Algorithms Approaches to AI

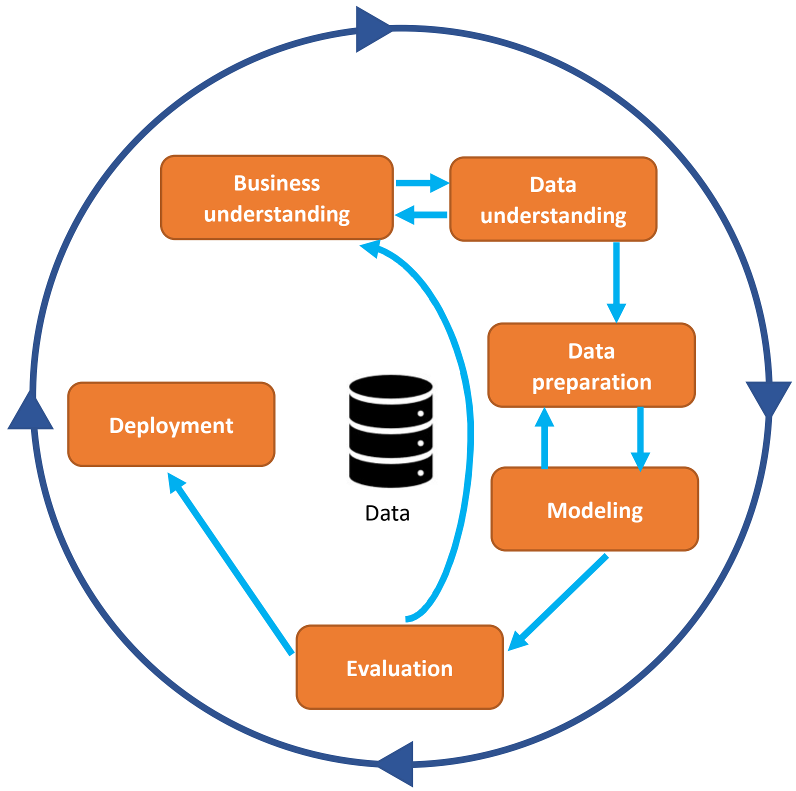
Lesson 5: Discussing the common algorithms approaches to achieving artificial intelligence

Lesson	Date	Topics
1	18-Apr-22	Introduction to AI
2	21-Apr-22	Basics of Python
3	25-Apr-22	Linear Algebra + Python for AI (I)
4	28-Apr-22	Python for AI (II)
5	**4-May-22**	**AI approaches**
6	6-May-22	AI concepts and techniques
7	9-May-22	AI concepts and techniques
8	12-May-22	AI concepts and techniques
9	17-May-22	Written Test
10	23-May-22	Future of AI, revision and review

AI approaches

- Processes
- Approaches over the years
- ML and DL

CRISP-DM

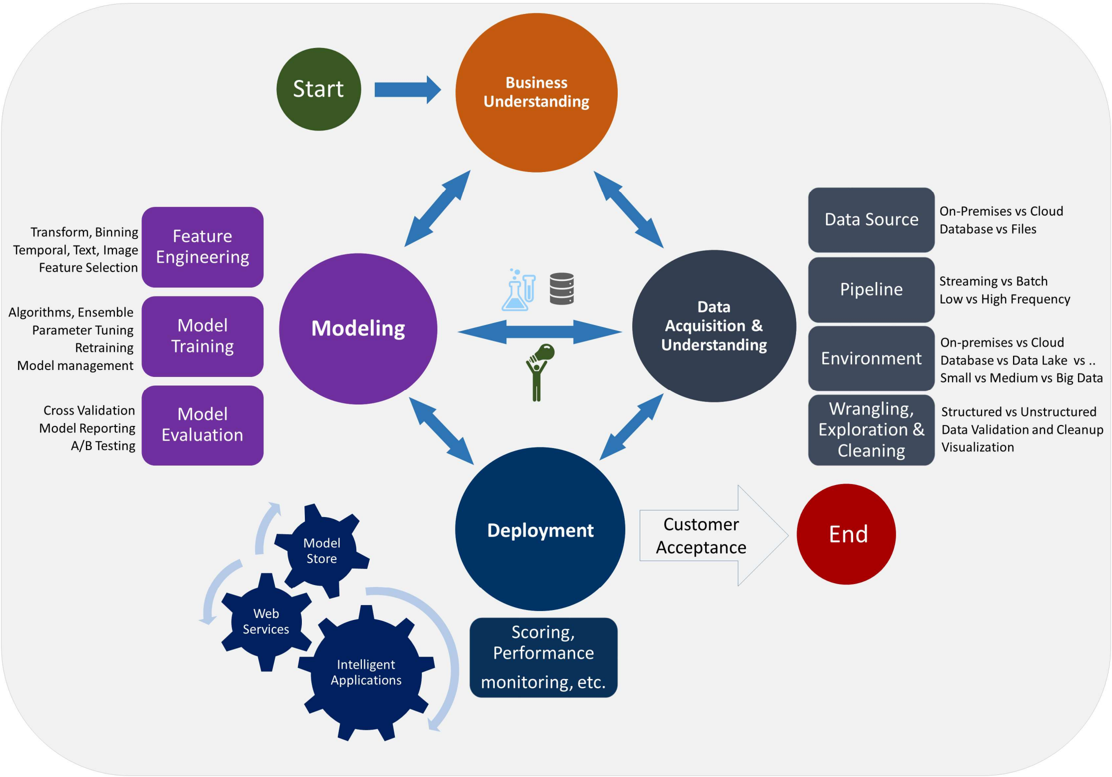


Business understanding
Data understanding
Data preparation
Modeling
Evaluation
Deployment
Data

CRISP-DM

1. Business understanding – What does the business need?
2. Data understanding – What data do we have / need? Is it clean?
3. Data preparation – How do we organize the data for modeling?
4. Modeling – What modeling techniques should we apply?
5. Evaluation – Which model best meets the business objectives?
6. Deployment – How do stakeholders access the results?

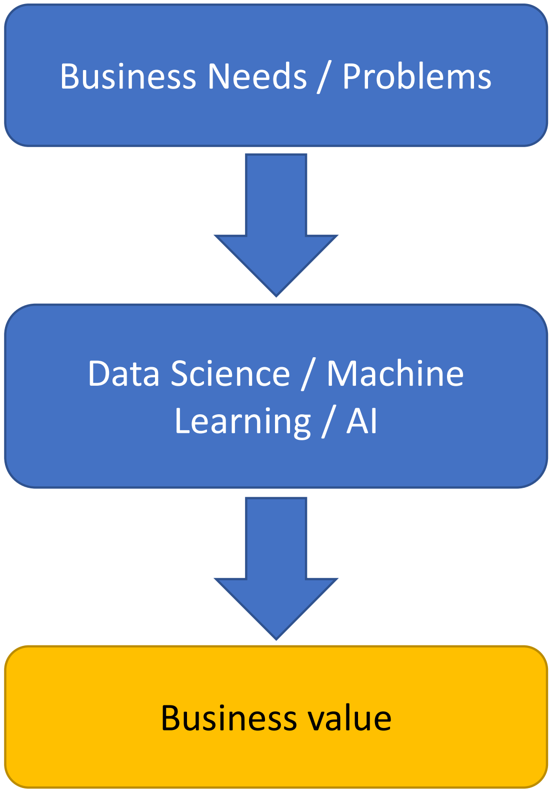
Team Data Science Process



How to succeed?

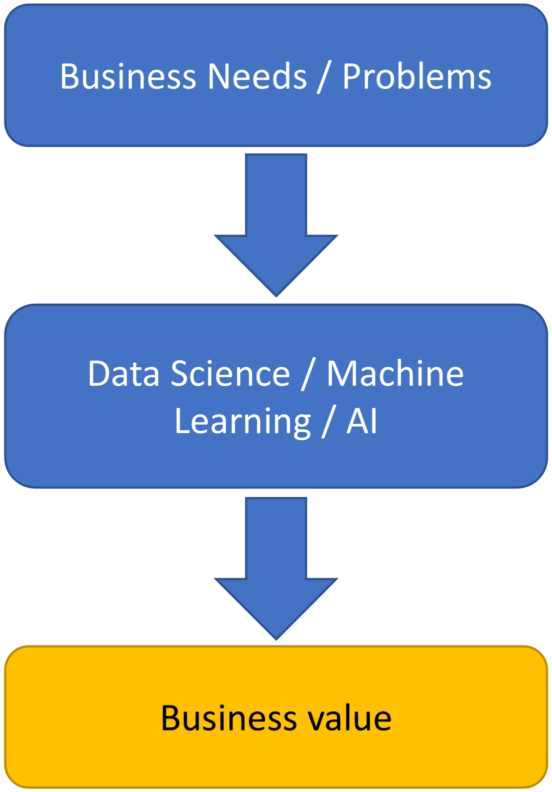
- Business/Domain expertise
- Technical expertise
  - Programming/Coding
  - Math
  - Algorithms
  - .....

- Working in teams
- Communicating insights/values



How to succeed?

- What am I trying to solve/improve?
- Collect data + clean data
- Explore the data
- Apply the algorithm / technique
- Which approach provided the best model?
- How does all this help the problem and the business?



4 Main Approaches

1. Cybernetics and brain simulation (1950s, 1960s)
2. Symbolic (mid 1950s)
3. Sub-symbolic (1980s)
4. Statistical (1990s)

4 Main Approaches

1. Cybernetics and brain simulation (1940s, 1950s)
2. Symbolic (mid 1950s)
3. Sub-symbolic (1980s)
4. Statistical (1990s)

1940s – 1950s
Use electronic networks to exhibit basic intelligence
Largely abandoned but some concepts were revisited in 1980s

Grey Walter's tortoises
https://www.youtube.com/watch?v=lLULRlmXkKo

4 Main Approaches

1. Cybernetics and brain simulation (1940s, 1950s)
2. **Symbolic (mid 1950s)**
3. Sub-symbolic (1980s)
4. Statistical (1990s)

explore the possibility that human intelligence could be reduced to symbol manipulation

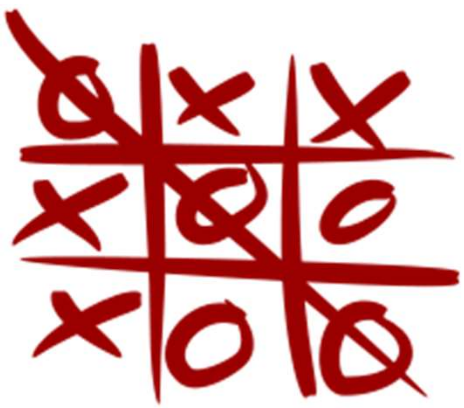
Good Old Fashioned AI (GOFAI)

Manipulate a symbolic representation to find solutions to problems

- Cognitive simulation
  - human problem-solving skills and attempt to formalize them
- Logic-based
  - abstract reasoning and problem-solving
  - formal logic to solve knowledge representation, planning and learning
- Anti-logic or scruffy
  - difficult problems in vision and natural language processing required ad-hoc solutions
- Knowledge-based
  - expert systems, stores facts and rules

Symbolic approach

- Rules engines, expert systems or knowledge graphs
- Rules connect symbols in a relationship similar to an If-Then
- Piles of nested if-then statements drawing conclusions about entities and their relations



1.If someone has a "threat" (that is, two in a row), take the remaining square. Otherwise,
2.if a move "forks" to create two threats at once, play that move. Otherwise,
3.take the center square if it is free. Otherwise,
4.if your opponent has played in a corner, take the opposite corner. Otherwise,
5.take an empty corner if one exists. Otherwise,
6.take any empty square

Symbolic approach

- Expert system processes the rules to make deductions and to determine what additional information it needs, i.e. what questions to ask, using human-readable symbols

Symbolic approach

- Difficulty of revising beliefs once encoded in a rules engine
- The more rules you add, the more knowledge is encoded in the system, but additional rules can’t undo old knowledge
- Computer itself doesn’t know what the symbols mean; i.e. they are not necessarily linked to any other representations of the world in a non-symbolic way

4 Main Approaches

1. Cybernetics and brain simulation (1940s, 1950s)
2. Symbolic (mid 1950s)
3. **Sub-symbolic (1980s)**
4. Statistical (1990s)

Many believed that symbolic systems would never be able to imitate all the processes of human cognition

Approach intelligence without specific representations of knowledge

Subsymbolic approach

Embodied intelligence

- interface agent, that interacts with the environment through a physical body
- E.g. a human or a cartoon animal, graphically conversational agents
- gesture, facial expression

Computational intelligence and soft computing

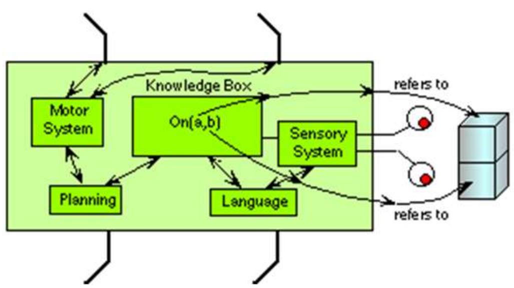
- solutions to problems which cannot be solved with complete logical certainty
- an approximate solution is often sufficient

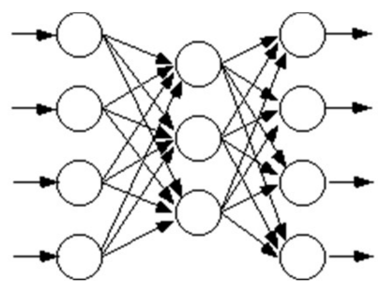
From Symbolic to Subsymbolic

- Symbolic
  - mid 1950s
  - classical artificial intelligence / Good Old Fashioned AI (GOFAI)
  - access to digital computers
  - reduce human intelligence to symbol manipulation
- Subsymbolic
  - 1980s
  - connectionist
  - work on a lower level than symbols
  - no symbolic representation of rules or properties
  - recognize pattern from data
- Hybrid of symbolic and subsymbolic

Comparison: Symbolic vs Subsymbolic

Symbolic	Subsymbolic
Explicitly represented by rules	Implicitly represented by experiences
Rules created by human intervention Logic and search to solve problem Knowledge is represented by sentences in formal languages	Establishes correlations between inputs and output Knowledge is represented by weights on connections in a network
Eg. IBM's Deep Blue win chess champion 1997	**Eg. Genetic algorithms, neural networks, machine learning, deep learning**





Comparison: Symbolic vs Subsymbolic

Symbolic	Subsymbolic
Easy to understand steps and reason behind conclusion	Difficult to understands steps and reasons behind conclusion May have undesired outcome in critical and high risk decisions
Require less data	Require large data
Rules have to be coded to learn	More robust against noise
	Easier to scale up
	Better performance

Images that block recognition of person

https://www.youtube.com/watch?v=MIbFvK2S9g8

4 Main Approaches

1. Cybernetics and brain simulation (1940s, 1950s)
2. Symbolic (mid 1950s)
3. Sub-symbolic (1980s)
4. Statistical (1990s)

Sophisticated mathematical tools to solve specific subproblems.

Results are both measurable and verifiable, and they have been responsible for many of AI's recent successes.

“Far better an approximate answer to the right *question, which is often vague, than an* exact *answer to the wrong question, which can always be made precise.”*

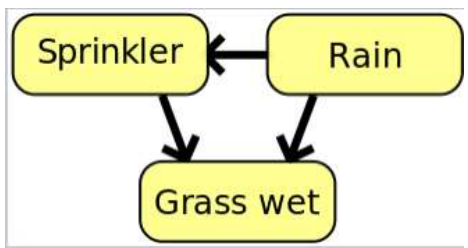
- John Turkey

Statistics approaches

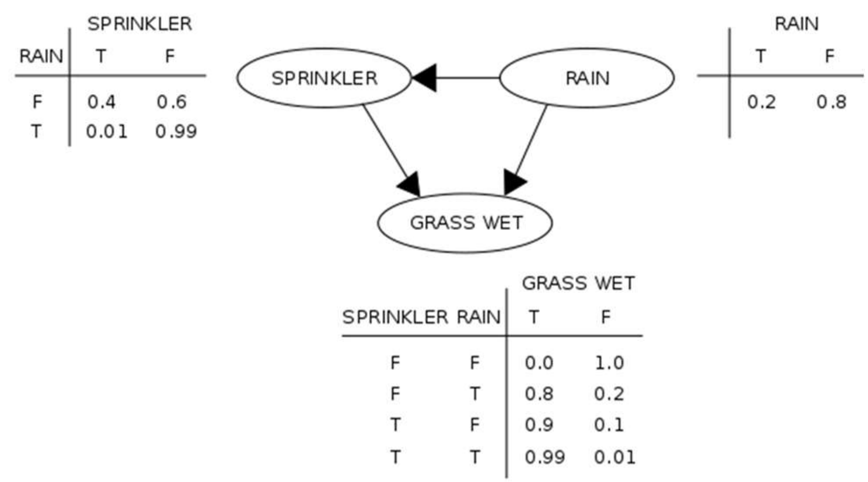
- Non-symbolic, sophisticated mathematical tools
- E.g. Bayesian, Hidden Markov model (HMM)
- A mathematical model that embodies a set of statistical assumptions concerning the generation of some sample data and similar data from a larger population
- E.g. six-sided dice
  - first statistical assumption: probability of each face (1, 2, 3, 4, 5, and 6) coming up is 1/6
  - alternative statistical assumption: probability of the face 1 coming up is 1/8 (because the dice are weighted)

Bayesian

- Probabilistic graphical model (a type of statistical model) that represents a set of variables and their conditional dependencies via a Directed Acyclic Graph (DAG)
- E.g. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases



A simple Bayesian network. Rain influences whether the sprinkler is activated, and both rain and the sprinkler influence whether the grass is wet.



A simple Bayesian network with conditional probability tables

A quick look at Bayes' Theorem

Let us play a game: Can you guess the job of Steve?

a) Writer
b) Engineer
c) Sales executive
d) Farmer

Steve is shy and quiet

Steve's income varies across the months

The population breakdown is as follows

- 1% are writers,
- 4% are engineers,
- 5% are sales executives, and
- 90% are farmers,

Bayes' Theorem

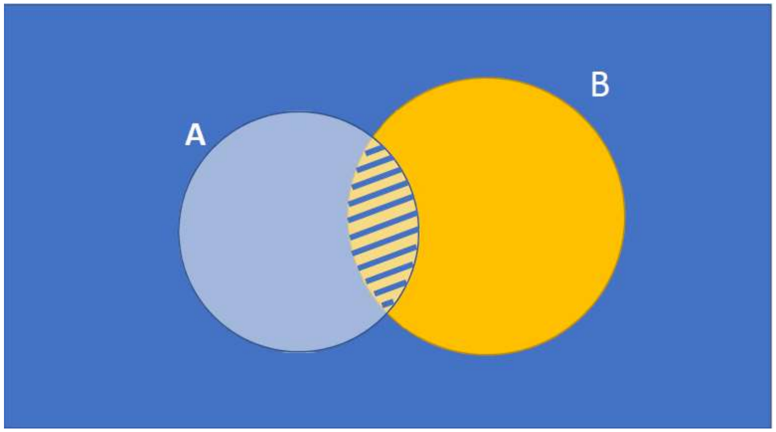
$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

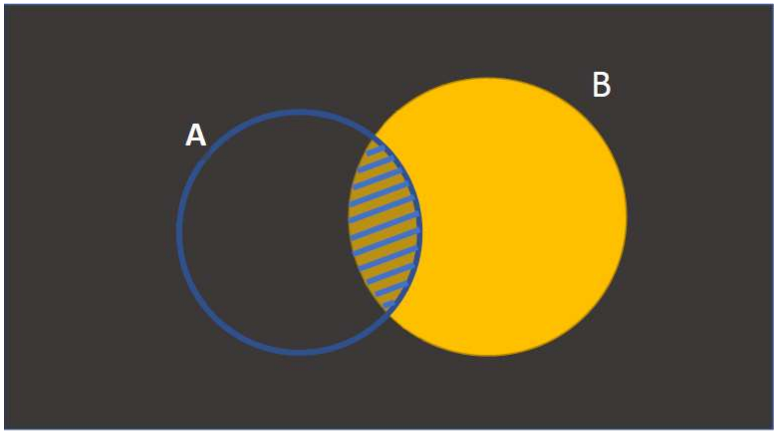
$$P(A \cap B) = \boxed{P(A|B) * P(B)}$$

$P(A \cap B) = P(B \cap A)$

$$P(B|A) = \frac{P(B \cap A)}{P(A)}$$

$$= \frac{\boxed{P(A|B) * P(B)}}{P(A)}$$





Example

- Weather
  - P(Rain) = 4/8
  - P(Sunny) = 4/8
- Traffic
  - P(Jam) = 3/8
  - P(No Jam) = 5/8
- Traffic
  - P(Rain | Jam) = 2/3

Day	Weather	Traffic Jam
0	Rain	Y
1	Rain	N
2	Rain	N
3	Sunny	N
4	Sunny	N
5	Rain	Y
6	Sunny	Y
7	Sunny	N

- $P(Jam|Rain) = \dfrac{P(Jam)\, P(Rain|Jam)}{P(Rain)}$

$$= \frac{1}{2}$$

Traffic

P(Jam|Rain) = 2/4

- 1 % of the population have a genetic disorder
- A test is 95% certain of correctly identifying the genetic disorder
- If John went for the test and was tested positive (have the genetic disorder), what is the probability that John has the disorder.

- T – tested positive

- $P(\text{has disorder} \mid T) = \dfrac{P(T \mid \text{has disorder}) * P(\text{has disorder})}{P(T)}$

$$= \frac{P(T \mid \text{has disorder}) * P(\text{has disorder})}{P(T \mid \text{has disorder}) * P(\text{has disorder}) + P(T \mid \text{has no disorder}) * P(\text{has no disorder})}$$

$$= \frac{0.95 \times 0.01}{0.95 \times 0.01 + 0.05 \times 0.99}$$

$$= 0.161$$

	Has Disorder	Do not have Disorder
Test Positive		
Test Negative		

- Considering the low numbers of occurrence in the population, the chance is 16% after the first positive test.

- 1 % of the population have a genetic disorder
- A test is 95% certain of correctly identifying the genetic disorder
- If John went for a second test and was tested positive (have the genetic disorder) again, what is the probability that John has the disorder.

- T2 – tested positive

- $P(\text{has disorder} \mid T2) = \dfrac{P(T2 \mid \text{has disorder}) * P(\text{has disorder})}{P(T2)}$

$$= \frac{P(T2 \mid \text{has disorder}) * P(\text{has disorder})}{P(T2 \mid \text{has disorder}) * P(\text{has disorder}) + P(T2 \mid \text{has no disorder}) * P(\text{has no disorder})}$$

$$= \frac{0.95 \times \underline{0.161}}{0.95 \times \underline{0.161} + 0.05 \times \underline{0.839}}$$

$$= 0.785$$

- Considering the second test was also positive, the probability of John having the disorder is 79%.

Application of Bayes' Theorem

- What is the probability an email is a spam given that it has the word "prize"

- $P(\text{ spam} \mid \text{"prize"}) = \dfrac{P(\text{"prize"} \mid \text{spam}) * P(\text{spam})}{P(\text{"prize"})}$

    - P("prize") = Possible outcome of "prize" in spam + Possible outcomes of "price" in non-spam emails

- $P(\text{spam} \mid \text{"}\mathbf{prize}\text{"} \cap \text{"}\mathbf{million}\text{"})$

$$= \frac{P(\text{"}\mathbf{prize}\text{"} \mid \text{spam}) * P(\text{"}\mathbf{million}\text{"} \mid \text{spam}) * P(\text{spam})}{P(\text{"}\mathbf{prize}\text{"} \cap \text{"}\mathbf{million}\text{"})}$$

$$= \frac{P(\text{"}\mathbf{prize}\text{"} \mid \text{spam}) * P(\text{"}\mathbf{million}\text{"} \mid \text{spam}) * P(\text{spam})}{P(\text{"}\mathbf{prize}\text{"} \mid \text{spam}) * P(\text{"}\mathbf{million}\text{"} \mid \text{spam}) * P(\text{spam}) + P(\text{"}\mathbf{prize}\text{"} \mid \text{legit}) * P(\text{"}\mathbf{million}\text{"} \mid \text{legit}) * P(\text{legit})}$$

Use of Hidden Markov Model (HMM)

- Popularly used in Natural Language Processing (NLP)
  - Part of speech tagging
  - Named entity recognition
  - Text chunking

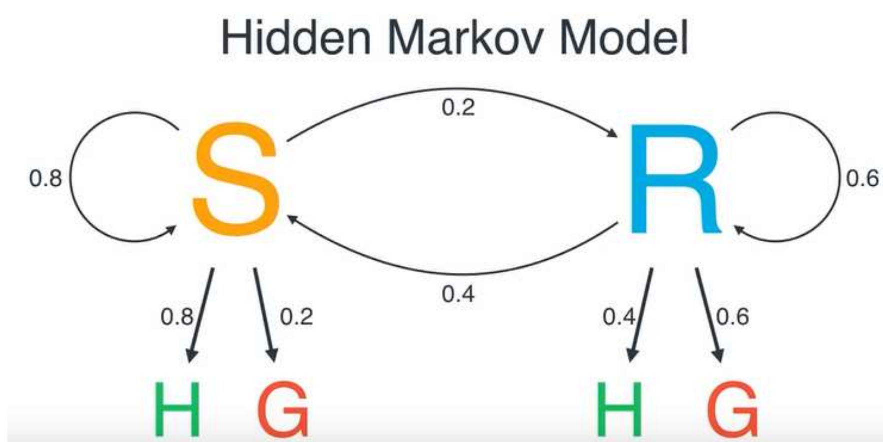
- Also popular in computational biology and central to speech recognition

Hidden Markov Model (HMM)

- System being modelled is assumed to be a Markov process with unobserved (i.e. hidden) states
- Represented as the simplest dynamic Bayesian network

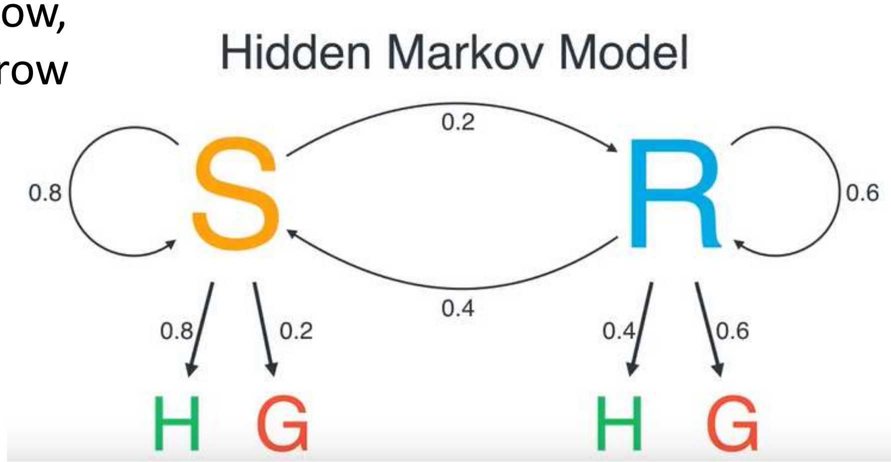
States (hidden)

Observations (visible)



Example

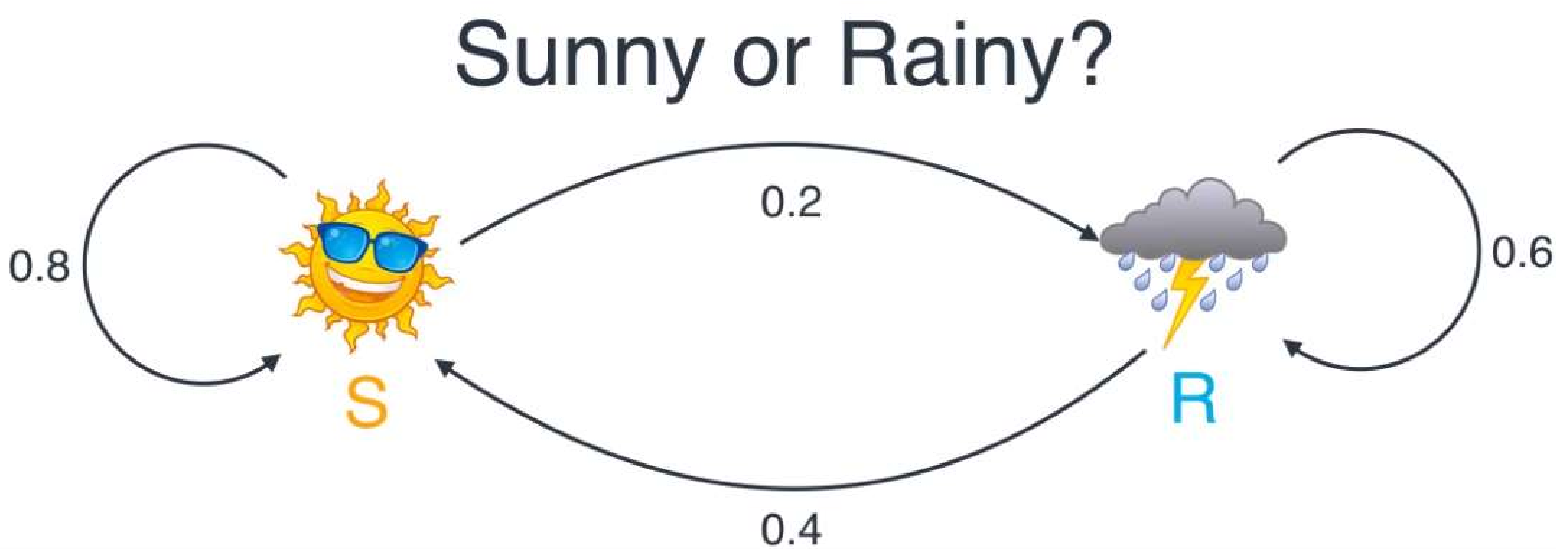
- states = ('Rainy', 'Sunny’)
- observations = (‘Happy’, ‘Grumpy’)
- start_probability = {'Rainy': 0.6, 'Sunny': 0.4}
- transition_probability = {
  'Rainy’ today : {'Rainy': 0.6, 'Sunny': 0.4} tomorrow,
  'Sunny' today : {'Rainy': 0.2, 'Sunny': 0.8} tomorrow
  }
- emission_probability = {
  'Rainy' : {‘happy’': 0.4, ‘grumpy': 0.6},
  'Sunny' : {‘happy’': 0.8, ‘grumpy': 0.2},
  }



Source: https://www.youtube.com/watch?v=kqSzLo9fenk

How did we find the probabilities?

Finding probability of Sunny or Rainy



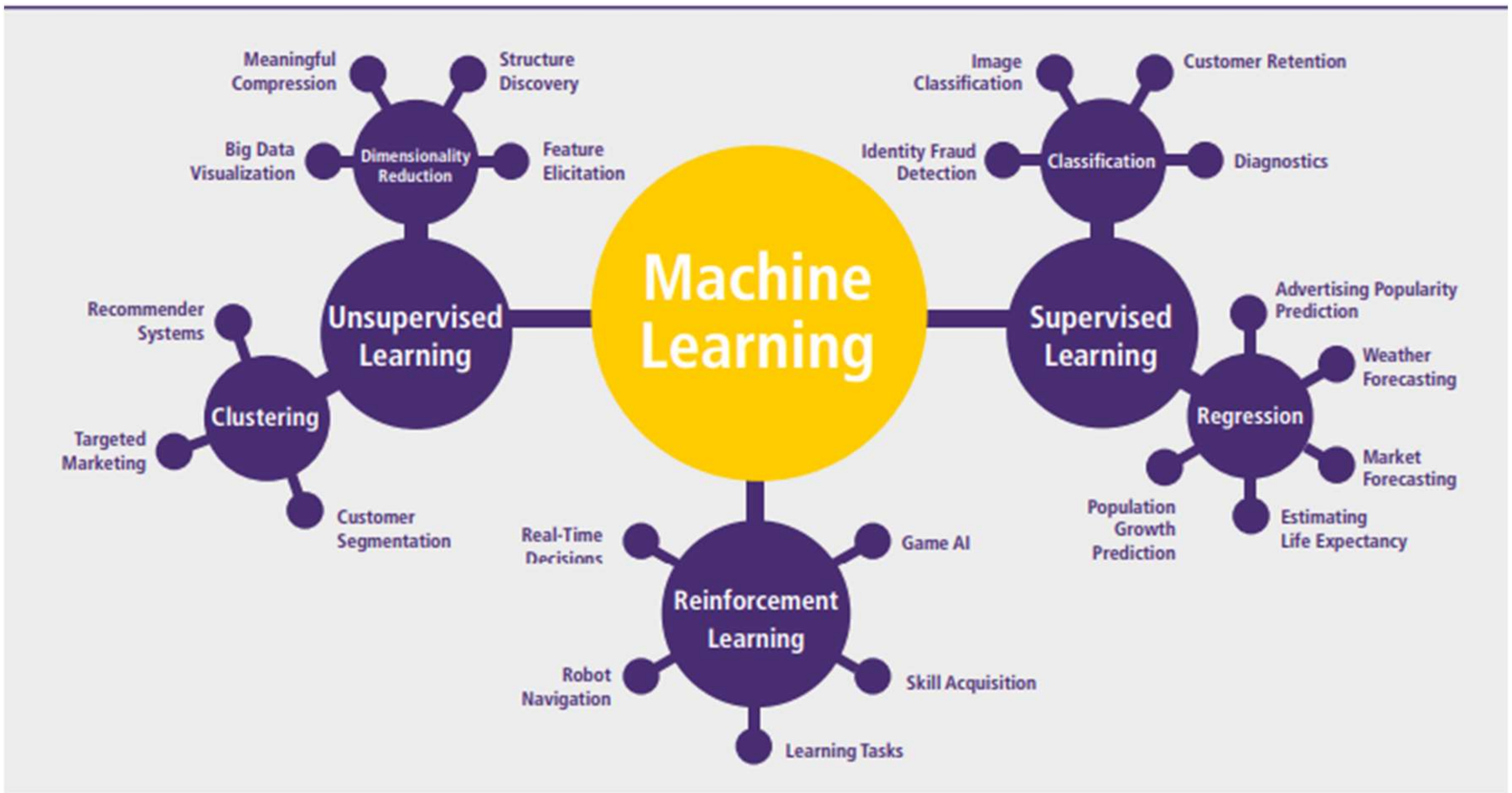
$S = 0.8S + 0.4R$ $\quad$ $S + R = 1$ $\quad$ $R = 0.2S + 0.6R$

$$\boxed{\begin{array}{l} S = 2/3 \\ R = 1/3 \end{array}}$$

Reference

- https://www.youtube.com/watch?v=kqSzLo9fenk
- 4.40: Hidden Markov Model
- 12.08: Bayes Theorem
- 22:48 Viterbi Algorithm
- 30:20 Applications

Back to ML, DL



Machine Learning
Unsupervised Learning
Dimensionality Reduction
Meaningful Compression
Structure Discovery
Big Data Visualization
Feature Elicitation
Clustering
Recommender Systems
Targeted Marketing
Customer Segmentation
Supervised Learning
Classification
Image Classification
Customer Retention
Identity Fraud Detection
Diagnostics
Regression
Advertising Popularity Prediction
Weather Forecasting
Market Forecasting
Estimating Life Expectancy
Population Growth Prediction
Reinforcement Learning
Real-Time Decisions
Game AI
Robot Navigation
Skill Acquisition
Learning Tasks

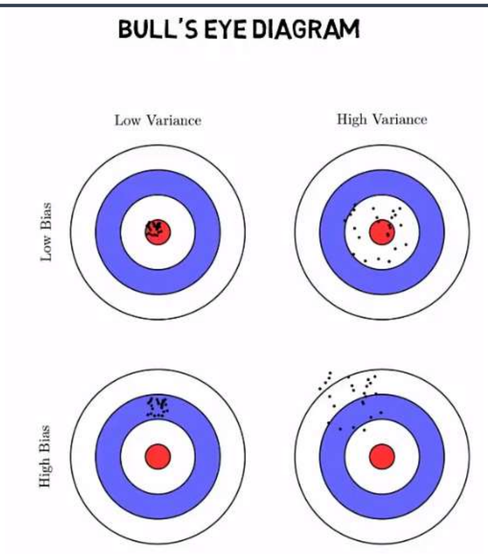
Supervised learning

- Each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal)
- Analyses the training data and produces an inferred function, which can be used for mapping new examples
- Requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way
- Algorithms
  - Support Vector Machines
  - Linear regression
  - Logistic regression
  - Naive Bayes

  - Linear discriminant analysis
  - Decision trees
  - K-Nearest Neighbour algorithm (KNN)
  - Neural Networks (Multilayer perceptron)

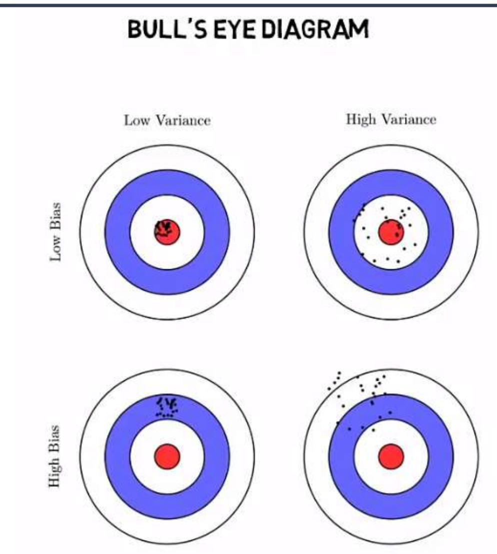
Issues for consideration

- Bias-variance trade-off
- Function complexity and amount of training data
- Dimensionality of the input space. Y=aX + bY +…
- Noise in the output values. E.g. image noise|lighting
- Heterogeneity of the data. i.e. different samples
- Redundancy in the data. i.e. repeated

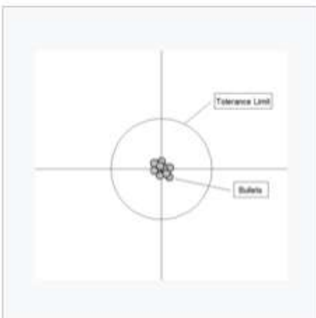


Issues for consideration

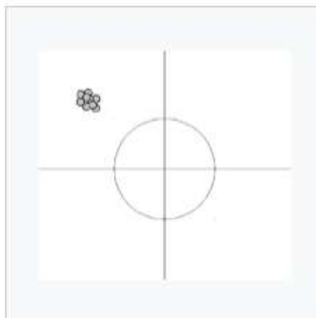
- Bias-variance trade-off
- Function complexity and amount of training data
- Dimensionality of the input space. Y=aX + bY +…
- Noise in the output values. E.g. image noise|lighting
- Heterogeneity of the data. i.e. different samples
- Redundancy in the data. i.e. repeated



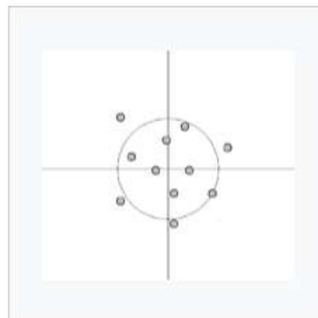
Bias-variance trade-off



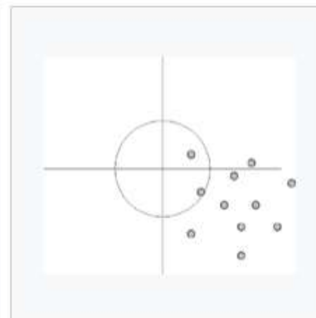
bias low, variance low



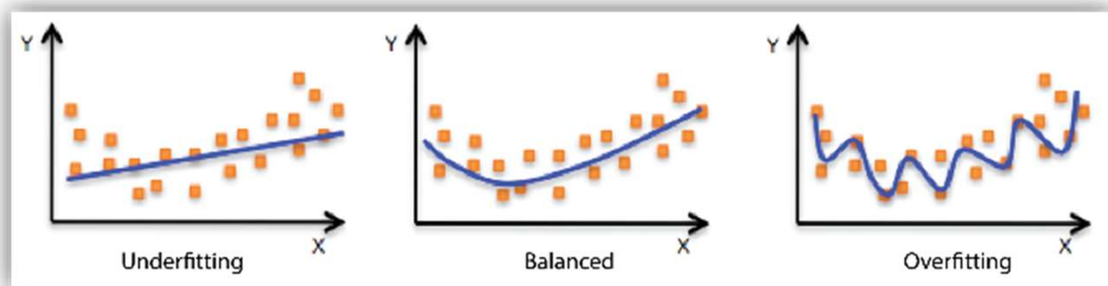
bias high,
variance low:



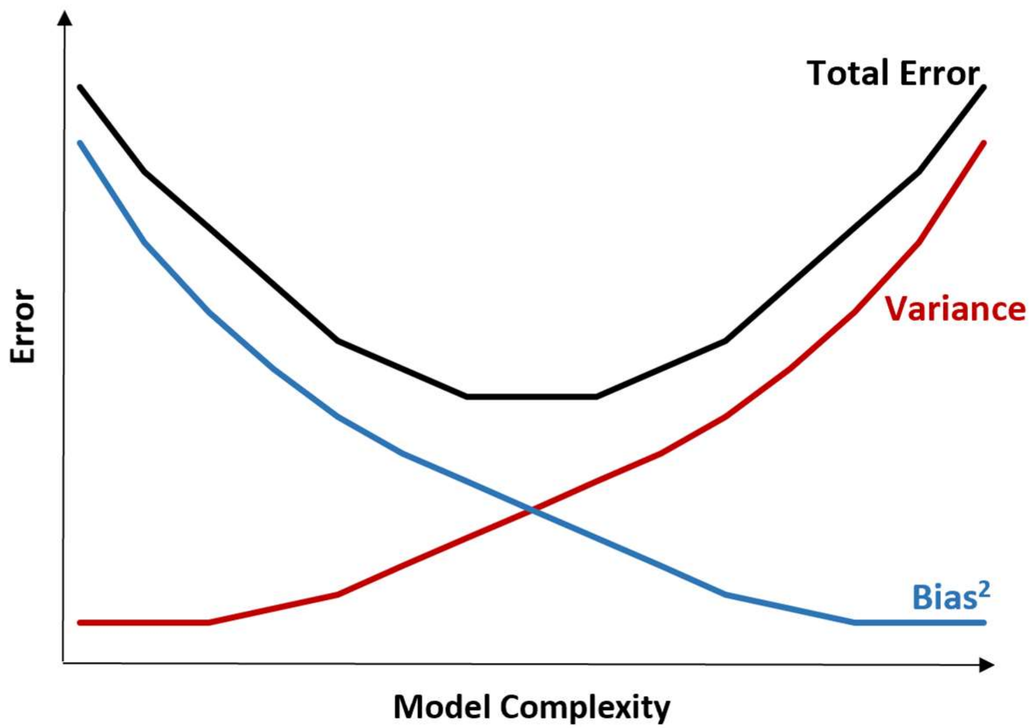
bias low,
variance high:

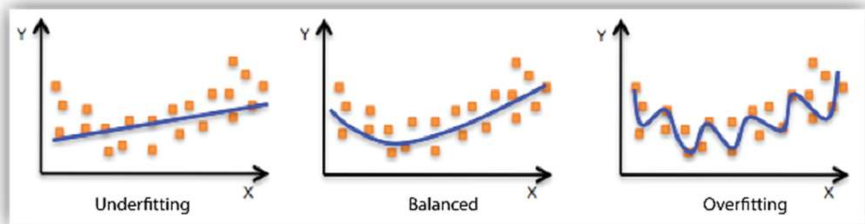


bias high,
variance high:



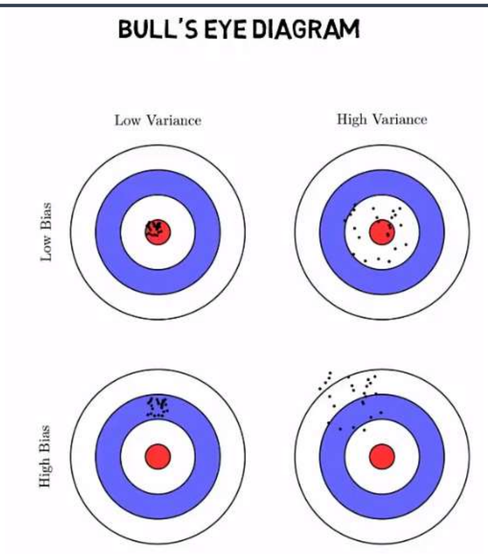
Bias-variance trade-off

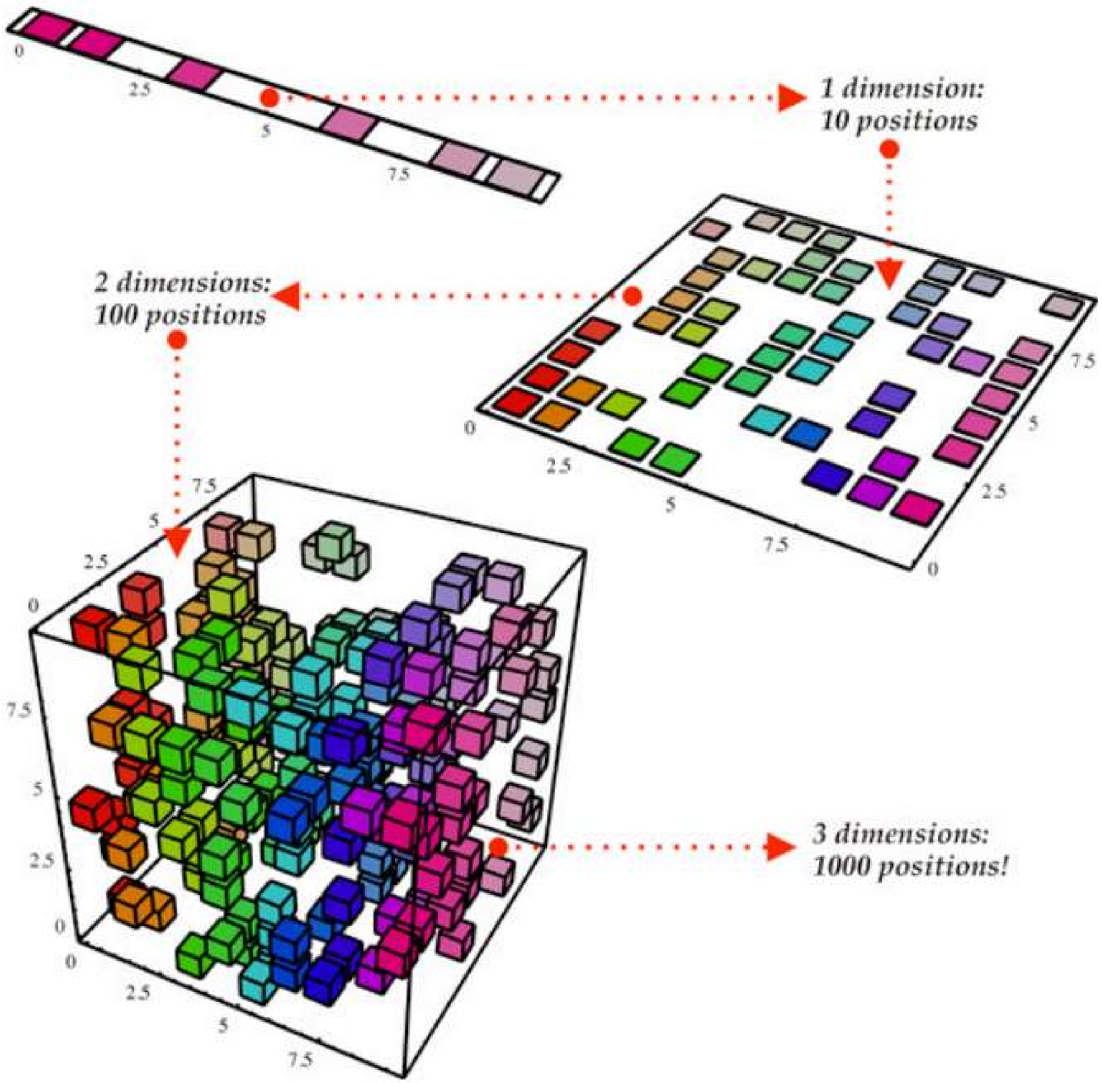




Issues for consideration

- Bias-variance trade-off
- Function complexity and amount of training data
- Dimensionality of the input space. Y=aX + bY +…
- Noise in the output values. E.g. image noise|lighting
- Heterogeneity of the data. i.e. different samples
- Redundancy in the data. i.e. repeated




1 dimension:
10 positions
2 dimensions:
100 positions
3 dimensions:
1000 positions!

Unsupervised learning

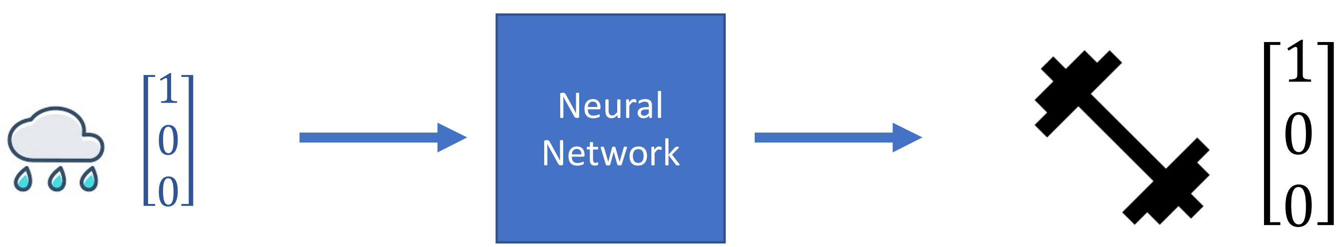
- Identifies commonalities in the data and reacts based on the presence or absence of such commonalities in each new piece of data
- Learn relationships between elements in a data set and classify the raw data without "help“
- Algorithms
  - Clustering
  - Anomaly detection
  - Neural Networks

Reinforcement learning

- Take actions in an environment so as to maximize some notion of cumulative reward
- Also known as approximate dynamic programming, or neuro-dynamic programming
- Agent interacts with its environment in discrete time steps
  - At each time, the agent receives an observation, which typically includes the reward
  - Agent chooses an action from the set of available actions, which is subsequently sent to the environment.
  - Environment moves to a new state and the reward associated with the transition is determined
  - Goal of a reinforcement learning agent is to collect as much reward as possible

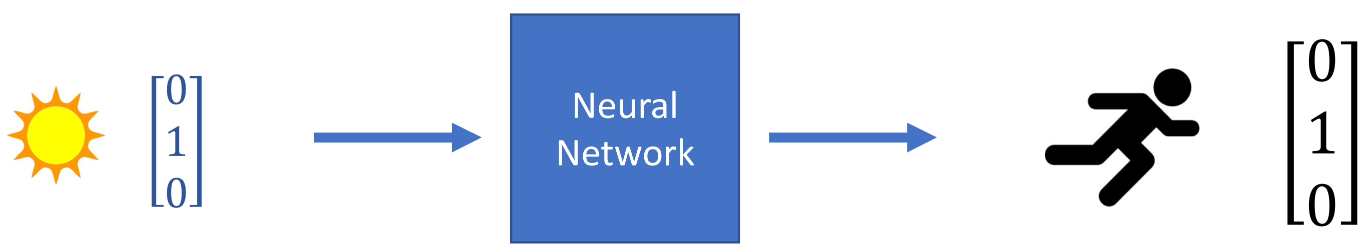
Basics of Neural Network

- If it is raining, John will go the gym
- If it is sunny, John will go running
- If it is overcast, John will go cycling



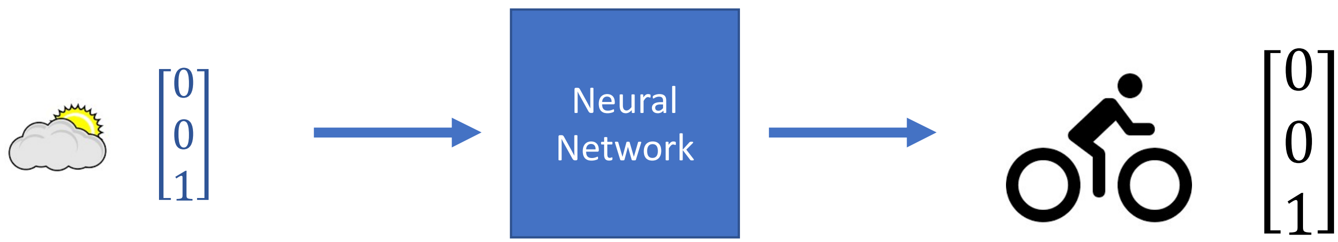
Basics of Neural Network

- If it is raining, John will go the gym
- If it is sunny, John will go running
- If it is overcast, John will go cycling



Basics of Neural Network

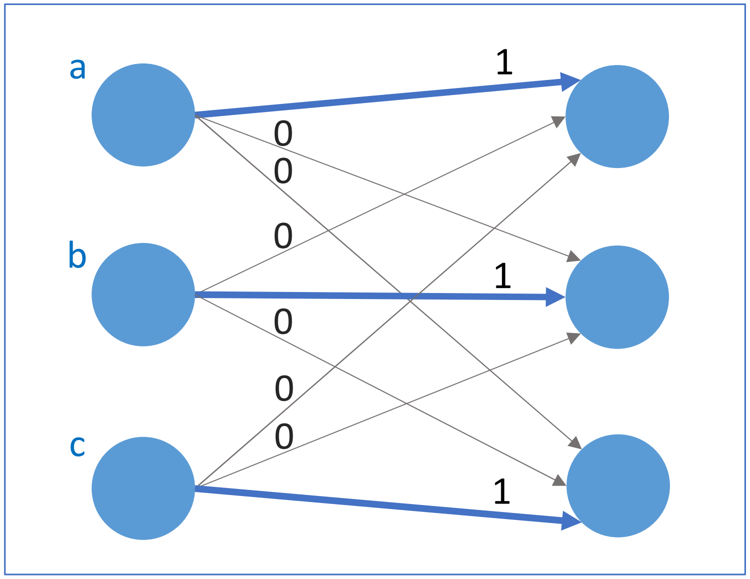
- If it is raining, John will go the gym
- If it is sunny, John will go running
- If it is overcast, John will go cycling



Basics of Neural Network

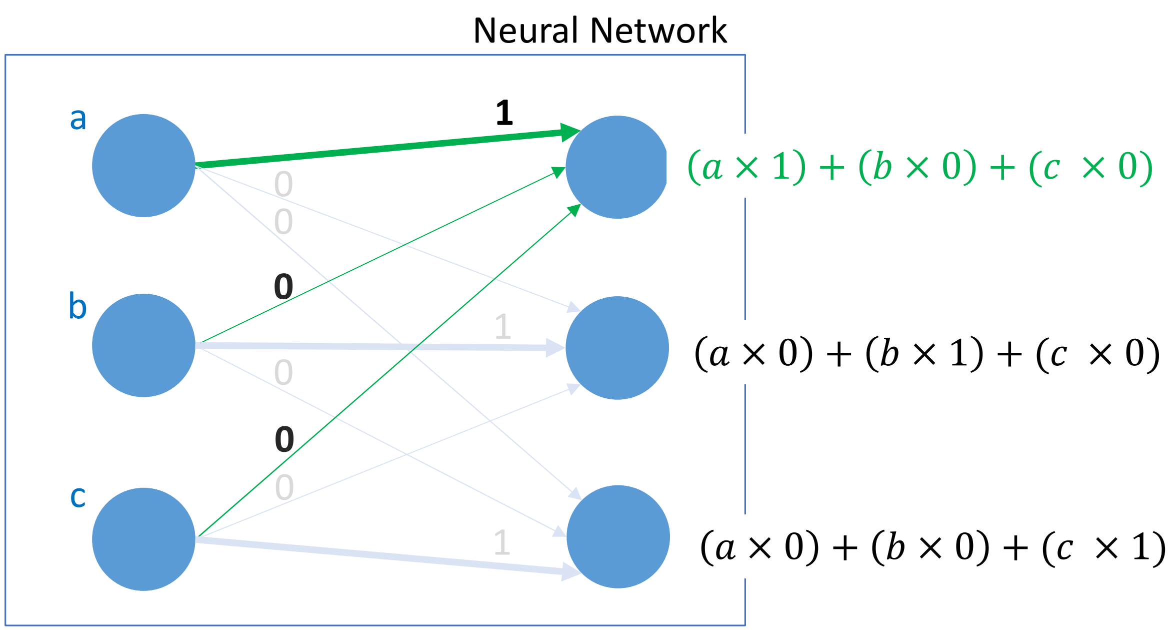
- If it is raining, John will go the gym
- If it is sunny, John will go running
- If it is overcast, John will go cycling

Neural Network



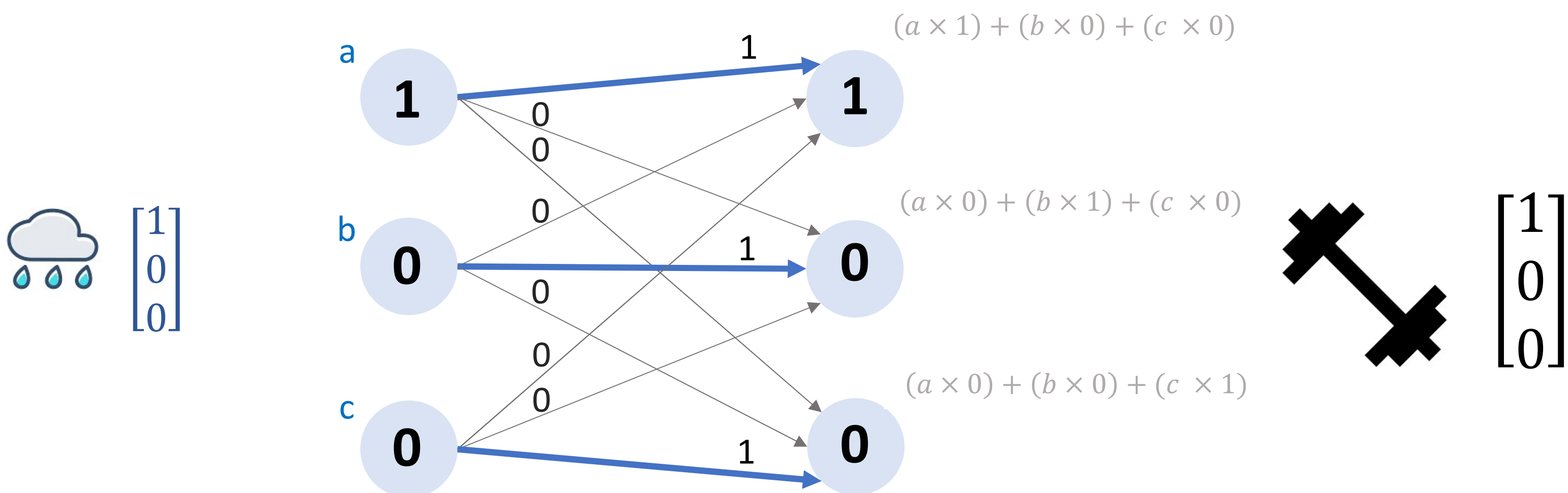
Basics of Neural Network

- If it is raining, John will go the gym
- If it is sunny, John will go running
- If it is overcast, John will go cycling



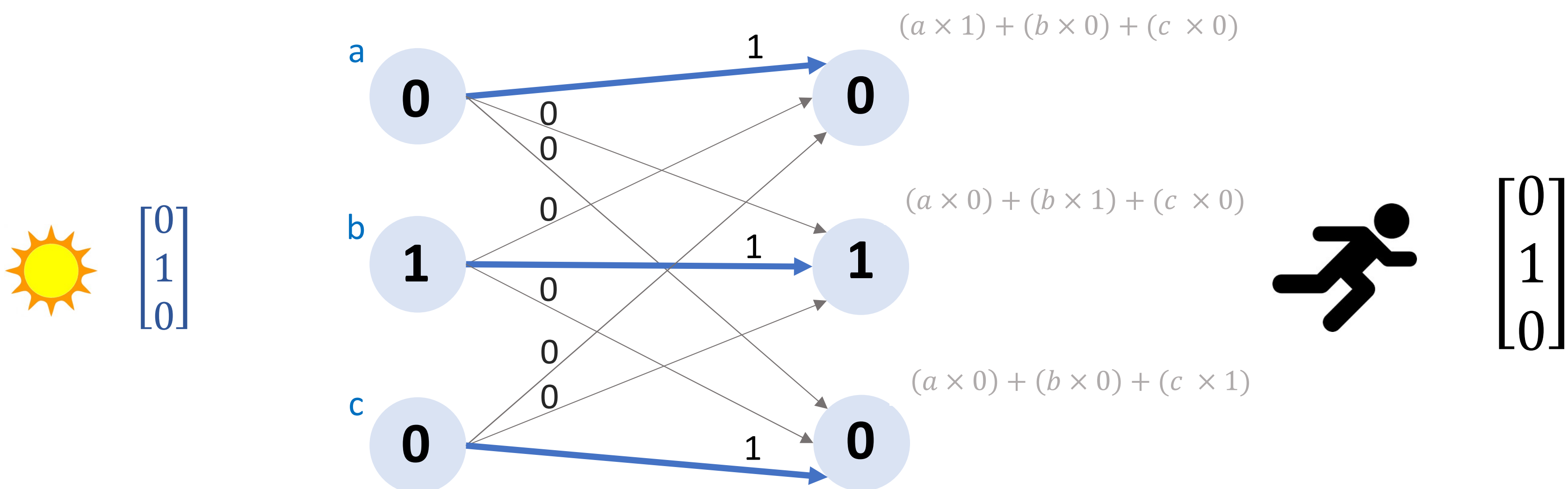
Basics of Neural Network

- If it is raining, John will go the gym
- If it is sunny, John will go running
- If it is overcast, John will go cycling



Basics of Neural Network

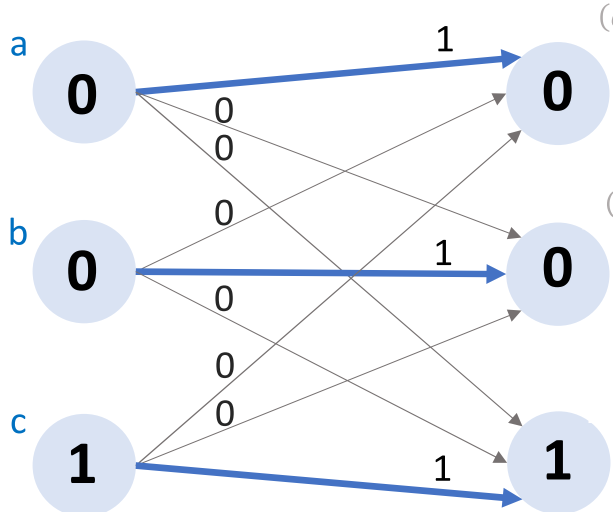
- If it is raining, John will go the gym
- If it is sunny, John will go running
- If it is overcast, John will go cycling



Basics of Neural Network

- If it is raining, John will go the gym
- If it is sunny, John will go running
- If it is overcast, John will go cycling

$$\begin{bmatrix}0\\0\\1\end{bmatrix}$$



$(a \times 1) + (b \times 0) + (c \ \times 0)$

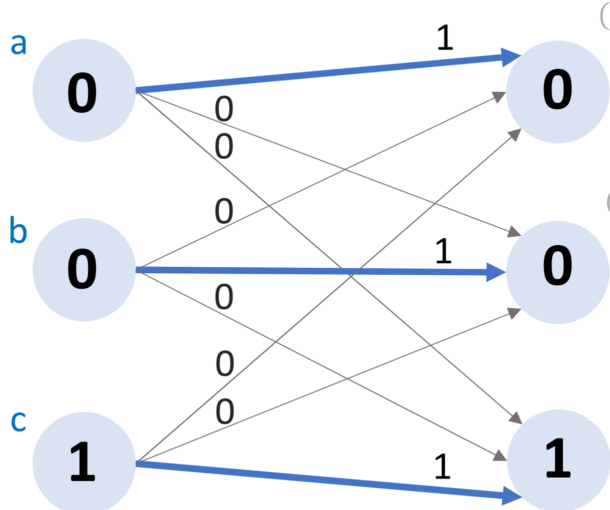
$(a \times 0) + (b \times 1) + (c \ \times 0)$

$(a \times 0) + (b \times 0) + (c \ \times 1)$

$$\begin{bmatrix}0\\0\\1\end{bmatrix}$$

Basics of Neural Network

- If it is raining, John will go the gym
- If it is sunny, John will go running
- If it is overcast, John will go cycling



$$(a \times 1) + (b \times 0) + (c \times 0)$$

$$(a \times 0) + (b \times 1) + (c \times 0)$$

$$(a \times 0) + (b \times 0) + (c \times 1)$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Basics of Neural Network

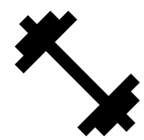
- If it is raining, John will go the gym
- If it is sunny, John will go running
- If it is overcast, John will go cycling

$$\begin{bmatrix}1\\0\\0\end{bmatrix}$$

$$\begin{bmatrix}0\\1\\0\end{bmatrix}$$

$$\begin{bmatrix}0\\0\\1\end{bmatrix}$$

$$\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\begin{bmatrix}0\\0\\1\end{bmatrix}=\begin{bmatrix}0\\0\\1\end{bmatrix}$$



$$\begin{bmatrix}1\\0\\0\end{bmatrix}$$

$$\begin{bmatrix}0\\1\\0\end{bmatrix}$$

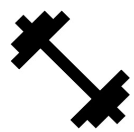
$$\begin{bmatrix}0\\0\\1\end{bmatrix}$$

Exercise

If John switched his routine and decided to

- If it is raining, John will go the gym
- If it is sunny, John will go **cycling**
- If it is overcast, John will go **running**

What is the new matrix representing the new network?

 $\begin{bmatrix}1\\0\\0\end{bmatrix}$  $\begin{bmatrix}1\\0\\0\end{bmatrix}$

 $\begin{bmatrix}0\\1\\0\end{bmatrix}$  $\begin{bmatrix}0\\0\\1\end{bmatrix}$

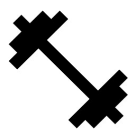
 $\begin{bmatrix}0\\0\\1\end{bmatrix}$  $\begin{bmatrix}0\\1\\0\end{bmatrix}$

$$\begin{bmatrix}? & ? & ?\\ ? & ? & ?\\ ? & ? & ?\end{bmatrix}$$

Additional Examples

- If it is raining, John will **go to the gym**
- If it is sunny, John will go running
- If it is overcast, John will **go to the gym**

 $\begin{bmatrix}1\\0\\0\end{bmatrix}$

 $\begin{bmatrix}1\\0\\0\end{bmatrix}$

 $\begin{bmatrix}0\\1\\0\end{bmatrix}$

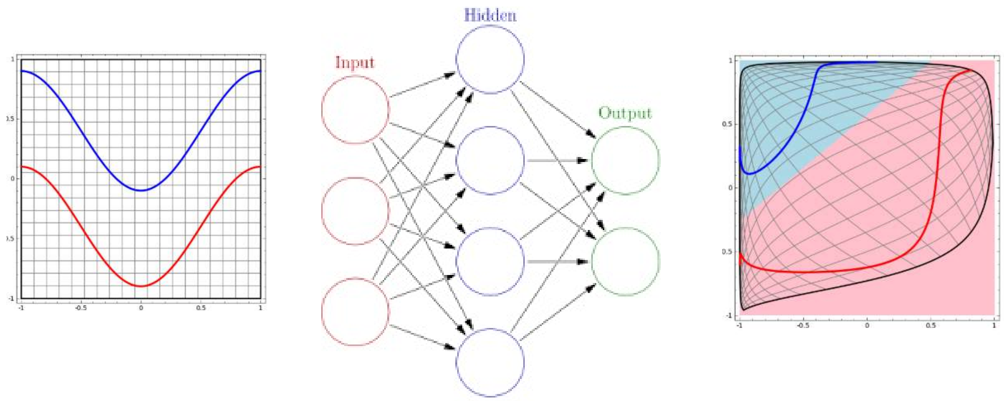
 $\begin{bmatrix}0\\0\\1\end{bmatrix}$

 $\begin{bmatrix}0\\0\\1\end{bmatrix}$

$$\begin{bmatrix}? & ? & ?\\ ? & ? & ?\\ ? & ? & ?\end{bmatrix}$$

Neural network as a classification algorithm

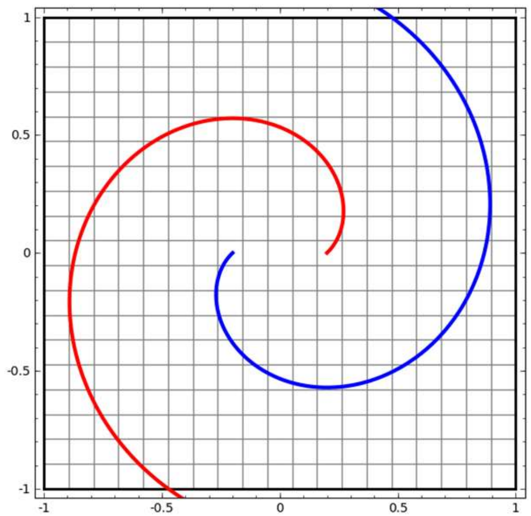
- In this sample dataset, classification algorithms will generally attempt to separate the 2 groups of data
- Neural network, have input, output and hidden layers
  - Layers worked to change representation of the data space
  - Each layer creates a new representation
  - Eventually leading to a final representation where a straight line can be drawn to separate the 2 groups of data



https://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

Neural network as a classification algorithm

- The following shows how the data space is being changed to allow a straight line to be drawn to separate two groups of data



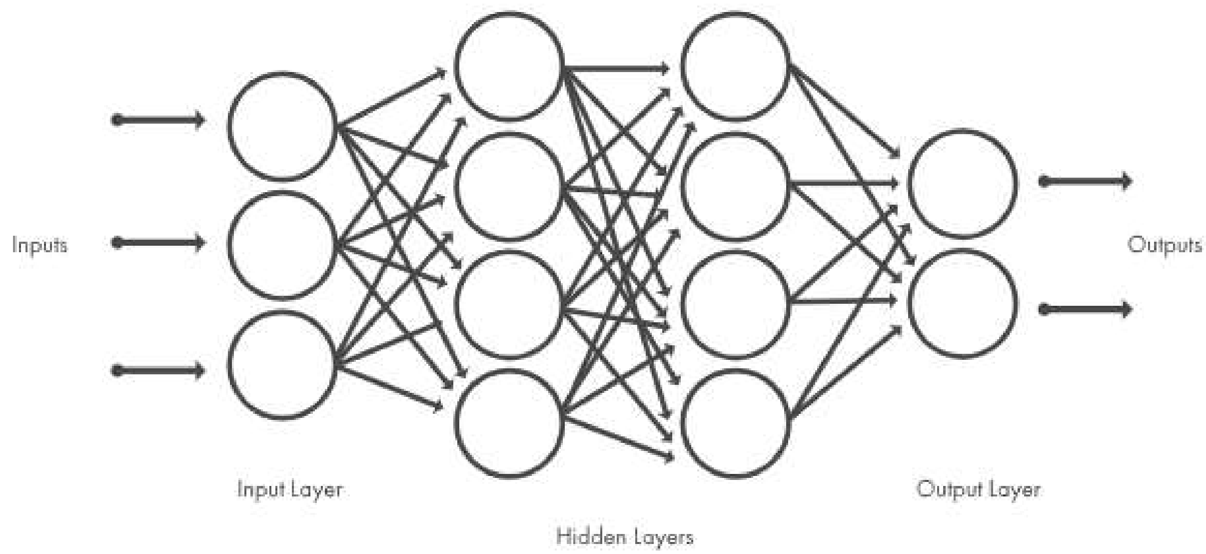
https://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

Deep learning approach (hierarchical learning)

- Usage: computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design and board game programs
- Inspired by information processing and communication patterns in biological nervous systems
- Use a cascade of multiple layers of nonlinear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as input
- Learn multiple levels of representations that correspond to different levels of abstraction; the levels form a hierarchy of concepts

How it works

- “deep” usually refers to the number of hidden layers in the neural network
  - Traditional neural networks only contain 2-3 hidden layers, while deep networks can have as many as 150



Deep learning approach

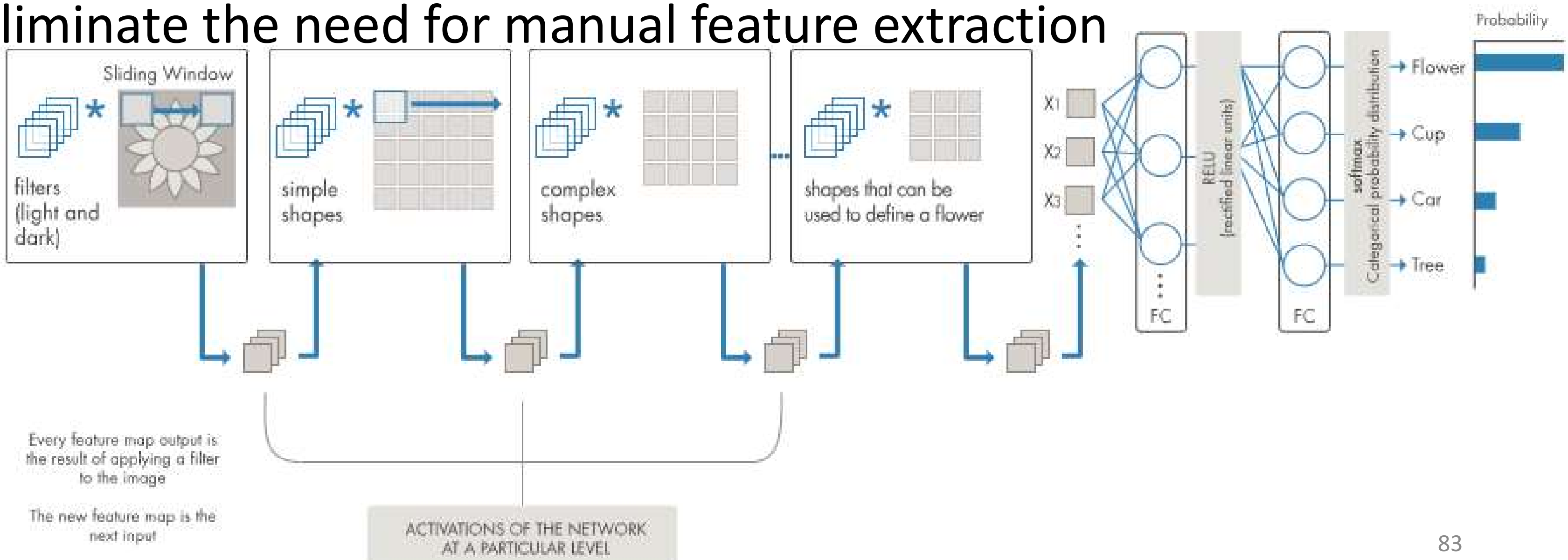
- Based on an Artificial Neural Network (ANN)
- Each level learns to transform its input data into a slightly more abstract and composite representation
  - E.g. image recognition application, raw input a matrix of pixels
  - first representational layer abstract the pixels and encode edges
  - second layer may compose and encode arrangements of edges
  - third layer may encode a nose and eyes
  - fourth layer may recognize that the image contains a face
- Requires large amount of labelled data
- Requires substantial computing power

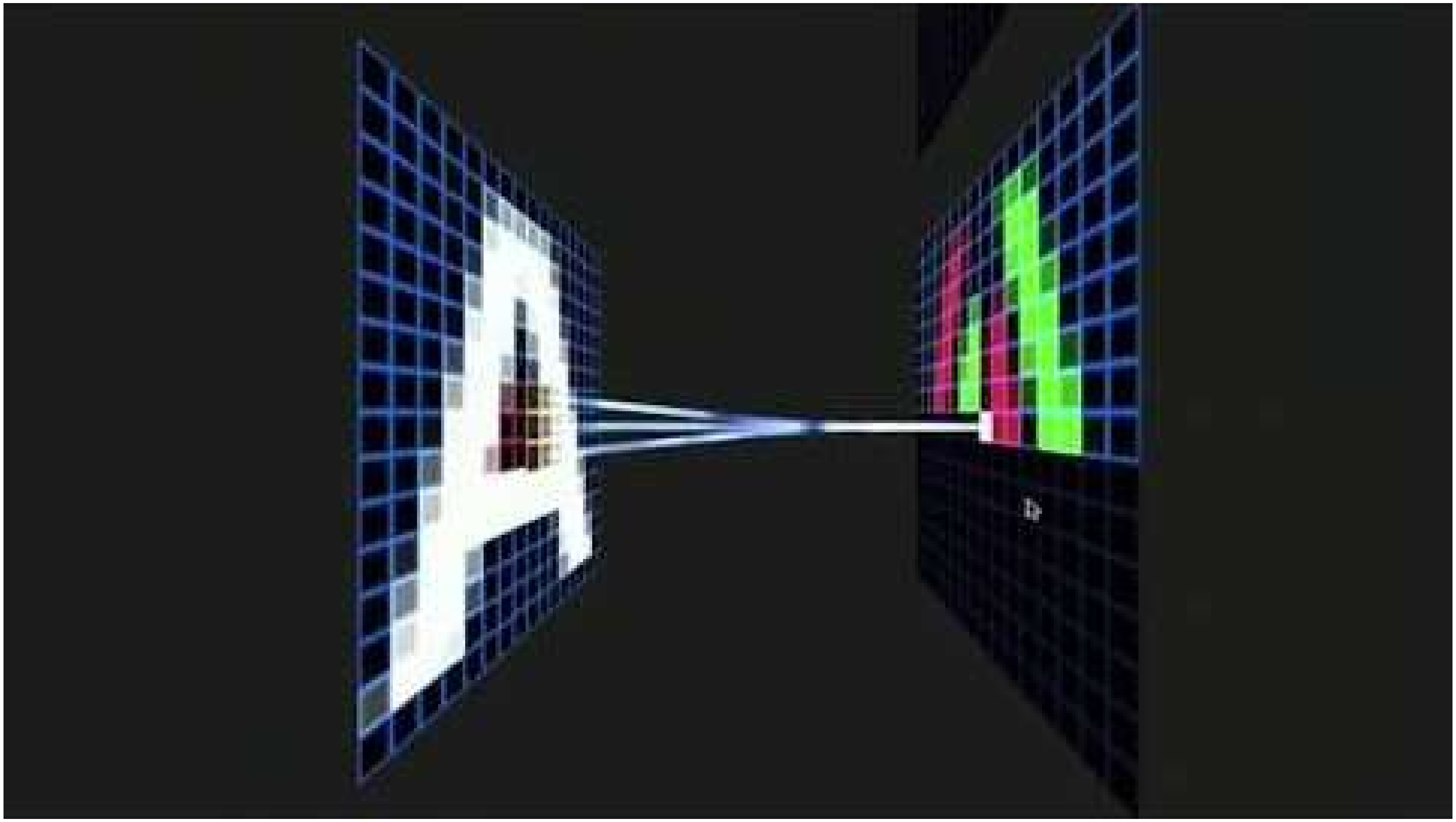
Convolutional Neural Networks (CNN or ConvNet)

- Convolutional layers
  - Transformation is a convolution
  - Early layer detect the simple geometric shapes
  - Deeper layers detect more sophisticated feature/object
- Image: Width x Height x Depth (3 channels (RGB))

Convolutional Neural Networks (CNN or ConvNet)

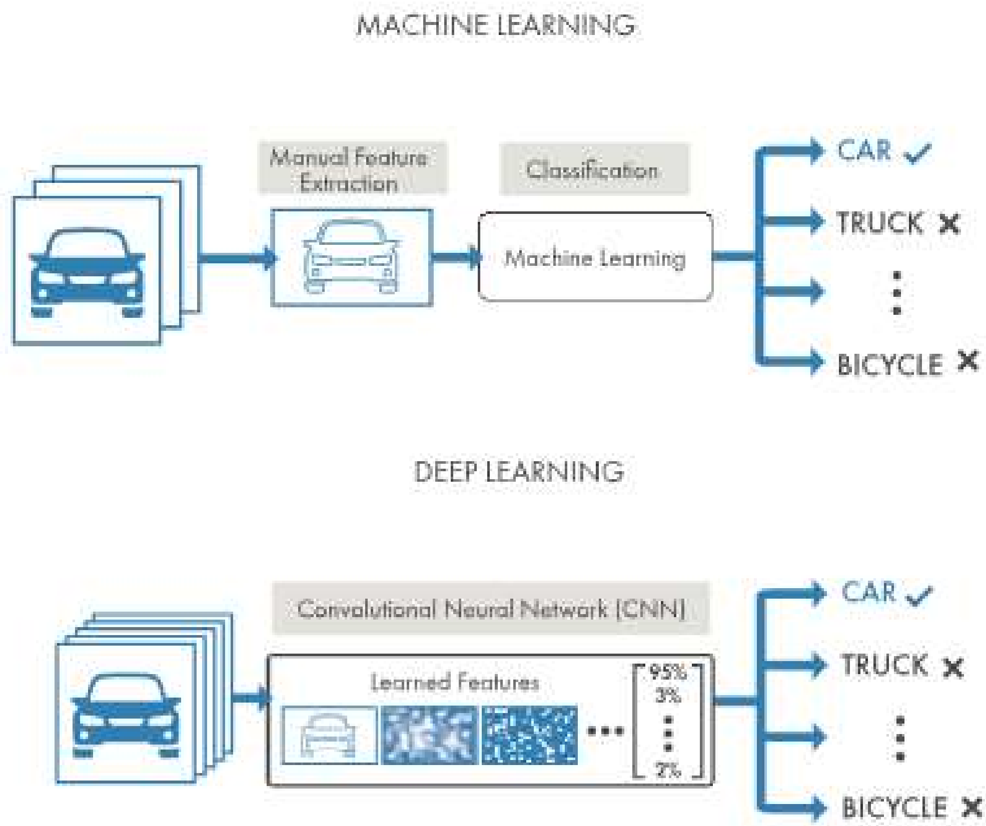
- Learned features with input data, and uses 2D convolutional layers, making this architecture well suited to processing 2D data, such as images
- [CNNs](#) eliminate the need for manual feature extraction





https://youtu.be/f0t-OCG79-U

Difference: Machine Learning & Deep Learning



Summary

- Symbolic (mid 1950s)
  - What is
  - Limitations/problems
- Sub-symbolic (1980s)
  - Embodied intelligence
  - Computational intelligence and soft computing
- Statistical (1990s)
  - Hidden Markov model (HMM), Bayesian
- ML, DL revisited

Self-Exploration

- A DARPA Perspective on Artificial Intelligence (16 mins)
  - https://www.youtube.com/watch?time_continue=6&v=-O01G3tSYpU&feature=emb_logo
- A friendly introduction to Bayes Theorem and Hidden Markov Models (32 mins)
  - https://www.youtube.com/watch?v=kqSzLo9fenk
  - 4.40: Hidden Markov Model
  - 12.08: Bayes Theorem
  - 22:48 Viterbi Algorithm
  - 30:20 Applications
- Reinforcement Learning – Deep Learning simplified (6 mins)
  - https://www.youtube.com/watch?v=e3Jy2vShroE
- Learning to drive in a day (through reinforcement learning) (2 mins)
  - https://www.youtube.com/watch?time_continue=129&v=eRwTbRtnT1I&feature=emb_logo
- AI learns to park – Deep reinforcement learning (11 mins)
  - https://www.youtube.com/watch?v=VMp6pq6_QjI