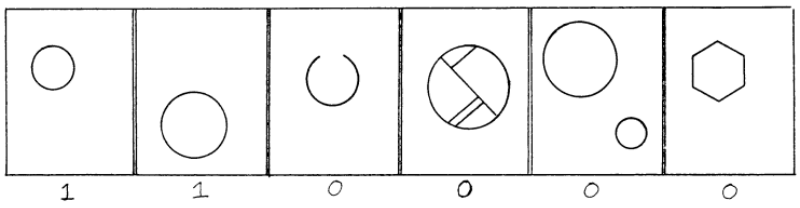


Figure 9.16: Results and values of the predicate ψ_{circle} on various geometrical shape. The detail of what gives the criterion is not mentioned, however implicitly defined to be naturally encoded. Taken from Minsky and Papert [1988].

For a convex figure, we have the according predicate ψ_{convex}, such is also

$$\psi_{\text{convex}} = \begin{cases} 1 & \text{X is a connected convex figure} \\ 0 & \text{otherwise} \end{cases} \tag{9.17}$$

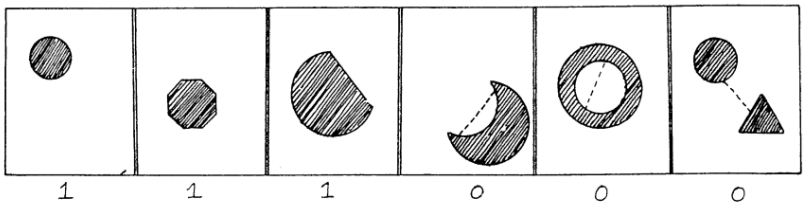


Figure 9.17: Results and values of the predicate ψ_{convex} on various geometrical shape. While still being implicitly defined, computationally this predicate takes more complexity than the circle predicate. Taken from Minsky and Papert [1988].

Additionally, we define some very much simpler predicates, in which for the sake of clearness, denoted by φ. So, for example, for $X \subset R$, the recognize predicate is that:

$$\varphi_p(X) = \begin{cases} 1 & p \in X \\ 0 & \text{o.w.} \end{cases}$$

or by extending this for an entire subset A:

$$\varphi A(X) = \begin{cases} 1 & A \subset X \\ 0 & \text{o.w.} \end{cases}$$

So, for a typical system, Marvin devised the scheme of working on a listing of **predicate**: functions and op-code that acts like logical predicate, for φ being those simpler predicate (the *ground* predicate), and ψ the constructed, more complex predicate. In the end, they would work together to devise specific structures from it.

The concept of 'locality'

There are still a few preliminaries before defining the Minsky-Papert Perceptron. Properties elementary to those predicate includes the concept of **locals**. We define it as original, which requires the definition of set convexity:

Definition 9.6.1 (Set-theoretic convexity on R). *A set X fails to be convex if and only if there exists three points such that q is in the line segment joining p and r, and:*

- $p \in X$.
- $q \notin X$.
- $r \in X$

Thus we can test for convexity by examining triplets of points. Because all the tests can be done independently, and the final decision made by such a logically simple procedure, unanimity of all tests, Minsky propose it as a kind of "local" conjunction.

We then define the notion of *conjunctively local*:

Definition 9.6.2 (Conjunctively local). *A predicate ψ is **conjunctively local** of order k if it can be computed by a set of Φ of predicates φ such that:*

- *Each φ depends upon no more than k points of R.*
- *The predicate $\psi(X)$ is evaluated by:*

$$\psi(X) = \begin{cases} 1 & \varphi(X) = 1 \forall \varphi \in \Phi \\ 0 & o.w. \end{cases} \tag{9.18}$$

In such sense, ψ_{convex} is conjunctively local of order 3.

Computed here refers to the analogous mention of *parallel computation*, in which is described by calculating $\psi(X)$ with the computation of *independent predicate* $\varphi_1(X), \varphi_2(X), \dots$ and then combine the result by means of a function Ω of n arguments to obtain ψ.

Now, the perceptron structure is defined as followed. First, we determine the linearity argument of the problem.

Definition 9.6.3 (Minsky, linearity of predicate). *Let $\Phi = \{\varphi_1, \dots, \varphi_n\}$ be a family of predicates. We will say that ψ is linear with respect to Φ if there exists a number θ and a set of numbers $\{\alpha(\varphi_1), \dots, \alpha(\varphi_n)\}$ such that*

$$\psi(X) = 1 \leftrightarrows \alpha(\varphi_1)\varphi_1(X) + \cdots + \alpha(\varphi_n)\varphi_n(X) > \theta \tag{9.19}$$

*In the same notion, θ is called the **characteristic threshold**, and α are called weights.*

The intuition of this is that each predicate Φ is supposed to provide some evidence about whether ψ is true for any figure X. A subject that can solve this problem is called a **perceptron**:

Definition 9.6.4 (Minsky, perceptron). *A **perceptron** is then a device capable of computing all predicates which are linear in some given set Φ of partial predicates.*

The family of perceptrons is numerous. Under such definition, there are plenty of

1. **Diameter-limited perceptrons**: For each φ in Φ, the set of points upon which φ depends is restricted not to exceed a certain *fixed diameter* in the plane.

2. **Order-restricted perceptron**: We say that a perceptron has order $\leq n$ if no member of Φ depends on more than n points.
3. **Gamba perceptron**: Each member of Φ may depend on all the points but must be a "linear threshold function" (that is, each member of Φ is itself computed by a perceptron of order 1, as defined.)
4. **Random perceptron**: These are the form most extensively studied by Rosenblatt: the φ are random Boolean functions.
5. **Bounded perceptron**: Φ contains an infinite number of φ, but all the parameters $\{\alpha\}$ lie in a finite set of numbers.

We have reviewed the concepts, the particular theories and development made in history, and the current usage of the treatment in the field of artificial intelligence and by extension to the modern machine learning theory. While there are much to be desired, it is imperative to note that we did not do a thorough overview, but the general theoretical interest.

With that, comes evaluation of the classical stance. Even though the classical theory can be seen as quite advanced and matured, in fact, it is not. Rather, it is endowed in the world of mathematics and empiricalism, for its advancements and development as present. Furthermore, which is one of the factor that inflicts the status of *naive* onto the classical treatment, is the overall no coherence structure - given how much and how rapid the changes have been - of the establishment, both practical-wise and theory-wise. Science itself, particularly, also has this problem. Other than that, there is also the fact that the current framework of theoretical artificial intelligence is not matured enough to take on various new questions and problems that are detrimental to their operation and mechanism. If one was to say, then they would be saying that while we dream of AGI (Artificial General Intelligence), we have no tools capable of reaching it aside from the dreams come true of wishes.

It is again, not superficially mentioned to downplay the impact and role of classical theory in the development and conceptualization of artificial constructs. It is, however, inadequate of the present, insufficient for the future, and inoperable in practice. We then, is inclined to accept and continue on toward a newer approach.

The following part of several chapters will be dedicated specifically to allow for the perhaps new formalism and theory to grow. In such, we will tackle those problems that the old scientist and researchers failed, those problems that was created from their insight (and failure), and lessons from successes. Fortunately for us, those are plenty, and we have to look not too further from the starting point to see the beacon where we will proceed.

Chapter 10. General principles

For much of the history, as well as the current conception of artificial intelligence designing, the design and implementation of AI has been influenced into the four main paradigm, as mentioned in Norvig's Artificial Intelligence. A modern approach (2011). Alternatively, it also follows the more general view toward either looking specifically at the structural components of any specimen of intelligence (human neurons and animal's), or *neurological approach*, also called *connectionism*; and those that follow the view of universal rationalism, capturing the essence behavioural constructs, and basing AI constructions on such universal justification, the *symbolic AI* approach. While those theories can be said to eventually be useful and perhaps, more groundbreaking than not, they are not so much desirable, as they are also very well-being specialized, unscalable, and was stuck in the paradigm of the more simpler tasks. In fact, whilst NLP, the chatbot that is prominent in the later date of the 2010s was hailed to be the epitome of development in AI (which indeed, perhaps it is), the issues with, for example, *common sense knowledge* is still present, and any attempt in fixing it often resulted in either the rough, uncertain and naive mimic of the 'real thing', or fails miserably. In one way or another of such, we can say that the current AI theory has no *framework*. Or at least a manageable one.

It is then of our interest to put up at least a view of what should be, and what can be said of artificial intelligence as a framework, a system for theories, and the principles in which one can say about the core essence of artificial intelligence, given an interpretation. In said following sections, we would allow ourselves to tread the thin line, and lay out the foundation of what to develop to a full-fledged artificial intelligence theory.

What should be expected of this outline? Not so much. The ultimate goal here is to formalize the principles, the working mechanism, the designing compass in which developments might ensue of a later date. Settle on matters that of said digression, will influence how the entire framework is formed, from the macroscopic view to the microscopic design elements, the point of view of the system, and more on the treatment of such system in certainty. Or probabilistically, which you can choose. In fact, the issues between determinism and uncertainty (probability interpretation) will also be touched upon, forgoes the huge amounts of works on two said interpretations ambiguous as it is possible to be.

10.1 Defining artificial intelligence

There are many ways to define artificial intelligence, either by the phenomenological one, or by the presumptuous, universal-assumed way of logicians in the old way. However, our definition, or at least mechanism of working on artificial intelligence, should be by then very different, as to not stumble upon the mistake of overestimating our predicate, and neglect the hardness of the problem itself.

What is then called AI, in our view? We treat AI as rather a more general system. First, we digress on the term **artificial**. We have talked about this in previous chapters, but, for the

moment, it is perhaps better to reformulate it. A construct, object, subject A is called *artificial*, if it fits the following rough conceptual definition.

Definition 10.1.1 (Artificial). *We say an object A is artificial only if it is not natural, or rather, it is intentionally created by meaning intentions, and not the general evolution of states, the natural interaction of the law of nature, or the natural transformation of biology.*

By such definition, artificial is then a quality to be separated from *natural intelligence* - of naturally made intelligent vessels or construct, existed because of natural, biological evolution and advancements by itself. A construct, then, is the main interest of our study. What about intelligence then, one might ask? The truth is, we don't know. We know roughly that intelligence is the exhibition of rationalism, the actions and demonstration of perhaps consciousness, of thinking. However, we have no way to define operationally, formally, and if not, conceptually sound of such aspect of intelligence. Let's assume we don't know. Then artificial intelligence is especially the realization and the discovery of intelligence itself. This is pretty much universal as we can get, out of the existing structure and predicament. Overall, the *universal argument* that would be utilized in developing our conceptual shell of artificial intelligence, shall not be too restrictive as for the universality of formal logic, for the world unable to be fully realized in formalism terms. We are now ready for a conceptual definition of artificial intelligence.

Definition 10.1.2 (Artificial intelligence). *A **construct** U, subjected to a system S is called **artificial intelligence** if it satisfies the condition of being artificial, whilst also satisfies a given criterion set of being autonomous, dynamic, and overall general. It is such that will give rise to intelligence construct.*

While it is dubious, we will further develop those points made above. However, we might as well want to justify the first point in all - the cogency of the intelligence criterion.

10.1.1 Intelligence criterion

We mentioned the notion of the criterion of intelligence. However, what should we define it? How should we know to even evaluate it, is a very hard question even that we did not (or unable to) fully realize yet, then what we want to do with it? This question is where a lot of things in the artificial intelligence research was based upon. For example, the (Total) Turing Test in which outlines possible outlook for intelligence, for capabilities that then defines the fields in which we are having nowadays, for example, computer vision for the capability of visual perception, natural language processing (NLP) for the capacity of language, and more. We also have various conceptual criterions in which people have been suggesting about the model of the intelligent being, for example, various set of criterions that outlines and includes even consciousness, some suggest behavioural conditions, some goes for the exhibition of *chain of thoughts*, and some even goes further than that, which is perhaps irrelevant aside from mentioned for example. Overall, it is perhaps a mess.

We still do not know what to come of criteria, or rather, in the quest of producing intelligence, we base ourselves onto it too much. As a species capable of intelligence and more sophisticated notion, we have the basis, and the advantage of being able to examine ourselves. By that, eventually, as the highest example of intelligent being, we use ourselves as standard, for examine, psychology, neurological behaviourism, neuroscience, applied onto the quest of going for artificial intelligence. Hence, there exists the total Turing test, and there exists the conflicts between various definitions and criterion of artificial intelligence. A mistake perhaps has been made, doubtfully so that one did not realize of such. While it is said that AI researcher has been working on, or at least researching on the general notion of artificial intelligence principles, it is, in fact, not so much of a principle, as we did not realize yet that what we are doing is

still the act of mimicking ourselves - creating a plane by replicating a bird. By phenomenologically absorb and construct architectures, models on the higher-level surface of what artificial intelligence constitute, the deeper construct is still non-existent. By copying the apparent capabilities of human and related intelligence being, biological rather than not, the core of which those behaviours occur, and facilitate the organs and observations made is perhaps, manifested. Ironically, while being too strict, wrongfully abhorrent to the fallacy of themselves, and too resistant to changes, symbolic approach got one of the right thing. If there exists intelligence, then it must be *universal by virtue*. That is, you cannot argue that alien from another universe is not intelligent, because they do not satisfy one of the criteria of the Turing test, just because such notion does not exist in such universe.

10.1.2 The should not of defining AI

Personally, I don't think we should, or we could define artificial intelligence, at least of this particular stage that we are in. Philosophically, being an armchair philosopher would not help in pursuing such notion, yet again because we are arguing on the basis of our own existence, and not the subject's matter viewpoint. There are problems related to it, also, of such that the mind and consciousness is arguably debatable in every given sense, of which no one seems to agree on the mundane notion that intelligence and consciousness come from chemical and the weird 'quantum effect' that would be then believed to be. And, truth to be told, we are not even endorsing such direction. In actuality, we don't even know what is intelligent, and also don't even know what can be of artificially made rather than matching mathematics.

On the flip side, computationally and neuroscientifically, the lack of formal treatment and overall encompassing knowledge conjunctions plague the construction and foremost attempt to do anything, simply because too many things have been said yet none can unify them together. Such is also to say different directions and different methodologies being conducted, yet they are so distinctively separated to be unable to conform one to another, despite them taking on the same object. Furthermore, there are a lot of assumptions given in computational theory, and the overall application thereof. As for anything, assumptions can be broken, and reinforced, for whatever it is being inconsistent as a virtue.

It is wise to remember that, for now with neuroscience being not advanced enough and in a perhaps different direction from what can be seen, while certainly for empirical science we can utilize neuroscience's knowledge, we should not take in the philosophical arguments and 'idea', including computational theory of mind. For empirical neuroscience, it is also not the fully-encompassing field that observe the brain from every angle, and observe consciousness of everything if ever, at least of the present. And, for the *philosophical* and idealistic view, only one thing can be said about such being "the lines on the map is made up".

10.1.3 Deferences in approach?

So, how should we approach this particular problem where one wants to create more than just logic in disguise, but also computational in nature, or else that no one can predict? Well, it is to generalize them . Simply speak, we do not go for the AI itself, but what can then constitute it. Granted, it is not similar to going blindfolded, or any kind of predisposition that protest the usage of the term and outlook on AI simply because perhaps of the impression above that we do not know what it is, hence no need for pursuing on such narrow road. But rather, to extract the fundamental facilities, concept, objects that in conjunction of knowledge that can be brought up or newly constructed, that is relevant of interest. And starting from ground zero with the modesty of assuming none and bias to minimum.

1. We do not create artificial intelligence. We create and investigate the *facilities*, the *theory* of which structures and objects can be utilized to create the framework that is not intelligent,

yet encompass more and might be able to give rise to intelligence. That is, for example, if we are to say that artificial intelligence can be constructed on a machine, then what can be said of the theory of machine? What is machine? What is the model for machine, and our understanding of such? What are machines that do not bear any similarity to the skewed vision of what is intelligent? And so is computer and its principle, and how we actually, formally, operate it? More so, we construct the subjects. The intelligent comes afterward. [1]

2. We do not, consequently, gauge intelligence as per metric or in terms of the skewed test that one can take, or metric that one get the R-value and so forth. It is rather somewhat baseless in such regard.
3. We stand on the assumption of *constructiveness* - there exists the *absolute minimum* of any given object that can be the basis for more advanced concept and construction. Not regarding such constructiveness often gives us headache in co-joining different constructs and ideas together, or simply interpreting machines and constructions of their characteristics and properties rather than just 'somewhat weird trick'. This is more in line of what would be considered a more computer-specific property, however, it is perhaps rather universal in consideration of modelling.
4. No model is perfect, if we ever create one.
5. We regard artificial intelligence to be consisted of two main things: the **facilities** that support its existence, and the **process** that support the subject matter that is examined. By this, then, instead of for example, thinking that intelligence only exhibits in subjects that have the learning property. Then we go on the reverse. What would happen to any given subject, that has something even remotely similar to learning, without considering the case of "if it can learn then it is intelligent" and of what capability? That is, to consider the action to be components itself, and not 'requirement'?

Overall, without stating more for out-of-scope reason (I do not intend for this note to go further into such discussion), we are trying to *construct and investigate* the formal foundational knowledge first, before even can utilize it to construct a generalized construct that encompass what is not intelligent, and what is then intelligent. Then, construct it with assumptions and constructiveness, plus the realization of both the building, and the lifetime of such building by itself. With this, perhaps we can then continue what was left behind, of what we might want to do and upgrade.

[1] It also can be considered loosely as to focus on the chaotic behaviours and the role of **percolation** plus **emergence**, rather than, well, specific construction. But a blend of both that and descriptive criteria is objectively better, rather than relying on the sole factor of randomness.

Chapter 11. The principle neural architecture

11.1 Neurons perspective

The formalization to be *atoms* comes from the biological inspiration of exactly the biological neurons in human and species' brain. However, the mechanism and how they operate is a bit more interesting.

For the biological more advanced equivalent, the brain consists of a very large number of highly connected elements, called *neurons*. The simplified structure of such neuron consists of three components: the *dendrite*, which is tree-like receptive network that carry signals (electrical) to the cell body; the *cell body* effectively processes these signals, usually as sums and threshold response; and the *axon*, a single long fibre that carries signal from the cell body to other neurons. The point of contact between an axon and a dendrite of another cell is called a synapse. The neural network, hence, is established by the synapses and various arrangements of the neuron.

This configuration of biological neuron is particularly powerful. Not only that they can process information allowing someone to write this text, but also the fact that of billions or neuron, they parallelly work at the same time. Furthermore, the intricacy lies further ahead than the above formation - the entire structure is of too much complexity, for the brain itself. However, a principle can be seen - they are contained of *building blocks* together, in which case is neuron. And **artificial networks** also take this approach.

Note 11.1.1. *From such, we can conclude that, we have to have some classification that requires the smallest component neuron to be the atoms that everything else is formed upon.*

One of the most fundamental thing of the structure of the *neural network formalism*, hence, is the notion of a *unit neuron*. Specifically, a neuron x is defined, and designed to be a discrete operating unit on its own. The word *operating* here means that it has all the facility requires for an input-output process - the most simple one includes the input receiver, the processing formulae, and the output transmitter.

The neural network formalism is best expressed by the following principle.

Theorem 11.1.1 (The fundamental theorem of neural formalism). *In the construction of an automated, artificial intelligence construct, the smallest singleton workable component must have the same size and dimension specified as the minimally defined neural structure, $\mathfrak{N}(\mathcal{N}, \mathcal{C})$.*

We begin by examining the model of a typical neuron. Note that, this is the simplest one, but also is the fundamental block. We have the following.

Definition 11.1.1 (Neuron). *Given the neural network formalism. Define a construct x. Then x is called a* **neuron** *if it belongs component-wise to the class $\mathcal{N}$ of all neurons, which is minimally expressed by $\mathcal{N}(n_i, n_o, M(\dots))$, where n_i is the input handler, n_o is the output handler, and $M(\dots)$ is the*

internal system.

From this, we then come up with the definition of the minimally defined neuron - the single most iteration of the above definition.

Definition 11.1.2 (Minimally defined neuron). *Given the neural network formalism. Then, for $x \in \mathcal{N}$, we call a neuron **minimally defined** if it belongs to the class $\mathcal{N}_0 \subset \mathcal{N}$ of $\mathcal{N}_0(n_i[\mathbf{w}], n_o[\mathbf{w}'], M(f, b))$ for $f : n_i[\mathbf{w}] \times b \to \mathcal{O}$ is the internal function of M. $\mathcal{O}$ is the domain of n_o in which it receives the value, and in cases, there exists no $\mathbf{w}'$ configured for the output channel.*

The notation $n_i[\mathbf{w}], n_o[\mathbf{w}']$ inherently indicate the notion of *control* of the neuron on the two gate of input and output. More specifically, a given neuron, aside from the upper-level expression of $\mathcal{N}(\dots)$, can also be realized by its *parameters*, or rather, its configuration of the neuron itself. By that, for $x \in \mathcal{N}$, the *configuration parameters* for the minimally defined neuron can be taken in the form $\mathcal{C}(\mathbf{w}_i, \mathbf{w}_o, \mathbf{w}_M)$, here we use instead the pairing notation. The letter $\mathbf{w}$ used here is historical by certain accounts: the original idea comes from the term **weight**, in which the neuron can conceptually influence the input, for example, the signal received or information received, by certain amount, either *downplay* it or *signify* it.

We clearly clarify the need for both $\mathcal{N}$ and $\mathcal{C}$ here. We know, that we want the neuron to be an *operating unit*. This means that for it to be fully realized, it needs to also operate, and hence, to be 'observed'. Hence, you need both the description of the parameters which defines it - taken the interpretation where each parameter and configuration is one specific building block on its own, then the block of all those parameters form the shape of the neuron. Similarly, the *operations* on such neuron assure the interaction and working mechanism of all those components together, inside the neuron, by itself. Hence, to fully, minimally express a minimalized neuron, you need to have both $\mathcal{N}$ and $\mathcal{C}$ as its minimality requirement. We then revise it to the following definition.

Definition 11.1.3 (Classical neuron class). *Given the neural network formalism. Then, we define a **neuron** A to be minimally defined if it is equivalent to any unit x of the class $\mathfrak{N}_m(\mathcal{N}, \mathcal{C})$, called a neuron class, where the 2-tuple expanded to $\mathcal{N}(n_i, n_o, M)$ and $\mathcal{C}(\mathbf{w}_i, \mathbf{w}_o, \mathbf{w}_M)$.*

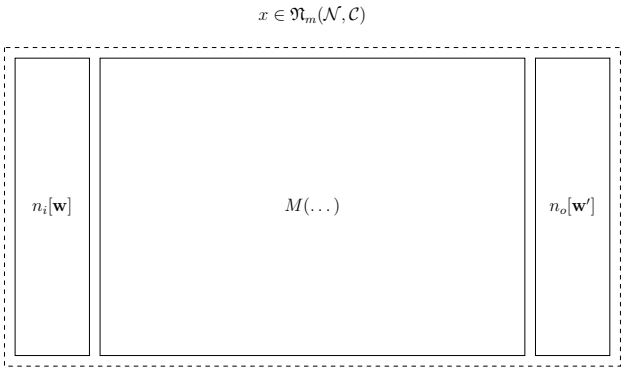


Figure 11.1: Minimal neuron structure

With this, we formalize the abstract neuron into two descriptions. Note that, however, the *minimally defined* neuron has a parameterized exposure protocol to the construction. In practice, for any neuron $a \in \mathfrak{N}$, the tuple $(\mathcal{N}, \mathcal{C})$ can be different, not to say complex. One of the strong

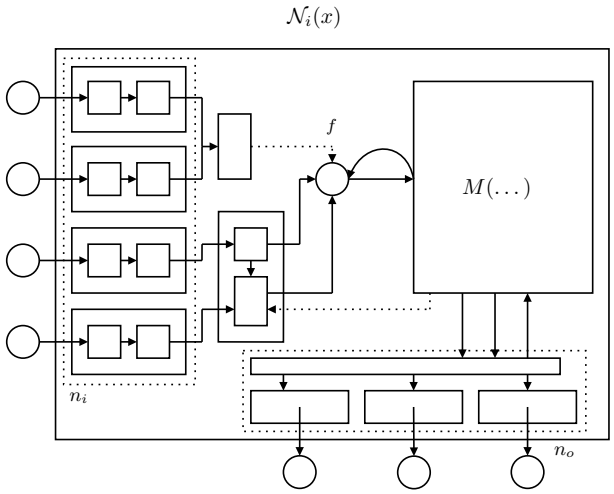


Figure 11.2: The compound structure construction. The same component can be seen, for n_i, n_o and M. Multiple consecutive components construct some components, and further outward. Also, we also reflect the complexity of $\mathcal{C}$ for a given architecture.

points of this construction of the neural class is that it is *recursive*. The following proposition might help aid in understanding why it is recursive.

Proposition 11.1.2 (Compound construction). *An abstract neuron A can be dissected into three most important compartments $(\mathrm{I}, \mathrm{O}, M)$, where I, O is respectively the in-out interface. Hence, a cluster of neurons $\{A_i\}_{i\leq n}$ for such $|\{A_i\}| \geq 3$ can be compartmentalized into a new neuron unit, that is, $\{A_i\} \in \mathfrak{N}$, if and only if $|\{A_i\}| \geq 3$.*

Proof. This mostly comes of as definition (2.1). Notice that for any structure of a neuron to be minimally defined, the tuple $\mathcal{N}$ must minimally contain (n_i, n_o, M). We want them to be discretely defined, then a cluster of neuron will have to have at least one neuron such that to satisfy the requirement of the 3-tuple component. If $n = |\{A_i\}| = 1$, it reduces to a singular neuron; for $n = 2$, at least one compartment do not have any component, hence minimally we need 3 neurons to effectively be arranged as a neuron unit. This can be illustrated using 11.2. □

Hence, we can recursively define neuron, either going up or down, if they satisfy such condition. However, usually, we will only go up, as we will have to define in detail the set $\{A_{M,i}\}$ of all possible *minimally defined* neuron. [1]

There are several assumptions and properties accompanied by this type of neuron structure that we would like to note.

1. This neuron structure works *sequentially*. A minimally defined neuron, such as in a practical setting, will have $\mathcal{C}$ to support sequential operation. What this means is present in the definition of the minimal neuron, such that $\mathcal{C} = (\mathbf{w}_i, \mathbf{w}_o, \mathbf{w}_M)$, and the operation follows $i \to M \to o$ for an input-process-output session.

2. By extension of the point above, the neuron order in a cluster will also be *fixed*. Usually, this follows a single dimension, or, for example, if there exists a singular span directional vector $\vec{dr}$ that indicates the direction of operation in the cluster.

[1] At this point, I am considering using *group theory* to aid such set compartment.

3. We suppose the *independence* of any components inside a neuron.

4. The neuron is expressed in a *parameterized manner*. That is, the behaviour of the neuron can be expressed, controlled, and designed by modifying and constructing a set of non-realized parameters. If this set of parameters is finite, we say that it is *closed parameterization*. If not—i.e., there can be infinitely many parameters—we call it *open parameterization*.

5. This neuron structure is called the *forgetful neuron*, simply because it works, in principle and in actuality, as a processing unit, in the minimally defined case. We will soon touch upon this definition for the detailed structure of the minimally defined neuron.

When we structure our neuron in this way, it is natural to ask whether the neural unit can be expressed similarly to a *finite automaton* or not. But aside from such an analogy, our minimal neuron is already very powerful. Certain clusters of such neurons, embedded with structure $\mathcal{L}$ of *layer-order clustering* (which can be thought of as a more formal notion for a layered neural network), can approximate any continuous representation scheme $c : \mathcal{X} \to [\cdot, \Sigma]$, given an arbitrary closed interval and to an arbitrary degree of accuracy. We often refer to this as the *universal approximation theorem* for the minimally defined neuron system, and it will serve as one of the focuses of the future analysis.

11.1.1 Analysis

The above construction of the fundamental neuron fundamental create the component model of the minimum processing unit which will be used in various collections of system of related mean. In mathematical modelling context, the neuron model is *descriptive*, meaning that it aims specifically not only for the input-output protocol, but also the unit neuron itself. To do this, we recall any given concrete, functional structure under mathematical formalism is given of the basis, on which abstraction and quantification (without physical realization). Hence, our model here aligns well with such mathematical formalism, in the sense that they end up with abstraction without the impromptu need to signify the inner physical realization - or existence itself. And doing this leaves us with formalizing the *minimally defined neuron* of the class $\mathfrak{N}(\mathcal{N}, \mathcal{C})$ into specific requirement.

The inspiration for the concept of a neuron is a biological one: it stems from the study of the brain or any given processing organs of species with specialized region to be called brain. Considering such, we found out that the brain mostly consists of the fundamental component called biological neuron and its supporting structures - other resource matters of the inner brain that helps to make it function. Without physical realization depends on many things, but most of the time, it is expressed by the proposition that is to remove the physical attachment from any given object of interest, giving it a more quantified look. For a neuron, it can be treated in such way by formalize the neuron component without its supporting structure, replacing electrical signal and other innate signalling mechanism with abstract 'data flow' and 'operations', without much worry about the actual physical realization of chemical reaction and channel - as we do not need such.

Back there, we all said of the neuron class. What does this mean? For starter, this section on the neuron is the relatively minimal construction we can get it onto. Recall that we say mathematical formalism is to put objects in quantifiable notions, and abstractions of the given subject; also, to express them in a language (or framework, depends on your choice of lexical sense) that support such. Hence, one of the first thing we would like to do, is to formalize the neuron into specific *description scheme*; that is, we would like to treat our objects - the neurons - specifically in certain descriptions that captures the essence of the neural unit, without much ambiguity, and an overhead view of the neuron. The tuple $(\mathcal{N}, \mathcal{C})$ did exactly that - it tells us

that the neuron is expressed by that separation of descriptions - in the language of mathematical quantification; for then, $\mathcal{N}$ stands for the quantitative *parameters* that characterize the resources, component scale, or the **mass** of a given neuron, which $\mathcal{C}$ stands for the operations that governs the neuron's internal dynamics. One feasible assumption here is that the neuron does not exhibit any of its operation, that is, interfering with the macroscopic world outside itself, aside from the input-output formalism. Those two descriptions, one governs the mass, one govern the internal mechanics that confounds the subject's behaviours, is what the mathematical description do - and it works very well as it is.

In such formalism, $\mathcal{N}$ is typically easier to specify than $\mathcal{C}$ - which is trivial since the internal mechanics is harder to define and construct, for there exists many relationship and connection between components in the same system. And, considering that for any working system, especially for an interactive one, which inherently dependent on the scale t, or given any reference point of operation to be referred upon, there can be either infinite configuration, or infinite type of ordering of the system itself. We may come to the analysis on *invariant structure* of the space of configuration, but otherwise, it is uncommon to have invariant structure presented in the majority inside a typical system, except fitting of certain criteria.

Overall, the neuron formalism of capturing them into a neural class also helps to figure out the *macroscopic configuration* and the *microscopic configuration* of any given neuron. Under a certain instance of the neuron class $\mathfrak{N}$, iterations of individual neurons can be vastly different, depends on their representation scheme $\mathcal{R}$ for individual sub-description of subcomponent - for example, if there exist the component called the *loss function*, then the macroscopic behaviour will only be configured up to certain accuracy, the rest is left for the detail microscopic setting. Hence, we can partition $\mathcal{N}$ into

$$\mathcal{N} \supset \mathcal{N}_{\mathcal{H}}(\dots) \times \mathcal{N}_{\mathcal{P}}(\dots), \quad \mathcal{N}_M \supset \mathcal{N}_{\mathcal{H},M}(n_i, n_o, M) \times \mathcal{N}_{\mathcal{P},M}(n_i, n_o, M)$$

of which here we use the word *hyperparameter* for $\mathcal{H}$ a bit different from its usage of literature, and $\mathcal{P}$ for microscopic *parameters*, under the proposition that the properties can be configured using parameters as abstraction. Similar notion goes for $\mathcal{C}$, though it is less apparent.

Overall, the minimally defined neuron, or what we can take as the **fundamental model of neural processing unit** works well, and provides us with the first iteration of the constructing brick for our larger, wider implementation. Though our aim is to advance this structure, first, we must take a look to the formalization and propositional-nation of the rudimentary concept of the neuron, even of the fundamental, minimal neuron. And this will happen at the end of this chapter. Until then, examining the maximal usage of the current structure will be more than enough as it aligns with historical literature's neuron and neural network formalism. Following sections will see how far we can go with this.

11.1.2 Why we call it minimal

It seems a bit weird why we call our construction on its own a *minimal construction*. To see this, though, we have to go back to the definition of a neuron, and the historical viewpoint on the constructing principle.

III. Drafts

A man who thinks all the time, has nothing to think except thoughts. Theoretical and philosophical, let the arm raises from the chair and dwindle up the reality that unfolds, for thoughts have no meaning if they are not manifested.

Chapter 12. Double Descent

12.1 Note

This is one of my first research - well, yeah, it started in early 2024, up until now. While there are many things to talk, overall, it is a fairly nice research topic. I wonder where would I go after this research, though, so that is definitely a problem, but for now, I think I will stick to this. When I first encountered this problem, I thought it would be plenty easy. I guess I was wrong, and it might take me more than two years to figure it out on my own at this rate, which is fairly troublesome after all.

This part of the manuscript contains two parts. The first part is concerned of the **draft and development notes** during research, and the second part (the one begin with the section abstract), is the **paper manuscript**.

12.2 Developing analysis

Okay, before running into the paper itself, uh, what the heck are we talking about? Based off our idea, we are targeting, well, **statistical learning theory** and the weird problem named **double descent**. Double descent has its notion from the dilemma of bias-variance tradeoff, a somewhat empirical hack to the problem of model selection. So, what is it?

12.2.1 Statistical learning theory

Roughly speaking, with machine learning being developed, there are petitions and pushes for the development of a further, more formal ground to explain and interpret the action of learning, and aside from heuristic, empirical design, a somewhat theoretical guarantee net for whatever we will be trying to do. This is where **learning theory** comes in, with specifically two disciplines - or rather two approaches: *Computational Learning Theory (CoLT)* which applies computational aspect to the learning problem, and the *Statistical Learning Theory (SLT)* which focus on the statistical interpretation of the learning problem. Albeit seems pretty different, they are actually very, well, interconnected, to the point that certain important notions can be almost the same.

Well, enough speaking by then, we will have the general idea and outward look of a more **formal system** of analysis. Let's hope we can deal with it.

12.2.2 Double descent

Double descent is tricky, in the sense that it is one of those phenomena that you perhaps would say to "break the theory", just like how lights and the Stern-Gerlach experiment broke classical physics' absolute view on determinism. Indeed, the thing that double descent broke is perhaps very important, in our opinion: *the bias-variance tradeoff*. What it means, and how it perhaps is important, we might be able to say.

While this sounds perhaps too good, it is indeed, one of the thing that I have to say about it as important.

Informally, bias-variance is an observation and perhaps analytical derivation of the behaviour of a learning system, within one consideration of the dynamic - complexity versus errors. With machine learning, for correctness reason of the *training session* conducted, error is a requirement. Bias-variance tradeoff connects this with the complexity of the model, by using the two proxies from statistical analysis, called bias and variance (statistical). Their result is pretty much as the following.

Theorem 12.2.1 (Bias-variance tradeoff). *For the expected loss of any given hypothesis h, the bias $\mathcal{B}(f, y)$ and variance $\mathcal{V}(f, y)$ is inversely proportional, that is, $\mathcal{B}(f, y) \propto \lambda^{-1}\mathcal{V}(f, y)$ for some proportionality λ that may or may not be constant. In the most general case possible, $\lambda = -1$ on the entire error range.*

We will have to define the notion of the bias $\mathcal{B}$ and the variance $\mathcal{V}$. In the classical derivation of Geman et al. [1992], it is defined in a pretty derived way. One can define the bias $\mathcal{B}$ as:

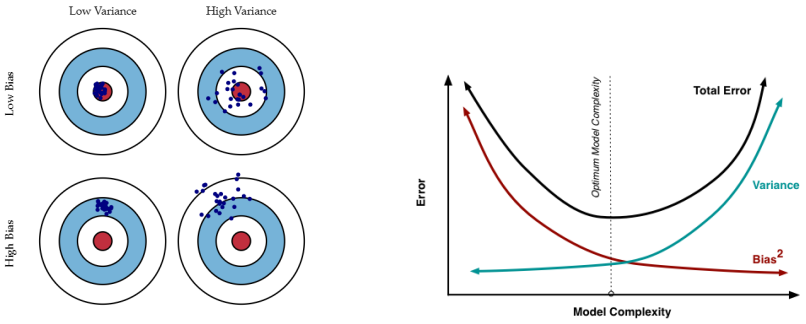
$$\mathcal{B}(f, y) = \underbrace{\{\mathbb{E}_{\mathcal{D}}[f(x; \mathcal{D})] - \mathbb{E}[y \mid x]\}^2}_{\text{bias}} \tag{12.1}$$

and the variance:

$$\mathcal{V}(f, y) = \underbrace{\mathbb{E}_{\mathcal{D}}\left\{(f(x; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[f(x; \mathcal{D})])^2\right\}}_{\text{variance}} \tag{12.2}$$

Of which both of them are derived from decomposing the supposed test error.

This results in the ultimate form of the bias-variance curve, which is usually portraited as the following famous inverse graph - there, you can also see the valley of optimality that is the ideal complexity - error ratio that is often wanted.



(a) (b)

Figure 12.1: (a) A typical example of bias-variance tradeoff in a statistical dataset. (b) When graphed into a continuous notion, we gain the complexity-error graph. Notice that it specifically goes for the *test error*, which fits - the representative problem of prediction.

So yes, it is indeed a problem. A very specific one. Especially since it is concerned with one of the main topic of machine learning - model selection. So, what can be said to solve the problem? Historically, not so much. But we will have to eventually take this pill and analyse it historically for now.

The first identification of the double descent phenomena dated back to the paper of Belkin - Belkin et al. [2019a], in which the title is literally "reconciling" modern machine learning practice and the bias-variance tradeoff. What is happening here? In modern machine learning

practice, or state-of-the-art developments, models are now bigger than ever. If to notice, we will see that currently models are inherently large, for example, a normal large language model will have from 900 millions (900M) to a few billions, for example 10 billions (10B) parameters. That is not taking into account the overall dynamics and structure of the model, which dictates the operating range and efficiency of the model itself. These model, based on the neural network architecture are somewhat trained to exactly fit (or interpolate) the data, almost certainly so that it turn from a prediction setting to an estimation setting. By statistical learning theory, this would be considered overfitting, and yet, they often obtain very high accuracy on test data. Is the test wrong, or it's just that we are missing something? No one knows for sure.

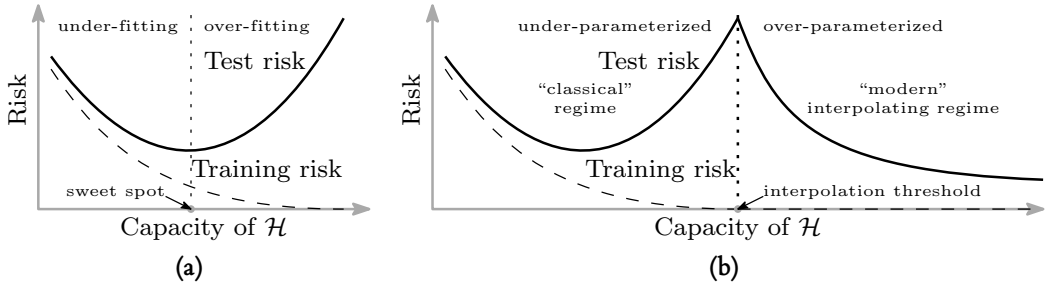


Figure 12.2: **Curves for training risk (dashed line) and test risk (solid line).** (a) The classical *U-shaped risk curve* arising from the bias-variance trade-off. (b) The *double descent risk curve*, which incorporates the U-shaped risk curve (i.e., the "classical" regime) together with the observed behaviour from using high capacity function classes (i.e., the "modern" interpolating regime), separated by the interpolation threshold. The predictors to the right of the interpolation threshold have zero training risk. Reproduced from Belkin et al. [2019a].

The main finding that Belkin found is a pattern for how the apparent performance on unseen data depends on model capacity and the mechanism underlying the emergence of double descent. When function class capacity is below the "interpolation threshold", learned predictors exhibit the classical U-shaped curve from Figure 12.2.

> *The bottom of the U is achieved at the sweet spot which balances the fit to the training data and the susceptibility to over-fitting: to the left of the sweet spot, predictors are under-fit, and immediately to the right, predictors are over-fit. When we increase the function class capacity high enough (e.g., by increasing the number of features or the size of the neural network architecture), the learned predictors achieve (near) perfect fits to the training data—i.e., interpolation. Although the learned predictors obtained at the interpolation threshold typically have high risk, we show that increasing the function class capacity beyond this point leads to decreasing risk, typically going below the risk achieved at the sweet spot in the "classical" regime.* (Belkin et al. [2019a])

Belkin tested on several contexts. The first one is with Random Fourier features, in which it is defined as followed.

Definition 12.2.1 (Random Fourier features). *The RFF model family $\mathcal{H}_N$ with N (complex-valued) parameters consists of functions $h\colon \mathbb{R}^d \to \mathbb{C}$ of the form*

$$h(x) = \sum_{k=1}^{N} a_k \phi(x; v_k) \quad \textit{where} \quad \phi(x; v) := e^{\sqrt{-1}\langle v, x\rangle},$$

and the vectors $v_1, \ldots, v_N$ are sampled independently of the standard normal distribution in $\mathbb{R}^d$. (We con-

sider $\mathcal{H}_N$ as a class of real-valued functions with $2N$ real-valued parameters by taking real and imaginary parts separately.)

This model is evaluated using the formulation of the classical statistical learning theory (refers to the respective chapter, if able), and is tested on the MNIST dataset. Their result? Pretty much the emergence property you observed of the figure above there, in which there exists double descent for both either zero-one loss or squared loss.

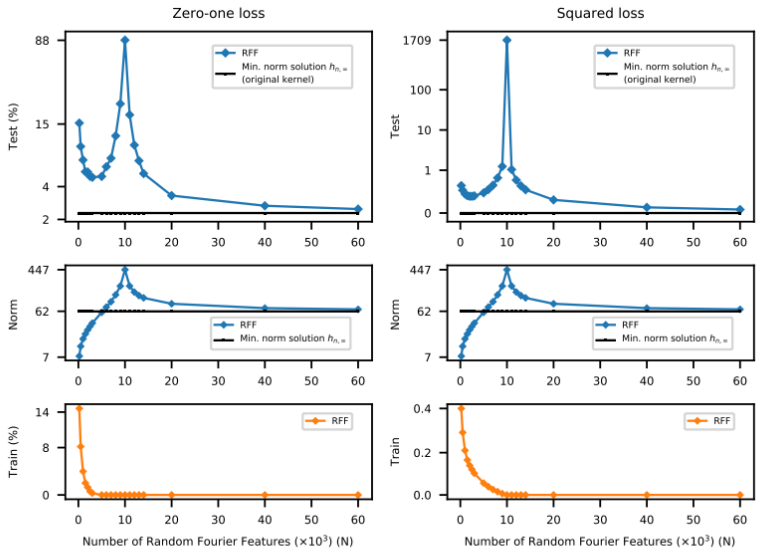


Figure 2: **Double descent risk curve for RFF model on MNIST.** Test risks (log scale), coefficient ℓ_2 norms (log scale), and training risks of the RFF model predictors $h_{n,N}$ learned on a subset of MNIST ($n = 10^4$, 10 classes). The interpolation threshold is achieved at $N = 10^4$.

Another models that they tested is the vanilla neural network, and decision trees and ensemble methods, in which gives you the following shape, similar to their double descent on RFF curve. Notice how a lot of the setting substitute the complexity to be the number of parameters or weight, in which case for decision tree, it is

$$\mathcal{C}(h_{DT}) = (d+1) \cdot H + (H+1) \cdot K$$

Which is still quite a lot, for H hidden units, dataset of K classes, and d is the dimension of the MNIST pool channel.

For now, perhaps we can see the apparent shape and identification of double descent. Empirically, double descent gives an interpolation point where then the test error 'goes downhill' very fast, almost better than the optimal point of the saddle in bias-variance tradeoff. Perhaps this is what happening inside large model, where absolute almost interpolation occurs and then the entire model is, well, so accurate that the test literally imploded. Or is it?

The second representative paper that analysed double descent, specifically in the setting of deep learning architecture, is Nakkiran et al. [2019]. Here, again, there is the issue with defining the model complexity of the model, and they offered us an understanding using the notion of the *effective model complexity*, $\text{EMC}_{\mathcal{D},\epsilon}(\mathcal{T})$ for a procedure $\mathcal{T}$ with parameters $\epsilon > 0$ and distribution $\mathcal{D}$.

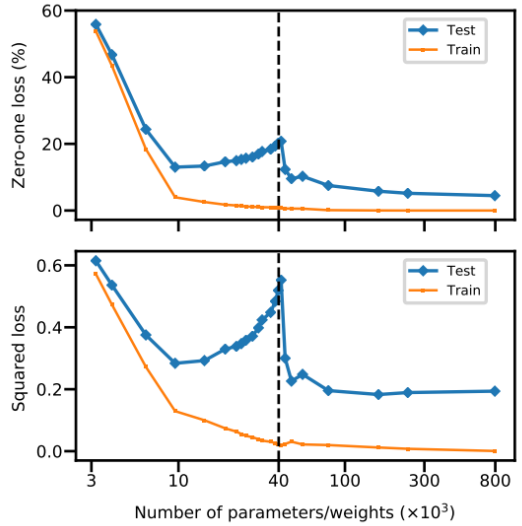


Figure 4: **Double descent risk curve for fully connected neural network on MNIST.** Training and test risks of network with a single layer of H hidden units, learned on a subset of MNIST ($n = 4 \cdot 10^3$, $d = 784$, $K = 10$ classes). The number of parameters is $(d+1) \cdot H + (H+1) \cdot K$. The interpolation threshold (black dotted line) is observed at $n \cdot K$.

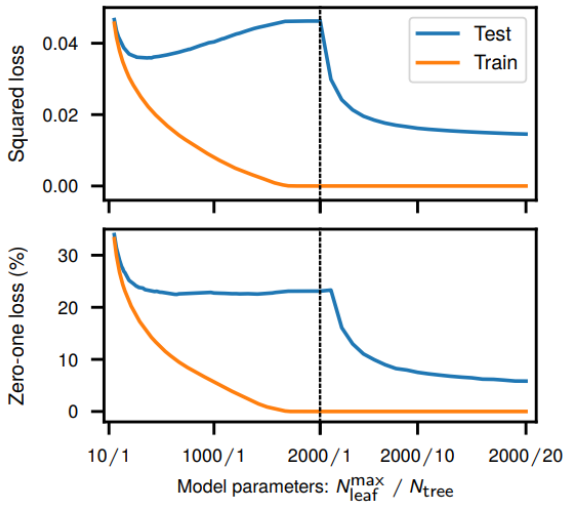


Figure 5: **Double descent risk curve for random forests on MNIST.** The double descent risk curve is observed for random forests with increasing model complexity trained on a subset of MNIST ($n = 10^4, 10$ classes). Its complexity is controlled by the number of trees N_{tree} and the maximum number of leaves allowed for each tree $N_{\text{leaf}}^{\text{max}}$.

Another prominent result to look at is Nakkiran et al. [2019], on the double descent of deep learning models. This is the first step toward identifying double descent to be perhaps, universal.

But first, we will have to see their beautiful illustration about deep learning's double descent. While being more complex and harder to analyse, they exhibit the same phenomena. Here, we call the region of changing dynamics from bias-variance to double descent relatively to *critical*

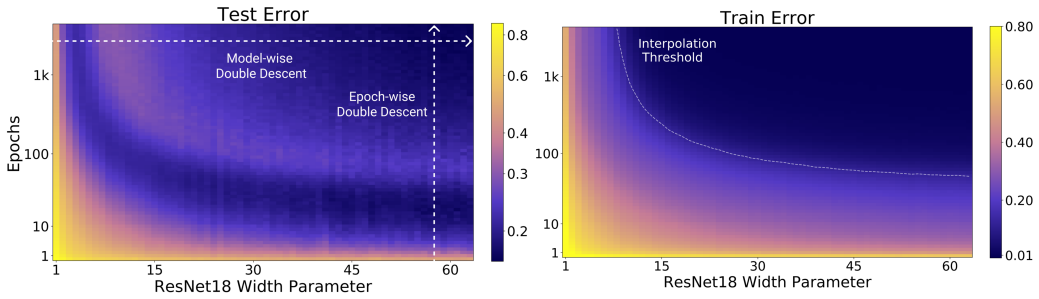


Figure 12.4: **Left**: Test error as a function of model size and train epochs. The horizontal line corresponds to model-wise double descent–varying model size while training for as long as possible. The vertical line corresponds to epoch-wise double descent, with test error undergoing double-descent as train time increases. **Right** Train error of the corresponding models. All models are Resnet18s trained on CIFAR-10 with 15% label noise, data-augmentation, and Adam for up to 4K epochs.

regime, and the place where the shift happens, is the *interpolation threshold*

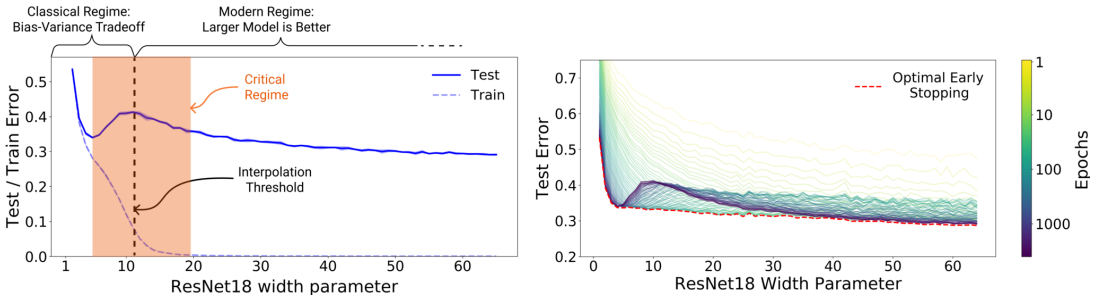


Figure 12.3: **Left**: Train and test error as a function of model size, for ResNet18s of varying width on CIFAR-10 with 15% label noise. **Right**: Test error, shown for varying train epochs. All models trained using Adam for 4K epochs. The largest model (width 64) corresponds to standard ResNet18. Reproduced from Nakkiran et al. [2019].

In such sense, we cannot seems to filter those figures or data received from the model by itself. Rather, because of such, the team above developed a hypothesis, of which relies on empirical observations at best.

They define *effective model complexity* of $\mathcal{T}$ (w.r.t. distribution $\mathcal{D}$) to be the maximum number of samples n on which $\mathcal{T}$ achieves on average ≈ 0 *training error*.

Definition 12.2.2 (Effective Model Complexity). *The* Effective Model Complexity *(EMC) of a training procedure $\mathcal{T}$, with respect to distribution $\mathcal{D}$ and parameter $\epsilon > 0$, is defined as:*

$$\mathrm{EMC}_{\mathcal{D},\epsilon}(\mathcal{T}) := \max\{n \mid \mathbb{E}_{S\sim\mathcal{D}^n}[\mathrm{Error}_S(\mathcal{T}(S))] \leq \epsilon\}$$

where $\mathrm{Error}_S(M)$ is the mean error of model M on train samples S.

Their main hypothesis can be informally stated as follows:

Hypothesis 12.2.1 (Generalized Double Descent hypothesis, informal). *For any natural data distribution $\mathcal{D}$, neural-network-based training procedure $\mathcal{T}$, and small $\epsilon > 0$, if we consider the task of predicting labels based on n samples from $\mathcal{D}$ then:*

Under-parameterized regime. *If* $\mathrm{EMC}_{\mathcal{D},\epsilon}(\mathcal{T})$ *is sufficiently smaller than* n*, any perturbation of* $\mathcal{T}$ *that increases its effective complexity will decrease the test error.*

Over-parameterized regime. *If* $\mathrm{EMC}_{\mathcal{D},\epsilon}(\mathcal{T})$ *is sufficiently larger than* n*, any perturbation of* $\mathcal{T}$ *that increases its effective complexity will decrease the test error.*

Critically parameterized regime. *If* $\mathrm{EMC}_{\mathcal{D},\epsilon}(\mathcal{T}) \approx n$*, then a perturbation of* $\mathcal{T}$ *that increases its effective complexity might decrease* **or increase** *the test error.*

As it turns out, even they themselves do not fully understand such open question, regardless of whatever was stated.

> Hypothesis 12.2.1 is stated informally as we are yet to fully understand the behaviour at the critically parameterized regime. For example, it is an open question what determines the width of the *critical interval*—the interval around the point in which $\mathrm{EMC}_{\mathcal{D},\epsilon}(\mathcal{T}) = n$, where increasing complexity might hurt performance. Specifically, we lack a formal definition for "sufficiently smaller" and "sufficiently larger". Another parameter that lacks principled understanding is the choice of ϵ. In our experiments, we use $\epsilon = 0.1$ heuristically.
>
> While in both the under-parameterized and over-parameterized regimes increasing complexity helps performance, the dynamics of the learning process seem very different in these two regimes. For example, in the over-parameterized regime, the gain comes not from classifying more training samples correctly but rather from increasing the confidence for the training samples that have already been correctly classified.

So overall, while actually pretty much reproducing results, they did not have a great step forward to the problem.

12.3 Issues

There are plenty issues with the above analysis.

12.3.1 The messiness of analysis

Truth to be told, the analysis is pretty much *kind of useless*. What I mean by that is despite all the research into it, not much progress has been made in understanding or trying to fix the problem that is there. Instead, it is the same fancy-styled interpretation and re-visualization of the concept by itself, while not asking the underlying question.

Even more so, it is the issue that *modern practices far outran theory, even classical one*. Furthermore, because of its clearly nature of being a multidisciplinary field, a lot of ideas are poured in from many fields out wide, such that it is very confusing, furthermore, very ambiguous and very informal when working with any of the formal system that analyze the learning system. This gives us the kind of disconnect that we perceived in the theoretical analysis, and the practical aspect of empirical designs and implementations.

12.3.2 The ambiguity of analysis

Analysis is often, very, very ambiguous. This also includes in the general setting of the experiment itself. Sometimes, even when the issue is with an unknown event like double descent, even the Gaussian white noise is not considered to be a potential factor in the fluctuation. Somehow, it is treated as trivial. Dissecting the system out, ideally there must exist a list of

factors and their importance weight according to how the influence the model, yet there seems to be no such insight about that.

Furthermore, a lot of terms, definitions are often hand-waived in papers or in discussions. This also led to the point that in Nakkiran et al. [2019], they have to somehow 'reinvent' another type of model complexity itself, albeit unsatisfying of a definition, is still a new kind of definition per insufficiently of such.

12.3.3 Stepping in the wrong direction

Perhaps we stepped in the wrong direction in most of the analysis and approaches to the problem? For example, are we taking the test error in a wrong way, or the dilemma is actually misunderstood? For example, while it is true that we are taking the test error, with the assumption of unseen data, what happens if the model is actually too flexible to the point that, it can capture almost perfectly the many cases that it is configured, with that substantially big amount of dataset getting in, guarantee it, because of the dataset, an absolute *concept capture* - that is, the true concept space and the observed space is very, very close together that there is simply not a *true unseen data* anymore?

Perhaps we should review all of it. From statistical learning theory to the interpretation of that itself. We might want to focus on such.

12.3.4 Main problems

For now, we can identify plenty problems what will have to be resolved during our works on bias-variance tradeoff.

- We need to make clear of the phenomena: bias-variance, double descent, the descent action.
- We need a discussion framework that is clear of context for previously arbitrary notion: inductive bias, model complexity, model flexibility, all kind of biases.
- Not only that, but we need also a framework to discuss the disconnect between empirical, practical approach, and a more formal technical ground of studies.
- Experimental setting and understanding. This will require us to develop a scheme to make sense of the complex system in practice, and to explain the empirical observation. Furthermore, we will also have to test the hypothesis that the test error is indeed, wrong of all accord in understanding of the statistical learning theory.

One of the typical problems observed when trying to solve double descent is exactly that - there are too many ambiguities, even in the sense and configuration of the experimental setting, as it adopts scientific research, yet is not well-rounded in their adoption.

12.3.5 Hypothesis

While it is not perfect, we have several hypothesizes for what to look out for, as well as the formulation needed to interpret certain phenomena and observations. This ranges from topics of re-evaluating the statistical learning theory, estimation and refactor of assumptions (or *inductive bias* when designing models), to different interpretations and representations that can better explain, model, or gives certain insights.

1. Stability of bias-variance measure is questionable, as it is in certain literature, loosely defined

by defining a 3-dimensional vector $\lambda = (\lambda_1, \lambda_2, \lambda_3)$ for such:

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}\left[d((f, \mathbb{E}[y \mid x])\right] &= \lambda_1 \text{Bias}(f, y) + \lambda_2 \text{Var}(f, y) + \lambda_3 \epsilon(\mathcal{D}) \\ &= \lambda_1 \underbrace{\{\mathbb{E}_{\mathcal{D}}[f(x; \mathcal{D})], \mathbb{E}[y \mid x]\}}_{\text{bias term}} + \lambda_2 \underbrace{\mathbb{E}_{\mathcal{D}}\left\{(f(x; \mathcal{D}), \mathbb{E}_{\mathcal{D}}[f(x; \mathcal{D})])\right\}}_{\text{variance term}} + \underbrace{\lambda_3 \epsilon}_{\text{irreducible error}}\end{aligned}$$

where $\epsilon(\mathcal{D})$ is the irreducible error of the system, depends (theoretically) on the intrinsic imperfection of the dataset $\mathcal{D}$. Assessment on stability of such decomposition is recommended.

2. We believe that the bias-variance decomposition is a poor estimation and measure for controlling, validating and choosing model from. Specifically, bias and variance term, as well as their supposed expression of decomposition in the standard loss measure of model effectiveness is not representative, nor expressive of the actual, underlying notion of *model complexity* $C(f)$ and its effectiveness. Furthermore, bias-variance decomposition is not straightforward of others type of loss functions, except only for some classes of loss function (loss on Bregman divergences and exponential class).

3. Inductive bias obviously plays a role, as mentioned by some literature [1], though we don't know exactly how the inductive bias can be formed to play any role in the central dynamic of the model operating process. Certain assumption, for example, the *convexity of the loss function* might actually affect the model as it is, and introduce unwanted patterns in the following process.

4. Random initialization of parameters, as well as the initial conditions of the model initialization process might also be taken into account. Obviously, this is not a new insight, and is rather a very old one, however, this approach still have varied potential for interpreting double descent. Scaling up/down this effect might result in new insight, as well as inspecting the uncertainty and diffusion patterns in large-scale model deployment, in which from an end-to-end perspective requires several space transformation operations in successions.

5. The PAC-learning framework, and its generaliy to *agnostic PAC-learning*, seperate the concept class c into two individual parts - either the assumption $c \in \mathcal{C}$ is true, or $\mathcal{C}$ is absent in the hypothesis, and therefore the inductive bias (assumption priori) is reduced of the underlying concept. However, most of the assumption and sepration comes around in the generalization is by assuming $\mathcal{X}$ is also random, and therefore unpredictable by the most random scenario. The most common justification of the generality of agnostic case, is by considering a non-unique nature of the input space's labels. However, usually, there is no assumption or any justification that species that the label, or the target class (of painter) must be *unique*. In fact, of the measure in binary classification, for $\mathcal{Y} = \{0, 1\}$, used in Mohri et al. [2012] for PAC-learning class, it is indeed a case of repeating label. The reason for this separation then is ambiguous and not totally founded. Though, it is reasonable to assume $\mathcal{X}$ is embedded with random probability, by separating the concept similar to what is illustrated in 8.2, there exists a distinct *internal relation*, for example, the probabilistic constraint of the input space $\mathcal{X}, p(X)$ unaccounted for c. However, grouping c by justifying properties of $\mathcal{X}$ is not so good. For example, consider c to be an automaton, either finite state automaton, or infinite state automaton (pushdown machine, Turing machine). Either c can be considered *deterministic* or, *non-deterministic* - for example, Probabilistic Turing machine (PTM), where state transition is justified of certain propabilistic measure embedded. Assuming so, we can

[1] Need to include some in here.

see that probability, if exists in such automaton, will only be defined upon the pair $(c, \mathcal{Y})$, and not $(\mathcal{X}, c)$ - the *ground space* (input) has no justification on c if specifically so.

6. The very ambiguous idea or treatment of statistical learning, as well as machine learning, is the idea of masked. Or, paraphrased, *hidden information* regarding the underlying process of estimation. Specifically, for example, we tend to assume things to not be able to grasp - either the probability $p_X(c)$ of the concept class, the distribution $\mathcal{D}$ specified to such input distribution, or the concept class $\mathcal{C}$, Markov properties, etc. We would like to formalize this notion as soon and as such in the most direct way.

7. The most outlandish one is that the test error is actually, to be seen as falsely interpreted by statistical learning theorists.

We cannot answer nor verify every single bit of the above hypothesis. However, by listing so gives us certain insight in which can be potentially refuted or agreed upon, it will base our understanding of the system in a more clear sight.

We have one more hypothesis on implementation.

- As we have seen with the disparity between SVM and other parameter-wise complexity measure, it is then to see that the measure of complexity is extremely lacking. It is also observed that many attempts has been made to categorize different types of complexity for a given model. It is then suggested, would there exists an algorithm or at least a behavioural control method that find the optimal, weighted complexity measure analysis? That is, assuming we have all the currently available complexity measure that we can find, then because we don't know what to consider "better" than the others, if we just simply weight them on, then tune the parameters of such tuning, then will we have a figuratively confident results that is applicable to a wide range of model selection procedure in such sense?

12.3.6 A rather simple solution

It can perhaps, be observed, that our entire description of the machine learning problem is informal and often not well-organized. Of course, in the face of progress little can be spared toward making things clean. After all, it is the pioneers who plunder and conquer distant lands. However, in the case of our machine learning research, it leaves a wide gap in the analysis and any potential solution at present, to some of the more meticulously difficult and obscure problems of the consequences of such rapid development, for example, our double descent. One of which is the consideration of the *system transparency problem*.

We may want to expand our notion of the mathematical modelling statements. A **mathematical modelling setting** consists of, consequently, three simplified components (S, Q, M). S contains the system and its object of interest, Q set up the problem aims, and M acts as information, assumptions, and constraints of the setting. Usually, even under such consideration, the problem setting is treated as an external source - an isolated system S of which contains the subject of interest *only*, and the model, on either extreme of being phenomenological or mechanistic. However, unknowingly, the aspect in which machine learning and subsequently deep learning, or any form of automating correlation known as **learning**, is doing things differently.

It is then theorized of a simple configuration. We extend S to encompass the entire hypothesis by itself. Though it is rather strange to do it in a formal sense, less of the cumbersome task for which it seems that we have already conducted. However, even if such line of argument is right, we have indeed, not configured our system accordingly. This is presentable in even the book Vel [2024], in which we treat the phenomenological model (neural network at foremost)

only as an outer method. This is then recognized, for the time being of the hypothesis, to be an insufficient conduct.

It turns out, however, we can think of our models to be subsequently realizing the internal mechanics by its own representation. As for the rather arbitrary definition of phenomenological and mechanistic model, it can be clearly seen: overall, those models differ only in their configurations of being either totally blind, or are given any sense of the underlying internal structure's interpretation of the events - factors and degree of freedoms, parameters and constants setting. Under multiple extreme assumptions, the notion of **transparency**, for if it is said that the problem is entirely exposed of its mechanics, or is totally unknown, is considered. Machine learning, and the learning action generally, would play as to mimic the mechanics by itself, regardless of internal mechanics transparency, by its own *representation* as we have been speaking of, with the condition that the acting space remains the same. This is partially guaranteed by the reduction of the observation space into an input-output channel observations. It is then, to be seen, if we can conjecture that any problem setting in transition to mathematical formalism, that is, including the act of observing, recording, and else that gives the dataset by itself, can be reduced nominally to various unknown, arbitrary, primitive parameters and of-no-knowledge models. With such, even differential equations can be somewhat observed, given the right representation space and enough encoding.

Why is this not so prominent then, in a more classical machine learning setting. It partially has something to do with the relative way both classical machine learning and deep learning are constructed, or rather, the structural constraints between the two. While indeed, such problems would be encountered in classical ML, the design of such models are rather intrinsically different, in the sense that they are restricted and of not so fluid form. Furthermore, they are also specialized, for example, in the original text of Vapnik [1999], to works on pattern recognitions, regression problems or density estimation. Such tasks are fairly limited and constrained of their setting and requirements, hence, nominally, the class of "pretty good and extraordinarily useful" model structures are also constrained small enough for the representation to be in classical form; and, given that the learning setting is often not so rigorous as many aspects were not linked together, it puts a fair strain onto such interpretation.

This is where we would theorize our idea. Rather than just as classical machine learning indicted, from certain observations of *neural network*, it can be said that the following conjecture will be stated.

Conjecture 12.3.1. *The learning problem can be considered a one-way description-representation learning problem between the concept structure $\mathcal{C}$ and the hypothesis $\mathcal{H}$.* [a]

[a] The structure and the underlying observations, plus information exchanges, are typically formed into input-output interpretation.

Conjecture 12.3.2. *Under deep learning structure, one assumes that every possible configuration and representation of certain concept $c \in \mathcal{C}$ can be decomposed to the standard representation components.*

It is then to see if such conjecture can be observed, or at least proved heuristically. Aside from this conceptual perspective, we also note that in most of its conception, the testing and theoretical working of the theory indeed also is based on the notion of transparency - for example, Agnostic PAC-learning where we assume no knowledge is possible about the class $\mathcal{C}$ of all concept class conceivable of the problem. Such assumption is the transparency window that we are talking about. Rather than doing experiments without considering such, it is then wised to perhaps analyze, how, we can reconstruct the setting, using an end-to-end (only the supervisor or the designer - we - know) phenomenological$\rightarrow$ mechanistic learning behaviours occurs.

This also perhaps will outline a controlled environment in testing out information theory and interpretation of KL-divergence.

12.4 The perspective of modelling theory

A normal configuration of a mathematical modelling system, in which the language of mathematic is applied, often consists of three components: The system S in which all the objects, the landscape, the resources, the parts that is considered lives in; The question, or the **objective** that is of interest, this is also called the setting, or the scenario in which the system is placed in, of a specific notion; and the statements M of which assumptions, statements, condition, restrictions are put on for the system, though not necessary so for the objective. Then, for every model and the construct that we want to make, we have to then specify those first, in the most rigorous manner possible, to disparage the ambiguity. This starts by, as the name of the section called, specifying and constructing the system S.

Before we continue, it is to note that these are some of the more **informal treatment** of the research direction. Later on, we will have to find something, or some mathematics to tackle the problem laid down here.

In a typical system S of machine learning, we assume there exists the following objects:

- The **concept object**, which is implicitly the object of study. This object has its own mathematical construct.
- The **hypothesis object**, which implicitly is the object in which objective is to study the concept object. Similarly, it has its own mathematical construct.
- The **evolutionary object**, which it aims to correct the hypothesis object to align with the concept object.

Those two first objects have their own mathematical construct, and live in a space in which we call the ground space, of which all their operation and behaviours will be observed. Here, we have the consideration of their *transparency*, that is, the amount of information we know about the object. Or rather, it is called the relative *priori knowledge* about each of them. Using this, we have **absolute black-box** - where the object is believed to only be an input-output process, without any priori about what is the underlying structure, ever. On the other hand, we have **total transparency**, in which the internal mechanics of the system is totally clear with respect to the object participated in the dynamic of the object. Before that, let's look at the structures of the two first important object.

12.4.1 Structures of object

Each object in this system would then have to have a very important component: its **representation**. This is trivial of interest, because as we said, mathematical modelling convert something into the mathematical formalism and language. In the case of learning a mathematical object by itself, there are also multiple interpretations, definitions, and representation of different objects in different cases, however, most of the time, we will observe a lot of object with very explicit representation of a class of itself. This relies on, for the reason to specify representation, the notion of **operable object** by itself. An object is *operable* if its internal structure and its nested substructures is arranged, organized, and represented such that there exists operators on which is specified as the action of the object. For example, a function is operable since its substructure includes sub-objects that are the blobs of parameters and the free-variable parameter b (as people always call it such), of which the operations and relations between them constructs to be an operable object by virtue of its observable being produced. For a graph, however, it is inherently static - its structure does not provide us with any actionable quality or operation.

Only by either a walk or a trail (which essentially the same thing). To provide or extract the object's structure in an effective sense, which can be said of to not just copy the entire graph, we need to have actions and operations that defines on the graph, to extract meaningful data from it. This is perhaps relevant of the neighbourhood aggregation in a GNN (graph neural network), as it too, needs to extract data by using neighbourhood properties. The graph itself is not equipped of any operable structure. So there is it.

The next component about an object in our consideration, is the utilization of the representation scheme to then make the class of all descriptions by itself. For example, a representation scheme consists of multiple neuron components - the parameters of each neuron, the path given, et cetera, will only be effective and useful if it is constructed in a structure by itself, that is, in a neural network architecture where neurons are composed of some of those parameters or resources given, with their value based on the field specified in the notation of a numerically-realized representation scheme. This will mostly be the more important notion of the description of the object, since it is where certain criteria, conditions, restrictions and overall description of the object is specified, using the restricted language of the representation scheme. We will see how this is done.

Representation

Then, what can be said about the representation of the hypothesis and the concept objects? Okay, this is harder than I thought. There are many ways to represent a structure or a concept of interest, yet, we will stick to this one, maybe. We will stick to the one that is most familiar: remember in mathematics where we have *sets* embedded with structures? Yeah, about that... we will do the same thing. The representation that is used to represent our structure will be called the **representation scheme**.

First, we have the following assumption, which will fall to the set of statements M, that is as followed.

Assumption 12.4.1.1. *The hypothesis and the concept are both represented in the same representation scheme.*

If the concept is not of the same representation scheme as the hypothesis, then there must be a **representation encoder** to handle such task. Depends on the restriction of the representation, we can figure out the total loss of information, generality of information in-between such encoding. We define the concept of the representation scheme as followed.

Definition 12.4.1 (Representation scheme). *For a system and its object, an object's* ***representation scheme*** $\boldsymbol{\mathcal{R}}$ *is a function* $\boldsymbol{\mathcal{R}} : (\Sigma \cup \mathcal{G})^* \to \mathcal{O}$*, where* Σ *is the operator alphabet, or relational structure,* $\mathcal{G}$ *are the components, and* $\mathcal{O}$ *is the object.* $(\Sigma \cup \mathcal{O})^*$ *is the coupling of* ***all*** *configuration of the components and the operators.*

In case where scalar numbers are presented, for example, in the axis-aligned rectangle case, we will have: $\boldsymbol{\mathcal{R}}^2_{\square} : (\Sigma \cup \mathcal{G} \cup \mathbb{F})^* \to$ Rec of all rectangle using this representation scheme, so

$$\boldsymbol{\mathcal{R}}_{\square} = \begin{cases} \Sigma = \{\leq, \wedge\} \\ \mathcal{G} = \{l_c, p_c, a\} \\ \mathbb{F} = \mathbb{R} \end{cases} \tag{12.3}$$

For the formula of all concepts c being axis-aligned rectangle as:

$$\sigma = \{c\} = \boldsymbol{\mathcal{R}}^2 = \{a(x_a, y_a) \mid l_c^{(1)} \leq x_a \leq p_c^{(1)} \wedge l_c^{(2} \leq x_y \leq p_c^{(2)}\} \tag{12.4}$$

Generally, this is called the *numerical representation scheme*, in which it is supported by a field (we do not care much about the dimension of the field, as long as we can decompose it to discrete scalar taking values), that is, $\boldsymbol{\mathcal{R}} : (\Sigma \cup \mathcal{G} \cup \mathcal{F}) \to \mathcal{O}$. Any representation specification $\sigma \in (\Sigma \cup \mathcal{G} \cup \mathcal{F})$ such that $\boldsymbol{\mathcal{R}}(\sigma) = c$ is a representation of c on $\boldsymbol{\mathcal{R}}$, denoted σ_c, and the set of all such specification $R_c = \{\sigma_c\}$ is called the **representation space** of c on $\boldsymbol{\mathcal{R}}$.

The set of all hypotheses h that is specified by certain representation scheme $\boldsymbol{\mathcal{R}}_h$ is called the **hypothesis class** $\mathcal{H}$, and $\boldsymbol{\mathcal{R}}_h$ is the **hypothesis class representation**. Similarly, for a concept c the set of all concepts that are represented by $\boldsymbol{\mathcal{R}}_c$ is called the **concept class** $\mathcal{C}$ for the **concept representation scheme** $\boldsymbol{\mathcal{R}}_c$.

The necessity of the description of a field is rather natural, especially since we are working on a numerical encoding. Generally speaking, the computer at large represents a very complex binary encoding of numerical logical operations, though via abstractions, most of them are 'cancelled out' of the fundamental details. We can almost make a *representation space* just similar to how we define vector space, though, it is more or less not so effective as it can.

Using this, we can specify **a lot** of object class. The first one though, we can specify the representation scheme of an input-output model. Now, for this type of definition, then the class of all **linear function representation** can be designed as

$$\boldsymbol{\mathcal{R}}_L^n = \left\{ \{x_1, \dots, x_n\}, y \middle| \sum_{i=1}^{n} w_i x_i + b \wedge w_i, x_i, b \in \mathbb{R} \right\} \tag{12.5}$$

From this, we can notice that there exists the notion of *size* for the object class. In one way or another, we have abstracted of the class of all objects that can be specified in such a way that their representation falls into the range of such representation class. Then, we might want to consider the concept of a **representation complexity**, and the size of the class, denoted $size(\boldsymbol{\mathcal{R}})$. Do note that this is not defining the operational complexity, but simply the structure by itself. Which is why we might want to have the definition of an object in such system that we are considering.

Two representations $\boldsymbol{\mathcal{R}}_A$ and $\mathcal{R}_B$, they are said to be **equivalent** if one can convert objects from the first representation to the second one, and vice versa. Then two representations are said to be **equal** if their size is the same, assume that the representation scheme is equivalent. This prompts us to define the notion of the size of the representation, but before that, a general insight will be the follow through - you have to be able to reduce one to the others, and reverse. By then, essentially, *you cannot compare apple to orange*, that is. Generally, the size of a representation class is defined the smallest representation of the object in the underlying representation. So, for your example of the linear function, then

$$size(c) = \min_{\sigma \in R_c} \{size(\sigma)\} \tag{12.6}$$

which again, prompt us *again*, to figure out the notion for $size(\sigma)$. Before then, and defining the object's descriptions, let's try to construct a few more 'mathematical construct' of the same type. A subtle remark can be made here, that the representation only, again, specify the parameters used to represent it, and the string accompanied by such representation. The overall total repetition, for example, of a certain variable, and operation succeedingly, is totally irrelevant. I just get the formula as a shorthand for that scheme up there, as it is. Which again, means that to specify it, you need both the components and how it is connected. Abstractly speaking, and generally speaking.

Note 12.4.1. *It is sufficient to note here that the representation scheme has its complexity more than just the count of all its subsequent operations. for example, if the class of the multiplication operation is included, it will, of course, drastically change the dynamic and presentation of the encoding possible for the description. If so, then, we have a very hard time in the future to try "ranking" these type of operators together. Furthermore, while not of the representation complexity itself, the combination of the components in the description also decides another type of complexity, which for now maybe we will call as expression complexity.*

Do note that even though the representation scheme is inherently denoted similar to an end-to-end formula would look like, it is not the case, usually. The representation scheme is not a fully mathematical-defined object, for such case then the object itself would have no meaning at all. In such sense, one can define a representation scheme that contains intricate operating structures instead, for example, a self-loop network with various potential-like mechanism and affectants.

Overall, the role of the representation is to specify the arbitrary language of working. While we are not taking the more diluted problem between the translation from physical, real setting to the mathematical world, and restrict ourselves to the part of which arbitrary mathematical settings take place, it is still of importance of the *language of mechanism* that underlies the structure of certain system pertaining to actions and modelling [2]. By doing so, it encompasses a variety of mathematical descriptions in which expresses a very similar aspect of the mathematical language, the configuration and largely operative nature of an object-dependent setting. While doing so, it also makes way for the more in-focus notion of *descriptions* to be defined, and such way encompass a lot of structures, for example, the binary class descriptions of a typical computer, for example. We shall examine this later, for its implications large.

Hence, from the definition and consideration of the representation space, we can at least organize and capture the:

1. Representation language: What type of language are considered and how to specify its component, at least in the operational sense of what we are dealing with.
2. Representation complexity: While the notion of *complexity* is inherently complex, it is still applicable of current knowledge to then approach the problem of *representation complexity*. This can be done by *representation infimum* and *representation supremum*, both of which will bounds the representation size as much as it could, except for representation anomalies.
3. The relative dynamic between *representation complexity* and *representation computational complexity*: In the subject of efficiency in working with representation, it also between computational cost and the richness of the representation space that is of interest. In which, we should encounter, and try to process.

We would likely want to give a pretty much different interpretation thereof, on one of the more primitive notion in mathematical analysis, which also looks just as similar in idea as the representation of subject to be: the **Stone-Weierstrass Approximation Theorem**, which states that for $f \in C([a,b],\mathbb{R})$, then there is a sequence of polynomials $p_n(x)$ that conveys uniformly to $f(x)$ to $[a,b]$, essentially represents the complex subject. We would ultimately, in actuality,

[2] This way of doing things are particularly similar to a philosophical mathematics's way of defining mathematics through axiomatizations: first they also have to define the symbols and the symbolism expression, as well as the descriptions of mathematical objects through such axiomatization and symbol conventions. This is why some branches of mathematics, where they pertain on a general notion like *set* and *logic* formalism, is used far wider than their own - the language of something like *set theory*, *category theory*, and *logic* are much more useful in other fields or others mathematical objects in which its general framework can be applied to express distinct and more specialized, descriptive objects.

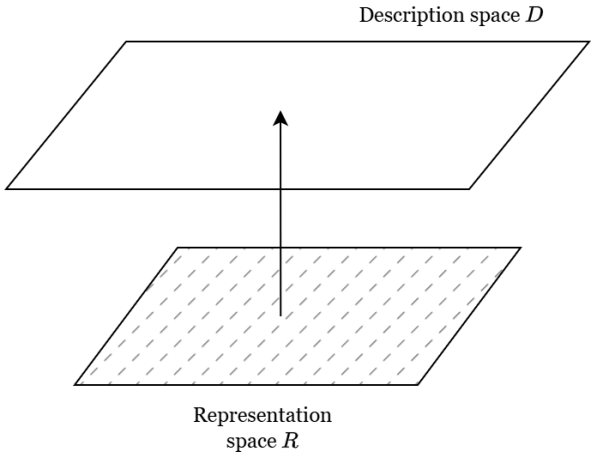


Figure 12.5: The representative order of representation and description. As of the name implied, in transition to a mathematical formalism and language, there must then exist a representation to each and every element of certain subject. The process of doing is this called *external encoding*, and is true also between portion of mathematical-encoded system to each other, if they are distinct. The reverse act is called again, *decoding*, and between mathematical subjects to each other might as well be called *internal encoding*, with respect to the mathematical language.

assume this to be true in our connection to practical problems and more complex subjects non-mathematically, and if this is not particularly true, then we are in plenty of trouble.

Description

While the representation space is helpful, it does not capture the entirety of the analysis as a whole. More specifically, it only specify at most the structure of the bricks laying the object, not the object itself.

Using such alphabet, we then can construct various structures of interest.

12.5 The perspective of modified learning setting

12.6 Bias-variance: A history

While a lot is known about bias-variance tradeoff, its history is pretty much indeterminate in the lens of a contriving analysis. Rather, controversies and theoretical development that have led to the appearance of double descent, and other phenomena like grokking and else, is of particular chaotic picture in the wider scene of the learning theory, especially of modern time learning. The original definition of bias-variance tradeoff by Geman et al. [1992] is first constructed using the means-square error, which is regarded as a normal measure in the real encoding space. Their approach is to justify bias-variance via decomposition of the loss function ℓ, for such to find an alternative reasonable form of such loss landscape. We will eventually notice a lot of the assumptions, and the breakdown of the nominal setting hidden inside this portion of subject matter.

Suppose of a regression problem to construct a hypothesis function $f(x)$ from $(x_1, y_1, \dots, x_N, y_N)$ for the purpose of generalization - that is, predicting unseen variational values for different pair $(x_j, ?)$ such that $? = y_j + \epsilon$ for a conceivable implicit error. To be explicit about the relation of this problem, or f on the given data $\mathcal{D} = \{(x_i, y_i) \mid i \leq N\}$, denote $f(x; \mathcal{D})$ instead of f, the natural mean-square measure as a predictor is:

$$\mathcal{M}(f, y) = \mathbb{E}\left[((y - f(x; \mathcal{D})))^2 \mid x, \mathcal{D}\right] \tag{12.7}$$

for $\mathbb{E}[\cdot]$ the expectation wrt to a distribution P. Decomposing the right-hand side, we have:

$$\mathcal{M}(f, y) = \mathbb{E}\left[((y - f(x; \mathcal{D})))^2 \mid x, \mathcal{D}\right] = \mathbb{E}\left[(y - \mathbb{E}[y \mid x])^2 \mid x, \mathcal{D}\right] + (f(x; \mathcal{D}) - \mathbb{E}[y \mid x])^2 \tag{12.8}$$

Here, $\mathbb{E}\left[(y - \mathbb{E}[y \mid x])^2 \mid x, \mathcal{D}\right]$ does not depend on $\mathcal{D}$, but simply the statistical variance of y given x. The term $(f(x; \mathcal{D}) - \mathbb{E}[y \mid x])^2$ is considered a natural measure of effectiveness on $\mathbb{R}^n$ as a singular predictor of y. Now, for $\mathbb{E}_{\mathcal{D}}\left[(f(x; \mathcal{D}) - \mathbb{E}[y \mid x])^2\right]$ which depends on the training set $\mathcal{D}$ in its computation, is decomposed into the form of *bias-variance decomposition* terms, by derivation:

$$\mathbb{E}_{\mathcal{D}}\left[(f(x; \mathcal{D}) - \mathbb{E}[y \mid x])^2\right] = \underbrace{\{\mathbb{E}_{\mathcal{D}}[f(x; \mathcal{D})] - \mathbb{E}[y \mid x]\}}_{\text{bias term}} + \underbrace{\mathbb{E}_{\mathcal{D}}\left\{(f(x; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[f(x; \mathcal{D})])^2\right\}}_{\text{variance term}} \tag{12.9}$$

We summarize this in the following statement.

Theorem 12.6.1 (Bias-variance tradeoff). *Suppsoe the model $f(x; \mathcal{D})$ for the data $\mathcal{D} = (x_i, y_i)$ and its parameter x is defined. For y_i of the target concept's responses y, and consider a regression problem with the loss measure $\mathcal{M}(f, y)$ of mean squared risk, the following statement is true:*

$$\mathbb{E}\big[\mathcal{M}(f, y)\big] = \mathcal{B}(f, y) + \mathcal{V}(f, y) + \mathbb{E}\Big[\mathbb{E}\left[((y - f(x; \mathcal{D})))^2 \mid x, \mathcal{D}\right]\Big] \tag{12.10}$$

for $\mathbb{E}[\cdot \mid x, \mathcal{D}]$ any expression with dependencies on x and $\mathcal{D}$. The bias and variance term is subsequently expressed by

$$\mathcal{B}(f, y) = \underbrace{\{\mathbb{E}_{\mathcal{D}}[f(x; \mathcal{D})] - \mathbb{E}[y \mid x]\}}_{\textit{bias}}, \quad \mathcal{V}(f, y) = \underbrace{\mathbb{E}_{\mathcal{D}}\left\{(f(x; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[f(x; \mathcal{D})])^2\right\}}_{\textit{variance}} \tag{12.11}$$

Proof. We present the original derivation of bias-variance tradeoff and its analysis from Geman et al. [1992]. Geman's paper is, by his own word, concerned of *parametric models*, i.e. hypothesis class without strong assumption of parameters defined, though the definition of bias-variance

tradeoff afterward and its justification is made in the sense of parametric model. We partially clarify[3] this with the following definition as customary.

Definition 12.6.1 (Parameterization). *A **parametric model** is one that can be parameterized by a fintie number of parameters. In general, for a hypothesis class $\mathcal{H}$of all parametric hypothesis is then expressed as:*

$$\mathcal{H} = \{f(x;\theta) : \theta \in \Theta \subset \mathbb{R}^d\} \tag{12.12}$$

*where Θ is called the parameter space. A **nonparametric model** is one which cannot be parameterized by a fix number of parameters.*

Suppose of a training dataset $\mathcal{D}$ of N 2-tuples $(\mathbf{x}_i, y_i)$, that is:

$$\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\} \quad |\mathcal{D}| = N, \mathbf{x} \in \mathbb{R}^d, y \in \mathbb{R} \tag{12.13}$$

The regression problem is to construct a function $f(\mathbf{x}) \to y$ based on $\mathcal{D}$, which is denoted $f(\mathbf{x};\mathcal{D})$ to show this dependency. Then,

$$\begin{aligned}
\mathbb{E}\left[(y - f(\mathbf{x};\mathcal{D}))^2 \mid \mathbf{x}, \mathcal{D}\right] &= \mathbb{E}\Big[\big((y - \mathbb{E}[y \mid \mathbf{x}] + (\mathbb{E}[y \mid \mathbf{x}] - f(\mathbf{x};\mathcal{D}))\big)^2 \mid \mathbf{x}, \mathcal{D}\Big] \\
&= \mathbb{E}\Big[(y - \mathbb{E}[y \mid \mathbf{x}])^2 \mid \mathbf{x}, \mathcal{D}\Big] + \Big(\mathbb{E}[y \mid \mathbf{x} - f(\mathbf{x};\mathcal{D})]\Big)^2 \\
&\quad + 2\mathbb{E}\big[(y - \mathbb{E}[y \mid \mathbf{x}]) \mid \mathbf{x}, \mathcal{D}\big] \cdot \big(\mathbb{E}[y \mid \mathbf{x}] - f(\mathbf{x};\mathcal{D})\big)^2 \\
&= \mathbb{E}\Big[(y - \mathbb{E}[y \mid \mathbf{x}]) \mid \mathbf{x}, \mathcal{D}\Big] + \big(\mathbb{E}[y \mid \mathbf{x}] - f(\mathbf{x};\mathcal{D})\big)^2 \\
&\geq \mathbb{E}\big[(y - \mathbb{E}[y \mid \mathbf{x}])^2 \mid \mathbf{x}, \mathcal{D}\big]
\end{aligned} \tag{12.14}$$

Hence, we decompose it to:

$$\mathbb{E}\left[((y - f(\mathbf{x};\mathcal{D})))^2 \mid \mathbf{x}, \mathcal{D}\right] = \mathbb{E}\left[(y - \mathbb{E}[y \mid \mathbf{x}])^2 \mid \mathbf{x}, \mathcal{D}\right] + (f(\mathbf{x};\mathcal{D}) - \mathbb{E}[y \mid \mathbf{x}])^2 \tag{12.15}$$

Where $\mathbb{E}\left[(y - \mathbb{E}[y \mid \mathbf{x}])^2 \mid \mathbf{x}, \mathcal{D}\right]$ is regarded to be a relative constant of variance on y given $\mathbf{x}$. Taking the expectation on $\mathcal{D}$, we gain:

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}\Big\{\mathbb{E}\left[((y - f(\mathbf{x};\mathcal{D})))^2 \mid \mathbf{x}, \mathcal{D}\right]\Big\} &= \mathbb{E}_{\mathcal{D}}\Big\{\mathbb{E}\left[(y - \mathbb{E}[y \mid \mathbf{x}])^2 \mid \mathbf{x}, \mathcal{D}\right] + (f(\mathbf{x};\mathcal{D}) - \mathbb{E}[y \mid \mathbf{x}])^2\Big\} \\
&= \mathbb{E}_{\mathcal{D}}\Big\{\mathbb{E}\left[(y - \mathbb{E}[y \mid \mathbf{x}])^2 \mid \mathbf{x}, \mathcal{D}\right]\Big\} + \\
&\qquad \mathbb{E}_{\mathcal{D}}\Big\{(f(\mathbf{x};\mathcal{D}) - \mathbb{E}[y \mid \mathbf{x}])^2\Big\}
\end{aligned} \tag{12.16}$$

The second term is of importance, and is further decomposed to:

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}}\Big\{(f(\mathbf{x};\mathcal{D}) - \mathbb{E}[y \mid \mathbf{x}])^2\Big\} \\
&= \mathbb{E}_{\mathcal{D}}\Big\{(f(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[f(\mathbf{x};\mathcal{D})] + \mathbb{E}_{\mathcal{D}}[f(\mathbf{x};\mathcal{D})] - \mathbb{E}[y \mid \mathbf{x}])^2\Big\} \\
&= \mathbb{E}_{\mathcal{D}}\Big[(f(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[f(\mathbf{x};\mathcal{D})])^2\Big] + \mathbb{E}_{\mathcal{D}}\Big[(\mathbb{E}_{\mathcal{D}}[f(\mathbf{x};\mathcal{D})] - \mathbb{E}[y \mid \mathbf{x}])^2\Big] + \\
&\qquad 2\mathbb{E}_{\mathcal{D}}\Big[f(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[f(\mathbf{x};\mathcal{D})]\Big] \cdot (\mathbb{E}_{\mathcal{D}}[f(\mathbf{x};\mathcal{D})] - \mathbb{E}[y \mid \mathbf{x}]) \\
&= \underbrace{\{\mathbb{E}_{\mathcal{D}}[f(x;\mathcal{D})] - \mathbb{E}[y \mid x]\}}_{\text{bias term}} + \underbrace{\mathbb{E}_{\mathcal{D}}\left\{(f(x;\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[f(x;\mathcal{D})])^2\right\}}_{\text{variance term}}
\end{aligned} \tag{12.17}$$

[3]Generally, there is no definite definition on what would be considered non-parametric. Though, an example might be drawn from the (vanilla) neural network and Gaussian Process Regression (GPR)

which yields the desired bias-variance tradeoff. □

Empirically, this expression assesses the concern of Geman et al. [1992] on the scenario where the optimal training pattern, either SRM or ERM, cannot approximate well depends on the particuliarity of the training set $\mathcal{D}$ - that is, the near-optimal predictor of y can be achieved in certain dataset, while some varies substantially with $\mathcal{D}$, or that the Bayes's optimizer is not near the optimal bound of f, y for the regression $\mathbb{E}[y \mid x]$.

12.6.1 Another approach - No-free-lunch

Even though bias-variance tradeoff is considered to be the standard rule of thumb in encounters of the dichotomy between *complexity* and *performance*, its form and analytical expression is not uniform throughout different literature of interest. Specifically, it is sometime considered to be equal to the estimation-approximation tradeoff, though in some case it is not. Hence, for an approach for the formal treatment, our consideration should start with the question of the existence of a universal approximator. The goal of the standard learning problem, aside from microscopic and specific details, deals with the construction of the solution to the problem of approximating a given observation set, providing that there exists a hidden relation or concept c that governs the observation itself. Then, the question is to ask if there exists a universal approximator that can approximate any concept c of the entire concept space $\mathcal{C}$, given sufficient time, complexity, and expression.

Conjecture 12.6.1 (General insight). *Bias-variance can be identified, under several contexts, to mean the following:*

- *For any model h and measure of its complexity $\mathcal{M}(h)$ $\mathcal{H} \to \mathbb{F}^k$, There exists a point ψ such that when*

This is expressed by the No-Free-Lunch theorem. We would then see why this leads to the tradeoff we are familiar. We state the No-Free-Lunch theorem as followed.

Theorem 12.6.2 (No-Free-Lunch). *Let A be any learning algorithm for the task of binary classification with respect to the $0-1$ loss over a domain $\mathcal{X}$. Let m be any number smaller than $|\mathcal{X}|/2$, representing the training set size. Thn, there exists a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$, such that:*

1. *There exists a function $f : \mathcal{X} \to \{0,1\}$ with $L_{\mathcal{D}}(f) = 0$.*
2. *With probability of at least $1/7$ over the choice of $S \sim \mathcal{D}^m$ we have that $R_{\mathcal{D}}(A(S)) \geq 1/8$.*

This theorem states that there exists no universal learner, for every learner there exists a task on which it fails, even though success can be achieved by another learner. In certain way or another, this theorem is in the profession of *impossibility theorem*, where one choose to forbid the model at present theory from being universal, that is, capable of doing everything. In particular, any algorithm that choooses its output from hypothesis in $\mathcal{H}$ will fail on some learning tasks.

Corollary 12.6.1. *Let $\mathcal{X}$ be an infinite domain, and let $\mathcal{H}$ be all functions from $\mathcal{X} \to \mathcal{Y}$. Then $\mathcal{H}$ is not PAC learnable.*

Draft proof for binary case. We first prove this for some class of binary classification functions $f : \mathcal{X} \to \{0,1\}$. Assume $\mathcal{H}$ is learnable. Choose for $\epsilon < 1/8, \delta < 1/7$. There, $\mathcal{H}$ is fixed. Then, by PAC learning, for some number $m = m(\epsilon, \delta)$, there must be some algorithm $\mathcal{A}$ and an integer $m = m(\epsilon, \delta)$, such that for any data-generating distribution over $\mathcal{X} \times \{0,1\}$, if for some function $f : \mathcal{X} \to \{0,1\}$, $R(f) = 0$, then with probability greater than $1 - \delta$ when A

is applied to samples S of size m, generated i.i.d. by $\mathcal{D}$, $R(A(S)) \leq \epsilon$. However, applying the No-Free-Lunch theorem, since $|\mathcal{X}| > 2m$, for every learning algorithm (and in particular for the algorithm A), there exists a distribution $\mathcal{D}$ such that with probability greater than $1/7 > \delta$, $R(A(S)) > 1/8 > \epsilon$, which leads to the desired contradiction. □

Main paper, draft 1

12.7 Abstract

The analysis of learning action, machine learning and related practices in the field theoretically has been made by utilizing Computational Learning Theory (CoLT) Valiant [1984] and Statistical Learning Theory (SLT) Vapnik [1999]. These two theories, while overlapped, provides a general framework in analysing and justifying learning actions and learning model constructions, aiding in the formation of modern practices. One of the famous insight using such framework is the *bias-variance tradeoff* Geman et al. [1992], which states that model complexity and generality inversely affect each other, thus guarantee the need for a safe bracket between them. However, recent literatures, Belkin et al. [2019a] has indicated the fallout of such dilemma by the new phenomenon called *double descent*, where there exists an interpolation threshold that renders the current statistical justification for bias-variance inconsequential to a given region. Various 'anomaly' has also been detected similarly, with varying degree of sophistication and potential. In this paper, we analyse the classical learning framework, investigating aspects and concepts related to the formation of the insight of bias-variance dilemma, double descent, and give a separated interpretation and explanation of the learning theory as well as double descent. Furthermore, we also interject with one of the particular experimental result on GNN - a special case where the observed double descent does not occur at all, yet.

12.8 Introduction

Machine learning and its modern practice has been developed and researched, of a substantial portion by empirical and heuristic approach, either by advancing new practices, architectures or by try-and-test modification. From its early onset of a regression estimator $\theta(x, y)$ on the linear regression problem, machine learning has developed substantially. Certain model concept with great successes includes regression-classification model, Bayesian modelling, generative learning model, support vector machine, Gaussian processes, and more. Of all such, the more formal and complex model architecture created, is the concept of a *neural network*. With increasingly sophisticated architecture, heuristic approach become popular, the fast-paced advancement of the field comes with new method, new results, new observations, and its far-reaching application which led to even bigger, larger scale deployment, there has been questions about the formation and status of a theoretical ground, a rigorous matter on the side of **theoretical machine learning**.

While rigorous and well-formulated in a sense, classical and theoretical machine learning was dwarfed by the modern advancement of machine learning as a whole, leading to several anecdotal problems regarding the interpretation of phenomena, the reevaluation of the theory to fit the more updated analysis, and explanation to more sophisticatedly designed system. This and many more, plus as present, many of such advancements and improvements are heuristic, and the general theory and conceptual understanding remain limited, led to the choice to often opt for analogies and empirical workaround.

12.8.1 Statistical Learning and Double Descent

Statistical learning theory (SLT) and computational learning theory (CLT) Vapnik [1999], Mohri et al. [2012], Shalev-Shwartz and Ben-David [2014], Hajek and Raginsky [2021], Bousquet et al. [2020] has been prominent in constructing a well-rounded formal theory surrounding learning problems, models, and machine learners analysis. Valuable insights have been dissected from treatment of statistical theory and mathematical modelling on models, including the *bias-variance tradeoff* Geman et al. [1992], Domingos [2000a], which serves as a bound for efficient

learning and model configuration (or complexity). However, recently there has been observations of *double descent* Belkin et al. [2019a], Schaeffer et al. [2023], Nakkiran et al. [2019], Lafon and Thomas [2024] which refute the famous tradeoff assumption, and hence brings question to the establishment of the theory, as well as several assumptions and insight in the framework. Further events and phenomena observed also includes grokking and triple, to n-descent Davies et al. [2023], d' Ascoli et al. [2020]. Furthermore, many problems of defining and formalizing notions used in designing and implementing machine learning models are inconclusive, such as, for example, *model complexity* and others.

The phenomena *double descent* itself has been investigated somewhat thoroughly, firstly introduced by Belkin et al. [2019a]. Further analysis was made by several literatures, particularly conjectured the existence of double descent to the concept of model complexity and inductive bias. Nakkiran et al. [2019] expanded the phenomena into deep neural network models. Their conclusion is reached by considering the perturbation of a learning procedure $\mathcal{T}$ on the effective model complexity $\mathrm{EMC}_{\mathcal{D},\epsilon}(\mathcal{T})$ defined in the paper, separating the eventual phenomena into regions of observations, thereby in one way or another, predicting the tendency of double descent. This is also the first, arguably, "concise" definition of double descent, even though its nature is an empirical definition, including the notion of model complexity. Preliminary works are done by Lafon and Thomas [2024], Schaeffer et al. [2023], and Liu and Flanigan [2023] on the role of optimization in double descent. Davies et al. [2023] attempted to unifying grokking with double descent, theorized a possibility of similarity during the generalization phase transition of the inference period, and Olmin and Lindsten [2024] attempted to explain epoch-wise double descent within the model of two-layer linear network. However, it is mentioned that it can also be expanded into deep nonlinear networks.

12.8.2 Relation to Graph Neural Network

Graph Neural Network (GNN), first introduced in 1996 by P. Baldi and Y. Chauvin [BA96], subsequently introduced in more attention by Scarselli et al. (2008b), Bruna et al. (2014), Gilmer et al. (2017), Kip & Welling (2017), Velickovic et al. (2017) is a type of specialized neural network, designed for graph and relational data processing. They are built upon the MLP architecture, inserting additional *Message Passing* (MP) operations amid FF layers (Kipf & Welling, 2017), puts up forth in recent years gains traction, specifically in the concern of solving graph-like data.

GNN from those that have emerged as a transformative paradigm in machine learning and artificial intelligence, with applications toward graph and connected data, clusters, social network analysis, and recommendation systems, marked the rapid evolution, remarkable capabilities in complex analysis of complex data structure (Bharti et al., 2024). Unlike self-supervised learning, the graph neural network is utilized for interpretation data and complex relational structure, resulting in the learning of *graph representation* (Cui et al., 2018a; Hamilton et al., 2017b; Zhang et al., 2018a; Cai et al., 2018) that learns to represent the edges or subgraphs of graph nodes by low-dimensional vectors. It is also good for non-Euclidean data, of which can generalize some architecture, such as CNNs on graphs, and extend deep neural models to non-Euclidean domains, by the work on geometric deep learning (Bronstein et al., 2017), which receives enormous attention. Furthermore, several hints also indicate the *'generalization'* power of GNN to generalize the current architecture of deep learning and neuro-topological structures in present models.

Investigating this phenomenon, and other observations that contradict the theoretical notion of

learning theory is important, as the theory is born to interpret conceived notion of machine learning model in practice, and to understand a free-guided, random process such as a learning model.

Hence, in this work, we investigated the theoretical machine learning formulation, its theoretical notions, rigours, and setting. This is coupled with the review on neural network-style architecture, the question of performance and large-scale organization of a GNN network, mostly focuses on a network of graph convolutional network (GCN) and graph attentions networks (GAN), and their hybrid forms. In effect, we study and identify the existence of the **double descent** phenomena within GNN, and provides reasonable analysis of it. However, since it is apparently from recent literature that there are no reports of double descent as of date, the question can also be inferred as investigating **why** the phenomena does not occur.

12.9 Outline

To address the proposed problems, and laying out the methodology and analysis given, the paper is structure in detail as the following outline. The order is relative.

[1] We wish to establish, and reaffirming the setting of the *learning theory*, and the construction of the *mathematical, computational modelling* that founded neural networks and other classical models. This includes the follow-up treatment of two inherent problems in the (machine) learning theory: **computational learning theory (CoLT)** and the problem of **statistical learning theory (SLT)**. Rigorous introduction and related text are also available.

[2] We focus on the example of *bias-variance tradeoff*, its establishments and the general dilemma surrounding it, as well as its weakness and particular anomaly such as *double descent* (Belkin, 2019), and a brief consideration of the generalization of double descent, n-descent. Overall, the establishment, the definition, usages and interpretation of the bias-variance tradeoff will be analysed, summarized, and given insights to the problem. To do this, conjunction with theory we also wish to use exclusively the *neural network formalism* to procure **test models**, such as polynomial regression, SVM, and RNN for analysis.

[3] We wish to pinpoint our problems and weakness in the above section when dealing with *bias-variance tradeoff* and subsequently, other phenomena occurred but without effective solution in solving the objective required. From there, we also outline the sufficient next-section development of theories and our own results, which both reflects recent modern theoretical works from various manuscripts, but also our own insight and treatments of the problem.

[4] Section four will bring in a particular abnormality in between the phenomena of study, the graph neural network (GNN). Particularly, GNN does not exhibit any trace of double descent, which is a very interesting point of study. We provide rigorous experiments and testing on GNN network.

12.10 Background

We discuss and summarize aspects of theoretical machine learning Hajek and Raginsky [2021], Mohri et al. [2012], Shalev-Shwartz and Ben-David [2014] used throughout the standard treatment of bias-variance tradeoff and double descent in literatures.

We are given the observations, or dataset of the form $\mathcal{S} = (\mathcal{X}, \mathcal{Y}) \subset \mathbb{R}^n \times \mathbb{R}^m$, of all 2-tuple pairs, assumed to be sampled or observed and governed by a distribution $\mathcal{D}$.

$$\mathcal{S} = \{(x_1, y_1), (x_2, y_2), \dots (x_n, y_n)\} \subset \mathcal{X} \times \mathcal{Y}$$

The dataset is assumed to be i.i.d. sampling according to $\mathcal{D}$, which is unknown by priori.

The learning problem is then formulated as followed. Given the machine learning model expressed a hypothesis h of the hypothesis class $\mathcal{H}$, the learning theory aims for creating a procedure to learn either elements of the concept class $\mathcal{C}$ of all concepts $c : \mathcal{X} \to \mathcal{X}$, or the function class $\mathcal{F}$ of all functions $f : \mathcal{X} \to \mathcal{Y}$, which is usually set to $\{0, 1\}$ or $[0, 1]$. This distinction is trivial, hence, if it is clear, we will talk about the concept class $\mathcal{C}$ only as representative form. The learner $\mathcal{L}(h)$ consider the set of possible hypothesis $\mathcal{H}$, in which might not coincide with $\mathcal{C}$. It receives a partial image of sample $S = (x_1, \ldots, x_2, \ldots, x_n)$ drawn i.i.d. according to $\mathcal{D}$ as well as the label $(c(x_1), \ldots, c(x_n))$. This constitutes the dataset $\mathcal{S}$, which are based on specific concept $c \in \mathcal{C}$ for the hypothesis to learn. The task is then to use (or *extract*) meaningful information to select a hypothesis $h_S \in \mathcal{H}$ that accurately mimic c, with marginal error Θ. The notation h_S stands for all hypothesis that can be inferred from the range of the dataset (the first argument, $\mathcal{X}$).

The marginal error Θ is considered of two parameters, the empirical error $\hat{R}(h)$ and the generalization error $R(h)$. We give the following definition of them.

Definition 12.10.1 (Empirical risk). *Given a hypothesis $h \in \mathcal{H}$, a target concept $c \in \mathcal{C}$, and a sample $S = (x_1, \ldots, x_m)$. For some particular $\epsilon > 0$, the* ***empirical error*** *or empirical risk of h is defined by*

$$\hat{R}_S(h) = \mathop{\mathbb{P}}_{x \in S \sim \mathcal{D}}[\ell\{h(x), c(x)\} \leq \epsilon] = \frac{1}{m} \sum_{i=1}^{m} \ell\{h(x_i), c(x_i)\} \tag{12.18}$$

Definition 12.10.2 (Generalization risk). *Given a hypothesis $h \in \mathcal{H}$, a target concept $c \in \mathcal{C}$, and an underlying distribution $\mathcal{D}$ on $\mathcal{X}$. For some particular $\epsilon > 0$, the generalization error or risk of h is defined by*

$$R(h) = \mathop{\mathbb{P}}_{x \sim \mathcal{D}}[\ell\{h(x), c(x)\} \leq \epsilon] = \mathop{\mathbb{E}}_{x \sim \mathcal{D}}[\ell\{h(x), c(x)\}] = \int_{x \in \mathcal{D}} \ell\{h(x), c(x)\} \, dP(x) \tag{12.19}$$

For fixed $\mathcal{H}$, for fixed and sufficiently large $\mathcal{S}$, and no observation (data) errors, the empirical risk is the generalization risk. These two measures between h and c constitute the learning problem, which can also be separated into both cases - either empirical learning or generalization learning, one to minimize $\hat{R}(h)$, and one to minimize $R(h)$.

Definition 12.10.3 (Empirical learning problem). *We present the formal form of the empirical learning. Suppose we have a target, $c \in \mathcal{C}$, where $\mathcal{C}$ is an arbitrary concept class that captures targets of the same type. Suppose we are provided a set of observations $\mathcal{S}$. The problem is to use certain algorithm $\mathcal{A}$ using $\mathfrak{D}$, to obtain a hypothesis h^* for a fixed $\mathcal{H}$ such that:*

$$R(h^*) = \min_{h \in \mathcal{H}} \hat{R}(h) = \min_{h \in \mathcal{H}} \mathop{\mathbb{E}}_{x \sim \mathcal{D}, x \in \mathfrak{D}} \ell\{h(x), c(x)\} \tag{12.20}$$

The hypothesis h^* is often called the *empirical best*, for it being the minimal, finite hypothesis of the lowest loss evaluation on the entire observation space $\mathcal{S}$. There exists no certified assumption regarding whether h^* aligns with the minimal generalization error.

Definition 12.10.4 (Generalization learning problem). *We present the formal form of the generalization learning problem. Suppose we have a target, $c \in \mathcal{C}$, where $\mathcal{C}$ is an arbitrary concept class that captures targets of the same type. Suppose we are provided a set of observations $\mathfrak{D}$. Supposed we have an algorithm $\mathcal{A}$ that for fixed hypothesis space $\mathcal{H}$, equation 8.5 holds true. The problem is to use certain algorithm $\mathcal{A}'$*

such that, under limited availability, to obtain $\boldsymbol{h}$, satisfies:

$$R(\boldsymbol{h}) = \min_{h \in \mathcal{H}} R(\boldsymbol{h}) \leq \{\epsilon\}, \quad \epsilon > 0 \tag{12.21}$$

For a set of risk bounds ϵ. If the setting is deterministic, then there exists $\epsilon = 0$.

The way of solving separately in essence, two learning problems is inherently to interpret statistical learning theory in its elementary focus of *prediction analysis*, which underlies the essence of generality such that the learnt concept is general, works for unseen observations and situations.

Solving this problem requires developing algorithms and learning procedures that can solve the smaller problem within the empirical dataset, and prepare for generalization for unseen data. This is conducted mostly by using **empirical risk minimization** (ERM) or structural risk minimization (SRM), which targets mostly the empirical data, and adding some criteria for ensuring generality. More heuristic approach (with the guarantee of convergence for the loss function and the overall objective) includes regularization of parameterized weighted model, or by analysis of trivial model complexity and reduction of said measure. Under such umbrella of choosing and optimizing model, there is the question between the generalization capability, overall accuracy, and the complexity of the hypothesis. This is formulated in classical and modern literature as the **bias-variance tradeoff**.

Before the next section, we would also want to discuss the classical notion or at least controlled confirmation between the *correlation* of c, for distribution $\mathcal{D}$, and c', for distribution $\mathcal{D}_{c'}$.

Theorem 12.10.1. *For fixed $\mathcal{H}$, for fixed and sufficiently large $\mathcal{S}$, and no observation errors, the empirical risk is the generalization risk:*

$$\mathop{\mathbb{E}}_{\mathcal{S} \sim \mathcal{D}^m} [\hat{R}_{\mathcal{S}(h)}] = R(h) \tag{12.22}$$

which effectively guarantee that the two concept between empirical and generalization setting converges to the same point. However, this might not be the case in general.

12.11 Bias-variance tradeoff

To understand bias-variance tradeoff, we first review the phenomena, which has been studied extensively by historical literature, and modern understanding which has become a rule of thumb in many cases. For more detailed analysis, we refer to noa [a], Hellström et al. [2020], Geman et al. [1992], Fortmann [2012]. Here we are only interested in the main aspect of the problem, and follow a historical, more and more rigorous formation of the quoted term.

12.11.1 Defining the bias and variances

Since the subject of the bias-variance tradeoff is the statistical interpretation of the statistical learning system, and by its name, requested the formulation of both terms *bias* and *variance*, it is imperative that we define them in our learning setting.

In terms of statistical bias of a specific model $h \in \mathcal{H}$, it is a measure of the model as a whole, throughout its operation.

In a loosely defined fashion, the **bias** of any estimator/hypothesis model measures the **central tendency** of such model, to the true function of f. Of such, we have the first definition:

Definition 12.11.1 (Bias, I). *Given a model $M[h, S]$, where $h \in \mathcal{H}$ and S is the associated data, the bias of h to the true concept $c \supset \{S\}$ is defined as the measure of estimation of the central tendency of*

the hypothesis to the true concept:

$$\text{Bias}(h_S) = \underset{x\sim\mathcal{P}}{\delta}\{\mathbb{E}[h_S(x)], \mathbb{E}[c(x) \mid x]\} \tag{12.23}$$

where $\delta(\cdot,\cdot)$ is the difference measure, of certain measure space associated with the hypothesis and concept, and $\mathbb{E}[c(x) \mid x]$ is the function (deterministic) of x that gives the mean value of $y = c(x)$ conditioned on x.

In a somewhat acceptable and identical manner, variance can be defined in the same way. Although the spirit of the variance term is quite different, in simple term, often interpreted as followed. Informally, it is a measure of fluctuation of a learner around its central tendency (again, expectation value), where, the fluctuations result from different sampling of the training set [B. Neal, 2019]. So how much of this is true? We first go for the formal definition of such:

Definition 12.11.2 (Variance, I). *Given a model $M[h, S]$, where $h \in \mathcal{H}$ and S is the associated data, the **variance** of h to the true concept $c \supset \{S\}$ is defined as the measure of **fluctuations** of the hypothesis (learner) around the central tendency to the true concept:*

$$\text{Var}(h_S) = \underset{x\sim\mathcal{P}}{\mathbb{E}}\left[(h_S(x) - \mathbb{E}[h_S(x)])^2\right] \tag{12.24}$$

A more generalized definition gives the term variance as

$$\text{Var}(h_S) = \underset{x\sim\mathcal{P}}{\mathbb{E}} d_M(h_S, \langle h_S\rangle) \tag{12.25}$$

Bias and variance seems to be two distinct concepts. However, historically, in general statistic, they are very much interconnected together by their behaviours between two polar opposite of the estimation theory. This resulted in the *bias-variance tradeoff* in classical statistics. The first derivation of this in terms of machine learning theory is made by Geman in 1992 Geman et al. [1992].

12.11.2 Precursor (Geman, 1992)

The original definition of bias-variance tradeoff by Geman et al. [1992] is first constructed using the means-square error, which is regarded as a normal measure in the real encoding space. Their approach is to justify bias-variance via decomposition of the loss function ℓ, for such to find an alternative reasonable form of such loss landscape. Suppose of a regression problem to construct a hypothesis function $f(x)$ from $(x_1, y_1, \ldots, x_N, y_N)$ for the purpose of generalization - that is, predicting unseen variational values for different pair $(x_j, ?)$ such that $? = y_j + \epsilon$ for a conceivable implicit error. To be explicit about the relation of this problem, or f on the given data $\mathcal{D} = \{(x_i, y_i) \mid i \leq N\}$, denote $f(x; \mathcal{D})$ instead of f, the natural mean-square measure as a predictor is:

$$\mathcal{M}(f, y) = \mathbb{E}\left[((y - f(x; \mathcal{D})))^2 \mid x, \mathcal{D}\right] \tag{12.26}$$

for $\mathbb{E}[\cdot]$ the expectation wrt to a distribution P. Decomposing the right-hand side, we have:

$$\mathcal{M}(f, y) = \mathbb{E}\left[((y - f(x; \mathcal{D})))^2 \mid x, \mathcal{D}\right] = \mathbb{E}\left[(y - \mathbb{E}[y \mid x])^2 \mid x, \mathcal{D}\right] + (f(x; \mathcal{D}) - \mathbb{E}[y \mid x])^2 \tag{12.27}$$

Here, $\mathbb{E}\left[(y - \mathbb{E}[y \mid x])^2 \mid x, \mathcal{D}\right]$ does not depend on $\mathcal{D}$, but simply the statistical variance of y given x. The term $(f(x; \mathcal{D}) - \mathbb{E}[y \mid x])^2$ is considered a natural measure of effectiveness on $\mathbb{R}^n$ as a singular predictor of y. Now, for $\mathbb{E}_{\mathcal{D}}\left[(f(x; \mathcal{D}) - \mathbb{E}[y \mid x])^2\right]$ which depends on the training

set $\mathcal{D}$ in its computation, is decomposed into the form of *bias-variance decomposition* terms, by derivation:

$$\mathbb{E}_{\mathcal{D}}\left[(f(x;\mathcal{D})-\mathbb{E}[y\mid x])^2\right]=\underbrace{\{\mathbb{E}_{\mathcal{D}}[f(x;\mathcal{D})]-\mathbb{E}[y\mid x]\}^2}_{\text{bias term}}+\underbrace{\mathbb{E}_{\mathcal{D}}\left\{(f(x;\mathcal{D})-\mathbb{E}_{\mathcal{D}}[f(x;\mathcal{D})])^2\right\}}_{\text{variance term}} \tag{12.28}$$

We summarize this in the following statement.

Theorem 12.11.1 (Bias-variance decomposition). *Suppose the model $f(x;\mathcal{D})$ for the data $\mathcal{D}=(x_i,y_i)$ and its parameter x is defined. For y_i of the target concept's responses y, and consider a regression problem with the loss measure $\mathcal{M}(f,y)$ of mean squared risk, the following statement is true:*

$$\mathbb{E}\big[\mathcal{M}(f,y)\big]=\mathcal{B}(f,y)+\mathcal{V}(f,y)+\mathbb{E}\Big[\mathbb{E}\left[((y-f(x;\mathcal{D})))^2\mid x,\mathcal{D}\right]\Big] \tag{12.29}$$

for $\mathbb{E}[\cdot\mid x,\mathcal{D}]$ any expression with dependencies on x and $\mathcal{D}$. The bias and variance term is subsequently expressed by

$$\mathcal{B}(f,y)=\underbrace{\{\mathbb{E}_{\mathcal{D}}[f(x;\mathcal{D})]-\mathbb{E}[y\mid x]\}^2}_{\textit{bias}},\quad \mathcal{V}(f,y)=\underbrace{\mathbb{E}_{\mathcal{D}}\left\{(f(x;\mathcal{D})-\mathbb{E}_{\mathcal{D}}[f(x;\mathcal{D})])^2\right\}}_{\textit{variance}} \tag{12.30}$$

The above decomposition principle is often expressed into a form where there exists the intrinsic noise Brown and Ali [2024]:

$$\begin{aligned}\mathbb{E}_D\left[\mathbb{E}_{xy}\left(y-\hat{f}(x)\right)^2\right]&=\mathbb{E}_x\left[\left(y^*-\mathbb{E}_D[\hat{f}(x)]\right)^2\right]+\mathbb{E}_x\left[\mathbb{E}_D\left(\hat{f}(x)-\mathbb{E}_D[\hat{f}(x)]\right)^2\right]\\&\quad+\mathbb{E}_{xy}\left[(y-y^*)^2\right]\end{aligned} \tag{12.31}$$

A main common theme of criticism toward bias-variance tradeoff is the fact that the decomposition is much more general, and intrinsic for the class of *mean squared loss*. However, it can also be shown Brown and Ali [2024], Pfau [2013] that it also holds for the class of Bregman divergence measure.

In general, bias-variance is typically presented. In fact, one of the reason that it became the rule-of-thumb for ML practitioner, as well as generally statistical learning (Lafon and Thomas [2024] provides a quite rigorous treatment of bias-variance tradeoff in the section on statistical learning theory) solidify the trade-off as a particular model selection principle.

Generally this tradeoff can be summarized as followed:

Theorem 12.11.2 (Bias-variance tradeoff). *For the expected loss of any given hypothesis h, the bias $\mathcal{B}(f,y)$ and variance $\mathcal{V}(f,y)$ is inversely proportional, that is, $\mathcal{B}(f,y)\propto\lambda^{-1}\mathcal{V}(f,y)$ for some proportionality λ that may or may not be constant. In the most general case possible, $\lambda=-1$ on the entire error range.*

The tradeoff is then of inverse proportionality. Indeed, statistically, we have such tradeoff on a statistical framework in a more concrete sense. For the bias to increase, variance will increase, of which the criterion is inverse - we would like to have more bias but lower variance, according to such theory.

12.11.3 Formalism issues and uncertainty

While the formulation of bias-variance is quite intuitive, the above bias-variance relationship leaves a lot of room for ambiguous interpretation. Understanding this, and find a way to formalize this relationship is crucial for further analysis into the subject matter. The concept that we now know as bias-variance tradeoff has a long history, which is based in statistic. As of classical statistics, the bias-variance tradeoff is already presented with the oldest account dates back to Grenander [1952].

Being a relatively simple insight from decomposing the general expected error of the system, bias-variance trade-off is often interpreted as the relationship between the proxy for model complexity, and the proxy of model stability. The term model stability is more ambiguous, and as definition for variance, it can be thought as the *measure of fluctuation* of the result of the hypothesis learner. While bias calculate the total amount of error to the given dataset, variance calculate the overall fluctuation of the result to the mean. This in turn, led to the application of bias-variance trade-off to be applied onto the notion of *underfitting and overfitting*, which is two observable phenomena occurred in practice.

Formulation and interpretations of the concept of bias-variance has been conducted, particularly for example, in Shalev-Shwartz and Ben-David [2014], the bias-variance tradeoff is connected to the No-free-lunch theorem, and its somewhat argued in the sense of approximation-estimation error tradeoff [4]. In Geman's work, it is formulated using a treatment of statistical learning on fixed-size parametric system.

Theorem 12.11.3 (No-Free-Lunch). *Let A be any learning algorithm for the task of binary classification with respect to the $0-1$ loss over a domain $\mathcal{X}$. Let m be any number smaller than $|\mathcal{X}|/2$, representing the training set size. Thn, there exists a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$, such that:*

1. *There exists a function $f : \mathcal{X} \to \{0,1\}$ with $L_{\mathcal{D}}(f) = 0$.*
2. *With probability of at least $1/7$ over the choice of $S \sim \mathcal{D}^m$ we have that $R_{\mathcal{D}}(A(S)) \geq 1/8$.*

This theorem states that there exists no universal learner, for every learner there exists a task on which it fails, even though success can be achieved by another learner. In certain way or another, this theorem is in the profession of *impossibility theorem*, where one choose to forbid the model at present theory from being universal, that is, capable of doing everything. In particular, any algorithm that chooses its output from hypothesis in $\mathcal{H}$ will fail on some learning tasks.

In famous books and articles, bias-variance is often implicitly recognized, and is often not dealt with. Statistical learning theory on the other hand opted for more complex and often different approach to said model selection principle, for example, by using approximation-estimation metric. Works has also been done to expand bias-variance tradeoff to more robust and rich family of loss functions, for example, the bias-variance decomposition for general setting Pfau [2013], Buschjäger et al. [2020a], uni. Nevertheless, the issue remains: there are no exact treatment of the bias-variance tradeoff, especially the correction of such theory when abnormality in their behaviours are spotted, for instance, grokking and double descent.

In fact, this goes even further in history. By accounts of Neal [2019], experiments and evidences shown that even in Geman's analysis, there are inconclusive evidences of the cracks of the bias-variance formalism. We quote their explanation of the inconclusive result.

[4] We find this observation helpful. Brown and Ali [2024] indeed raised a very good point on the misleading nature of the approximation-estimation tradeoff in conjunction with its misunderstood usage with bias-variance tradeoff.

> The basic trend is what we expect: bias falls and variance increases with the number of hidden units. The effects are not perfectly demonstrated (notice, for example, the dip in variance in the experiments with the largest numbers of hidden units), presumably because the phenomenon of overfitting is complicated by convergence issues and perhaps also by our decision to stop the training prematurely. (Geman et al., 1998)

Overall, there are many contradictions and edge cases for bias-variance tradeoff, as well as observations and formulation being misrepresented and misunderstood, particularly with other measures in machine learning theory. It is, however, noted that bias-variance is not entirely wrong - it is a very useful measure and indicator for machine learning models, historically, despite its weakness and flaws that we now observe.

12.11.4 Approximation-Estimation tradeoff

Another closely related notion to bias-variance is the concept of approximation-estimation tradeoff. We refer to Mohri et al. [2012], Lafon and Thomas [2024] for some analysis and mentions.

Let $\mathcal{H}$ be a family of functions mapping $\mathcal{X} \to \{1, -1\}$. This is the particular case of **binary classification**, in which can be straightforwardly extended to different tasks and loss functions. The *excess error* of a hypothesis $h \in \mathcal{H}$, is the difference between its error $R(h)$ and the Bayes error R^*. This can be decomposed to be the following:

$$R(h) - R^* = \Big(R(h) - \inf_{h \in \mathcal{H}} R(h)\Big) + \Big(\inf_{h \in \mathcal{H}} R(h) - R^*\Big) \tag{12.32}$$

The first bracket contains the **estimation error**, and the second bracket contains what is called the **approximation error**. The estimation error depends on the hypothesis h selected. It measures the error of h with respect to the infimum of the error achieved by hypotheses in $\mathcal{H}$, or that of the best-in-class hypothesis h^* when that infimum is reached. The approximation error measures how well the Bayes error can be approximated using $\mathcal{H}$. It is a property of the hypothesis set $\mathcal{H}$, a measure of its richness.

Model selection consists of choosing $\mathcal{H}$ with a favourable trade-off between the approximation and the estimation error. However, in the most general case, this will be done, but not in practice, as it requires the underlying distribution $\mathcal{D}$ to be known to determine R^*, which is not possible. In contrast, the estimation error can be bounded, or can be analysed, using particular metric and analysis.

Is however, worth to note that bias-variance and approximation-estimation have a very complicated relationship, for example, in Brown and Ali [2024] shown that they are indeed not the same, and is in fact two different decomposition, where one is the others' component.

12.12 Double descent

Double descent, as mentioned, refuses the dichotomy of bias-variance tradeoff as the standard statistical rule of thumb for optimality of the statistical efficiency of the model, which is based on the nevertheless true to certain amount empirical observations. To do any analysis further down, we need clarification, definition of double descent. Shalev-Shwartz and Ben-David [2014], Mohri et al. [2012] provides most of the formal literature regarding this situation, though we would utilize a range of empirical evidence in such analysis.

The first paper to report it is Belkin et al.'s paper Belkin et al. [2019a] on reconciling the practice of bias-variance tradeoff, with new empirical evidences.

12.13 The break-off between theoretical and modern practice

While analysing statistical learning theory, modelling theory, the theory and implementation of double descent and bias-variance tradeoff, theoretical conjectures, and overall theme of previous analysis on similar topic, we perhaps have seen a pretty observable disconnect and break-off, or lacking in and between machine learning theory and modern application and heuristic practice.

Analysing statistical learning theory and overall landscape of statistical learning theory, and the learning theory as a whole, encountering new problems like double descent revealed its weakness, and ultimately, perhaps one of the reason why it is ineffective against such problem. First, there are simply too many assumptions made, too many formulations made during said process. Furthermore, there are also unclear notions and concepts, of which make it even harder to analyse or fully formalize. Secondly, there are inherent conflicts and uncertainty within those theories, formulations and notions by itself. For example, in one sense, the No-free-lunch theorem is considered to be representative and true, whilst also simultaneously being considered the opposite of such. And amidst abnormality behaviours of the old bias-variance formulation, we also find distinctive weakness in our theory, for example, the concerning difficulty in defining the notion of *model complexity* in various contextual ways. To analyse double descent, perhaps we also need a new theory or formulation to support it.

Furthermore, most of the general solution and bounds created by statistical learning theory is often in a very simplistic system. For example, if we are to utilize the Rademacher complexity measure, most of the time we will have to compute it through the growth function $\Pi_{\mathcal{H}}(m)$ for m points, on the standard finite hypothesis class $\mathcal{H}$ such that

$$\Pi_{\mathcal{H}}(m) = \max_{x_1,\dots,x_m \in \mathcal{X}} |\{(h(x_1), \dots, h(x_m)) \mid h \in \mathcal{H}\}| \tag{12.33}$$

Most of our problems resolve to binary classification, or rather, the problem of pattern recognizing discrete, reduced classification form. It is not so sure for now if all problems can be reduced to such way, so we cannot draw conclusive analysis that is not diluted of mathematical formulation for complex systems. That is not to count the computationally intensive operation required to calculate the supposedly classical measure, while not entirely of itself holds any substantial reasonable information about the internal dynamics of the model itself.

It is then suggested to instead revitalize a particular portion of the theory, or transition to another new theory on itself.

12.14 Preliminary experiments

For understanding and analysing double descent and bias-variance trade-off, and furthermore in later section on identifying double descent, we would like to use several test models specifically for exhibiting bias-variance, as well as testing hypothesis and forming theoretical conjectures. Specifically, we would like to use *polynomial regression model*, the standard architectural description of *support vector machine* (SVM), the vanilla *neural network*, and *recurrent neural network*. Our goal is pretty conclusive.

12.14.1 Polynomial model

Most of the examples often seen with bias-variance tradeoff is with the famous example of polynomial regression. Indeed, in the range of interpretable, observable model results, polynomial regression with expressive capacity increased is one of the more famous problem setting, which also lies in the regression learning task. Goodfellow et al. [2016] also used polynomial regression in his book to illustrate the problem of bias-variance tradeoff, and several textbooks, such as ISLR James et al. [2013].

Informally, polynomial regression considers the set of all hypotheses h of the polynomial hypothesis class $\mathcal{H}_p$. In the single, univariate case of the hypothesis representation, the polynomial $p_n(x) \in \mathcal{H}_p$ is expressed by:

$$p_n(x) = \sum_{i=0}^{n} c_i x^i$$

where c_i is the associated constant for each term. The bias term here is controllable, and is part of the polynomial as the mathematical formulation holds. As always, since it is a model, x argument cannot be controlled. The only degree of freedoms provided is the set $\{c_i\}$ of all constants. Hence, we can conceptualize this as always, as a bunch of unit processing embedded each unit, with a scaling factor by i, and control them by the weight. The form $p_n(x)$ in a polynomial regressor will often be

$$M[p(n, x)] = \mathbf{W} f(\mathbf{x})^\top$$

where $f : \mathbb{R}^{n+1} \to \mathbb{R}^{n+1}$ function which scales by applying power per argument. As standard, we will use the typical stochastic gradient descent [5], though it is important to note why we have to use it.

Experimentally, we would likely have to present certain interpretation and configuration to the setting, depends on particular subject of interest. So far, this is most presentable using diagram. So I will have to make one.

12.14.2 Support Vector Machine (SVM)

Support vector machine, first formally originated from Vapnik [1999], is a model inherently, specifically defined for pattern recognition task, and binary classification via an *optimal separating hyperplane*. There are two variants for SVM, namely, for linear and nonlinear hyperplane.

The SV machine implements the following two precursor ideas: It maps the input vectors x, supposed of the setting, into a high-dimensional feature space Z through some nonlinear

[5] As for why it is not pure vanilla gradient descent, we notice that for a gradient descent algorithm in an informal setting, it is wise to notice that for any given configuration dataset space, the path itself is deterministic based on all the description of the dynamical system (hyperparameters like learning rates, the algorithm itself, the hypothesis's parameters and representation, and the data assumptions - for example, without white noise or not). Removing determinism can be done using stochastic gradient descent, even though for now a formal treatment of this 'stripping off deterministic behaviour' is not fully formulated.

mapping, chosen a priori. In this space, an optimal separating hyperplane is constructed. By statistical learning theory, Vapnik restricted the function class (as for infinite hypothesis it is impossible to learn) to the class of hyperplanes by

$$\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = 0; \quad \mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R} \tag{12.34}$$

which $\mathbf{w}$ is the controlling weight, $\mathbf{x}$ is the input space to the space of all binary $\{-1, +1\}$ category. Hence, this basically divide the input space into two: one part containing vectors of the class -1 and the others being $+1$. If there exists such a thing, then it is said to be *linearly separable*. To find the class of a particular vector $\mathbf{x}$, we use the following decision function

$$f(\mathbf{x}) = \operatorname{sgn}[\langle \mathbf{w} \cdot \mathbf{x} \rangle + b] \tag{12.35}$$

This is called the **hyperplane classifier** class. As can be understood simply from such, there exists many hyperplanes that can correctly classifies the classes. It has been then shown that the hyperplane that guarantees the best generalization problem performances is the one with the maximal margin of separation between two classes Cristianini and Shawe-Taylor [2000]. The above form of finding final classification can then be presented in dual form, which then depends only on dot products between vectors, as

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{\ell} y_i \alpha_i \langle \mathbf{x} \cdot \mathbf{x}_i \rangle + b \right) \tag{12.36}$$

where $\alpha_i \in \mathbb{R}$ is a real-valued variable that can be viewed as a *measure* of how much informational value $\mathbf{x}_i$ has (for x_i the support vectors we get from training)[6]

The second idea is of the *kernel method*. This particular method, useful and well-known of the time SVM was created, gain a more prominent position among other techniques, including neural network. Specifically, this method is characterized by the mapping of the input vectors into a richer (usually high-dimensional) feature space where they satisfy *linear separable* criteria. This prompted the possibility to solve nonlinear problem through such mapping, and indeed yields a nonlinear decision surface in the input space, which is linear in the feature space. A **kernel** (function) is then a function $k(\mathbf{x}, \mathbf{y})$ that given two vectors in input space, return the dot product of their images in feature space such that $k(\mathbf{x}, \mathbf{y}) = \langle \phi(x), \phi(y) \rangle$ for the nonlinear mapping ϕ. The general form of SVM is then

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{\ell} y_i \alpha_i k \langle \mathbf{x} \cdot \mathbf{x}_i \rangle \right) \tag{12.37}$$

for any particular final categorization. One of the major problem for analysing SVM, is in the properties of it potentially having infinitely many parameters, depends on the size of the dataset. This makes the curve of both bias-variance and double descent not applicable in such regard - infinitely many parameters complexity, yet still finite complexity class.

From such observation, it would seem imperative that the notion of complexity based solely on the

[6] The *support vector* of a particular hyperplane $\mathbf{y}$ is the collections of all points that lies closest to the hyperplane, which is specified as condition for maximization on both side of maximal margin vector machine.

12.15 Experiments

To confirm and observe particular insights on the problem, experimentally, we focus more on the results given by the analysis on a particular type of neural network, the graph neural network (GNN). However, preliminary experiments are also needed, for example, with various complexity and function class, though most of them are algorithm based on ERM, which typically can be called derivation of gradient descent.

According to major literature, for example, Shi et al. [2024], double descent is absent usually in GNN, for example, GCN networks. However, in fact, it is relatively unstable, as for certain papers and researches point out its existence and variation of it being ubiquitous to the network, some reports none of such occurrence in their experiments. Thereby, our first strategy would be to encounter this absence, and the way to force it to exhibit itself, if the phenomenon is perceived non-existent. In experiments by Shi et al. [2024], Buschjäger et al. [2020a], we know that graph network are inherently sensitive to data configuration. The graph MPNN in Hamilton, for example, depends heavily on the configuration of data to present its neighbourhood aggregation for each layer. However, we would like to first claim the dichotomy of bias-variance holds for it to be presented as the first interpolation point.

We would use, and utilizes a hybrid setting between MPNN, GCN and GAN (attention network), to configure out network in itself. However, heterogeneous network - either consists of only MPNN or GCN layers, is somewhat preferred for ease of analysis, and potency of experimental learning control. Because double descent lies between the concept of test error and training error, either supervised or semi-supervised would be used. According to Shi et al. [2024], semi-supervised setting gives better flexibility and overall 'range' of operation. Other than GNN, certain more abstract, 'vanilla' neural network formalism as specified in the above section treatment will also be conducted, in a supplementary manner.

12.15.1 Main result

Because of the irregularity in the statement that GNN does not exhibit any phenomena that is related to double descent Shi et al. [2024], we would be investigating this phenomena the most. This particularly means that a lot of our focus will be to analyze the GNN network by itself.

Quick introduction to graph theory

A **graph** G is a 2-tuple (V, E) where V is the set of **vertices**(or nodes) and E is the set of **edges**. The set $ne[n]$ stands for the neighbour of vertex n, while $co[n]$ is the set of edges that have n as vertex. Edge is often denoted by (u, v) for vertices u and v. We said that (u, v) **joins** u and v, and it can be directed or undirected. A graph is called a *directed graph* if all edges are direct or *undirected graph* if all edges are *undirected*. The **degree** of vertices v, denoted by $d(v)$, is t number of edges connected with v. The graph data can be loosely (not specifically in cases) as followed. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ be a graph, where $\mathcal{V} = \{1, \ldots, n\}$ is the set of nodes, $\mathcal{E} = \{1, \ldots, M\} \subseteq \mathcal{N} \times \mathcal{N}$ is the set of edges, and $\mathcal{W} : \mathcal{E} \to \mathbb{R}$ is the edge's weight function. There is also the optional weight $\mathcal{W}' : \mathcal{V} \to \mathbb{R}$ for each vertex. In this case, it is for certain problem like TVP, where all cities have certain properties for the path. We say that a data sample $\mathbf{x}$ is a graph data, if its entries are related through the graph $\mathcal{G}$.

Our 2-tuple (V, E) and the collection of all such tuple forms a group of *simple graph* (undirected) and *directed graph*, where the only change is that $(u, v) \neq (v, u)$ for any given edge on a graph. From such, a typical setting, if we are to apply a machine learning setting on graph-theoretical problems, can be described as the following

Definition 12.15.1 (Graph-theoretical edge learning). *Given a graph $G(V, E, \psi_E)$ for ψ_E the edge proprietary classification, there exists an encoding space $\mathsf{ENC}(G)$ that the graph lives in, and a function $\phi : (E, \psi_E) \to (E', \psi'_E)$ such that to change the configuration of the connector space. The* ***learning problem*** *is for a learner $\mathcal{L}$ to learn a function ϕ that is appropriate of the intended use case, such that for any given data $D(V^*, \odot)$, either:*

1. *Assign E^* and transform to ψ^*_E.*
2. *For existing E^*, transform to ψ^*_E.*

All within the marginal error evaluation of L.

The reason for choosing an encoding can be justified as followed. Typically, graphs can be classified and categorized into a specific type of data representation, called *non-Euclidean data*. More specifically, there exists a topological space encapsulating the graph, but exists no fundamental measure on such topological space housing it. For example, there exists no notion of a *discrete distance* on a graph, but only the induced notion of *graph distance* by counting the shortest path from one node to another in a specific graph. Because machine learning works on the assumption that the space of the system gives *measurable space*, a natural response is to encode the system into an encoding space of arbitrary meaning. One, for example, simple encoding is the degree map, where nodes are mapped into a $n \times n$ matrix of nodes and valued by the number of neighbour they have.

Graph Neural Network

We target the graph neural network structure (GNN), specifically a neural network implementation for solving graph data problems. A more detailed description can follow from Hamilton, Scarselli et al. [2009]. Typically, graph neural network (Scarselli et al. [2009], Veličković [2023], Tanis et al. [2024], Lopushanskyy and Shi [2024]) follows the instruction flow of the *encoder-decoder* architecture. A GNN is formulated and structured by analysing a graph data system by conceptually apply a *neighbour-dependent* neural input arrangement on top of the data. That is, in principle,

Structurally, the description of a GNN is defined by the overall *message-passing neural network* (MPNN), defined by:

$$\mathbf{x}_i^{(k)} = \gamma^{(k)} \left(\mathbf{x}_i^{(k-1)}, \bigoplus_{j \in \mathcal{N}(i)} \phi^{(k)} \left(\mathbf{x}_i^{(k-1)}, \mathbf{x}_j^{(k-1)}, \mathbf{e}_{j,i} \right) \right), \tag{12.38}$$

for $\bigoplus$ the differentiable, permutation invariant function, usually called the aggregator, $\mathbf{x}_i^{(k-1)} \in \mathbb{R}^F$ the node features of node i in passing layer $k-1$, $\mathbf{e}_{j,i} \in \mathbb{R}^D$ the optional edge features from node j to node i. Additionally, γ denotes the nonlinearity differentiable function (usually ReLU, or sigmoid), and ψ denote the MLP layer accompanied. A simplified example of this structure in Scarselli et al. [2009], Hamilton as

$$\mathbf{x}_i^{(k)} = \sigma \left(\mathbf{W}_{\text{self}}^{(k)} \mathbf{x}_i^{(k-1)} + \mathbf{W}_{\text{neigh}}^{(k)} \sum_{v \in \mathcal{N}(i)} \mathbf{x}_v^{(k-1)} + \mathbf{b}^{(k)} \right) \tag{12.39}$$

where $\mathbf{W}_{\text{self}}^{(k)}, \mathbf{W}_{\text{neigh}}^{(k)}$ are trainable parameter matrices, and σ denotes an elementwise nonlinearity, and an optional bias term $\mathbf{b}^{(k)}$. Different flavours of the differentiable aggregator

create the *graph convolutional networks* (GCNs), defined by:

$$\mathbf{x}_i^{(k)} = \sigma\left(\mathbf{W}^{(k)} \sum_{v \in \mathcal{N}(i) \cup \{i\}} \frac{\mathbf{x}_v}{\sqrt{|\mathcal{N}(i)||\mathcal{N}(v)|}}\right) \tag{12.40}$$

A somewhat popular approach is to apply attentional layer and weights to facilitate neighbourhood attention, which is called graph attention network.

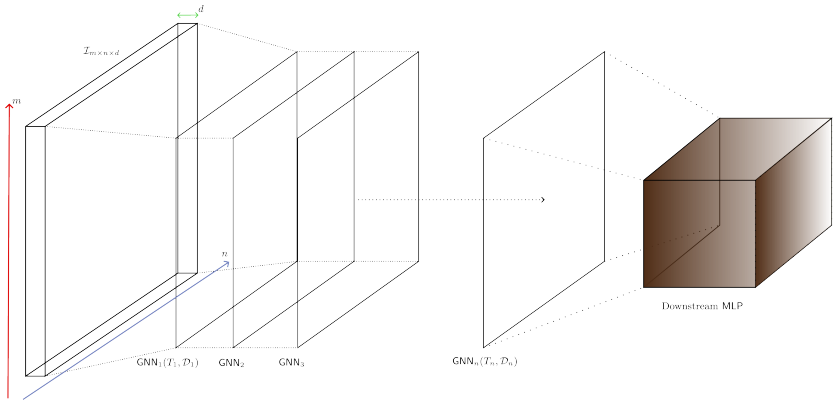


Figure 12.6: A conceptual illustration on the running flow of an n-layer GNN on particular structure of interest. Note that the data section itself has particular embedding structure on its own.

A GNN is, when fitting into the framework of learning system and modeling, is a feature mask generation and adaptation mechanism of the modelling pipeline. What this means is that GNN *assumes* every data point has its own feature-embedded space, and according relationships. By this, we further mean that it creates an embedding space of the aggregator, or the GNN-embedding space within such. While input embedding space captures and represents the best possible interpretation of the input space, GNN-embedding space captures what is required of the aggregation properties that the GNN layers specify. Hence, we can assume relative fallacy in such embedded space. Those n-embedded space of data aggregated at the nth final layer, would then be fed into another straightforward - task-based and structured network, such as a pass to a single-layer sigmoid network, Hamming network, or generally an FCN.

In the supervised learning domain, there are a few direct scenarios and learning problems as accordance to its nature - both can be divided to the *node* V problem, *edge* E problem, or aggregation-specific problem like embedding space-related tasks, for example.

12.15.2 Analysis of GNN

12.15.3 Experiment 1: Identifying bias-variance in GNN

Experiment 1 is the preliminary confirmation of bias-variance tradeoff in graph neural network, before jumping in empirical analysis or forcing appearance of double descent.

12.16 Conclusion

12.17 Related works

Performance and Evaluation The question of performance and broad evaluation of GNN on supervised or semi-supervised tasks has been investigated in some recent papers Oono and Suzuki [2020], Shi et al. [2024]. However, at of date, there are no sufficient evidence that double descent appears on GNN from any recent literature Oono and Suzuki [2020]. The task considered here is *node classification*, due to its sufficient large scale compare to graph classification, with the majority of similar literature focus on the analysis of Graph Convolutional Layer in such task.

Double descent The phenomena *double descent* itself has been investigated somewhat thoroughly, firstly introduced by Belkin et al. [2019a]. Further analysis was made by several literatures, particularly conjectured the existence of double descent to the concept of model complexity and inductive bias. Nakkiran et al. [2019] expanded the phenomena into deep neural network models. Their conclusion is reached by considering the perturbation of a learning procedure $\mathcal{T}$ on the effective model complexity $\mathrm{EMC}_{\mathcal{D},\epsilon}(\mathcal{T})$ defined in the paper, separating the eventual phenomena into regions of observations, thereby in one way or another, predicting the tendency of double descent. This is also the first, arguably, "concise" definition of double descent, even though its nature is an empirical definition, including the notion of model complexity. Preliminary works are done by Lafon and Thomas [2024], Schaeffer et al. [2023], and Liu and Flanigan [2023] on the role of optimization in double descent. Davies et al. [2023] attempted to unifying grokking with double descent, theorized a possibility of similarity during the generalization phase transition of the inference period, and Olmin and Lindsten [2024] attempted to explain epoch-wise double descent within the model of two-layer linear network. However, it is mentioned that it can also be expanded into deep nonlinear networks.

Graph Analysis The graph analytical problems and overall setting lies in the framework of *geometric deep learning*, covered and described in Bronstein et al. [2021, 2017]. It is also related to the problem of learning on *non-Euclidean* data as opposed to the normally Euclidean-embedded numerical data.

GNN Graph Neural Network has been extensively studied - first formulated by Scarselli et al. [2009] on the principle of subjects on which a graph can be studied, and the message passing principle. A more recent, fairly comprehensive resource on the treatment of graph neural network is Hamilton.

Chapter 13. Deconstruction of Neural Network Architecture

13.1 Abstract

We revise and reevaluate the theory of artificial neural networks, with the cornerstone notion of *forward process* and *perceptron operations*. This is the framework which gives rise to *neural network architecture* and furthermore, the *deep learning architecture* of multilayer perceptrons (MLP). In particular, we construct a novel structures and analytical component sets, which serves to decompose the neural network architecture to the smallest working component, and within general principle of working.

13.2 Introduction

With the development of neurological models since the early 1940s [McCulloch, Pitss, 1943], the works done in 1950s to the ends of 1970s [Hebb, 1949] [Rosenblatt, 1958] [Bernard, Hoff, 1959] [Ivakhnenko, Lapa, 1965] [Shun'ichi Amari, 1967,1972], [Seppo, 1970] [Kely, 1960] by many people before the period called the AI winter, and progress made of which formed the discipline of *deep learning* up to present, machine learning has been mostly considered, and conducted, within the *neural network architecture*. This architecture is based on its most basic component of an *artificial neuron*, most often described and constructed as a *perceptron*. Many progresses has been made using this architecture involves *forward neural network* (FNN), *backpropagation principle*, *perceptron learning* and *multilayer perceptron* (MLP), *Hebbian learning*, *Hopfield networks*, *recurrent neural network* (RNN) and *long-short term memory* (LSTM) network, *convolutional neural network* (CNN), up to *encoder-decoder*, *belief networks*, and then *transformer (sub)architecture*, most prominent in applications of the state-of-the-art operations in the present time.

However, we still do not know entirely how the neural network architecture actually works, even from those successes conceded. The nature of the neural network architecture is partially unknown, typically described as a black box. Its working mechanism is well-known, but the decision making and information processing is untrackable. Works has been conducted to deal with this problem, specifically through viewing the neural network, and its subsequent construction called *deep learning* via theory and traceback to neurological concepts [Samuel et al., 2023] [Daniel, Sho, Boris, 2021v2] [Timothy, Konrad, 2019] [Zhanghao, Ding, 2021], or by practical experiments from different perspective [Jason et al., 2015] [Nguyen et al., 2019] [1]. However, those researches conceived gained various insight, not an overall rework of the hypothesis, and most of the time is constrained by the rigid construction already presented. Theories are taken in turns, but some of the works focus on rather the general case, the bounds and conceptual limits, but not the exact phenomenological architecture itself. One example of such is the PAC-learning theory, PAC-Bayesian, VC-Theory, and else. An analysis of the actual

[1] We do not specify all papers related to such issues, but those are some highlights.

optimizer (taken in terms of PAC(-Bayesian) theory *objective*) is very much the main objective of this paper.

Hence, this paper focus, and **goal**, is on the *deconstruction* and a novel construction of the main schema of artificial intelligence researches - the neural network architecture, and presents it in a more conceptual-practical and transparent (if able) cases. This also includes various insights, propositions, conjectures and incorporation of different concepts into the analysis. The obvious goal, *learning*, however, is not in the schedule to be made, or rather, is not considered in such case.

13.3 Constructions

Neural network architecture was based around the construction of the singular processing unit, called *neuron*. This stems from neuroscience researches and knowledge, of which identifies the main processing construction is the *biological neuron* embedded in the development of the brain. McCulloch and Pitts first introduced, the first mathematical model of a biological neuron. This model consists of the classical separation with preactivation and *activation function*, specifically the Thresholding Logic Unit (TLU):

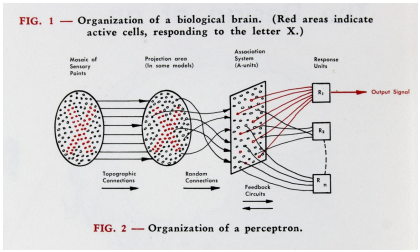
$$y_{in} = \sum_{i=1}^{n} y_{in_i} \; \forall x_n \in \{0, 1\} \tag{13.1}$$

$$f(y_{in}) = \begin{cases} 1 & y_{in} \geq \theta \\ 0 & y_{in} < \theta \end{cases} \tag{13.2}$$

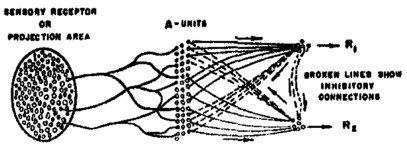
where θ is the threshold and y_{in} is the total net input signal received. The construction of this network follows propositional logics, and hence, it is theorized to simulate and construct *boolean logic handler*, for OR, AND, NAND or others. And with the parameter θ, its name as the TLU is common in electronics and computer systems. In principle , in their original paper *A logical calculus of the ideas immanent in nervous activity, 1943*, it is remarked that, their idea is to "[He] therefore attempted to record the behaviour of complicated nets in the notation of symbolic logic of propositions", in which the all-or-none principle of such time infers the absolute certainty of propositional logic. The goal is then apparent - to use symbolic and propositional logic to express the working of every networks, and hence applies it to the main structure.

This idea of a computational unit with receiver and activation patterns continues from then. The later idea includes Rosenblatt's perceptron (1957,1958), of which was based on works of McCulloch, the MCP or TLU, Culbertson (1956), Minsky (1956), Hebb (1949), and some of Von Neumann (1951, 1956). In their construction, per example for a photo-perceptron (optical signals), the organization of a perceptron is pretty much complex, based on three components:

1. The retina with localized connections, containing sensor points, called S-points.
2. The *projection area* A_I of which is described to be connected subsequently to the *sensor area* (this can be omitted, however, in case the next layer is directly connected to the retina), which receives a number of connections from the sensory points, and those formed a set called the *origin points* of such A-unit.
3. Between the projection area and the next layer here, we call as *association area*, connections are assumed to be random, scattered at random throughout the projection area.
4. The responses $R_1, R_2, \ldots$ are cells which respond in much the same fashion as the A-unit. Each response has a certain count of origin points located at random (similar to between A_I and A_{II}) in the A_{II} net.



(a)



(b)

Figure 13.1: (a) Figure of the original organization of the biological model of the brain functions. (b) Specifically, note that it is specifically for optical case, but can be extended to others type. Furthermore, the layer between last A-unit and the response units, there exists a pattern of feedback loop.

Descriptively, we describe Rosenblatt's perceptron as followed.

Definition 13.3.1 (Rosenblatt, 1957). *A perceptron (or linear classifier) is a function*

$$\begin{aligned} \text{lin} : \mathbb{R}^d &\to \mathbb{R} \\ \mathbf{x} &\to \mathbf{w} \cdot \mathbf{x} + b \end{aligned} \tag{13.3}$$

where the parameters to be learned are $\mathbf{w} \in \mathbb{R}^d$ *and* $b \in \mathbb{R}$.

A intricate detail to observe, is the fact that between A_{II} and the receptive field, there exists feedback loops, directly connected back to each other, based on firing principles and thereof. There exists inhibitory and excitatory feedback connections, which separated the propagation phase into the *predominant* and *postdominant* phases. Effectively, this brings the idea of tangent optimization to it for one of the first network to do so. A note on Rosenblatt's perceptron, though, is that there are no *activation function* in the network. Most notably, Rosenblatt's and McCulloch's perceptrons are often deferred to as either linear classifier, or *binary classifier* on linear space [Freund, Schapire, 1999].

Albeit the role of activation function is dubious, it was presented in the very first TLU as a threshold unit, effectively transformed TLU into a 2-partition boundary unit within the parameter space. In one way or another, we can think of activation function belongs to a collection of main operating mechanism of which the neuron is thought to be modelled as.

13.3.1 Multilayer perceptron (MLP)

Moving on, the limits of perceptron is well-known, at least in its concurrent form. By the standard linear-binary classifier, the "XOR problem" is unsolvable by the typical perceptron system. In fact, any data of which is *linearly inseparable* will render perceptron useless, albeit we still have the perceptron cycling theorem [Block, Levin, 1970]. Thereby, a different architecture is involved, namely the *multilayer perceptron architecture* (MLP).

It is here that we should note that the name multilayer perceptron brings a lot of controversy to the name on itself. The name directly (and incorrectly) infers to the fact that it contains only one more component: a lot of perceptrons in layers. However, the name is talking about the architecture of a single neuron, and we can see the idea of **layers** having introduced earlier on in Rosenblatt's perceptron model. Rather, it is the fact that multilayer perceptron is the architecture that comprises many components into layers, but not every component is a perceptron, and rather, they are in one way or another, *neuron-like* processing unit. Many of today's architecture relies on such interpretation, including the latest structures. Those, for example, Transformer,

CNN, and else, can be considered partially heuristic when considered to generality of the neurological processing operations, taking advantage of certain data configurations.

There are no definite definition of a multilayer perceptron. This is mainly because of its configuration and archetypical nature of being the 'way of construction' of meaningful compositions for smaller neural-like units, the name is given toward being an architecture instead on its own, in typical literature. [Ramchoun et al., 2016] [Roberts, Yaida, Hanin, 2021v2]

We would use MLP as a general interest in this paper.

13.3.2 Remark

Section 2 introduced the basic ground works for generally, all the currently constructed neural network system, per history and timeline of ideas. It leads to several questions:

1. Activation function is essentially to the formation of MLP. However, their appearance is not generally dated. Before that, MCP and Rosenblatt's perceptron also has activation functions, albeit they are threshold units containing logical syllogism of either 0 or 1, except for ADALINE with linear activation function.

   Multilayer perceptron is imperatively different, because it uses *non-linear* activation function. As we recalled, TLU can only solve the binary boundary problem, and while perceptron is better, it is ill-suited for problems that need non-linear solutions. Generally, it is true that linear unit can only interpret linear patterns and situations. Thereby, the non-linear activation function is considered a breakthrough, of which is prominent in the multilayer perceptron system. However, the question remains. What is the potential of an activation function?

2. Early in section 2, we refer to components of multilayer perceptron as neural-like components. But what exactly are them? In a loose sense, they can be considered to be processing unit, with input-output flow of operation, and certain operation mutating the input to certain form - or rather, mapping it from the input space $\mathcal{I}$ to the processor's space $\mathcal{P}$. But what is the true construction of such neural-like components. And further, what can be their concrete definition and theories?

3. What can be the representative mathematical object and operations that partially represents and interpret a single perceptron unit in actual system?

4. What is the *axiom* or assumption of the multilayer perceptron (for example, this question somewhat refers to Hebb's rule as one of the axiomatic condition of evolution of neural net, which then could be called as Hebbian neural architecture)?

5. What is exactly meant by *neural-like components*? From Rosenblatt to Marvin, their idea of components networks or *Society of Mind* in Marvin's case, but what exactly constitute the neural-like properties, and what constitute those components to works together?

For that, the next section will discuss the formal construction of *multilayer perceptron*.

IV. Appendix

Supplementary sections, containing further experimental results, more quantitative result handling, further discussion, additional details, analysis, unpublished opinion, and additional foundational comprehension.

Index

$\mathbb{R}$ standard neuron., 167
$\mathcal{N}_0$ simplex, 167
$\mathcal{N}_1$ simplex, 172

artificial, 19
axon, 156

brain lateralization, 156

certainty factor, 11
characterization, 38
composition, 47
compressed form, 53
computational rationality, 5
concentration inequalities, 83
conjunctions, 32
connectionism, 7
contradiction, 34

dendrites, 156
disjunction, 32

empirical risk, 113
ERC, 137
expert system, 10

function, 46

generalization risk, 113
gray model, 99
growth function, 142

halfspace, 144
Hebbian learning, 4
heliocentric topology, 162

image (function), 46
inference engine, 10
input unit, 165
Intelligence, 17
Intelligent, 17
intelligent agent, 6

knowledge database, 10

layer, 163
Logic Theorist, 6
logical equivalence, 35
logical operation, 32
logical quantification, 36
logical system, 32
logics, 31

martingale methods, 83
mathematical model, 94
mechanical unit, 165
mechanistic model, 98
minimized structure, 166
model, 90

Neural Doctrine, 159
neural vacancy path, 156
neuroglia, 157
neuron chaining, 169
noise, 133

open sentence, 32
output unit, 166

PAC learning, 124
phenomenological model, 98
physical model, 93
preimage, 48
proofs, 31
pseudometric space, 77

quantified statement, 36

Rademacher complexity, 137
Rademacher variables, 137
reduced system, 95
ReLU, 169

representation scheme, 126
Reticular Theory, 158
rogue class, 165
Russell's paradox, 41

scalar, 52
shattering, 145
sigma-algebra, 79
simulation, 92
single chaining, 169
standard class, 167
statement, 31
synapse, 157
system, 92
system parameter, 95

tautology, 34
theoretical model, 93

vector, 52

List of transfer functions

In our book, we mentioned a lot about specifically transfer functions in the neural construction. For convenience, here we summarize almost all transfer function of interest. What they are for? No idea (read the book, god-damn it!), but we will just search for them in here for a while, I guess.

Classical transfer functions

A lot of transfer function, or rather the modern derivative of it in the classical sense, can be taken from Demuth et al. [2014]. Here, the transfer function can be linear or nonlinear function, in which it is used to transform the input-output characteristic of a single-input neuron into the variety of the transfer function.

Hard limit (`hardlim[x]`)

The hard limit transfer function, used in distinctive categorical sorting, has its input-output characteristic as:

$$\texttt{hardlim}[x] = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases} \tag{13.4}$$

If we allow modification of the inhibitory value, that is, the zero, by putting it to another

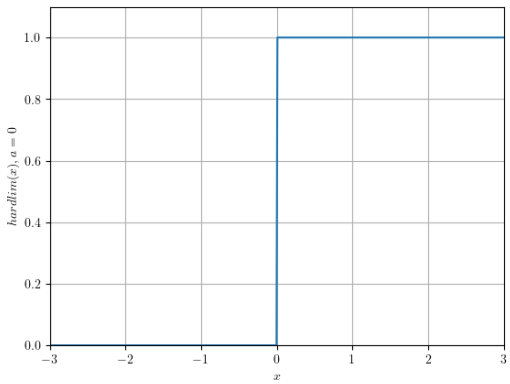


Figure 1: The typical hard limit transfer function with fixed a, and fixed range for x in $[0, 1]$.

variable a, the function turns into the dynamic hard limit function,

$$\texttt{varhardlim}[x] = \begin{cases} 0 & x < a \\ 1 & x \geq a \end{cases} \tag{13.5}$$

Actually, you can even give the function the value range of absolute jump to be more than $[0, 1]$, though fundamentally, according to the logical design, it is not interpretable to anything substantial.

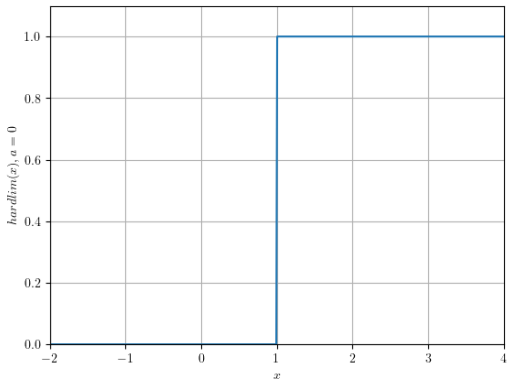


Figure 2: The typical hard limit transfer function with variable inhibition a, and fixed range for x in $[0, 1]$.

Symmetric hard limit (`hradlims[x]`)

This one is a variation of `hardlim`, in which the discrete binary channel is $\{-1, +1\}$ instead. Hard limit is more fitting for probability of logic setting, while symmetric hard limit is more favourable in certain specification, for example, for fuzzy logical domain or directed value functions. Coincidentally, this is also the range that certain sigmoidal variation takes place. Symmetric hard limit is then defined by

$$\texttt{hardlims}[x] = \begin{cases} -1 & x < 0 \\ +1 & x \geq 0 \end{cases} \tag{13.6}$$

Linear family (`satlin[x]`, `satlins[x]`, `purelin[x]`)

There are many ways to structure the linear input-ouput processing node. Usually, we will have the pure linear channel `purelin`, the saturating linear channel `satlin`, and the symmetric variation of the saturating linear channel `satlins`. Because they belong to the same family. The linear one is simple.

$$\texttt{purelim}[x] = x \tag{13.7}$$

For saturating linear, we have its signal inhibited toward the two ends:

$$\texttt{satlin}[x] = \begin{cases} 0 & x < 0 \\ x & 0 \leq x \leq 1 \\ 1 & x > 1 \end{cases} \tag{13.8}$$

A generalization of this is taken in the form of a functional $f(x)$ enclosed within this range. That is,

$$\texttt{varsatlin}[x] = \begin{cases} 0 & x < 0 \\ f(x) & 0 \leq x \leq 1 \\ 1 & x > 1 \end{cases} \tag{13.9}$$

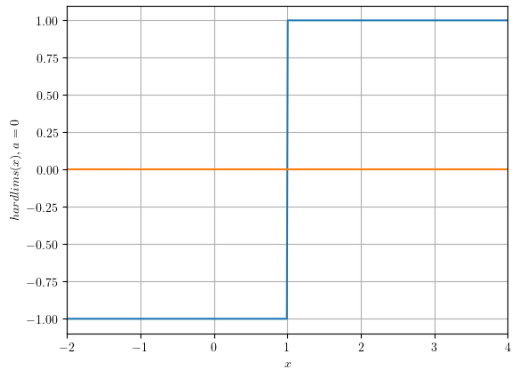


Figure 3: The typical symmetric hard limit transfer function with static inhibition a, and fixed range for x in $[-1, +1]$. As specified, this is the normal-extended range.

which might lead to undesire behaviours or simply non-continuous values, but we will have to resolve that later on. If ever. And finally, the symmetric version of the saturating linear functional,

$$\texttt{satlin}[x] = \begin{cases} -1 & x < 0 \\ x & 0 \leq x \leq 1 \\ 1 & x > 1 \end{cases} \tag{13.10}$$

which will also have the same generalized form.

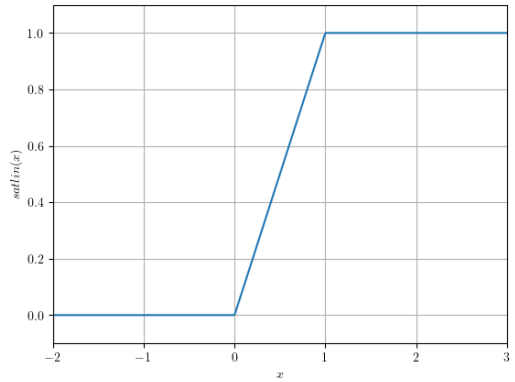


Figure 4: The saturating linear with linear region of $[0, 1]$. A smoother variation would be something like sigmoidal functions, that is.

Sigmoid (`sigmoid[x]`) and log-sigmoid (`logsigmoid[x]`)

The sigmoid function is fairly simple. Instead of giving piecewise saturating condition, we find the expression that gives pairwise, two-sided contiuously saturated function, expressed by:

$$\texttt{sigmoid}[x] = \frac{1}{1 + e^{-x}} \tag{13.11}$$

A fairly complicated and often reductive version of it is the log-sigmoid function, as

$$\texttt{logsigmoid}[x] = \log\left(\frac{1}{1 + e^{-x}}\right) = -\log\left(1 + \exp\left(-x\right)\right) \tag{13.12}$$

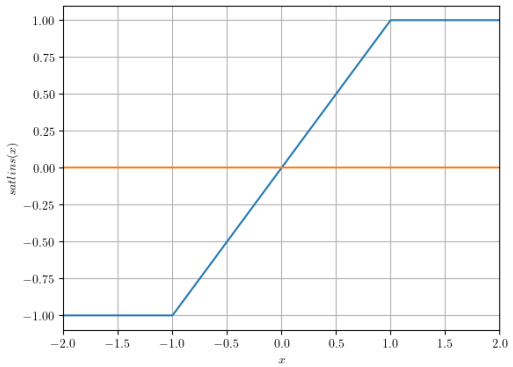


Figure 5: The symmetric saturating linear with linear region of $[-1, 1]$, a positive-negative variation of the saturating linear.

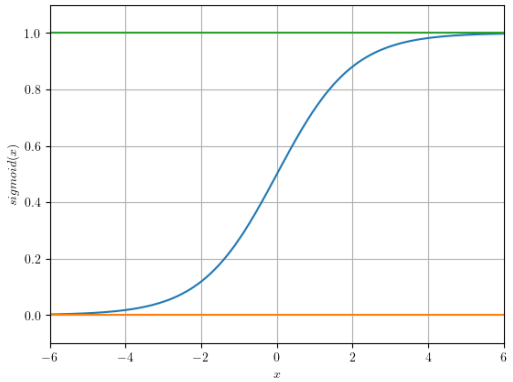


Figure 6: The sigmoidal function channel

Interestingly, the differentiation operator on log-sigmoid gives the sigmoid function, while sigmoid's differentiation gives $\texttt{sigmoid}[x](1 - \texttt{sigmoid}[x])$.

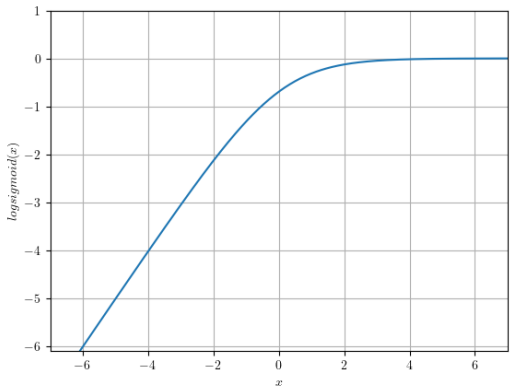


Figure 7: The logarithmic sigmoidal function channel. Notice that the range of `logsigmoid` is $[-\infty, 0]$, making it somewhat weird of a choice for a transfer function.

Hyperbolic tangent (`tansig[x]`)

The hyperbolic tangent the adoption of the hyperbolic function to be transfer function. As such, its range also lies in $[-1, 1]$, making it on par with variations of symmetric saturation function. Normally, we would regard this as the somewhat narrow (by width) symmetric version of sigmoid. It is formulated as:

$$\texttt{tansig}[x] = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(x)} \tag{13.13}$$

Also interestingly, hyperbolic tangent is self-referential, evidentual of the derivative:

$$\frac{d}{dx}\tanh(x) = 1 - \tanh^2(x)$$

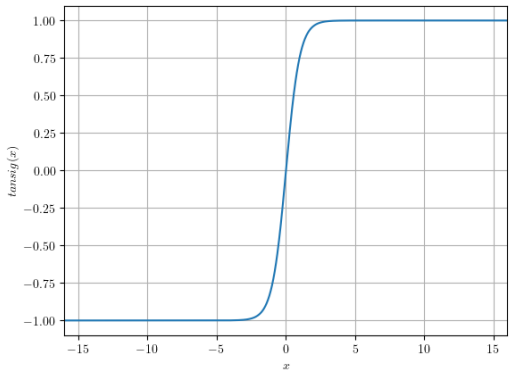


Figure 8: The hyperbolic tangent transfer function channel.

The uses of those function can be interpreted to be quite similar to how we can formulate the binary classification, or binary categorization problem-solving solution.

Bibliography

Bias-variance tradeoffs in program analysis | Proceedings of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, a. URL https://dl.acm.org/doi/10.1145/2535838.2535853.

Neural Networks and the Bias/Variance Dilemma | MIT Press Journals & Magazine | IEEE Xplore, b. URL https://ieeexplore.ieee.org/document/6797087.

[PDF] A Unifeid Bias-Variance Decomposition and its Applications | Semantic Scholar, c. URL https://www.semanticscholar.org/paper/A-Unifeid-Bias-Variance-Decomposition-and-its-Domingos/e1ed9d24db5e8f7ab326aeb797e965a94f5ad6d3.

[PDF] A Unifeid Bias-Variance Decomposition and its Applications | Semantic Scholar. URL https://www.semanticscholar.org/paper/A-Unifeid-Bias-Variance-Decomposition-and-its-Domingos/e1ed9d24db5e8f7ab326aeb797e965a94f5ad6d3.

Mathematical Modeling and Simulation: Introduction for Scientists and Engineers, 2nd Edition | Wiley, 2024. URL https://www.wiley.com/en-us/Mathematical+Modeling+and+Simulation%3A+Introduction+for+Scientists+and+Engineers%2C+2nd+Edition-p-9783527839407.

Kumar Abhishek, Sneha Maheshwari, and Sujit Gujar. Introduction to concentration inequalities, 2019. URL https://arxiv.org/abs/1910.02884.

Stephen Andrilli and David Hecker. Chapter 1 - vectors and matrices. In Stephen Andrilli and David Hecker, editors, *Elementary Linear Algebra (Fourth Edition)*, pages 1–77. Academic Press, Boston, fourth edition edition, 2010. ISBN 978-0-12-374751-8. doi: https://doi.org/10.1016/B978-0-12-374751-8.00001-9. URL https://www.sciencedirect.com/science/article/pii/B9780123747518000019.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine learning practice and the bias-variance trade-off. *Proc. Natl. Acad. Sci. U.S.A.*, 116(32):15849–15854, August 2019a. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1903070116. URL http://arxiv.org/abs/1812.11118. arXiv:1812.11118 [cs, stat].

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine learning practice and the bias-variance trade-off. *Proc. Natl. Acad. Sci. U.S.A.*, 116(32):15849–15854, August 2019b. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1903070116. URL http://arxiv.org/abs/1812.11118. arXiv:1812.11118 [cs, stat].

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, February 2013. ISBN 978-0-19-953525-5. doi: 10.1093/acprof:oso/9780199535255.001.0001. URL https://doi.org/10.1093/acprof:oso/9780199535255.001.0001.

Olivier Bousquet, Steve Hanneke, Shay Moran, Ramon van Handel, and Amir Yehudayoff. A theory of universal learning, 2020. URL https://arxiv.org/abs/2011.04483.

Gordon Briggs. Machine ethics , the frame problem , and theory of mind. 2014. URL https://api.semanticscholar.org/CorpusID:14954096.

Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, July 2017. ISSN 1558-0792. doi: 10.1109/msp.2017.2693418. URL http://dx.doi.org/10.1109/MSP.2017.2693418.

Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021. URL https://arxiv.org/abs/2104.13478.

Gavin Brown and Riccardo Ali. Bias/variance is not the same as approximation/estimation. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=4TnFbv16hK.

Sebastian Buschjäger, Lukas Pfahler, and Katharina Morik. Generalized Negative Correlation Learning for Deep Ensembling, December 2020a. URL http://arxiv.org/abs/2011.02952. arXiv:2011.02952 [cs, stat].

Sebastian Buschjäger, Lukas Pfahler, and Katharina Morik. Generalized Negative Correlation Learning for Deep Ensembling, December 2020b. URL http://arxiv.org/abs/2011.02952. arXiv:2011.02952 [cs, stat].

Nello Cristianini and John Shawe-Taylor. An introduction to support vector machines and other kernel-based learning methods. 2000. URL https://api.semanticscholar.org/CorpusID:60486887.

Stéphane d' Ascoli, Levent Sagun, and Giulio Biroli. Triple descent and the two kinds of overfitting: where & why do they appear? In *Advances in Neural Information Processing Systems*, volume 33, pages 3058–3069. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1fd09c5f59a8ff35d499c0ee25a1d47e-Abstract.html.

Xander Davies, Lauro Langosco, and David Krueger. Unifying Grokking and Double Descent, March 2023. URL http://arxiv.org/abs/2303.06173. arXiv:2303.06173 [cs].

Howard B. Demuth, Mark H. Beale, Orlando De Jess, and Martin T. Hagan. *Neural Network Design*. Martin Hagan, Stillwater, OK, USA, 2nd edition, 2014. ISBN 0971732116.

Rene? Descartes. *Discourse on Method*. Harmondsworth, Penguin, Harmondsworth,, 1950.

Pedro M. Domingos. A unified bias-variance decomposition for zero-one and squared loss. In *AAAI/IAAI*, 2000a. URL https://api.semanticscholar.org/CorpusID:2063488.

Pedro M. Domingos. A Unifeid Bias-Variance Decomposition and its Applications. In *Semantic Scholar*, June 2000b. URL https://www.semanticscholar.org/paper/A-Unifeid-Bias-Variance-Decomposition-and-its-Domingos/e1ed9d24db5e8f7ab326aeb797e965a94f5ad6d3.

Hubert L. Dreyfus. From micro-worlds to knowledge representation : Ai at an impasse. 1979.

Chris Smith et al. The history of artificial intelligence. Technical review, 2006.

Molavi et al. Model Complexity, Expectations, and Asset Prices | The Review of Economic Studies | Oxford Academic. URL https://academic.oup.com/restud/article-abstract/91/4/2462/7222145?redirectedFrom=fulltext&login=false.

Scott Fortmann. Understanding the Bias-Variance Tradeoff, 2012. URL https://scott.fortmann-roe.com/docs/BiasVariance.html.

Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992. doi: 10.1162/neco.1992.4.1.1.

Joseph von Gerlach. Ueber die structur der grauen substanz des menschlichen grosshirns. vorläufige mitheilung. *Centralblatt für die medizinischen Wissenschaften*, 10:273–288, 1872. URL https://www.booklooker.de/B%C3%BCcher/Joseph-Gerlach%2BUeber-die-Structur-der-grauen-Substanz-des-menschlichen-Grosshirns-Vorl%C3%A4ufige/id/A02ohXHY01ZZu. Introduced the reticular theory of the nervous system.

Ashok Goel. Looking back, looking ahead: Symbolic versus connectionist ai. *AI Magazine*, 42(4):83–85, Jan. 2022. doi: 10.1609/aaai.12026. URL https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/15111.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT Press, 2016.

Ulf Grenander. On empirical spectral analysis of stochastic processes. *Arkiv för Matematik*, 1:503–531, 1952. URL https://api.semanticscholar.org/CorpusID:122878699.

Jarek Gryz. The frame problem in artificial intelligence and philosophy. *Filozofia Nauki*, 21:15–30, 06 2013.

Bruce Hajek and Maxim Raginsky. *Statistical Learning Theory*, volume 1. 2021. URL https://maxim.ece.illinois.edu/teaching/SLT/.

William L. Hamilton. Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159.

Thomas Hellström, Virginia Dignum, and Suna Bensch. Bias in Machine Learning – What is it Good for?, September 2020. URL http://arxiv.org/abs/2004.00686. arXiv:2004.00686 [cs].

Xia Hu, Lingyang Chu, Jian Pei, Weiqing Liu, and Jiang Bian. Model Complexity of Deep Learning: A Survey, August 2021. URL http://arxiv.org/abs/2103.05127. arXiv:2103.05127 [cs].

Gareth James, Trevor Hastie, Robert Tibshirani, and Daniela Witten. *An introduction to statistical learning : with applications in R*. New York : Springer, [2013] ©2013, 2013. URL https://search.library.wisc.edu/catalog/9910207152902121.

Eric R. Kandel, John D. Koester, Sarah H. Mack, and Steven A. Siegelbaum. McGraw Hill, New York, NY, 2021. URL accessbiomedicalscience.mhmedical.com/content.aspx?aid=1180370208.

Michael J. Kearns and Umesh V. Vazirani. *An introduction to computational learning theory*. MIT Press, Cambridge, MA, USA, 1994. ISBN 0262111934.

Mohammad Emtiyaz Khan and Håvard Rue. The Bayesian Learning Rule, June 2024. URL http://arxiv.org/abs/2107.04562. arXiv:2107.04562 [cs, stat] version: 4.

Marc Lafon and Alexandre Thomas. Understanding the Double Descent Phenomenon in Deep Learning, March 2024. URL http://arxiv.org/abs/2403.10459. arXiv:2403.10459 [cs, stat].

Chris Yuhao Liu and Jeffrey Flanigan. Understanding the role of optimization in double descent, 2023. URL https://arxiv.org/abs/2312.03951.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks, 2025. URL https://arxiv.org/abs/2404.19756.

Dmytro Lopushanskyy and Borun Shi. Graph neural networks on graph databases, 2024. URL https://arxiv.org/abs/2411.11375.

Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, December 1943. ISSN 1522-9602. doi: 10.1007/BF02478259. URL https://doi.org/10.1007/BF02478259.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A Survey on Bias and Fairness in Machine Learning, January 2022. URL http://arxiv.org/abs/1908.09635. arXiv:1908.09635 [cs].

Kostas Metaxiotis and J-E Samouilidis. Expert systems in medicine: Academic illusion or real power? *Information Management & Computer Security*, 8:75–79, 05 2000. doi: 10.1108/09685220010694017.

Marvin L. Minsky and Seymour A. Papert. *Perceptrons: expanded edition*. MIT Press, Cambridge, MA, USA, 1988. ISBN 0262631113.

Isra Mishqat. The Neuron Doctrine (1860-1895) | Embryo Project Encyclopedia. URL https://embryo.asu.edu/pages/neuron-doctrine-1860-1895.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012. ISBN 026201825X.

Pooya Molavi, Alireza Tahbaz-Salehi, and Andrea Vedolin. Model Complexity, Expectations, and Asset Prices. *The Review of Economic Studies*, 91(4):2462–2507, July 2024. ISSN 0034-6527. doi: 10.1093/restud/rdad073. URL https://doi.org/10.1093/restud/rdad073.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep Double Descent: Where Bigger Models and More Data Hurt, December 2019. URL http://arxiv.org/abs/1912.02292. arXiv:1912.02292 [cs, stat].

Brady Neal. On the bias-variance tradeoff: Textbooks need an update, 2019. URL https://arxiv.org/abs/1912.08286.

A. Newell and H. Simon. The logic theory machine–a complex information processing system. *IRE Transactions on Information Theory*, 2(3):61–79, 1956. doi: 10.1109/TIT.1956.1056797.

Amanda Olmin and Fredrik Lindsten. Towards understanding epoch-wise double descent in two-layer linear neural networks, 2024. URL https://arxiv.org/abs/2407.09845.

Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1ldO2EFPr.

David Pfau. A generalized bias-variance decomposition for bregman divergences. Technical report, 2013.

Dale Purves, George J. Augustine, David Fitzpatrick, William C. Hall, Anthony-Samuel LaMantia, James O. McNamara, and S. Mark Williams, editors. *Neuroscience, 3rd ed*. Neuroscience, 3rd ed. Sinauer Associates, Sunderland, MA, US, 2004. ISBN 978-0-87893-725-7. Pages: xix, 773.

Victor Quétu and Enzo Tartaglione. Can we avoid Double Descent in Deep Neural Networks? In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 1625–1629, October 2023a. doi: 10.1109/ICIP49359.2023.10222624. URL http://arxiv.org/abs/2302.13259. arXiv:2302.13259 [cs].

Victor Quétu and Enzo Tartaglione. Can we avoid Double Descent in Deep Neural Networks? In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 1625–1629, October 2023b. doi: 10.1109/ICIP49359.2023.10222624. URL http://arxiv.org/abs/2302.13259. arXiv:2302.13259 [cs].

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408, 1958. URL https://api.semanticscholar.org/CorpusID:12781225.

Jairo A. Rozo, Irene Martínez-Gallego, and Antonio Rodríguez-Moreno. Cajal, the neuronal theory and the idea of brain plasticity. *Front Neuroanat*, 18:1331666, February 2024. ISSN 1662-5129. doi: 10.3389/fnana.2024.1331666. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10910026/.

Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, USA, 3rd edition, 2009. ISBN 0136042597.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. doi: 10.1109/TNN.2008.2005605.

Rylan Schaeffer, Mikail Khona, Zachary Robertson, Akhilan Boopathy, Kateryna Pistunova, Jason W. Rocks, Ila Rani Fiete, and Oluwasanmi Koyejo. Double Descent Demystified: Identifying, Interpreting & Ablating the Sources of a Deep Learning Puzzle, March 2023. URL http://arxiv.org/abs/2303.14151. arXiv:2303.14151 [cs, stat].

William Seager. Frame problems, emotions and axiological projectionism. *Philosophical report*, 2010s (or older).

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, USA, 2014. ISBN 1107057132.

Murray Shanahan. The Frame Problem. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Spring 2016 edition, 2016.

Rahul Sharma and Alex Aiken. Bias-variance tradeoffs in program analysis. In *Proceedings of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '14, pages 127–137, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 978-1-4503-2544-8. doi: 10.1145/2535838.2535853. URL https://doi.org/10.1145/2535838.2535853.

Cheng Shi, Liming Pan, Hong Hu, and Ivan Dokmanić. Homophily modulates double descent generalization in graph convolution networks, 2024. URL https://arxiv.org/abs/2212.13069.

Tom F. Sterkenburg. Statistical learning theory and occam's razor: The core argument. *Minds and Machines*, 35(1), November 2024. ISSN 1572-8641. doi: 10.1007/s11023-024-09703-y. URL http://dx.doi.org/10.1007/s11023-024-09703-y.

Masashi Sugiyama. *Introduction to Statistical Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2015. ISBN 9780128023501.

James H. Tanis, Chris Giannella, and Adrian V. Mariano. Introduction to graph neural networks: A starting point for machine learning engineers, 2024. URL https://arxiv.org/abs/2412.19419.

L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, November 1984. ISSN 0001-0782. doi: 10.1145/1968.1972. URL https://doi.org/10.1145/1968.1972.

Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer: New York, 1999.

Petar Veličković. Everything is connected: Graph neural networks. *Current Opinion in Structural Biology*, 79:102538, April 2023. ISSN 0959-440X. doi: 10.1016/j.sbi.2023.102538. URL http://dx.doi.org/10.1016/j.sbi.2023.102538.

Janet F. Werker and Richard C. Tees. Cross-language speech perception: Evidence for perceptual reorganization during the first year of life. *Infant Behavior and Development*, 7(1):49–63, 1984. ISSN 0163-6383. doi: https://doi.org/10.1016/S0163-6383(84)80022-3. URL https://www.sciencedirect.com/science/article/pii/S0163638384800223.

Zitong Yang, Yaodong Yu, Chong You, Jacob Steinhardt, and Yi Ma. Rethinking Bias-Variance Trade-off for Generalization of Neural Networks, December 2020. URL http://arxiv.org/abs/2002.11328. arXiv:2002.11328 [cs, stat].

Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. Dive into deep learning, 2023. URL https://arxiv.org/abs/2106.11342.