Discrete Choice Analysis: micro-econometrics and machine learning approaches

Machine learning approaches for discrete choice analysis

Sander van Cranenburgh

Associate Professor

CityAI lab
Section of Transport and Logistics
Faculty of Technology, Policy and Management
Delft University of Technology

Course set-up

<u>Motivation</u>

- Machine learning is increasingly used for choice behaviour modelling

<u>Learning objectives</u>

After this course, you can:

- Explain and apply some key concepts of Machine Learning (ML).
- Discuss on the strength and weaknesses of ML for choice behaviour modelling
- Discuss how ANNs and theory-driven discrete choice models can, and cannot, be brought together (do's and don'ts)
- Use ANNs and XAI for analysing your own choice data

<u>Teaching method</u>

- Oral lectures
- Lab sessions with **Jupyter notebooks** (some coding skills needed)

Course set-up

Focus

- Artificial Neural Networks (ANNs)
- Choice data (Not per se big Data!)
- Travel choice behaviour

Lost in translation:

- I use **two** vocabularies: Machine learning & Discrete choice modelling.

    Observation = Sample, Instance
    Estimation = Training
    Attribute = Feature
    Logit function = Softmax
    Beta = weight

Target audience

- Intermediate level of knowledge on choice modelling
- Limited knowledge on machine learning
- Basic Python programming skills

Course set-up

Expectations:

- Frontier of the field; forward looking; not a polished textbook story

- Balanced account on strength and weaknesses of theory-driven (micro econometric) and data-driven (machine learning) methods for discrete choice analysis

- Not complete in any regard w.r.t. machine learning

Course material:

1. Van Cranenburgh, S., Wang, S., Vij, A., Pereira, F., & Walker, J. (2022). Choice modelling in the age of machine learning-discussion paper. *Journal of Choice Modelling*, *42*, 100340.
2. Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.".
3. Sifringer, B., Lurkin, V., & Alahi, A. (2020). Enhancing discrete choice models with representation learning. *Transportation Research Part B: Methodological*, *140*, 236-261.
4. Lazar, A., Jin, L., Brown, C., Spurlock, C. A., Sim, A., & Wu, K. (2021, December). Performance of the Gold Standard and Machine Learning in Predicting Vehicle Transactions. In *2021 IEEE International Conference on Big Data* (Big Data) (pp. 3700-3704). IEEE.
5. Molnar, C. (2020). *Interpretable machine learning*. Lulu. com

Programme (day 1)

Lecture: Introduction to machine learning for choice modellers

10:00 – 10:30   Machine learning fundamental

10:30 – 11:15   Theory-driven vs data-driven approaches & Knowledge discovery

11:15 – 11:30   ☕

Lecture: Artificial Neural Networks & Training

11:30 – 12:00   Artificial Neural Networks

12:00 – 12:30   Training and hyperparameter tuning & Performance metrics

Lunch:

12:30 – 13:30

Lab session 1: Artificial Neural Networks for Discrete Choice Analysis

13:30 – 15:30   Artificial neural networks

15:30 – 16:00   Discussion/reflection on the use of ANNs for choice behaviour analysis

Programme (day 2)

Lecture: Behavioural insights from ANNs

10:00 – 10:30   eXplainable AI (XAI)

10:30 – 11:00   Hybrid models and SHAP values

11:00 – 11:15   ☕

Lecture: Data requirements for ANNs

11:15 – 12:00   How many observations do I need, and more

Lunch:

12:00 – 13:00

Lab session 2: Artificial Neural Networks for Discrete Choice Analysis - Behavioural insights

13:00 – 15:00   Behavioural insights

                        A. The hybrid ANN-MNL model

                        B. Using SHAP values to improve model specifications + COMPETITION

15:00 – 15:30   Reflection and closure

Introduction to machine learning for choice modellers

What is it, and what is it not?

Machine learning fundamentals

Machine learning fundamentals

Machine Learning (ML) is the field of study that gives computers the ability to learn <u>without being explicitly programmed</u>." – Arthur Samuel, 1959
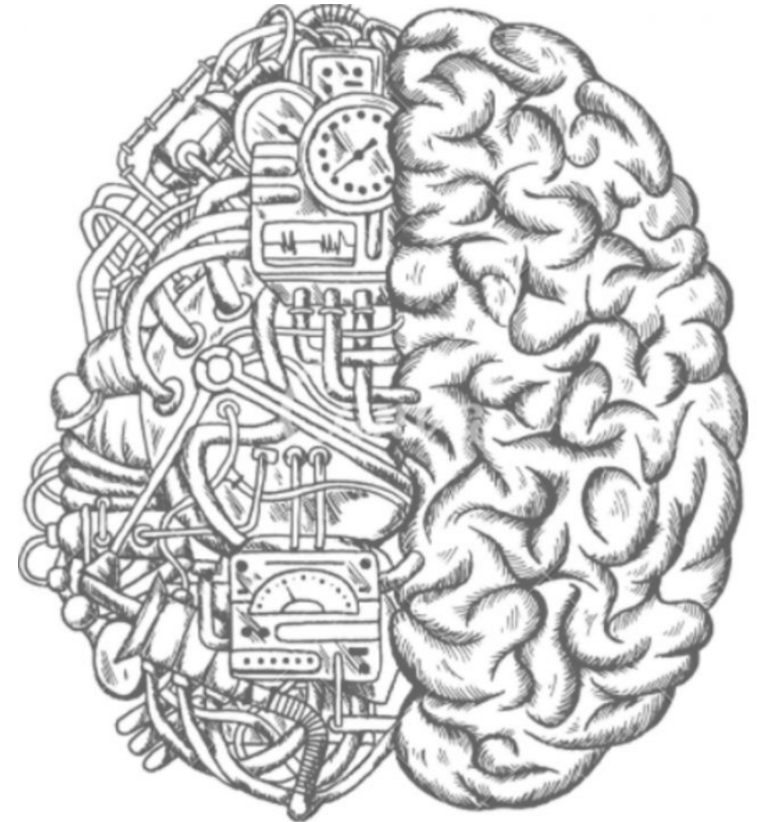
Examplar applications of ML
- Spam filter for email
- Chat bots
- Credit card fraud detection
- Placing Ads on websites
- Recommendation systems
- …

1. Why can't we use 'classic' statistical approaches for these applications? 🤔

2. What does "without explicitly programmed" mean?

3. Choice modelling is different from the exemplar applications, in what ways?

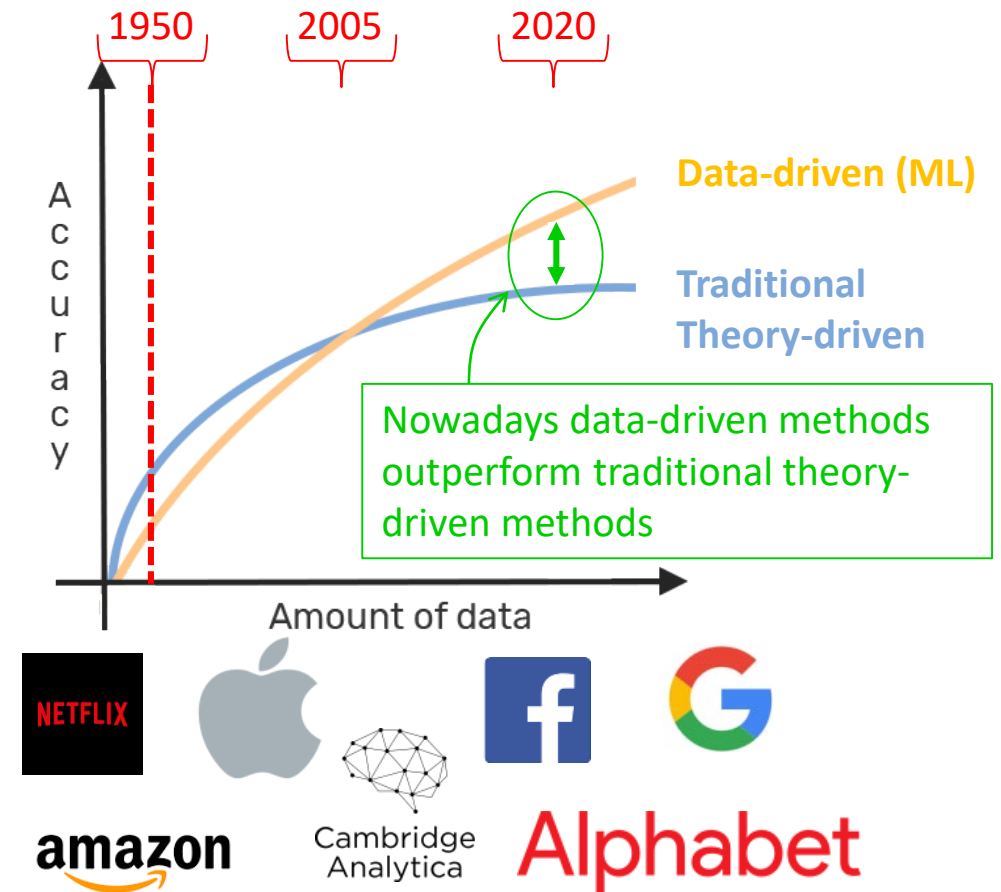# Machine learning fundamentals

Popular ML approaches include:

- Regression
- Logistic regression*
- Decision tree learning
- Random Forest
- Artificial Neural Networks
- Support vector machines
- Gradient Boosting
- Clustering
- Bayesian networks
- Association rule learning
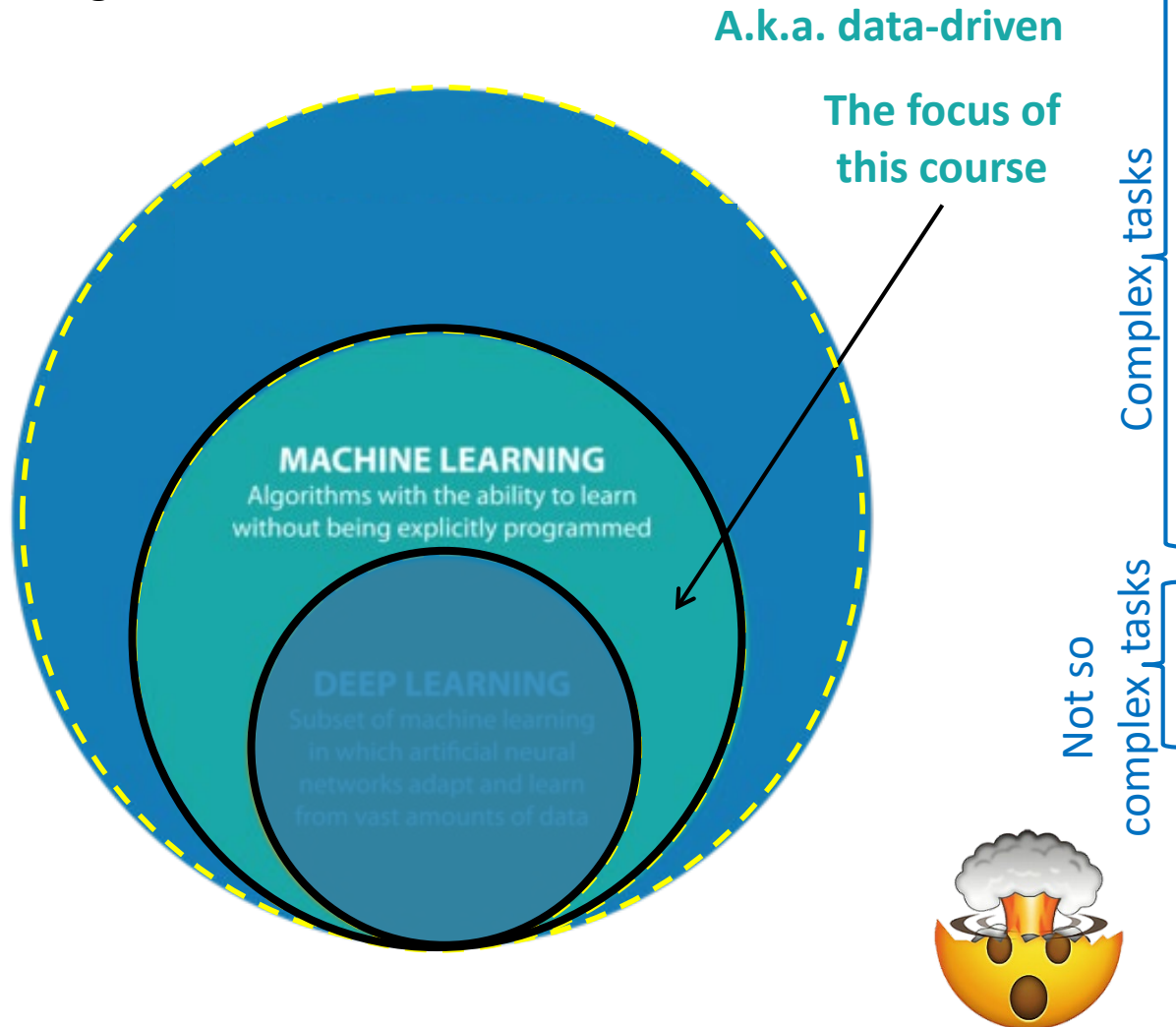- ...
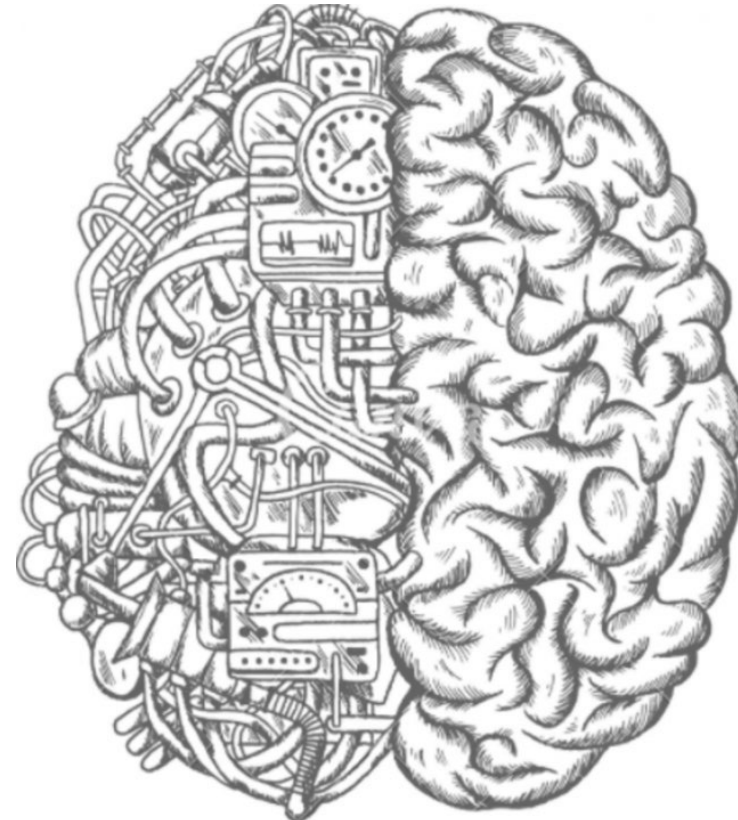
Machine learning fundamentals

Why focus on ANNs?

- Proven to be powerful in combination with large data + massive computational power

- Suitable classifiers for prediction choice behaviour

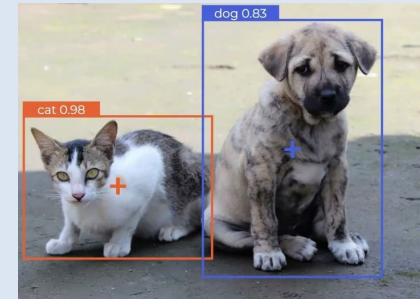- Most often used ML model in state-of-the- art choice modelling

Machine learning fundamentals

Categories within data-driven AI

A.k.a. data-driven

The focus of this course

MACHINE LEARNING
Algorithms with the ability to learn without being explicitly programmed

DEEP LEARNING
Subset of machine learning in which artificial neural networks adapt and learn from vast amounts of data

Deep learning excels on highly **complex tasks**

Complex tasks

$f := X \rightarrow Y$
1 x cat
1 x dog

*"This is penny wise, pound foolish"* $f := X \rightarrow Y$ *"Dit is verkeerde zuinigheid"*

Not so complex tasks

|	Route A	Route B	Route C
Travel time (one-way)	23 minutes	27 minutes	35 minutes
Travel cost (one-way)	€ 6	€ 4	€ 3

$P(Y)$

→Deep learning is **NOT** the right tool for (classic) modelling choice behaviour because choice behaviour is **NOT complex enough**

Machine learning fundamentals

- **Learning** refers to inferring a **function** g that maps an input to an output based on example input-output pairs
- When a machine can perform a task better after having been presented new data, we say the machine has **learned**
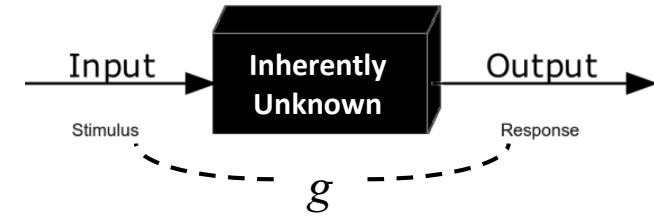
→ **Learning** is related to, but not similar to **estimation**. Estimation focusses on inference of **parameters**
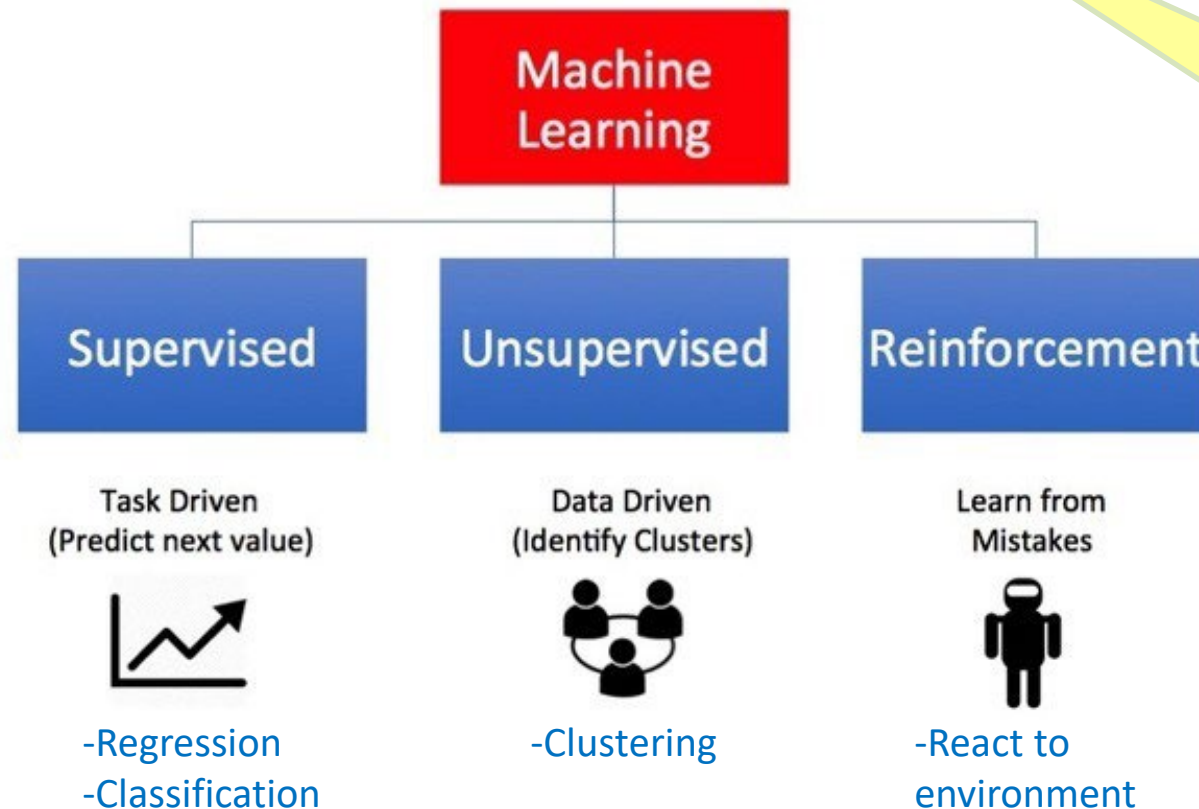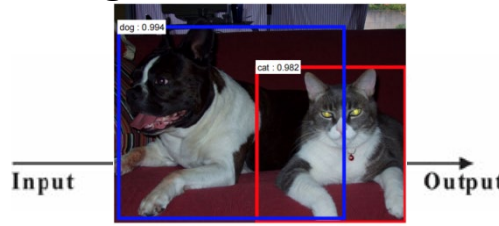
Machine learning fundamentals

Supervised Learning

- Labelled data (X,Y)

cat   cat   cat   dog   dog

- Training the network involves, learning to give the right answer
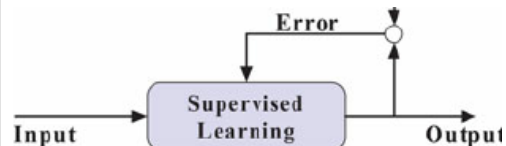
- Job - to replicate the right answers

Input   Output

Machine learning fundamentals

Supervised Learning	Unsupervised learning
• Labelled data (X,Y) cat cat cat dog dog	• Unlabelled data (X) e.g. travel patterns
• Training the network involves, learning to give the right answer	• Job - to find structure in the data
• Job - to replicate the right answers	

Machine learning fundamentals

Supervised Learning	Unsupervised learning	Reinforcement learning
• Labelled data (X,Y)	• Unlabelled data (X) e.g. travel patterns	• There is no target data
• Training the network involves, learning to give the right answer	• Job - to find structure in the data	• Reinforcement works with positive / negative rewards after winning / losing
• Job - to replicate the right answers		• Multiple decisions over time

Machine learning fundamentals

Inverted pendulum

https://youtu.be/J7E6_my3CHk

The laws of gravity are <u>learned</u> by the algorithm!

Reinforcement learning

- There is no target data

- Reinforcement works with positive / negative rewards after winning / losing

- Multiple decisions over time

Machine learning fundamentals

Supervised Learning

- Labelled data (X,Y)
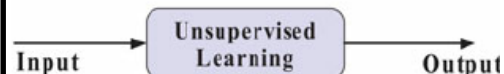

cat   cat   cat   dog   dog

- Training the network involves, learning to give the right answer

- Job - to replicate the right answers

Unsupervised learning

- Unlabelled data (X) e.g. travel patterns

- Job - to find structure in the data
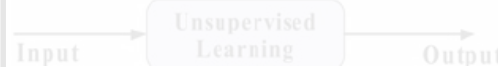
Reinforcement learning

- There is no target data

- Reinforcement works with positive / negative rewards after winning / losing
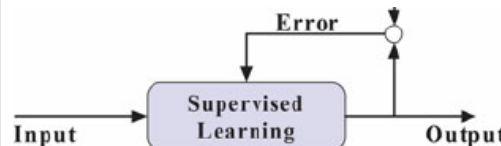
- Multiple decisions over time

18

Theory-driven vs data-driven models
(for supervised learning)

Theory-driven vs data-driven

So…. how is ML different from (theory-driven) discrete choice modelling?

Looking at the formulas for a **logistic regression model** and a (linear-additive) **RUM-MNL model** we see they **are <u>exactly</u> the same**. **Yet**, logistic regression is called ML (data-driven), while RUM-MNL is called theory-driven

Why?

$$P_{in} = \frac{e^{Vi}}{\sum_J e^{V_j}} \qquad V_i = \sum_m \beta_m x_{imn}$$

Theory-driven vs data-driven

Theory-driven

Guiding principle
Data generating process is a stochastic model

→ Analyst imposes structure **based on theory**

Input → $f := X \rightarrow Y$ → Output
Stimulus   Response

→ Set of parameters, sign. levels

How does this work out in choice modelling?

Choice task → Choice
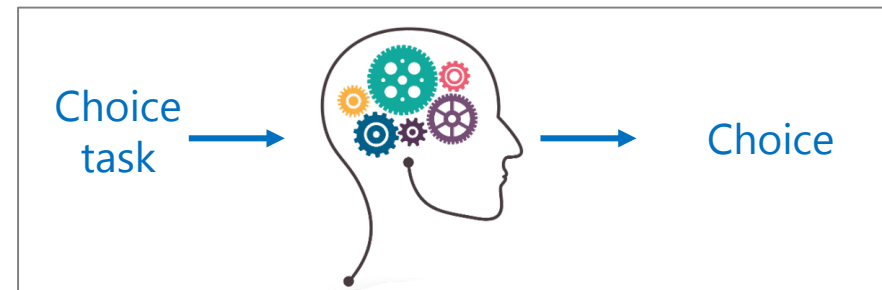
Random Utility Theory: →Samuelson 1948→Luce 1959 → Lancaster 1966 → McFadden, 1974

- Alternative i is chosen if and only if $U_i > U_j \ \forall j \neq i$

- $U_i = V_i + \varepsilon_i$

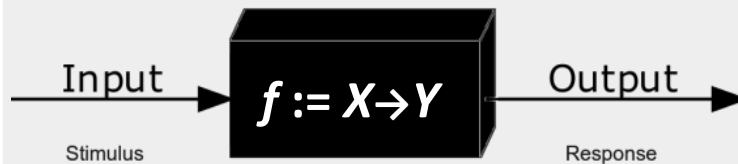- $V_i = f(x_{im}, \beta), \ \varepsilon_i \sim G(\circ)$

- $V_i = \sum_m \beta_m x_{im}$

Theory-driven vs data-driven

Theory-driven

Guiding principle
Data generating process is a stochastic model

→ Analyst imposes structure based on theory

Input → $f := X \rightarrow Y$ → Output

Stimulus   Response

→ Set of parameters, sign. levels

How does this work out in choice modelling?

Choice task → → Choice
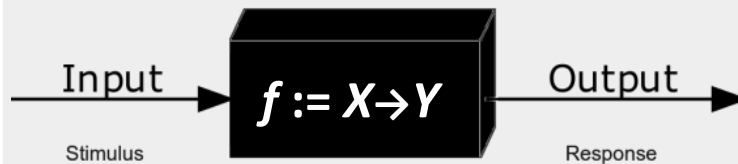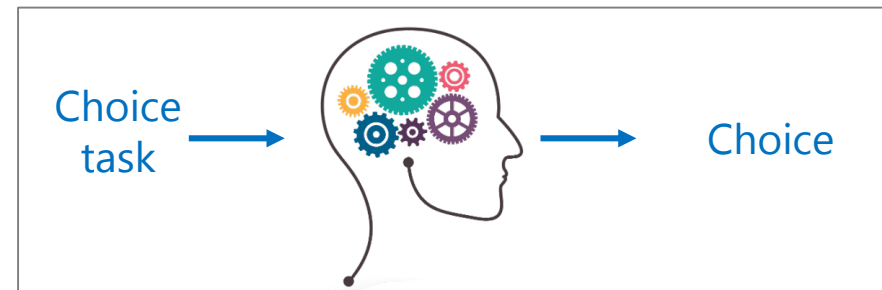
Behavioural __THEORY__ provides:
1. **Meaning to model parameters: e.g. β = marginal utility**
2. **Guidance in model specification**
3. **Economic outputs: WtP, Value-of-time → Appraisal, e.g. Cost-Benefit Analysis (CBA)**
4. **Solid ground for extrapolation → long-term forecasting**
5. **Elegant communication to policy-makers**

Theory-driven vs data-driven

Theory-driven
(discrete choice models)

Guiding principle
Data generating process is a stochastic model

→ Analyst imposes structure based on theory

Input → $f := X \rightarrow Y$ → Output

Stimulus   Response
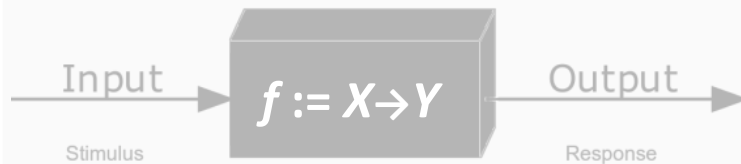
→ Set of parameters, sign. levels

Data-driven
(supervised machine learning)

Guiding principle
Data generating process is complex and cannot be known

→ Analyst does not impose structure: g needs to be learned from the data

Input → **Inherently Unknown** → Output

Stimulus   Response

g

→(Very good) prediction

Theory-driven vs data-driven

Strengths and weaknesses

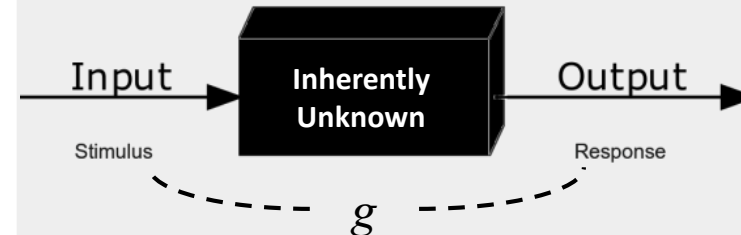|	Theory-driven (discrete choice models)	🙂 🙃	Data-driven (supervised machine learning)

Theory-driven
(discrete choice models)

Data-driven
(supervised machine learning)

🙂 🙃

Strengths
- Good interpretation, e.g.
  – Model parameters
  – Statistical measures for model fit comparison
- Transparent
- Solid ground for extrapolation
- Economic outputs (RUM)

Weaknesses
- Assumption-rich
- Restrictive on data
- Low prediction performance

Weaknesses
- Poor interpretation:
  – No model parameters
  – No statistical measures of model fit comparison
- Opaque
- Shaky extrapolation
- Unclear economic outputs

Strengths
- Assumption-poor
- Not restrictive on data
- High prediction performance

Important for discrete choice analysis?

Theory-driven vs data-driven

Strengths and weaknesses

Theory-driven
(discrete choice models)

Strengths
- Good interpretation, e.g.
  - Model parameters
  - Statistical measures for model fit comparison
- Transparent
- Solid ground for extrapolation
- Economic outputs (RUM)

Weaknesses
- Assumption-rich
- Restrictive on data
- Low prediction performance

Example

1	Route A	Route B	Route C
Average travel time (minutes)	45	60	75
Percentage of travel time in congestion (%)	10 %	25 %	40 %
Travel time variability (minutes)	±5	±15	±25
Travel costs (Euros)	€12,5	€9	€5,5
YOUR CHOICE	☐	☐	☐

	Beta	SE	t-value	p-value
average travel time	-0.0673	0.00192	-35.13	0.000
% in congestion	-0.0273	0.00157	-17.39	0.000
travel time variability	-0.0316	0.00266	-11.86	0.000
travel costs	-0.173	0.00803	-21.52	0.000
number of obs.	3510			
0-LogLik	-3856			
final-LogLik	-2613			
rho-squared	0.322			

Theory-driven vs data-driven

Strengths and weaknesses

Highly policy relevant insights!

Theory-driven
(discrete choice models)

Strengths
- Good interpretation, e.g.
  - Model parameters
  - Statistical measures for model fit comparison
- Transparent
- Solid ground for extrapolation
- Economic outputs (RUM)

Weaknesses
- Assumption-rich
- Restrictive on data
- Low prediction performance

Interpretation of ratio:

- WtP for reduction in average travel time: 23 €/hour

- WtP for decrease in time spent in congestion: 1.5 €/10%

- WtP for reduction in travel time variability: 11 €/hour

	Beta	SE	t-value	p-value
average travel time	-0.0673	0.00192	-35.13	0.000
% in congestion	-0.0273	0.00157	-17.39	0.000
travel time variability	-0.0316	0.00266	-11.86	0.000
travel costs	-0.173	0.00803	-21.52	0.000

number of obs.	3510
0-LogLik	-3856
final-LogLik	-2613
rho-squared	0.322

Theory-driven vs data-driven

Strengths and weaknesses

Theory-driven
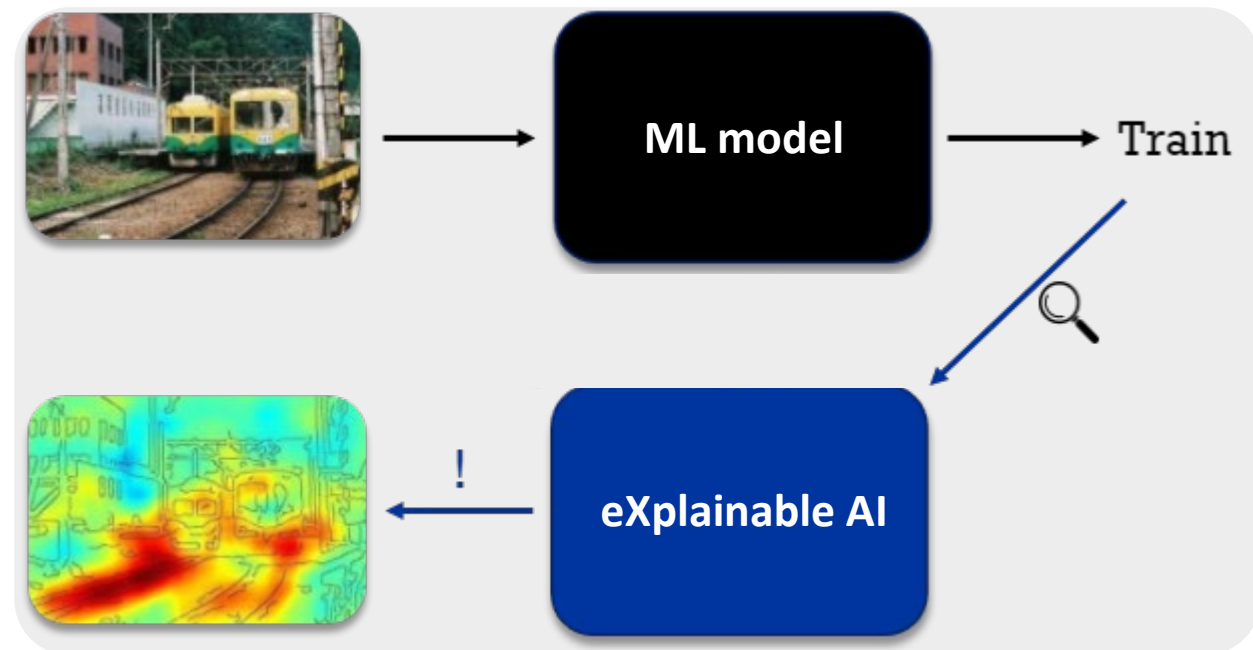(discrete choice models)

Strengths
- Good interpretation, e.g.
  - Model parameters
  - Statistical measures for model fit comparison
- Transparent
- Solid ground for extrapolation
- Economic outputs (RUM)

Weaknesses
- Assumption-rich
- Restrictive on data
- Low prediction performance

Interpretation of ratio:
- WtP for reduction in a   23 €/hour
- WtP for decrease in t   stion: 1.5 €/10%
- WtP for reduction in t   ....ty: 11 €/hour

ML model → Train

eXplainable AI

Theory-driven vs data-driven

Strengths and weaknesses

Theory-driven (discrete choice models)	Data-driven (supervised machine learning)

🙂 🙃

Theory-driven
(discrete choice models)

Strengths
- Good interpretation, e.g.
  – Model parameters
  – Statistical measures for model fit comparison
- Transparent
- Solid ground for extrapolation
- Economic outputs (RUM)

Weaknesses
- Assumption-rich
- Restrictive on data
- Low prediction performance

Data-driven
(supervised machine learning)

Weaknesses
- Poor interpretation:
  – No model parameters
  – No statistical measures of model fit comparison
- Opaque
- Shaky extrapolation
- Unclear economic outputs

Strengths
- Assumption-poor
- Not restrictive on data
- High prediction performance

Important for discrete choice analysis?

Theory-driven vs data-driven

What assumptions?

**Theory-driven
(discrete choice modelling)**

Important?

The Decision rule: $U_i > U_j \ \forall j \neq i$
- Utility maximisation (RUM)
- Regret minimisation (RRM)
- Lexicography
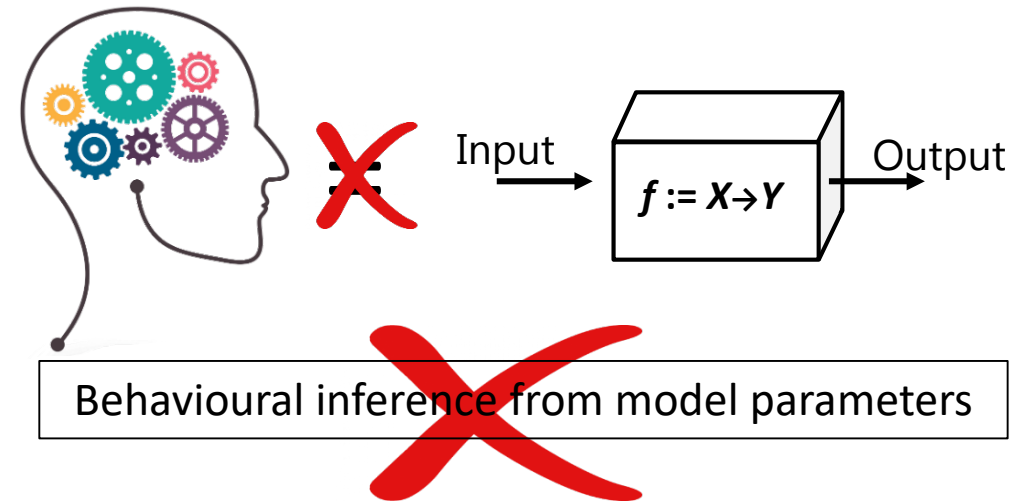- Satisficing
- Prospect theory

Utility specification: $V_i = ...$
- Attributes
- Interactions between attributes
- Linearity, additivity of part-worth utilities

Error term assumptions $\varepsilon_i \sim N(0,1)$
- Shape of distribution e.g. normal, gumbel, ...
- Correlation structure

Input

Output

$f := X \rightarrow Y$

Behavioural inference from model parameters

**Accuracy of behavioural inferences conditional on the theory accurately describing the true behaviour
%%
A model is as only good as its assumptions; Garbage In, Garbage Out (GIGO)**

Theory-driven vs data-driven

Strengths and weaknesses

Theory-driven
(discrete choice models)

🙂 🙃

Data-driven
(supervised machine learning)

Strengths
- Good interpretation, e.g.
  - Model parameters
  - Statistical measures for model fit comparison
- Transparent
- Solid ground for extrapolation
- Economic outputs (RUM)

Weaknesses
- Assumption-rich
- Restrictive on data
- Low prediction performance

Weaknesses
- Poor interpretation:
  - No model parameters
  - No statistical measures of model fit comparison
- Opaque
- Shaky extrapolation
- Unclear economic outputs

Strengths
- Assumption-poor
- Not restrictive on data
- High prediction performance

Important for discrete choice analysis?

Theory-driven vs data-driven

Strengths and weaknesses

Theory-driven
(discrete choice models)

Strengths
- Good interpretation, e.g.
  – Model parameters
  – Statistical measures for model fit comparison
- Transparent
- Solid ground for extrapolation
- Economic outputs (RUM)

Weaknesses
- Assumption-rich
- Restrictive on data
- Low prediction performance

Year	Month	Category	Type	Sales	Cost
2001	January	Fruit	Apples	12	10
2001	January	Fruit	Pears	21	13
2001	January	Fruit	Bananas	29	26
2001	January	Vegetables	Cucumber	9	6
2001	January	Vegetables	Tomatoes	13	11
2001	January	Vegetables	Lettuce	22	20
2001	February	Fruit	Apples	11	9
2001	February	Fruit	Pears	21	14
2001	February	Fruit	Bananas	31	27

… But many choice situations can hardly be summed up in a table

A	B	C
26 yr	29 yr	28 yr
Shop	?	?
Pop music	?	Rock
Travel	Book	?

e.g. Tinder

But also consumer choices such as buying a pair of shoes

31

Theory-driven vs data-driven

Strengths and weaknesses

Theory-driven
(discrete choice models)

Strengths
- Good interpretation, e.g.
  - Model parameters
  - Statistical measures for model fit comparison
- Transparent
- Solid ground for extrapolation
- Economic outputs (RUM)

Weaknesses
- Assumption-rich
- Restrictive on data
- Low prediction performance

Year	Month	Category	Type	Sales	Cost
2001	January	Fruit	Apples	12	10
2001	January	Fruit	Pears	21	13
2001	January	Fruit	Bananas	29	26
2001	January	Vegetables	Cucumber	9	6
2001	January	Vegetables	Tomatoes	13	11
2001	January	Vegetables	Lettuce	22	20
2001	February	Fruit	Apples	11	9
2001	February	Fruit	Pears	21	14
2001	February	Fruit	Bananas	31	27

… But many choice situations can hardly be summed up in a table

Theory-driven models cannot really handle 21st century (big) **data**, such as
- Images
- Video
- Audio
- Text

Theory-driven vs data-driven

Best of both

**Theory-driven
(discrete choice)**

+ Interpretation
+ Transparent
- Prediction
- Assumption-rich
- Restrictive

**Data-driven
(ANN)**
- Interpretation
- Opaque
+ Prediction
+ Assumption-poor
+ Versatile / flexible

In sum

- **Both modelling paradigms have their strengths and weaknesses**

- **Ideally, we can get the best of both**

Knowledge discovery

Theory-driven vs data-driven

Knowledge discovery

- Scientific advancement in social sciences almost always involves the **discovery** of **causal relations** to help understanding our world and beyond

- *Theories* describe the **first principles** explaining how and why *X* **causally** relates to *Y*
    - Theory of biological evolution (Darwin, 1859)
    - Utility theory (Bentham, 1788)
    - Big Bang theory (Lemaitre, 1931)
    - Theory of Planned Behaviour (Ajzen, 1985)
    - String theory ()

- **'Strong'** theories (1) can explain & predict *beyond* the support of the data, (2) are testable and (3) parsimonious (Occam's razor)
    - Einstein's theory on relatively explains & predicts the existence of Black holes, Gravitational Waves;
    - Standard Model explains & predicts the Higgs Boson, etc.
    - Adam Smith's Free market theory explains & predicts supply, demand and prices;
    - Game theory explains & predicts how firms behave in an oligopoly market, etc.

- *Models* [theory-driven] are **instantiations** of **theories**, providing a (local) mathematical description or understanding of the phenomenon

Knowledge discovery

THEORY-DRIVEN PARADIGM

- Theory development is a continuous process, involving several steps:

Example

1. Identification of phenomena of interest
   - Compromise effect: decision makers seem to more often choose a compromise alternative than would be expected based on (random) utility maximisation theory

2. Formulation of proto theory
   - Possibly this behaviour is driven by regret: the emotion experienced by a decision maker when one or more of the nonchosen alternatives performs better than a chosen one

3. Development of formal model

   - RRM2008 model: $R_{in} = \max\left\{\sum_m \max\left\{0, \beta_m \cdot \left(x_{jm} - x_{im}\right)\right\}\right\}$

4. Checking adequacy of the formal model
   - Able to explain and model the compromise effect $\longrightarrow$ ***μRRM2015***
   - Issues with estimation and identification $\longrightarrow$ ***RRM2010***

5. Evaluation of worth of the overall theory
   - Adds some behavioural intuition, but at a cost: ↓ welfare econ foundation

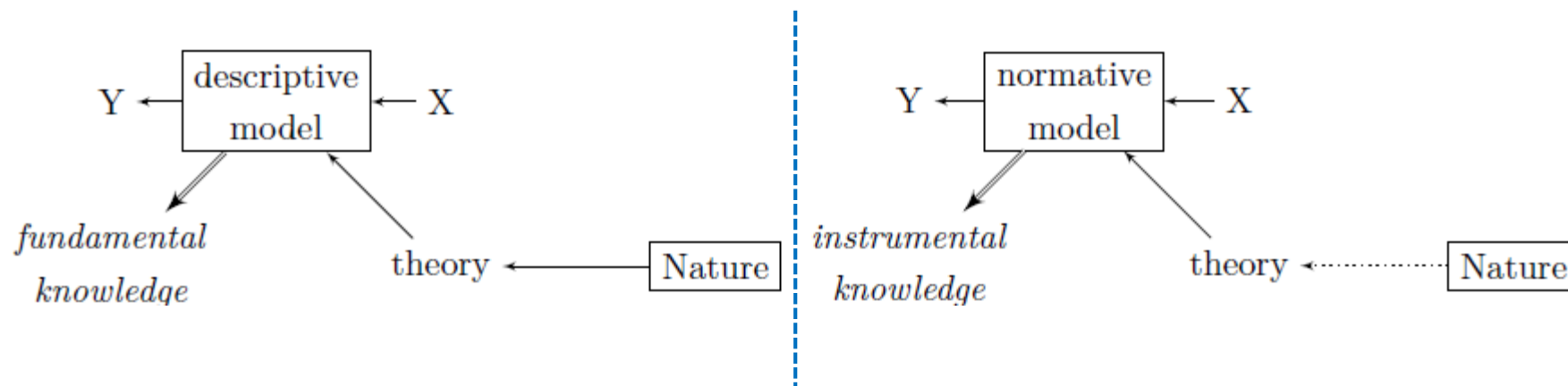*Accumulation of empirical evidence leads to new or revised theory*

Knowledge discovery

- In **choice behaviour modelling**, there are two flavours of theory-driven models: **descriptive** and **normative** models

- **Descriptive model** provide **fundamental knowledge** on human behaviour:
  - Prospect theory, regret theory
  - Endowment effect, Anchoring effect, Decoy effect, Gambler's fallacy, Loss aversion, etc.

- **Normative models** provide **instrumental knowledge** on human behaviour:
  - Models focus on what _should_ be done given a set of **axioms** e.g. $A >> B$, $B >> C \rightarrow A >> C$ _'Transitivity'_
  - E.g. assuming rational choice theory, we can derive Willingness to pay, which stipulates which policy intervention is theoretically maximises benefits to society

Knowledge discovery

CRITIQUES TO THE THEORY-DRIVEN PARADIGM

1. Theories, and by extension the models, are often too simplistic

2. Number of models possible theoretical models is large and quickly grows infeasible to test
   → Ad hoc model selection
   → **Irrelevant theories**
     – Model search leads to conflated outcomes
     – Weak theories (no real value beyond data)

3. Relies on hand-engineered high-level semantically meaningful variables as it cannot handle low level raw data

Field's response in choice modelling

1. More of the same: estimating even more complex models
2. Recently, development of hybrid models **(i.e. combining theory and data driven models)**

Choice task → Choice

Do you reall͏... ͏our is really that simple?

- Alternative *I* is chosen if and only if $U_i > U_j \, \forall \, j \neq I$
- $U_i = V_i + \varepsilon_i$
- $V_i = \sum_m \beta_m x_{im}$

if you torture the data long enough, it will confess to anything [Ronal Coase]

Knowledge discovery

DATA-DRIVEN PARADIGM
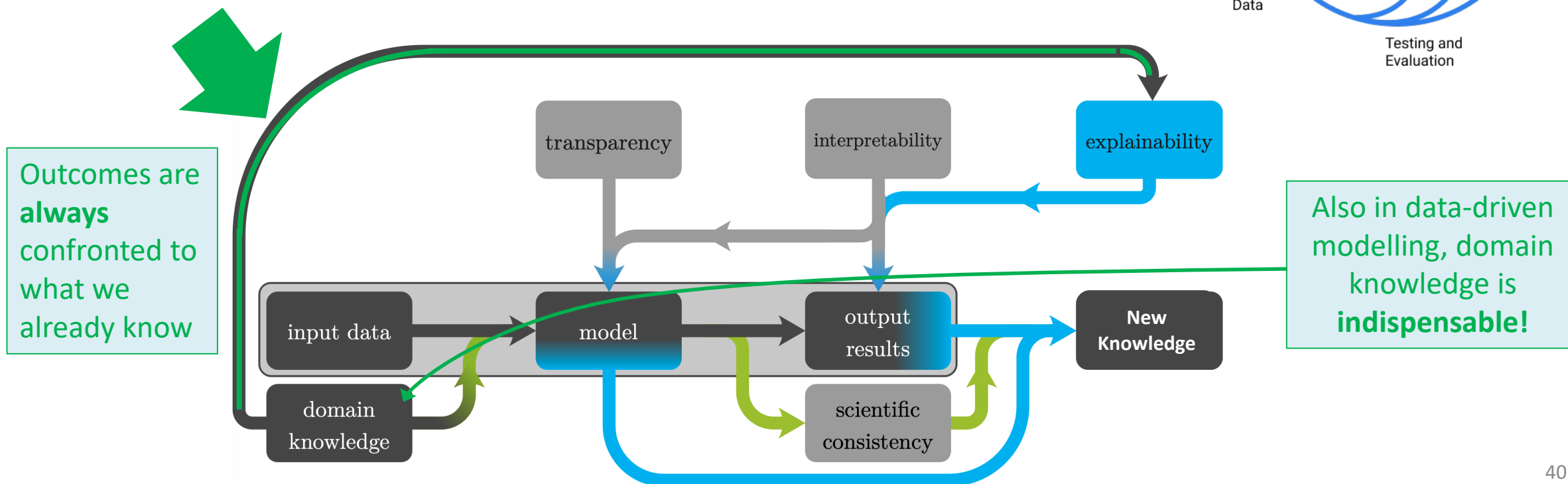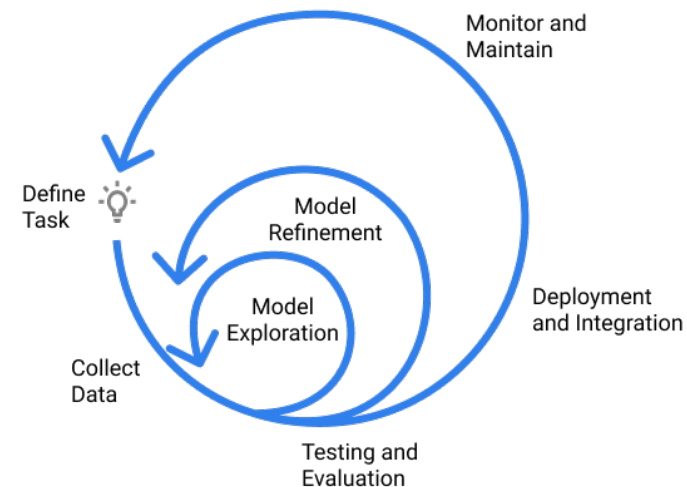
- Central idea: true data generating process is complex and cannot be known → function g is learned

- But, with **lots** of truly i.i.d. data the model can <u>learn</u> the '**true'** statistical relations

- Strong focus on in-distribution **prediction**

- Examples of key achievement of ML for scientific discovery
  - Predicting how proteins fold → new medicine
  - Healthcare diagnostics, e.g. based on MRI
  - Identifying previously missed exoplanets
  - Predicting new materials and properties
  - New moves / game strategies in AlphaGo
  - Assisting in with designing semi-conductors

$$ Y \leftarrow \boxed{\text{algorithm}} \leftarrow X $$

prediction
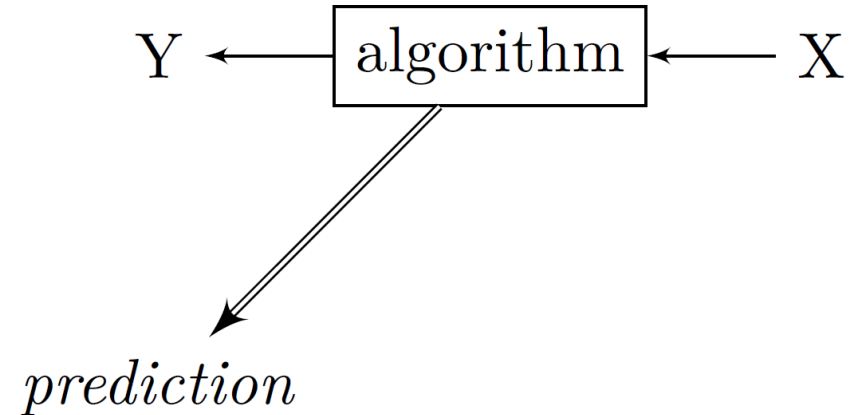
Knowledge discovery

**DATA-DRIVEN PARADIGM model development**

- Iterative process (as a modeller)
- Fast turn-over of fashionable models and topics (as a field)
- Tracking progress heavily relies on:
  - Standard data sets (benchmarks)
  - Software building blocks

Outcomes are **always** confronted to what we already know

Also in data-driven modelling, domain knowledge is **indispensable!**

40

Knowledge discovery

CRITIQUES TO THE DATA-DRIVEN PARADIGM :

- Sensitive to *causal fallacy*

- Unfit for out-of-distribution forecasting, such as new policy interventions (if A then B)
  - Even if it has learned the true causal relationships, it is still unclear how a model responds beyond the support of the data

Support of "quality" in the data
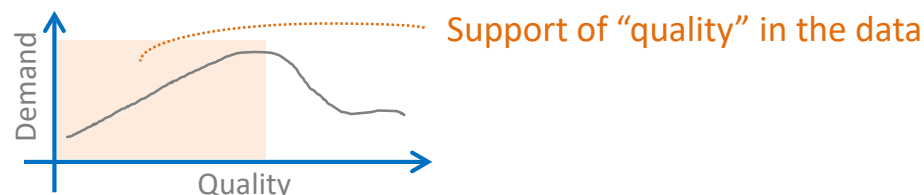
- Lack of trust for designing policies/interventions due to opaque nature (i.e. weights do not carry meaning provided by a theory)

- Good predictions ≠ knowledge

Field's response in ML

- More of the same: using even more data in the hope to attain i.i.d. observations
- Increasing interest in (1) model explanation (XAI) and (2) structural models **(i.e. combining theory and data models).**
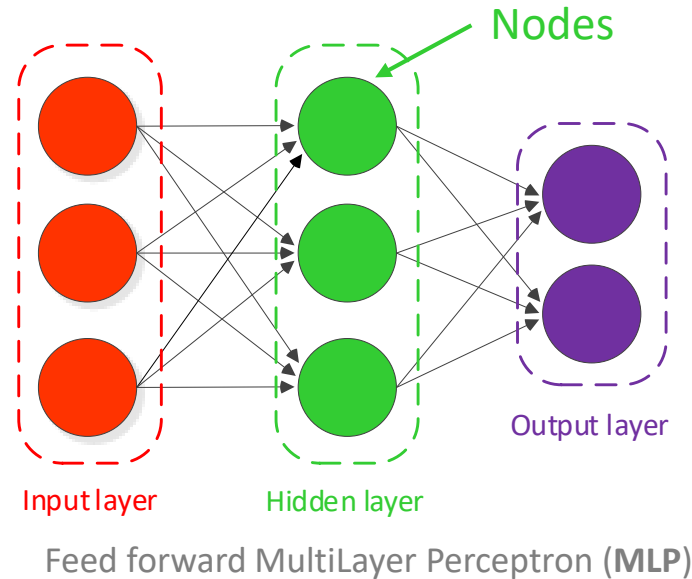
Knowledge discovery

In sum,

- Both theory- and data-driven paradigms can, if successful, lead to knowledge discovery

- Neither is fundamentally more promising to lead to knowledge discovery (major pitfalls)

- For BOTH paradigms causality is an key challenge

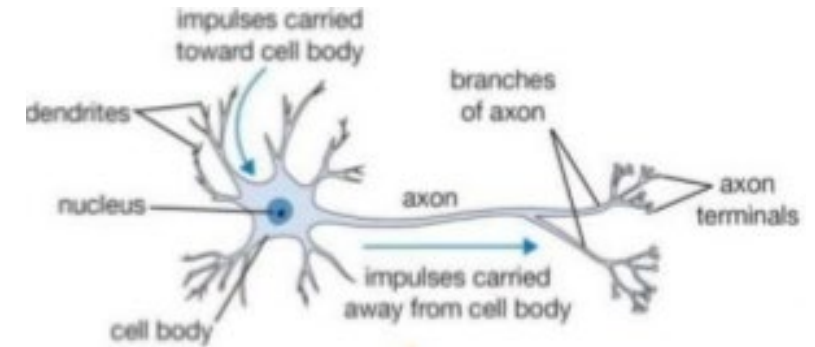- In both fields there is increasing recognition that cross-pollination is promising to deal with
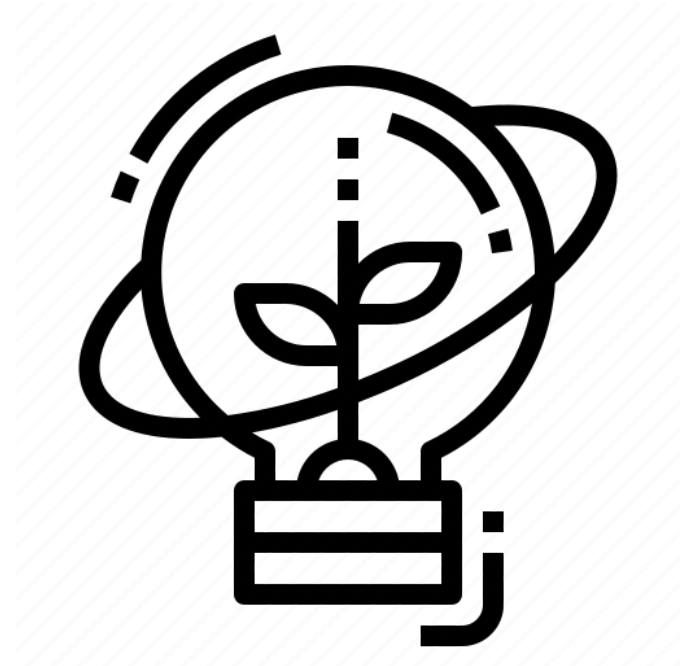
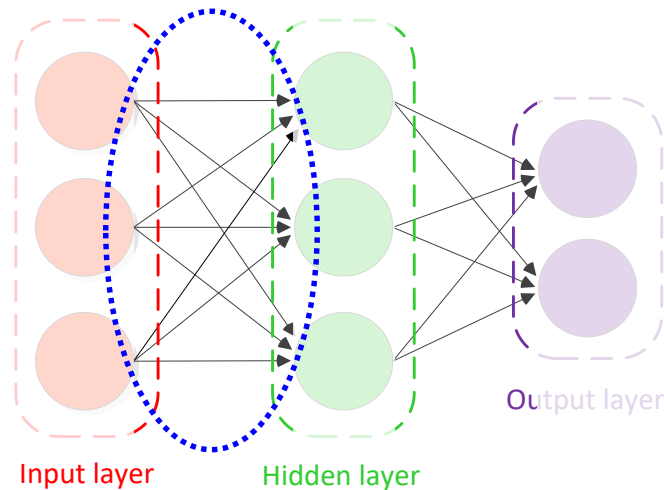# Artificial Neural Networks

Artificial neural networks

Layers

Loosely inspired by the brain

Nodes

Output layer

Input layer Hidden layer

Feed forward MultiLayer Perceptron (**MLP**)

impulses carried
toward cell body

branches
of axon

dendrites

nucleus

axon

axon
terminals

impulses carried
away from cell body

cell body

- **Input layer** consists of the features (explanatory variables, e.g. travel cost and travel time)

- **The output** layer consists of the target variable (dependent variable, e.g. the **choice**)

- **Hidden layer(s)** consists of a finite number hidden nodes (set by the analyst)
    - ➢ 1 hidden layer → Shallow neural network
    - ➢ 2 or more hidden layers → Deep neural network

44

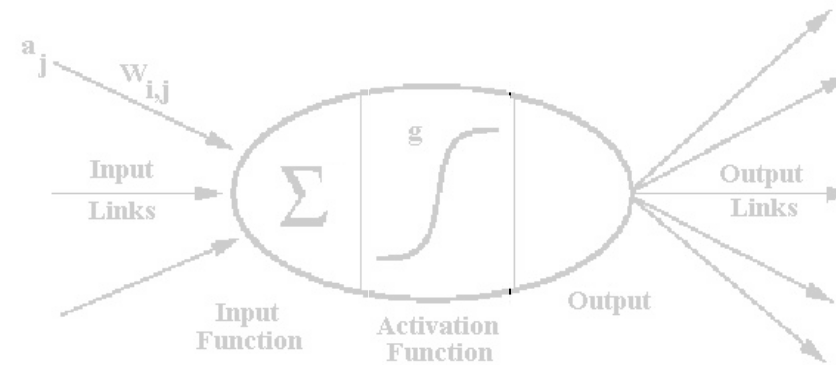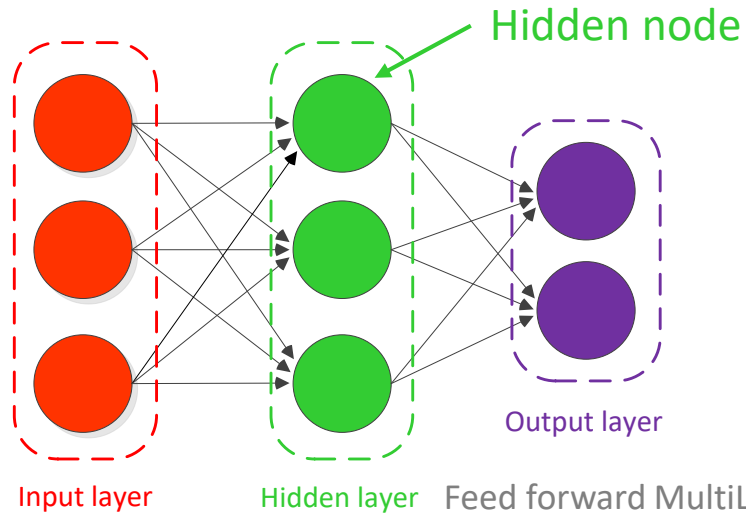# Artificial neural networks

Connections

Feed forward MultiLayer Perceptron (MLP)

- The links (black arrows) contain the weights w that need to be trained

- Fully connected network (common, but not strictly necessary)

- Even for this **very simple** network already 3 x 3 + 3 x 2 = 15 weights

Artificial neural networks

Nodes

Hidden node

Input layer   Hidden layer   Output layer

Feed forward MultiLayer Perceptron (MLP)

- At each hidden node:
  - ➤ The sum of the inputs is computed
  - ➤ A transfer (a.k.a. activation) function *g* is applied (set by the analyst)
  - ➤ Output is forwarded to the nodes in the next layer

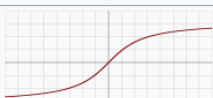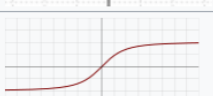$$a_j = \sum_i w_{ji} x_i$$

w_{ji} **are the trainable weights**

$$z_j = g\left(a_j\right)$$

Artificial neural networks

Transfer functions

Name	Plot	Equation
Identity		$f(x) = x$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Logistic (a.k.a. Sigmoid or Soft step)		$f(x) = \sigma(x) = \dfrac{1}{1 + e^{-x}}$ [1]
TanH		$f(x) = \tanh(x) = \dfrac{(e^x - e^{-x})}{(e^x + e^{-x})}$
ArcTan		$f(x) = \tan^{-1}(x)$
Softsign[7][8]		$f(x) = \dfrac{x}{1 + |x|}$
Inverse square root unit (ISRU)[9]		$f(x) = \dfrac{x}{\sqrt{1 + \alpha x^2}}$
Rectified linear unit (ReLU)[10]		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$

The most suitable transfer function depends problem, e.g. depth of network, and amount of data one works with

Source: Wikipedia

And many more
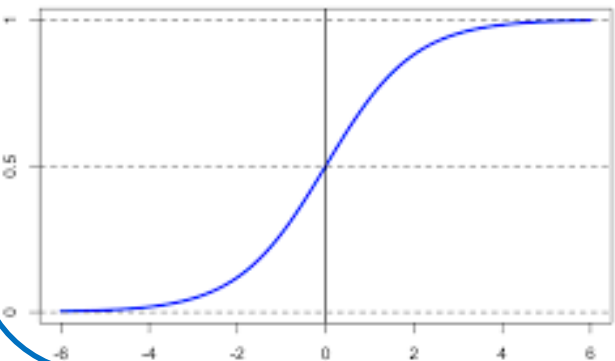
47

Artificial neural networks

Bias nodes
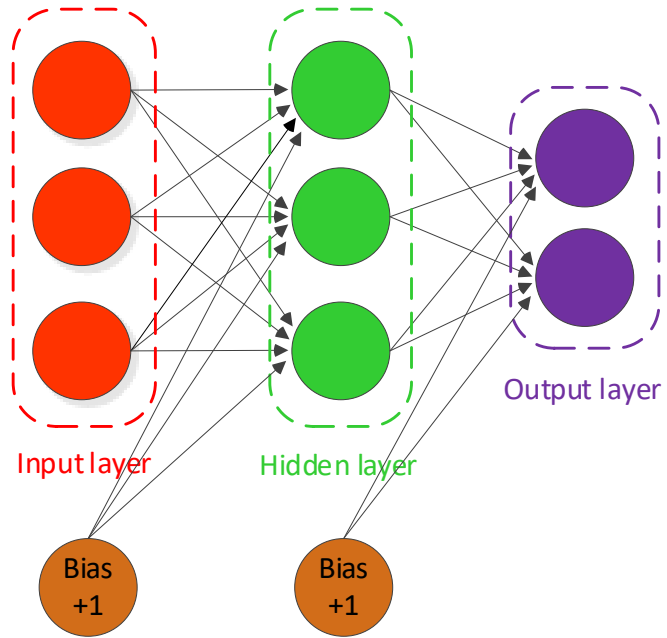
- The bias nodes ensure input, even if input space is 0

- The weights associated with the bias nodes on the output layer can be conceived as Alternative Specific Constants (ASCs) or Intercepts.

Artificial neural networks

Output layer function

- Regression: Linear

- Binary class classification: Logistic   a.k.a. **Binary Logit** in choice modellers' slang

- Multiclass classification: Softmax   a.k.a. **Multinomial Logit** in choice modellers' slang

$$f\left(y = j \mid x\right) = \frac{e^{\mathbf{x}^{\mathbf{T}}\mathbf{w_j}}}{\sum_{k=1...K} e^{\mathbf{x}^{\mathbf{T}}\mathbf{w_k}}}$$

Matrix notation
dot (inner) product

Softmax function

Artificial neural networks

Cost function (aka loss function)

- The cost function measures the performance of the model, for given data. It quantifies the error between predicted values and expected values and presents it in the form of a single real number.

  ➢ Regression: MSE, RMSE, …

  ➢ Classification: Cross-entropy, Hinge, …

$$-\frac{1}{N}\sum_{n}\sum_{j}y_{nj}\log p_{jn}$$

y_{nj} – Binary indicator (0 or 1), indicating the correct classification for example (observation) n and class (alternative) j

p_{jn} – Predicted probability in example n for class j

N – Total number of examples

- Minimisation of the cross-entropy is **equivalent** to maximisation of the (log-)likelihood

→ **Minimising** the **cross-entropy** implies **maximising** the (log-)**likelihood** of the data, given the model

Artificial neural networks

Types

- ANNs come in many different types
  - **Shallow** (this course) and Deep

  - Convolution neural nets (e.g. computer vision)

  - Recurrent Neural networks (e.g. text, speech)
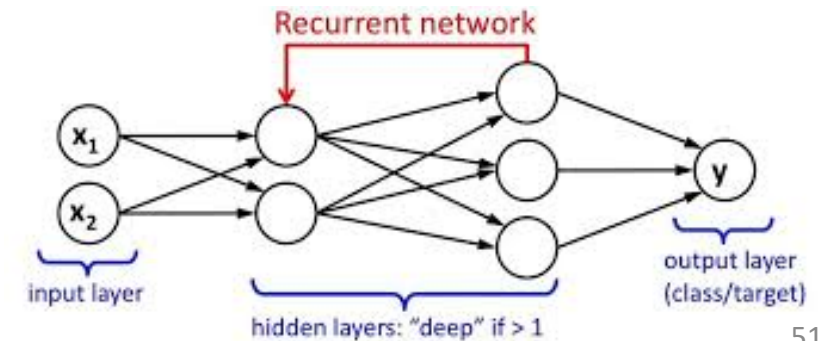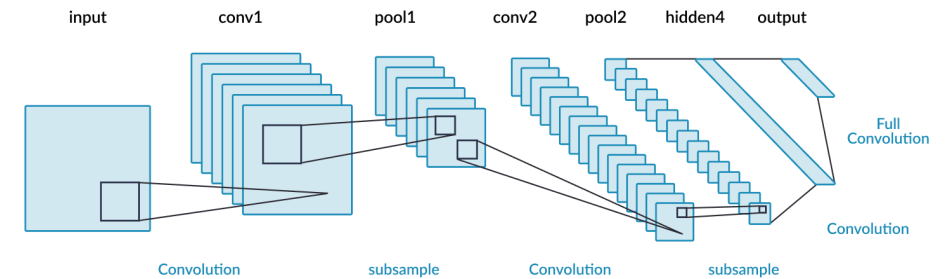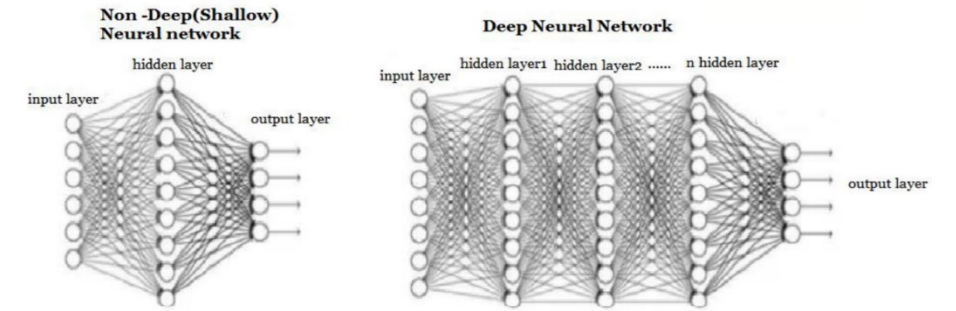
Artificial neural networks

- **Deeper networks (in theory) enable capturing more complex behaviours, such as**
  - Nonlinearity in utility
  - Interactions between attributes and e.g. socio demographics (beta_cost $\times$ income level)
  - Decision rules and heuristics (if cost>30, then Y==A, else …)

- Depth can be obtained by having more nodes per layer, or more layers
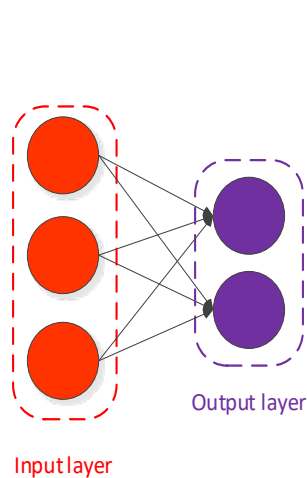
No hidden layers →
logistic regression model
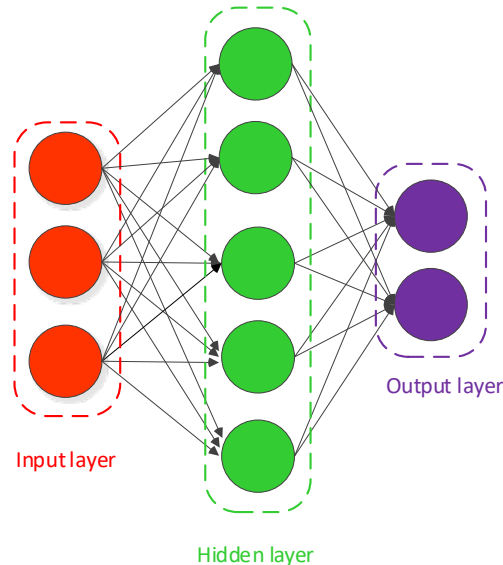
MLP with 1 hidden layer
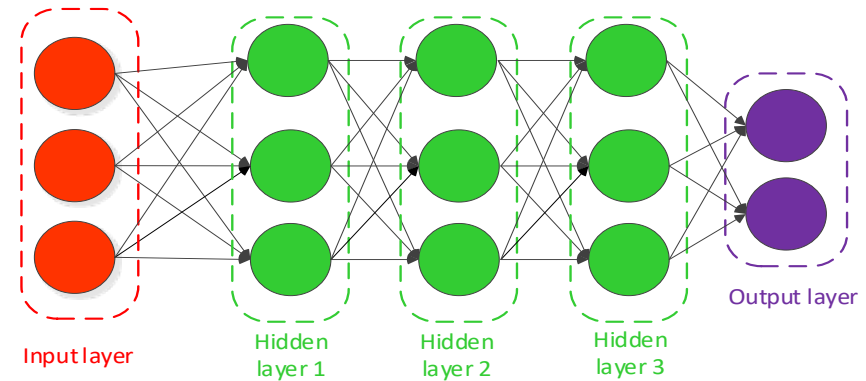
MLP with 3 hidden layers

- But, more depth increases the number of weights
- But, more depth will only pay-off when the DGP is complex AND with **(1) sufficient and (2) high quality data**

Artificial neural networks

Universal Approximation Theorem (UAT) (a.k.a. Cybenko's theorem)

- UAT states that simple MLP ANNs *can* approximate any continuous function given appropriate network dimensions (under mild assumptions on the activation function)

- But, the UAT does not touch upon what is needed to reach a good approximation (e.g. how much data).

Would You Like To Make $1099.20 Per Day Using My One-Click Software?...

Training ANNs

Training ANNs

- Training ANNs involves finding the weights *w* that **minimise** the **cost function** (aka loss function)

$$w = \arg\min_{w} L\left(Z \mid w \right) \quad where\, Z = \left(X, Y \right)$$

- Mathematically training an ANN is an **ill-posed** problem ("singular")

- That is, ANNs are not **uniquely identifiable**: i.e. there are numerous sets of weights that give the same output probability distribution on any data.

  → No stable, unique solution

- **But, lack of identifiability is not a problem: the model weights cannot be interpreted anyway** ☺

- A + B = 4   not uniquely identifiable: A=0,B=4
                                             A=1,B=3
                                             A=2,B=2
                                             A=3,B=1
                                             A=4,B=0

- A + B = 4   uniquely
- A − B = 2   identifiable   **A=3,B=1**

Training ANNs

- **Backward propagation** is commonly used algorithm to train ANNs

- Back propagation computes how the total loss is generated from each node in each layer, and how the total loss would change by changing the weights (in one forward-backward pass)
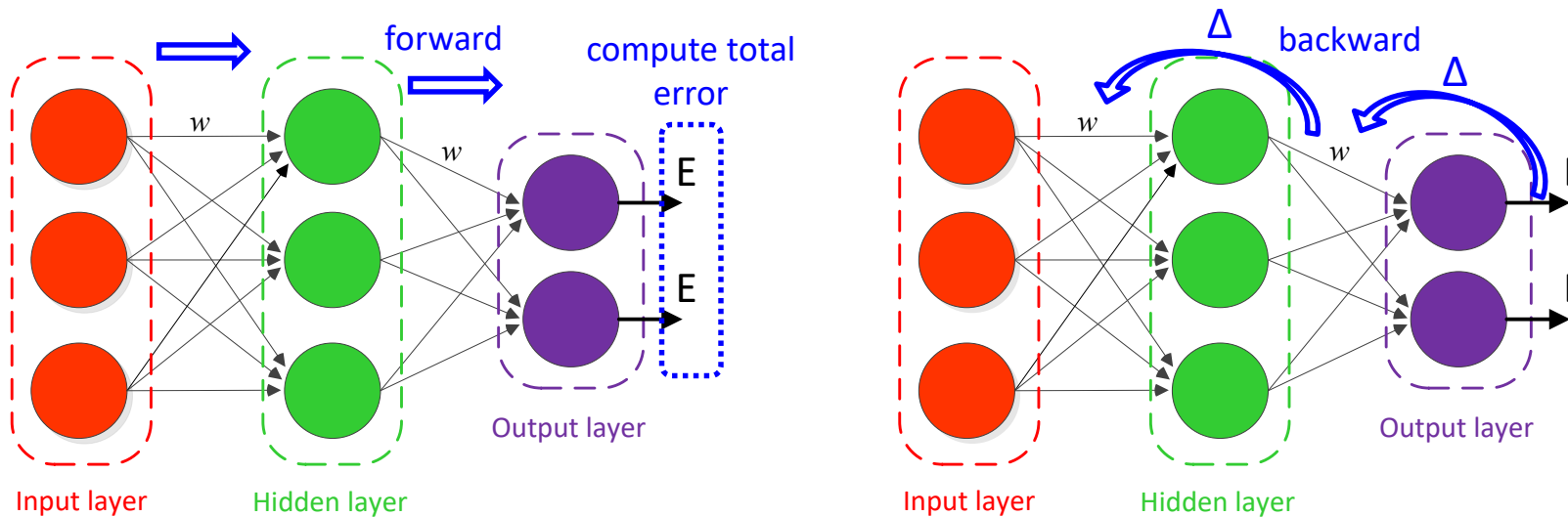
- Having the derivatives the weights are improved layer-by-layer, moving towards the minimum

- A forward-backward pass is called an **iteration**

Training ANNs

- To efficiently train ANNs input data need to be **scaled** (aka normalised)

- Scaling reduces getting stuck in **poor** local solutions

- There are various scaling practices
  - SciKit-learn's StandardScaler shifts the mean each feature to zero and its standard deviation to 1:

```
X_train.describe()
✓ 0.1s
```

	COST_1	SIZE_1	STORAGE_1	CAM_1	COST_2	SIZE_2	STORAGE_2	CAM_2	COST_3	SIZE_3	STORAGE_3	CAM_3	GENDER	INC
count	6700.0	6700.0	6700.0	6700.0	6700.0	6700.0	6700.0	6700.0	6700.0	6700.0	6700.0	6700.0	6700.0	6700.0
mean	530.2	6.1	106.8	2.3	529.1	6.1	104.8	2.3	529.6	6.1	105.8	2.3	0.5	1.7
std	109.4	0.2	82.7	1.1	110.4	0.2	82.3	1.1	109.6	0.2	82.3	1.1	0.5	0.6
min	400.0	5.8	32.0	1.0	400.0	5.8	32.0	1.0	400.0	5.8	32.0	1.0	0.0	1.0
25%	400.0	5.8	32.0	1.0	400.0	5.8	32.0	1.0	400.0	5.8	32.0	1.0	0.0	1.0
50%	500.0	6.0	64.0	2.0	500.0	6.0	64.0	2.0	500.0	6.0	64.0	2.0	0.0	2.0
75%	600.0	6.2	128.0	3.0	600.0	6.2	128.0	3.0	600.0	6.2	128.0	3.0	1.0	2.0
max	700.0	6.4	256.0	4.0	700.0	6.4	256.0	4.0	700.0	6.4	256.0	4.0	1.0	3.0
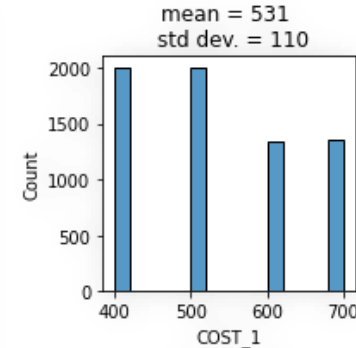
```
X_train_scaled.describe()
✓ 0.1s
```

	COST_1	SIZE_1	STORAGE_1	CAM_1	COST_2	SIZE_2	STORAGE_2	CAM_2	COST_3	SIZE_3	STORAGE_3	CAM_3	GENDER	INC
count	6700.0	6700.0	6700.0	6700.0	6700.0	6700.0	6700.0	6700.0	6700.0	6700.0	6700.0	6700.0	6700.0	6700.0
mean	-0.0	0.0	0.0	0.0	0.0	-0.0	-0.0	0.0	0.0	0.0	-0.0	-0.0	-0.0	-0.0
std	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
min	-1.2	-1.2	-0.9	-1.2	-1.2	-1.2	-0.9	-1.2	-1.2	-1.2	-0.9	-1.2	-1.0	-1.1
25%	-1.2	-1.2	-0.9	-1.2	-1.2	-1.2	-0.9	-1.2	-1.2	-1.2	-0.9	-1.2	-1.0	-1.1
50%	-0.3	-0.3	-0.5	-0.3	-0.3	-0.3	-0.5	-0.3	-0.3	-0.3	-0.5	-0.3	-1.0	0.5
75%	0.6	0.6	0.3	0.6	0.6	0.6	0.3	0.6	0.6	0.6	0.3	0.6	1.0	0.5
max	1.6	1.5	1.8	1.6	1.5	1.5	1.8	1.5	1.6	1.6	1.8	1.5	1.0	2.1

mean = 531
std dev. = 110

mean = -0
std dev. = 1
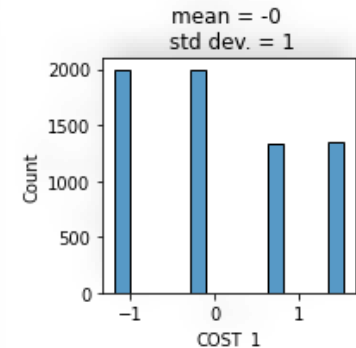
Training ANNs

- When training the researcher needs to set the **hyperparameters**
  1. Batch size
  2. Optimisation algorithms
  3. Learning rate
  4. Regularisation

- **Hyperparameter tuning** often involves the **finding the optimal combination of hyperparameters**, and often can considerably improve the model performance

- To tune the hyperparameters often a simple **grid search** is used:
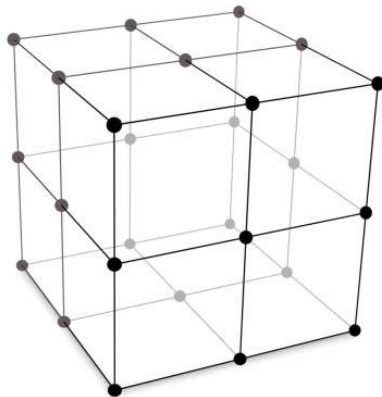
- But, hyperparameter tuning can quickly become computationally excessive…

Training ANNs

Hyperparameters

1. <u>Batch size</u>: number of observations (aka instances) utilized in **1 iteration**

   – All instances (i.e. all data)

   Batch gradient descent

   – One instance randomly sampled

   Stochastic gradient descent

   – *N* instances randomly sampled

   Mini batch gradient

— Batch gradient descent
— Mini-batch gradient Descent
— Stochastic gradient descent

Training ANNs

Hyperparameters

1. <u>Batch size</u>: number of observations (aka instances) utilized in **1 iteration**

Stability

Robustness towards saddles and local minima

Robustness of different optimisation algorithms
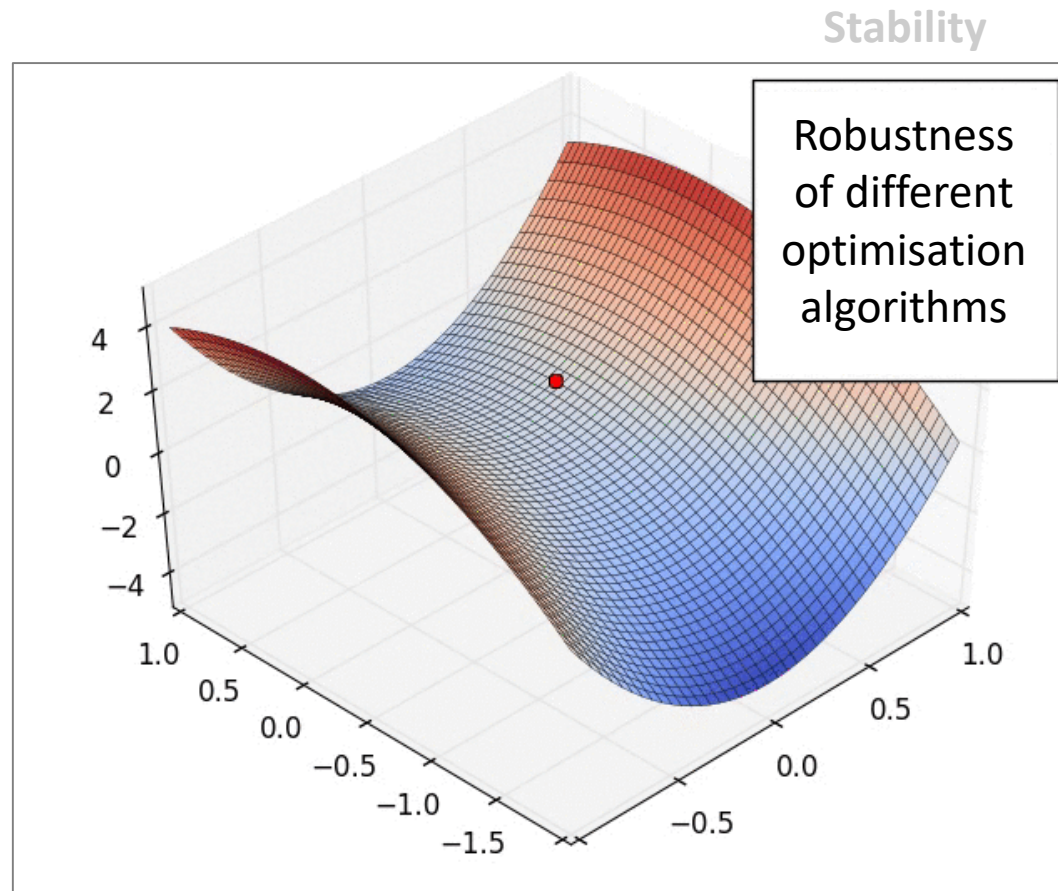
Not so robust

Robust

Fairly robust

Training ANNs

Hyperparameters

1. <u>Batch size</u>: number of observations (aka instances) utilized in **1 iteration**
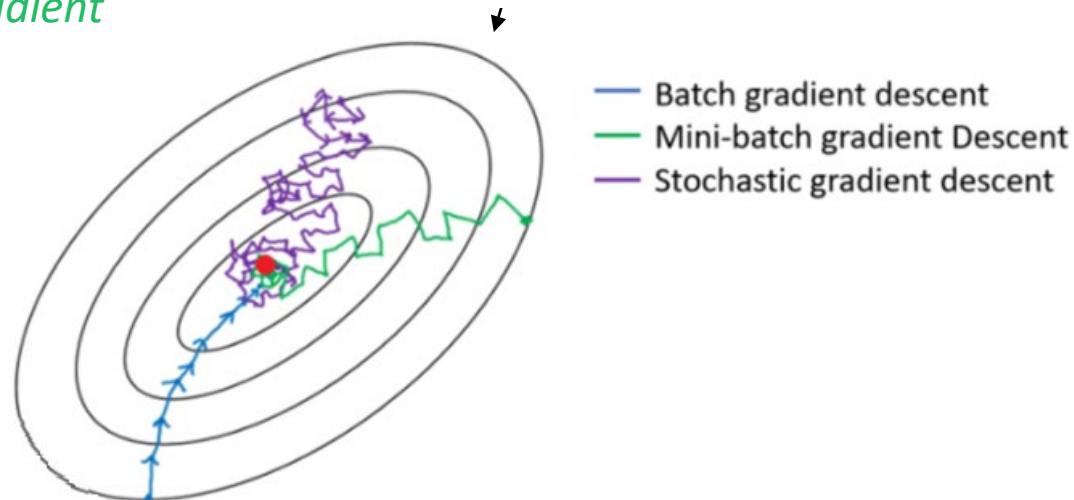
		Stability	**Robustness towards saddles and local minima**
–	All instances (i.e. all data)	Very stable	Not so robust
	Batch gradient descent		
–	One instance randomly sampled	Very instable	Robust
	Stochastic gradient descent		
–	*N* instances randomly sampled	Fairly stable	Fairly robust
	Mini batch gradient		

In 1 **epoch** all the observations (instances) in the data has been utilized once

Relationship batch size, iteration, epoch
N = 10,000 **observations** ⟹ 10 **iterations**
BS = 1,000 **(mini)batch size** per **Epoch**

Training ANNs

Hyperparameters

2. Optimisation algorithms: procedure for finding the weights that result in the minimum or maximum output of the function

   ➤ Gradient Descent (GD) [most commonly used: e.g. **Adam**, Adagrad, RMSprop]

   ➤ L-BFGS

   ➤ Levenberg–Marquardt (combination ↑)

• Optimisation algorithms differ from one another in terms of:

   ➤ Speed to converge to a solution

   ➤ Robustness towards **local minima and saddle points**

Local minima   **Saddle point**
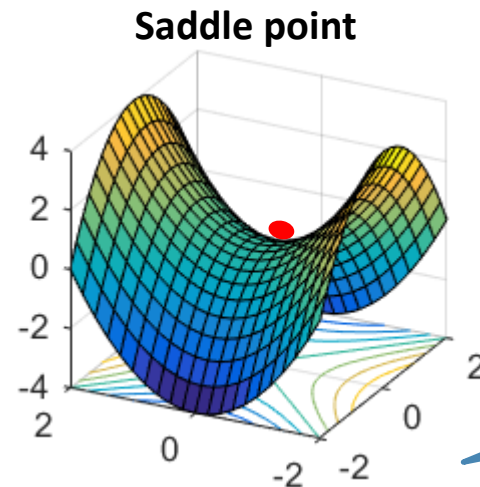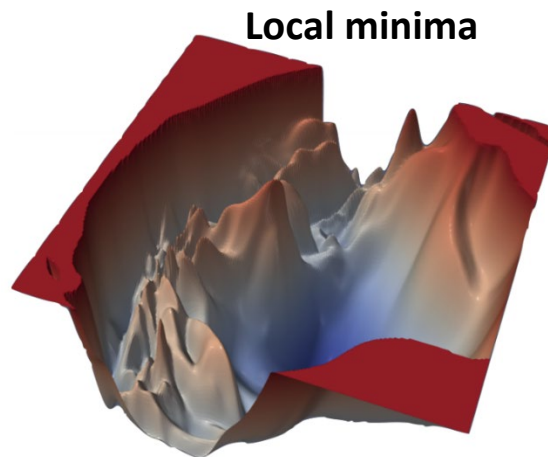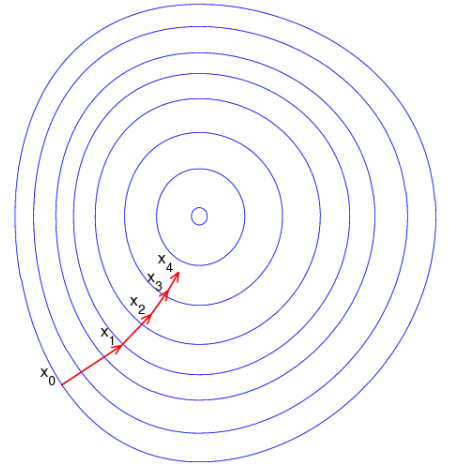
While the global minimum is seldom found, this is not necessarily a problem.

A 'good' local minimum can still be good

Training ANNs

Hyperparameters

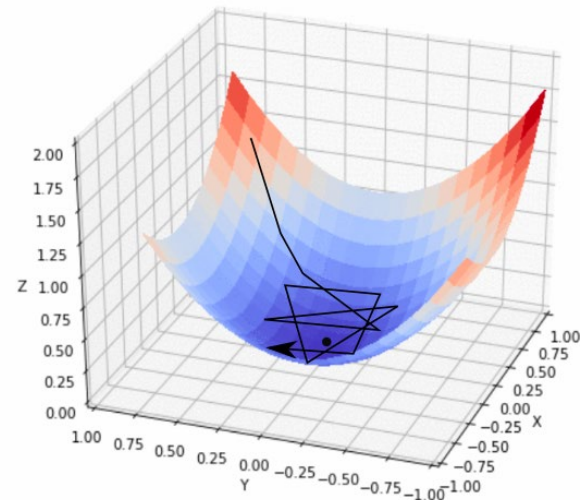3. <u>Learning rate</u>: determines the step size at an iteration

Large learning rate   **Small learning rate**

Too big →
Overshooting (instability)
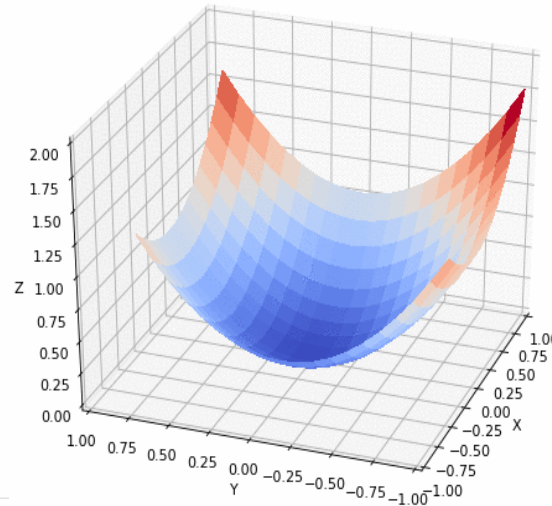
Too small →
Slow to converge

Training ANNs

Hyperparameters

4. <u>Regularisation</u>: **penalises** the magnitudes of the **weights**, to avoid overfitting

    ➤ L1 regularisation (aka Lasso) penalises the sum of the absolute values of the weights $\sum \lVert w \rVert$

    ➤ L2 regularisation (aka ridge) penalises the sum of the squared values of the weights $\sum \lVert w \rVert^2$

- The **penalty** is added to the cost (loss) function:

Cross-entropy

$$Loss = CE + \alpha \sum \lVert w \rVert \qquad (L1)$$

Penalty

$$Loss = CE + \alpha \sum \lVert w \rVert^2 \qquad (L2)$$

- The hyperparameter **α** governs the strength of the **penalty** relative to the **cost function**

    ➤ Too high α → risks underfitting

    ➤ Too low α → risks overfitting

Generalisation error

Generalisation error

- Often the objective of ML models is to **generalise** well
  - That is, the model does well in making predictions for **new data points** (from same data generating process)
  - In other words, the objective is often develop a model that minimises the **generalisation error**

← In-sample fit

	Underfitted "bias"	Good Fit/Robust	Overfitted "variance"
In-sample perf.	Poor	Good	Very good
Out-of-sample perf.	Poor	Good	Poor
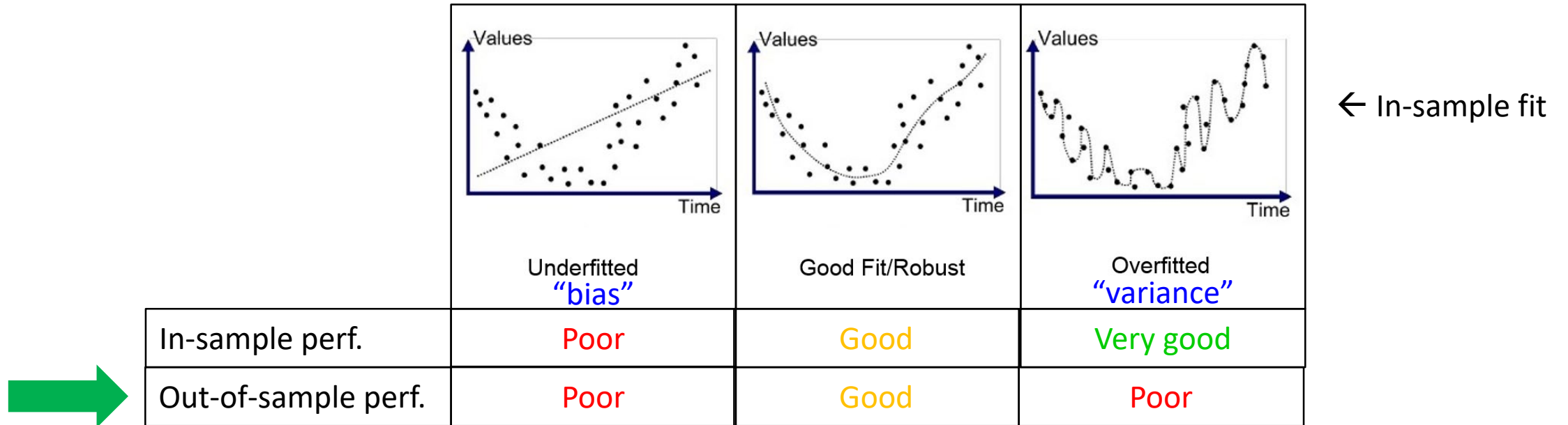
- Therefore, in machine learning we always assess the **out-of-sample performance,** a.k.a. **generalisation error (aka test/validation loss)**

- **So, how to determine the what is the 'best' model?**

66

Generalisation error

- Data must be always **split** in two parts:
  1. Train set
  2. Test set

- The ANN is trained on the **training set**
  - ➢ After each epoch, the performance of the model is **evaluated on test set**

  - ➢ The training stops when the performance on the **training set** no longer improves

Learning curve plot
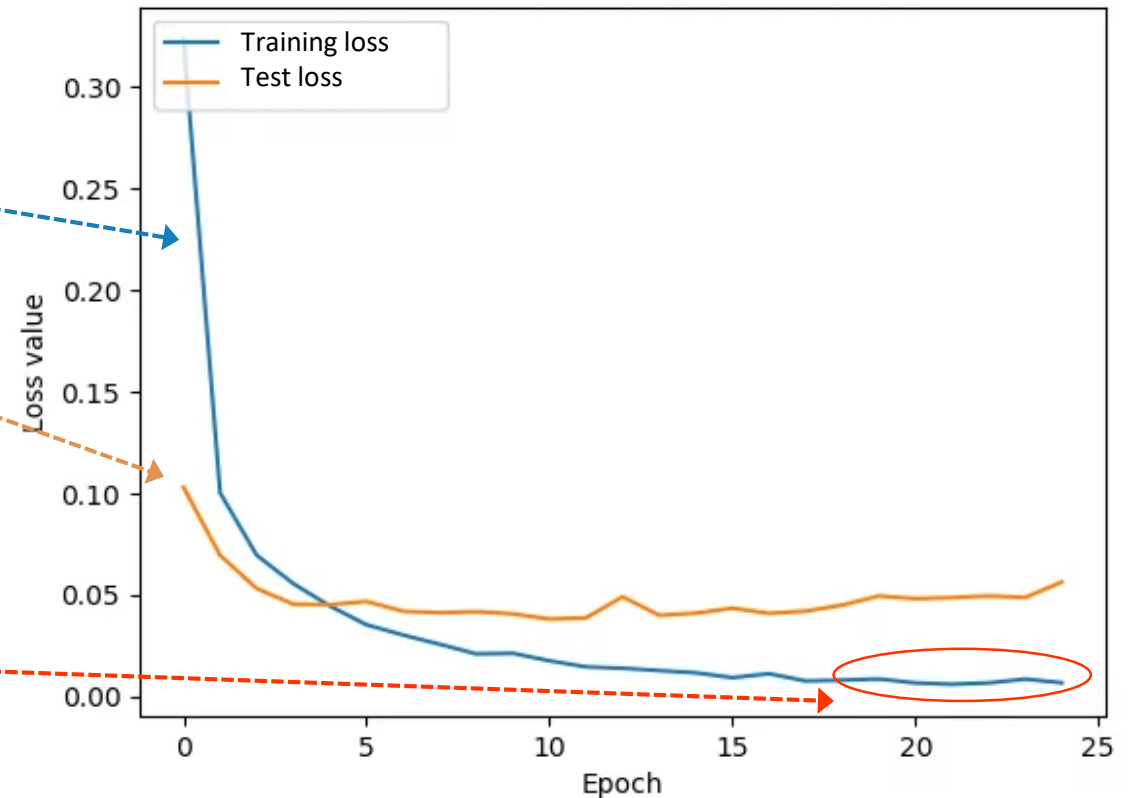
Generalisation error

The Bias-variance –trade-off

A.k.a. out-of-sample prediction performance (on test data set)

Objective = minimise generalisation error

"Out of sample predictions"
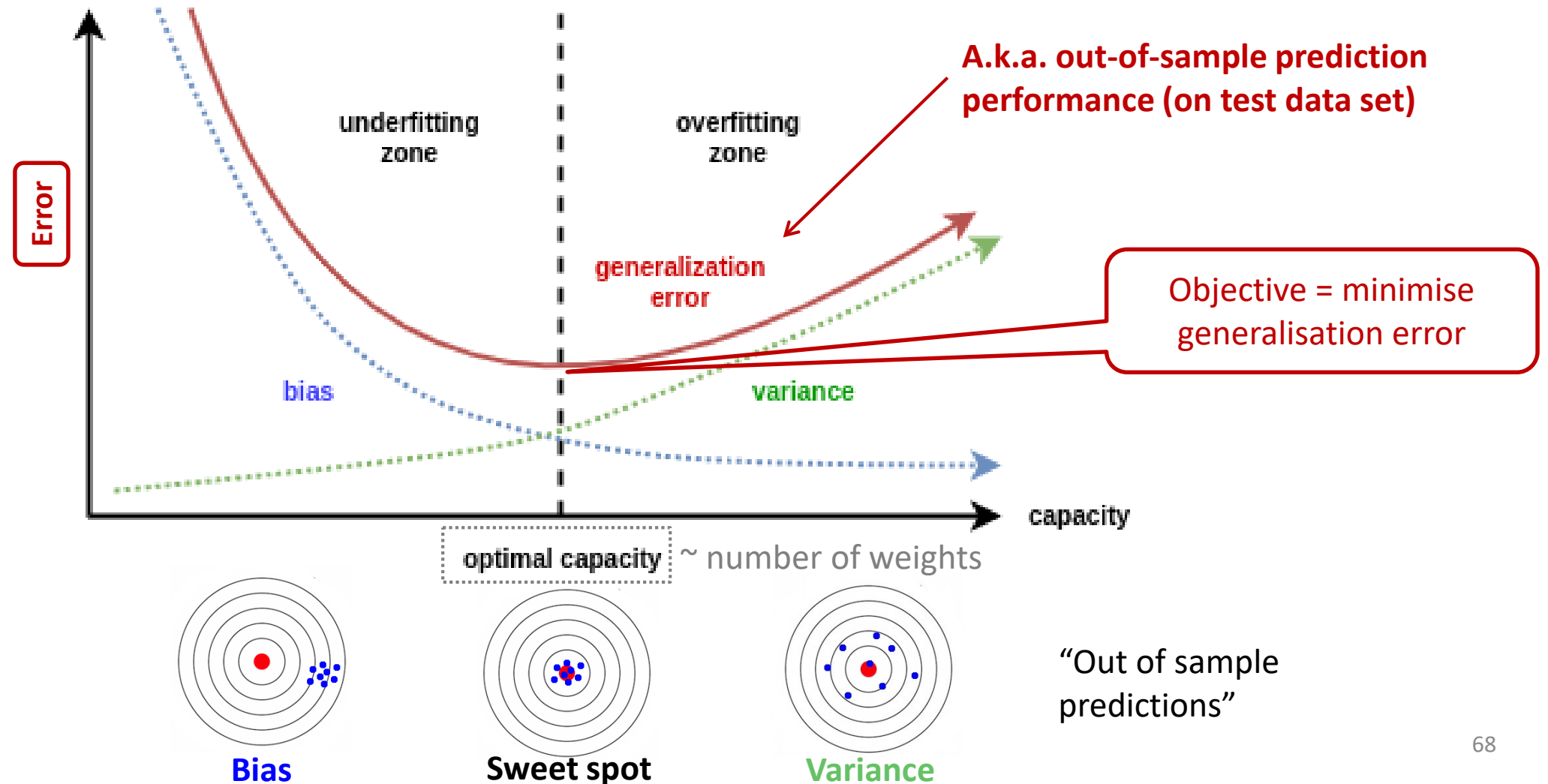
Generalisation error

K-fold cross validation

- K-fold cross validation is a method to more rigorously evaluate the generalisation performance (vs train/test split)

- K-fold cross validations is often used when tuning the hyperparameters

- When tuning hyperparameters on a simple train/test split, one risks overfitting on the particular manifestation of the test-split

- **K-fold cross validation procedure for hypertuning:**
  1. Split data in K-folds
  2. For $m = 1...M$ (models with hyperparameter settings)
     3. For $k = 1...K$
        I. Train the model, while leaving fold k out
        II. Evaluate the model performance on fold k
     4. Compute avg. performance of model m across all K folds
  5. Identify hyperparameters based on the best avg. performance across folds
  6. For deployment: train model with optimised hyperparameters on all data
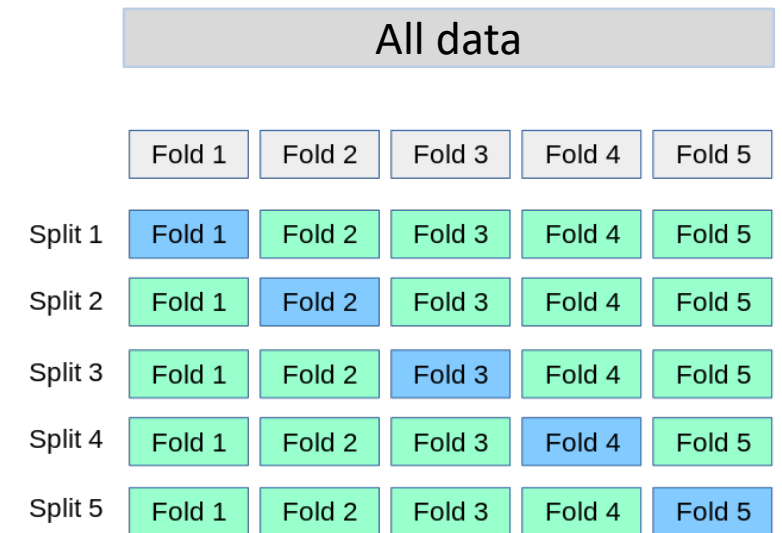
Generalisation error

In sum

- **Performance is determined** based on **out-of-sample** generalisation

- **K-fold** cross validation is a more rigorous way to test the generalisation performance than a simple train/test split

- As we use out-of-sample performance, **the number of parameters consumed is not relevant** (in contrast to theory driven models, such as in choice modelling)

- If possible, we may also want to account for the stochasticy in the performance when determining the best model typology (# nodes / hidden layers)

  →We may even need to conduct k-fold CV ~ 100 times

Early stopping

- To avoid **overfitting**, a technique called **early stopping** is commonly used

- Early stopping is in principle **not** meant to evaluate or improve the generalisation error

- Early stopping splits the data in **3 parts**:

Unseen by the network
(hold-out samples)

Early stopping

Steps:

1. **Train** the weights of network based on the **training data**

2. **Evaluate** the performance on the **validation data** & **test data**

3. **Stop** training when the performance on the **validation data** no longer improves

4. Use the **test data** for a final **unbiased evaluation** of a final model performance

Note that often no split is made between test and validation data. Only "test data" used
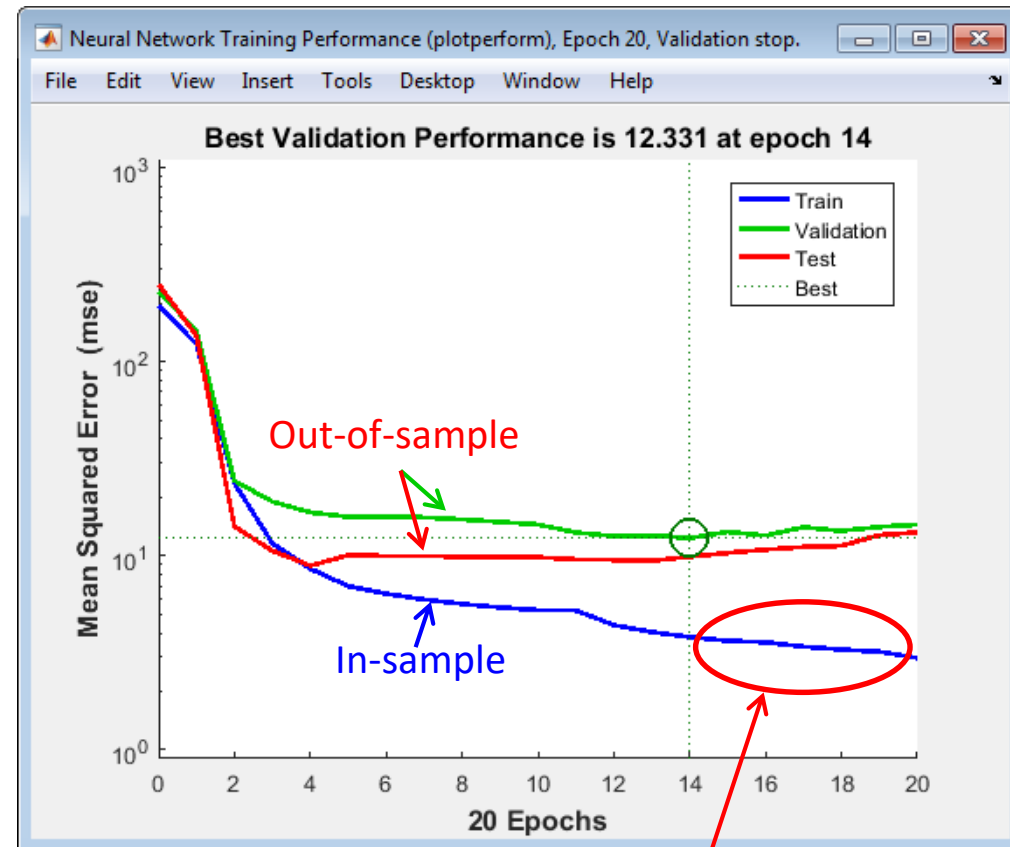
Model performance metrics

Model performance metrics

- **Next to** generalisation error, machine learning researcher often use various other **metrics to evaluate and compare models**

- The following metrics are commonly used in ML:
  - Accuracy
  - Precision
  - Recall
  - F1 score
  - Matthew's coefficient

- These metrics are computed based on the **confusion matrix**

Model performance metrics

Confusion matrix (2 classes)

- Shows counts from predicted and actual outcomes.
  - Predictions are based on **highest probability** alternative

$$P_{in} = [0.21 \ \ 0.42 \ \ 0.37] \rightarrow [0 \ \ 1 \ \ 0]$$

- Counts on the diagonal are correctly classified outcomes

- Counts on the off diagonal elements are the misclassified outcomes

		True outcome		
		Positive	Negative	
Predicted outcome	Positive	30	70	100
	Negative	20	100	120
		50	170	220

Model performance metrics

Confusion matrix (2 classes)

- True positive
  - ➤ The model correctly predicts the positive class

- False positive
  - ➤ The model incorrectly predicts positive class

- True negative
  - ➤ The model correctly predicts the negative class

- False negative
  - ➤ The model incorrectly predicts negative class

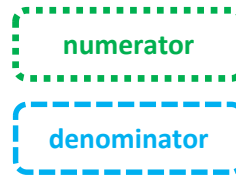|		True outcome	
		Positive	Negative
Predicted outcome	Positive	True Positives (TP)	False Positives (FP)
	Negative	False Negatives (FN)	True Negatives (TN)

Model performance metrics

Confusion matrix (2 classes)

– Accuracy [0,1]

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

Ratio of correctly predicted observations to the total observations

Model performance metrics

Confusion matrix (2 classes)

– Accuracy [0,1]

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

$$ACC = \frac{30 + 100}{30 + 100 + 70 + 20} = \frac{130}{220} = 0.59$$

Example
Mode choice prediction: TRAIN or BUS

		True outcome		
		TRAIN	BUS	
Predicted outcome	TRAIN	30 (TP)	70 (FP)	100
	BUS	20 (FN)	100 (TN)	120
		50	170	220

Model performance metrics

Confusion matrix (2 classes)

– Precision [0,1]

$$Precision\left(positive\right) = \frac{TP}{TP + FP}$$

numerator

denominator

$$Precision\left(negative\right) = \frac{TN}{TN + FN}$$

Ratio of correctly predicted positive (negative) observations to the total predicted positive (negative) observations

Model performance metrics

Confusion matrix (2 classes)

– Precision [0,1]

$$Precision\left(positive \right) = \frac{TP}{TP + FP}$$

$$Precision\left(negative \right) = \frac{TN}{TN + FN}$$

$$Precision\left(train \right) = \frac{30}{30 + 70} = 0.30$$

$$Precision\left(bus \right) = \frac{100}{100 + 20} = 0.83$$

The question that this metric answer:
*Of all travellers **predicted** to take the TRAIN (BUS), how many actually took the TRAIN (BUS)?*

Example
Mode choice prediction: TRAIN or BUS

Model performance metrics

Confusion matrix (2 classes)

− Recall [0,1]

$$Recall\left(positive\right) = \frac{TP}{TP + FN}$$

numerator

denominator

$$Recall\left(negative\right) = \frac{TN}{TN + FP} \quad \leftarrow \text{(a.k.a. specificity)}$$

Ratio of correctly predicted positive (negative) observations to the all observations actually in that class

Model performance metrics

Confusion matrix (2 classes)

– Recall [0,1]

$$Recall\left(positive\right) = \frac{TP}{TP + FN}$$

$$Recall\left(negative\right) = \frac{TN}{TN + FP}$$

$$Recall\left(positive\right) = \frac{30}{30 + 20} = 0.60$$

$$Recall\left(negative\right) = \frac{100}{100 + 70} = 0.58$$

The question that this metric answer:
Of all the passengers that truly took the TRAIN (BUS), how many did we predict?

		True outcome		
		TRAIN	BUS	
Predicted outcome	TRAIN	30 (TP)	70 (FP)	100
	BUS	20 (FN)	100 (TN)	120
		50	170	220

82

Model performance metrics

Confusion matrix (2 classes)

– F1 score [0,1]

$$F1\big(positive\big) = \frac{2TP}{2TP + FP + FN}$$

$$F1\big(negative\big) = \frac{2TN}{2TN + FP + FN}$$

F1 score is the **harmonic mean** of **precision** and **recall**. Especially in case of **imbalanced** class distribution and in case of Precision and Recall are important, F1 is preferred over Accuracy

numerator

denominator

		True outcome		
		Positive	Negative	
Predicted outcome	Positive	TP	FP	
	Negative	FN	TN	

Model performance metrics

Confusion matrix (2 classes)

- F1 score [0,1]

$$F1\left(positive\right) = \frac{2TP}{2TP + FP + FN}$$

$$F1\left(negative\right) = \frac{2TN}{2TN + FP + FN}$$

$$F1\left(positive\right) = \frac{60}{60 + 70 + 20} = 0.40$$

$$F1\left(negative\right) = \frac{200}{200 + 70 + 20} = 0.69$$

Example
Mode choice prediction: TRAIN or BUS

		True outcome		
		TRAIN	BUS	
Predicted outcome	TRAIN	30 (TP)	70 (FP)	100
	BUS	20 (FN)	100 (TN)	120
		50	170	220

Model performance metrics

Confusion matrix (2 classes)

– Matthew's correlation coefficient

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}}$$

MCC is similar in use and purpose as F1 score, but has the advantage that it gives a single number (not per class)

		True outcome		
		TRAIN	BUS	
Predicted outcome	TRAIN	30 (TP)	70 (FP)	100
	BUS	20 (FN)	100 (TN)	120
		50	170	220

Model performance metrics

Confusion matrix (2 classes)

− Matthew's correlation coefficient

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}}$$

$$MCC = \frac{30 \cdot 100 - 70 \cdot 20}{\sqrt{(30 + 70) \cdot (30 + 20) \cdot (100 + 70) \cdot (100 + 20)}}$$

$$MCC = \frac{1600}{\sqrt{5000 \cdot 20400}} = 0.158$$

		True outcome		
		TRAIN	BUS	
Predicted outcome	TRAIN	30 (TP)	70 (FP)	100
	BUS	20 (FN)	100 (TN)	120
		50	170	220

86

Model performance metrics

Confusion matrix (2 classes)

– Matthew's correlation coefficient

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}}$$

Derive the minimum and maximum value that MCC can attain: [a,b]? 🤔

		True outcome		
		Positive	Negative	
Predicted outcome	Positive	TP	FP	
	Negative	FN	TN	

Model performance metrics

Confusion matrix (classes > 2)

– Same logic (called mico-averaging)

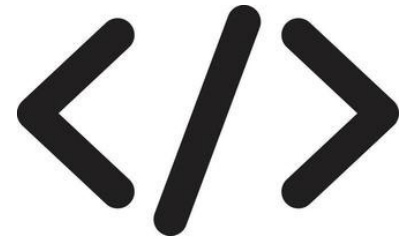$$ACC = \frac{30 + 100 + 80}{320} = 0.66$$

$$Precision(class1) = \frac{30}{30 + 40 + 30} = 0.30$$

$$Recall(class1) = \frac{30}{30 + 10 + 10} = 0.60$$

$$F1(class1) = \frac{2 \cdot 30}{2 \cdot 30 + \left(40 + 30\right) + \left(10 + 10\right)} = 0.40$$

		True outcome			
		Class 1	Class 2	Class 3	
Predicted outcome	Class 1	30	40	30	100
	Class 2	10	100	10	120
	Class 3	10	10	80	100
		50	150	120	320

Lab session 1

Artificial Neural Networks for Discrete Choice Analysis

Lab session 1

Lab sessions aim to

- Show and reinforce how the ML models and ideas presented in class are put to practice.
- Help you gather hands-on machine learning skills.

Learning objective for lab session 1

1. Prepare (choice) data for training Artificial Neural Networks
2. Train MultiLayerPerceptron (MLP) neural network for a classification task
3. Tune the hyperparameter and network architectures to improve the model performance
4. Assess the performance of competing models, based on various performance measures, including confusion matrices, and Precision, Recall, F1 score and Matthew's coefficient

Exercises

- The notebooks contain several exercises
- Do not just run the entire notebook at once and declare victory

Lab session 1

Practicalities

- Jupyter notebooks and **instructions** can be downloaded from GitHub

- Work in Google Colab or local

- You can get support from TAs (Francisco and Lucas), and fellow students

- Duration: **2 hr**

Answers

- Jupyter notebooks with codes and answers to exercises are distributed at the end of the day

Lab session 1

Programming language </>

- We use Python because it is the most commonly used language for ML

- In Python there are numerous packages available to train ANNs, e.g.
    - Scikit-learn (Python library)   → Beginner & intermediate users
    - Tensorflow (Python library)   → Advanced users
    - PyTorch (Python library)   → Advanced users

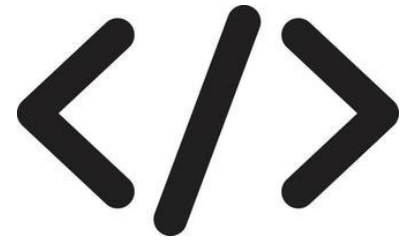- There are also good packages for e.g. R

Lab session 1

Data

- Swiss Metro data (widely used benchmark data set)

- Stated Choice data

- Straightforward experimental design
  - 3 alternatives: Train, SM and Car
  - Two generic attributes: Travel cost, Travel time

	Train	SwissMetro	Car
Cost [CHF]	48	52	65
Time [minutes]	112	63	117
Headway time [minutes]	120	20	N/A

Lab session 1

Let's GO!