WARWICK
THE UNIVERSITY OF WARWICK

MA930 Data Analysis & Machine Learning

Lecture 7: Machine Learning for Data Analysis I

Haoran Ni

Introduction to Machine Learning

- ML: Category of algorithms that allow software applications to become more accurate at predicting outcomes without being explicitly programmed.

- Receive input data, use statistical analyses to predict an output while updating outputs as new data become available.

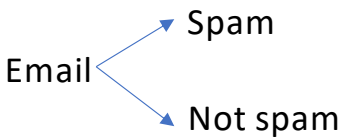
- Three types of ML: Supervised learning, Unsupervised learning, Reinforcement learning

Introduction to Machine Learning

- **Supervised learning:** Present algorithm with data that are labelled.

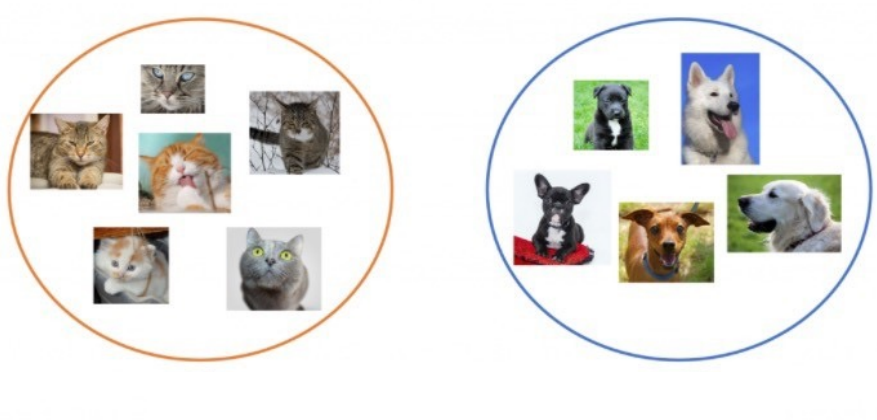
- Goal: Approximate mapping function so that, given new input data, the output variables can be predicted better

- Examples: Classification (output is categorical), Regression (output is real value – e.g. GBP)



- **Unsupervised learning:** Present algorithm with data that are not labelled.
- Goal: Separate data into different types

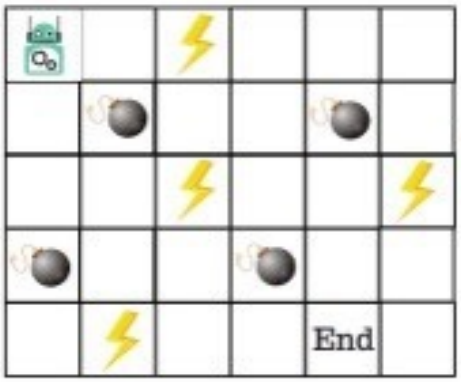
- Example: Clustering (find groupings in data)



Introduction to Machine Learning

- **Reinforcement learning:** Algorithm learns through interactions with the environment (e.g. being told by a human that choice is correct or wrong)

- Involves balance between exploration (of unchartered territory) and exploitation (of current knowledge)



1. The robot loses 1 point at each step. This is done so that the robot takes the shortest path and reaches the goal as fast as possible.
2. If the robot steps on a mine, the point loss is 100 and the game ends.
3. If the robot gets power ⚡, it gains 1 point.
4. If the robot reaches the end goal, the robot gets 100 points.


Cat


Dog

- **Reinforcement learning:** Algorithm learns through interactions with the environment (e.g. being told by a human that choice is correct or wrong)

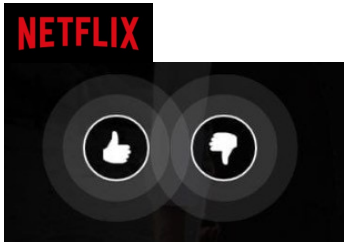
- Early autonomous cars are a good example

- Would make predictions and compare against a human who is actually driving, to "learn" how to drive

Introduction to Machine Learning

- **Reinforcement learning:** Algorithm learns through interactions with the environment (e.g. being told by a human that choice is correct or wrong)

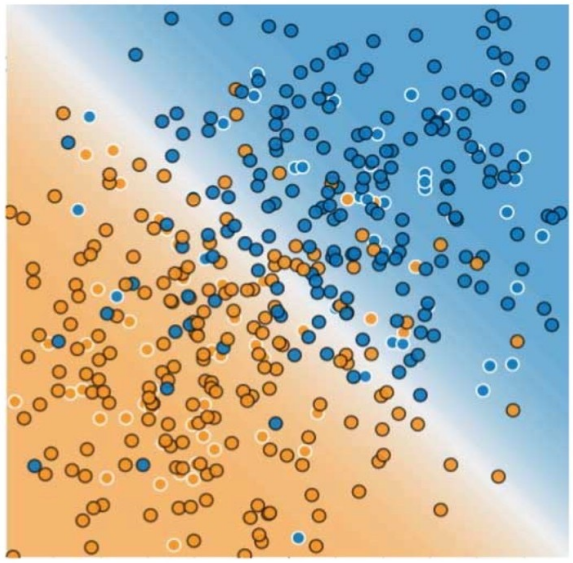
- Often used to refine previously trained ML models. E.g. to train a model for classifying different animal species on a new species



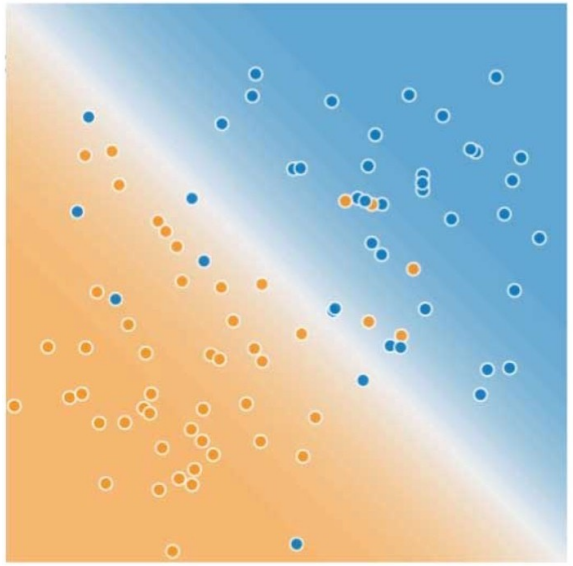
https://openai.com/blog/emergent-tool-use/

Validating ML models: As in statistics, ML models are often tested by splitting the original data into two parts.

Validating ML models: As in statistics, ML models are often tested by splitting the original data into two parts.

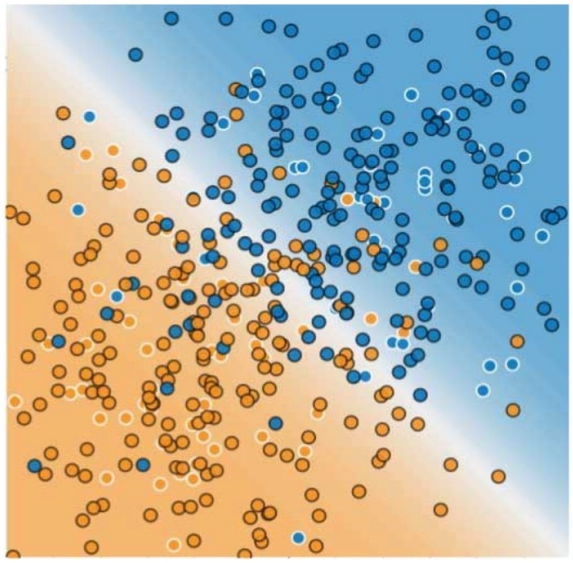


Training Data

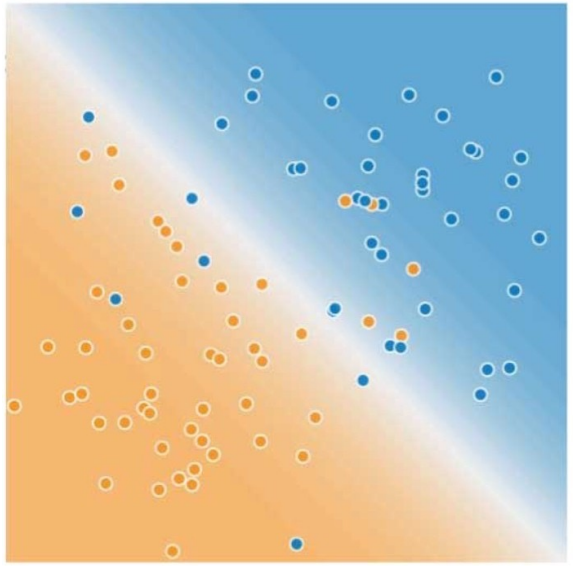


Test Data

Validating ML models: As in statistics, ML models are often tested by splitting the original data into two parts.



Training Data



Test Data

Measures precision (here: 95% precision) on test data

Gradient descent

WARWICK
THE UNIVERSITY OF WARWICK

- Optimisation algorithm (often used when training machine learning algorithms)

- Goal: Find (local) minimum of cost function $C(w_1, w_2)$
- Move in (w_1, w_2) space so that $C(w_1^{\text{new}}, w_2^{\text{new}}) < C\left(w_1^{\text{old}}, w_2^{\text{old}}\right)$

Gradient descent

- Optimisation algorithm (often used when training machine learning algorithms)

- Goal: Find (local) minimum of cost function $C(w_1, w_2)$
- Move in (w_1, w_2) space so that $C(w_1^{\text{new}}, w_2^{\text{new}}) < C\left(w_1^{\text{old}}, w_2^{\text{old}}\right)$

- $C(w_1 + \Delta_1, w_2 + \Delta_2) \approx C(w_1, w_2) + \Delta_1 \frac{\partial C}{\partial w_1} + \Delta_2 \frac{\partial C}{\partial w_2}$

$$= C(w_1, w_2) + \boldsymbol{\Delta} \frac{\partial C}{\partial \boldsymbol{w}}$$

Where $\boldsymbol{\Delta} = (\Delta_1, \Delta_2), \frac{\partial C}{\partial \boldsymbol{w}} = \left(\frac{\partial C}{\partial w_1}, \frac{\partial C}{\partial w_2}\right)$.

Gradient descent

Minimising the previous expression requires $\boldsymbol{\Delta}\,\frac{\partial C}{\partial \boldsymbol{w}}$ to be minimised, i.e. $\boldsymbol{\Delta}$ and $\boldsymbol{w}$ to be in opposite directions. Choose $\boldsymbol{\Delta} = -\alpha\,\frac{\partial C}{\partial \boldsymbol{w}}$

I.e. gradient descent involves doing:

$$w_k^{\text{new}} = w_k^{\text{old}} - \alpha\,\frac{\partial C}{\partial w_k}.$$

Gradient descent

Exercise 7.1 Gradient descent

$$C(w_1, w_2) = 1 - \frac{1}{1 + 2(w_1 - 3)^2 + (w_2 - 2)^2}$$

Find the minimum of this function by gradient descent, starting from $(w_1, w_2) = (2, 0.5)$ with $\nu = 50$ updates and update rate $\alpha = 0.3$.

i) Plot the "phase plane" (i.e. the points $\left(w_1^{(k)}, w_2^{(k)}\right)$ for $k = 1,2, \ldots 50$

ii) Plot the value of C at each of the steps

Gradient descent and linear regression

Recall: Linear regression involves N data points (t_n, x_n). Involves finding line of best fit $f_n = k_1 t_n + k_0$. Minimise error $\mathrm{E} = \frac{1}{2N}\sum_{n=1}^{N}(f_n - x_n)^2$.

Choose $k_0 = \frac{\langle x\rangle\langle t^2\rangle - \langle t\rangle\langle xt\rangle}{\langle t^2\rangle - \langle t\rangle^2}$, $k_1 = \frac{\langle xt\rangle - \langle x\rangle\langle t\rangle}{\langle t^2\rangle - \langle t\rangle^2}$.

Gradient descent and linear regression

WARWICK
THE UNIVERSITY OF WARWICK

Recall: Linear regression involves N data points (t_n, x_n). Involves finding line of best fit $f_n = k_1 t_n + k_0$. Minimise error $\mathrm{E} = \frac{1}{2N}\sum_{n=1}^{N}(f_n - x_n)^2$.

Choose $k_0 = \frac{\langle x\rangle\langle t^2\rangle - \langle t\rangle\langle xt\rangle}{\langle t^2\rangle - \langle t\rangle^2}$, $k_1 = \frac{\langle xt\rangle - \langle x\rangle\langle t\rangle}{\langle t^2\rangle - \langle t\rangle^2}$.

Alternatively: Let $w_1 = k_1$, $w_2 = k_0$ and minimise

$$C(w_1, w_2) = \frac{1}{2N}\sum_{n=1}^{N}(w_1 t_n + w_2 - x_n)^2$$

using gradient descent.

Gradient descent and linear regression

WARWICK
THE UNIVERSITY OF WARWICK

Exercise 7.2 Gradient descent and linear regression

For the curve $x = 2t + 1$, generate $N = 10$ data points between $t = 0 \ldots 1$, with Gaussian noise added with zero mean and standard deviation $\sigma = 0.2$.

Use gradient descent to find the line of best fit, starting from initial guess $w_1 = k_1 = 0.1$ and $w_2 = k_0 = 0.2$, with $\nu = 100$ updates and update rate $\alpha = 0.1$.

Plot: i) the $\left(w_1^{(k)}, w_2^{(k)}\right)$ phase plane;

ii) The value of the cost function at each step;

iii) The true line, the fitted line, and the data points.

Gradient descent and linear regression

WARWICK
THE UNIVERSITY OF WARWICK

(Yet) another approach: In the previous example, the cost function was

$$C(w_1, w_2) = \frac{1}{2N}\sum_{n=1}^{N}(w_1 t_n + w_2 - x_n)^2$$

Recall the likelihood function, L: the probability (density) of seeing the data given the model parameters. Assuming the distance between the predicted line and the data is also Gaussian (with mean zero, s.d. σ), then can pick (w_1, w_2) to maximise the log-likelihood.

$$L(w_1, w_2) = \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\left(\frac{w_1 t_n + w_2 - x_n}{\sigma}\right)^2\right)$$

Gradient descent and linear regression

WARWICK
THE UNIVERSITY OF WARWICK

Exercise 7.3 Gradient descent and linear regression (ctd)
Find the values of w_1, w_2 that maximise $\log(L)$, and add the resulting line to the third plot you made in exercise 6.3.

Logistic regression

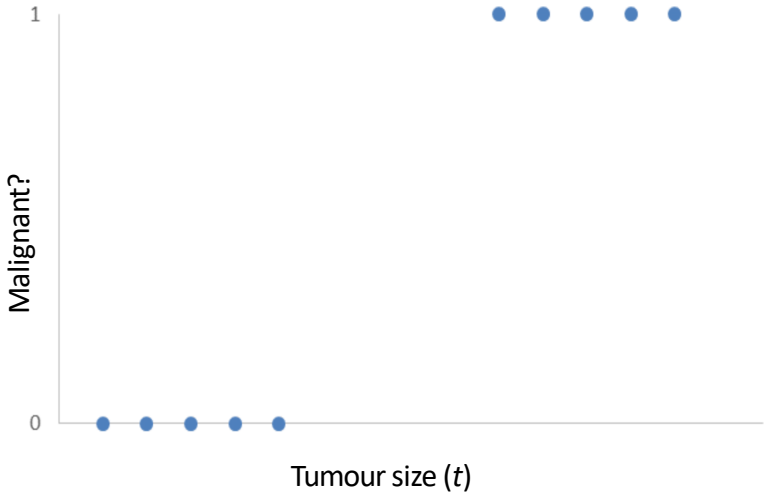
WARWICK
THE UNIVERSITY OF WARWICK

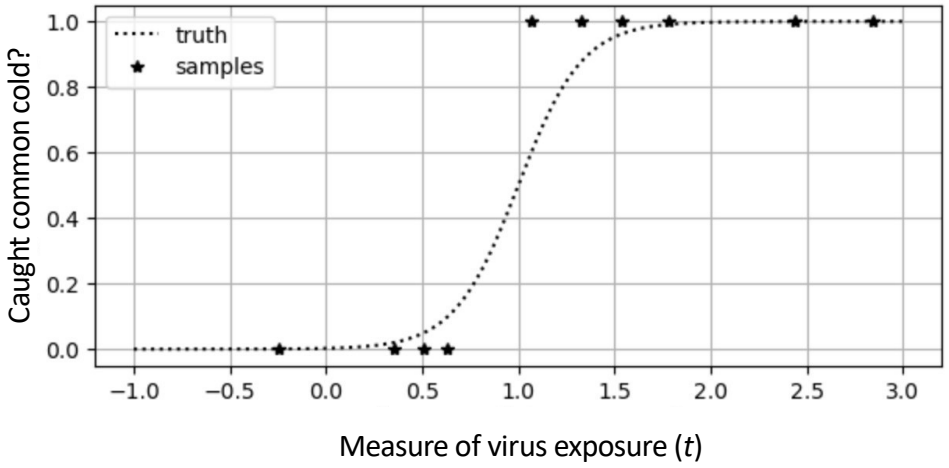
Recap (yesterday): Linear/polynomial fits. Target data: real numbers.

In ML, often have categorical data. Want to predict whether whether data point is a member of category ($x = 1$) or not ($x = 0$). I.e. a binary classification problem

Prediction p can take a value in [0,1], and can be interpreted as the probability that $x = 1$ is seen.

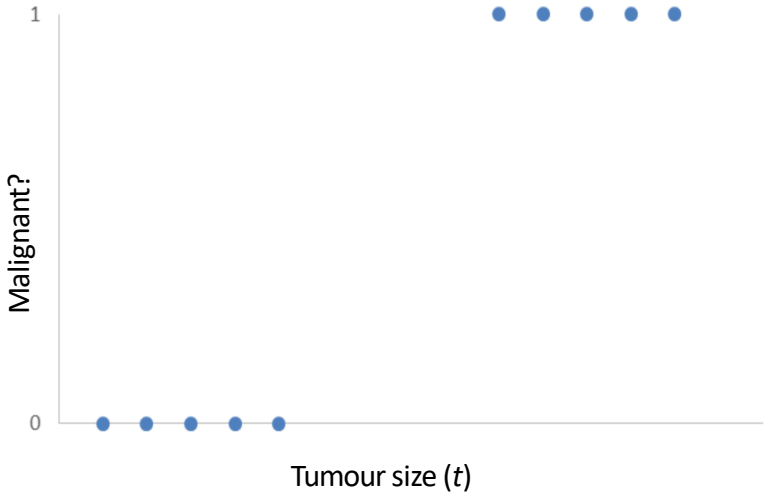
Logistic regression

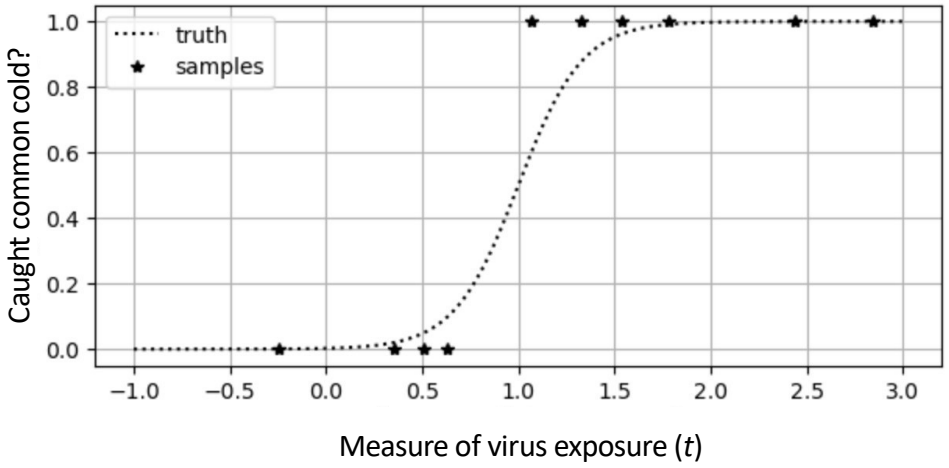




- Clear what a person might do in these examples.
- But how would a machine come up with a prediction p? (particularly if there is a slight overlap in the data)

Logistic regression





Answer: logistic regression

Fit a logistic (sigmoid) function, $p(t) = \frac{1}{1+\exp(-(w_1 t + w_2))}$

Logistic regression

- Fit a logistic (sigmoid) function, $p(t) = \frac{1}{1+\exp(-(w_1 t + w_2))}$

- Data (t_n, x_n) is binary (i.e. x_n is either 0 or 1)

  Likelihood to see $x_n = 1$ is $p(t_n)$.

  Likelihood to see $x_n = 0$ is $1 - p(t_n)$.

$$L(w_1, w_2) = \prod_{n=1}^{N} p(t_n)^{x_n} (1 - p(t_n))^{1-x_n}$$

- w_1 and w_2 enter this function through the function p. Choose w_1, w_2 to maximise the log-likelihood.

Logistic regression

- Can use gradient descent to minimise –log(L).

- Cost function:

$$C(w_1, w_2) = \sum_{n=1}^{N} (1 - x_n)\,(w_1 t_n + w_2) + \log(1 + \exp(-(w_1 t_n + w_2)))$$

Check this!

Logistic regression

- Recall: the logistic prediction function was $p(t) = \frac{1}{1+\exp(-(w_1 t + w_2))}$
- To make a binary prediction of an observation t, the standard choice is to predict that $x = 0$ if $p < 0.5$ and $x = 1$ if $p > 0.5$.
- The value of t for which $p(t) = 0.5$ is the *classification boundary*.
- Question: Find the classification boundary in terms of w_1 and w_2. (see board)

Logistic regression

WARWICK
THE UNIVERSITY OF WARWICK

Exercise 7.4 Gradient descent and logistic regression

Generate $N = 100$ data points (t_n, x_n) between $t = [-1, 3]$, sampling a binary value for x each time using a Bernoulli RV with probability $\tilde{p}(t_n) = \frac{1}{1+\exp(-6t+6)}$.

Use gradient descent to estimate the prediction function $p(t)$, starting from initial guess $w_1 = 1$ and $w_2 = -2$, with $\nu = 1000$ updates and update rate $\alpha = 0.01$.

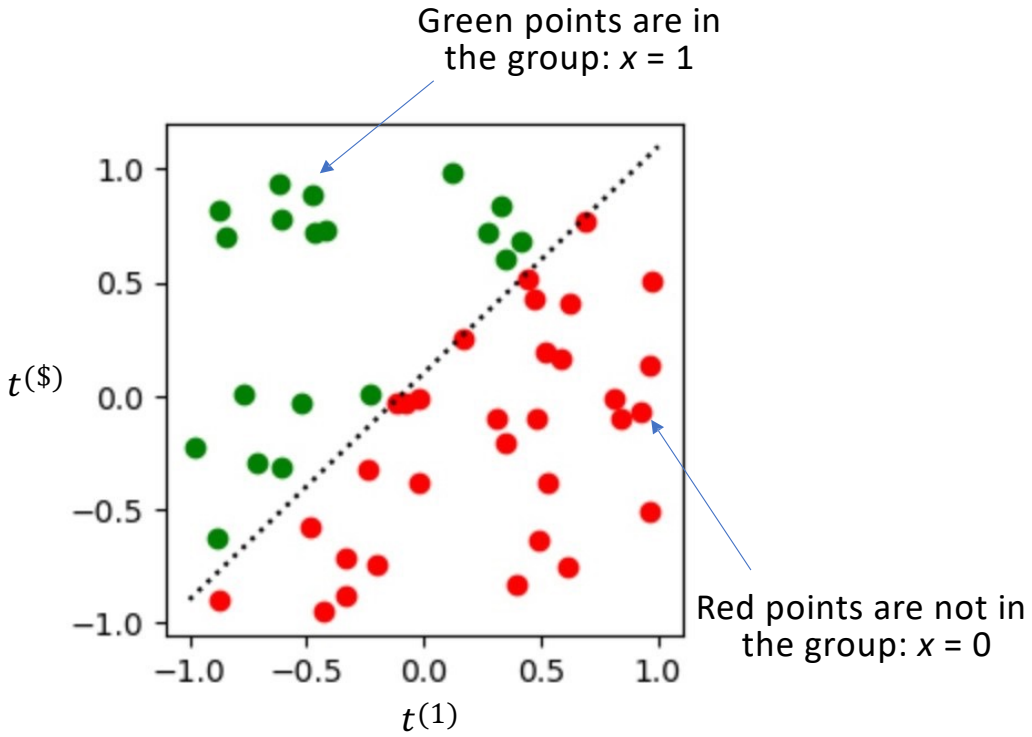
For each iteration, plot

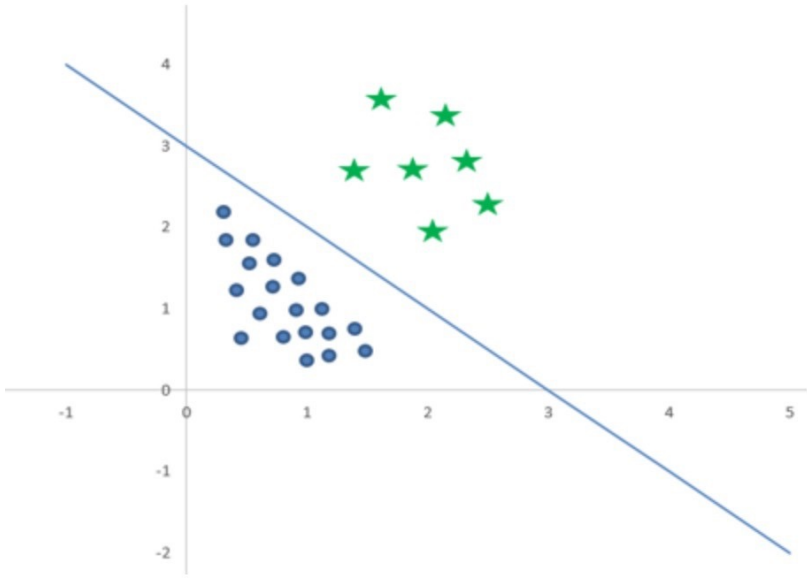
i) The classification boundary (i.e. the position of 50:50);
ii) The value of the cost function.

On a single graph, plot the true probability function used to generate the data $(\tilde{p}(t))$ the prediction $p(t)$ and the data

Extension to data with multiple variables

Example: classifier for linearly separable data





Extension to data with multiple variables

WARWICK
THE UNIVERSITY OF WARWICK

- Previously, we had input t and binary output x, so we fitted the logistic function $p(t) = \frac{1}{1+\exp(-(w_1 t+w_2))}$.

- Now, we have input $(t^{(1)}, t^{(2)})$ and binary output x, so we fit the logistic function $P\left(t^{(1)}, t^{(2)}\right) = \frac{1}{1+\exp(-(w_1 t^{(1)}+w_2 t^{(2)}+w_3))}$.

Extension to data with multiple variables

- Following the same logic as before, the likelihood is:

$$L(w_1, w_2, w_3) = \prod_{n=1}^{N} p(t_n^{(1)}, t_n^{(2)})^{x_n} (1 - p\left(t_n^{(1)}, t_n^{(2)}\right))^{1-x_n}$$

Extension to data with multiple variables

WARWICK
THE UNIVERSITY OF WARWICK

- Following the same logic as before, the likelihood is:

$$L(w_1, w_2, w_3) = \prod_{n=1}^{N} p(t_n^{(1)}, t_n^{(2)})^{x_n} (1 - p\left(t_n^{(1)}, t_n^{(2)}\right))^{1-x_n}$$

- Can use gradient descent to minimise –log(L).

- Cost function:
  $C(w_1, w_2, w_3)$ **Check this!**

$$= \sum_{n=1}^{N} (1 - x_n)\left(w_1 t_n^{(1)} + w_2 t_n^{(2)} + w_3\right) + \log(1 + \exp(-(w_1 t_n^{(1)} + w_2 t_n^{(2)} + w_3)))$$

Logistic regression

WARWICK
THE UNIVERSITY OF WARWICK

Exercise 7.5 Linear classifier on the plane

Generate $N = 50$ random data points $(t_n^{(1)}, t_n^{(2)}, x_n)$, where each t-variable lies between -1...1, and $x_n = 1$ if $t_n^{(2)} > t_n^{(1)} + 0.1$ (and $x_n = 0$ otherwise). Plot that dividing line on the same axes. Colour points with different colours based on whether $x_n = 0$ or $x_n = 1$.

Use gradient descent to estimate the prediction function $p(t_n^{(1)}, t_n^{(2)})$, starting from initial guess $w_1 = 0$, $w_2 = 1$ and $w_3 = 0$, with ν=1000 updates and update rate $\alpha = 2$.

By doing an exhaustive sweep over possible combinations of $t_n^{(1)}$ and $t_n^{(2)}$, create a new plot showing the region of $(t_n^{(1)}, t_n^{(2)})$ space that would be classified as "x=1" .

Summary

- Different types of ML algorithm exist
- Model validation (Test vs training data)
- Gradient descent
- Classification algorithms based on labelled data (type of supervised learning) using logistic regression
  - Single input, binary output
  - Multiple inputs, binary output

WARWICK
THE UNIVERSITY OF WARWICK

MA930 Data Analysis & Machine Learning

Lecture 8: Machine Learning for Data Analysis II

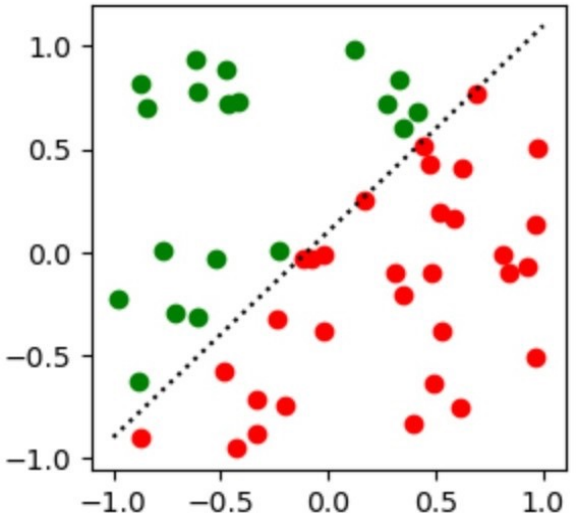
Haoran Ni

Recap

WARWICK
THE UNIVERSITY OF WARWICK

Last time, we explored:

- Gradient descent: to find (local) minima
- Logistic regression
- Classifying linearly separable data



Outline

- K Nearest Neighbour classification
- K-Means Clustering
- Neural Networks

Instance-based Learning

WARWICK
THE UNIVERSITY OF WARWICK

- Classification is not always so straightforward: Data may not be straightforwardly separable (e.g. linear)
- An alternative to parametric models is non-parametric models
- Simple non-parametric models can be used to approximate more complex target functions
- Learning amounts to simply storing training data
- Test instances classified using similar training instances

Instance-based Learning

WARWICK
THE UNIVERSITY OF WARWICK

Key assumptions:

- Output varies smoothly with input
- Data lie within (or near) data used to train model

Instance-based Learning: KNN classification

WARWICK
THE UNIVERSITY OF WARWICK

K-nearest neighbour classification:

Idea: Value of the target function is estimated from the known values of the nearest training examples

Instance-based Learning: KNN classification

WARWICK
THE UNIVERSITY OF WARWICK

K-nearest neighbour classification:

Data in Euclidean space: $\boldsymbol{t} \in \mathbb{R}^d$

Distance typically defined to be Euclidean:

$$\text{distance}(\boldsymbol{t}, \boldsymbol{s}) = \|\boldsymbol{t} - \boldsymbol{s}\| = \sqrt{\sum_{i=1}^{d} (t_i - s_i)^2}$$

Instance-based Learning: KNN classification

WARWICK
THE UNIVERSITY OF WARWICK

Simplest case is 1-nearest neighbour classification:

Data of the form $(\boldsymbol{t}, x)$ where x is the output variable

ALGORITHM

1. Find example $(\boldsymbol{t}^*, x^*)$ from the stored training set that is closest (in $\boldsymbol{t}$-variable) to the new data $(\boldsymbol{t}, x)$. I.e.

$$\boldsymbol{t}^* = \text{argmin}_{s \in \text{training set}}[\text{distance}(\boldsymbol{t}, \boldsymbol{s})]$$

2. Classify $x = x^*$

Instance-based Learning: KNN classification

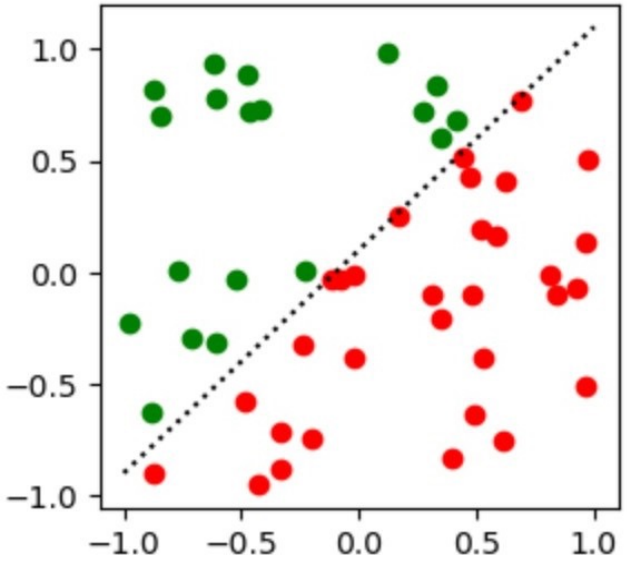
WARWICK
THE UNIVERSITY OF WARWICK

1-nearest neighbour classification:

Can, in principle, also compute classification boundaries

(like for linearly separable data that we saw yesterday)

Yesterday:

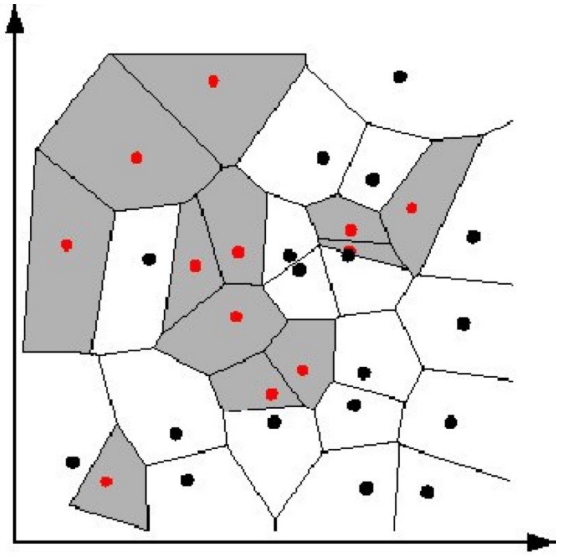


Instance-based Learning: KNN classification

1-nearest neighbour classification:

Can, in principle, also compute classification boundaries
(like for linearly separable data that we saw yesterday)

Today:



Voronoi diagram

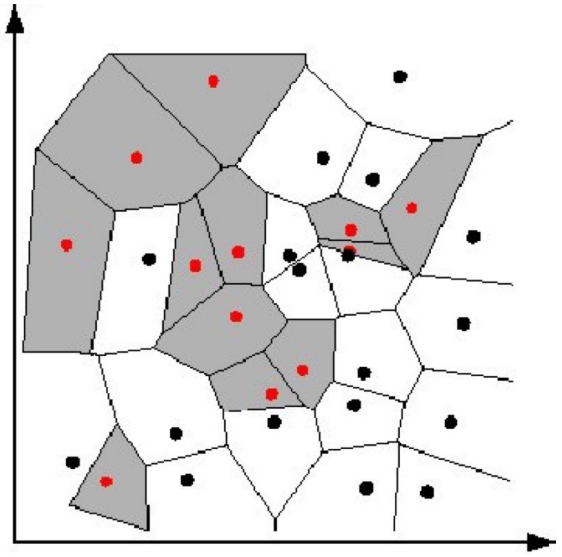
2-d decision boundaries

Instance-based Learning: KNN classification

1-nearest neighbour classification:

Can, in principle, also compute classification boundaries
(like for linearly separable data that we saw yesterday)

Today:



Voronoi diagram

On each line, equidistant from two nearest points

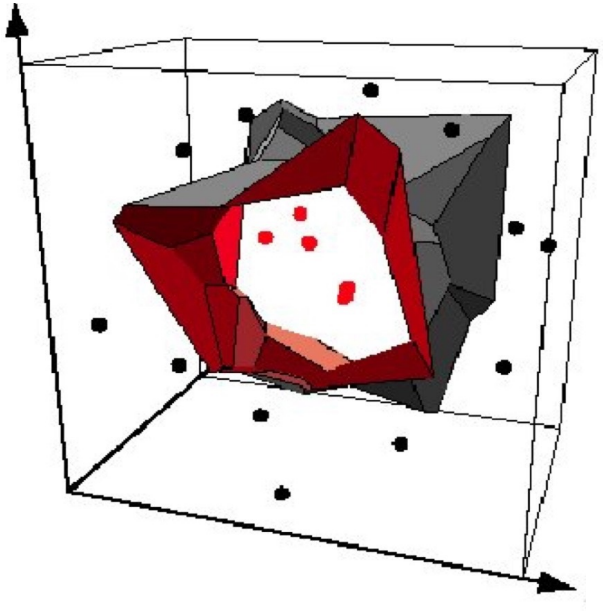
2-d decision boundaries

Instance-based Learning: KNN classification

1-nearest neighbour classification:

Can, in principle, also compute classification boundaries
(like for linearly separable data that we saw yesterday)

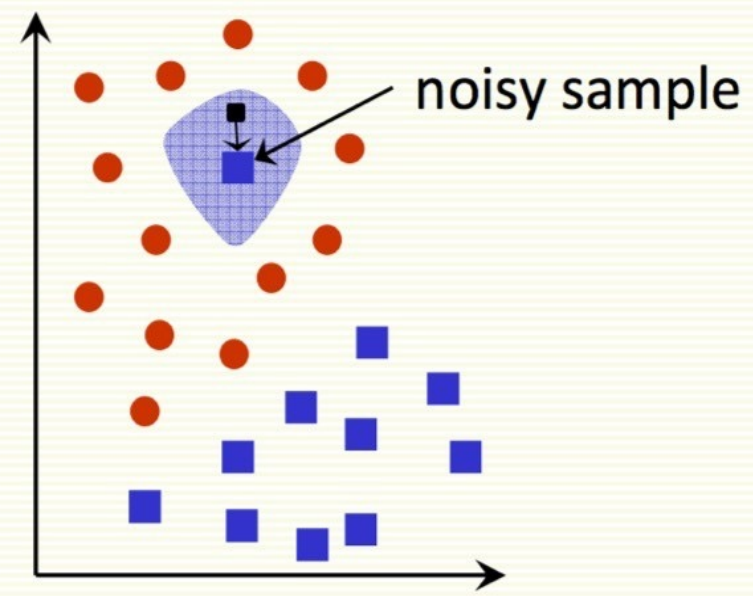
Today:



3-d decision boundaries

Instance-based Learning: KNN classification

Issue with 1-nearest neighbour: Highly sensitive to mislabelled training data



Every point in shaded region will be classified as blue

Evidence really points towards red...

Instance-based Learning: KNN classification

WARWICK
THE UNIVERSITY OF WARWICK

Solution: K-nearest neighbour

The K nearest training datapoints ”vote” for classification of the new data point $(\boldsymbol{t}, x)$

ALGORITHM

1. Find K-examples $\{(\boldsymbol{t}_i^*, x_i^*)\}$from the stored training set that are closest (in $\boldsymbol{t}$-variable) to the new data $(\boldsymbol{t}, x)$.
2. Classify $x = x^*$, where $x^* = \text{argmax}_{x_z}(\sum_{r=1}^{K} \delta(x_z, x_r^*))$

Instance-based Learning: KNN classification

WARWICK
THE UNIVERSITY OF WARWICK

How to choose K?

- If K is too small, then misclassified training data may lead to errors

- If K is too large, then data will be classified based on training data that are far away (e.g. extreme example where $K = N$, where N is the size of the training set)

- Rule of thumb in the literature: set $K \approx \sqrt{N}$

Instance-based Learning: KNN classification

WARWICK
THE UNIVERSITY OF WARWICK

How to choose *K*?

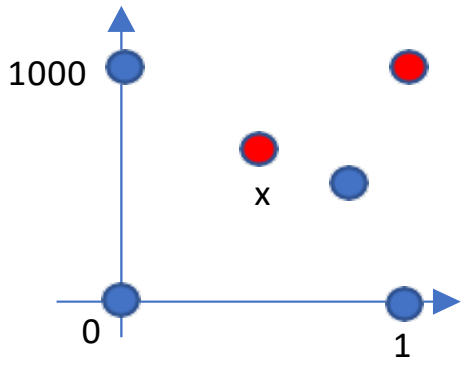
- One possible approach is to split the original data into a “training set” and a “validation set” (rather than simply using all the original data as the training set).
- The value of *K* can be chosen that gives the best predictive accuracy in the validation set.

Instance-based Learning: KNN classification

WARWICK
THE UNIVERSITY OF WARWICK

Potential issue:

- The range of some ***t*** variables may be different to others. This leads to giving some of the variables more importance...



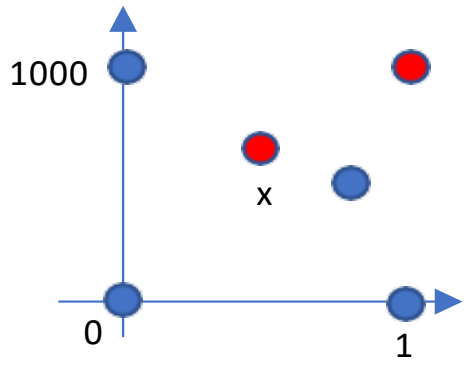
Instance-based Learning: KNN classification

Potential issue:

- The range of some ***t*** variables may be different to others. This leads to giving some of the variables more importance...

Possible solutions:

Option 1: Normalise the data so that all variables run 0...1– outliers sometimes then have a big effect



Instance-based Learning: KNN classification

WARWICK
THE UNIVERSITY OF WARWICK

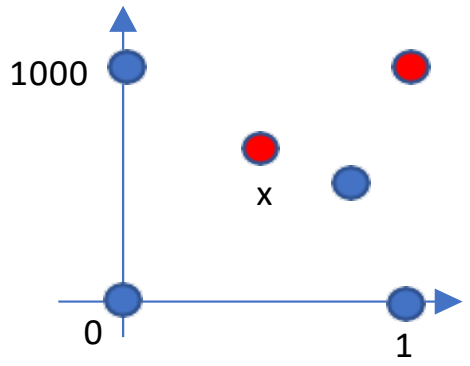
Potential issue:

- The range of some ***t*** variables may be different to others. This leads to giving some of the variables more importance...

Possible solutions:

Option 1: Normalise the data so that all variables run 0...1– outliers sometimes then have a big effect

Option 2: Other possible scaling – Scale the data so that each dimension has IQR running 0...1?



Instance-based Learning: KNN classification

WARWICK
THE UNIVERSITY OF WARWICK

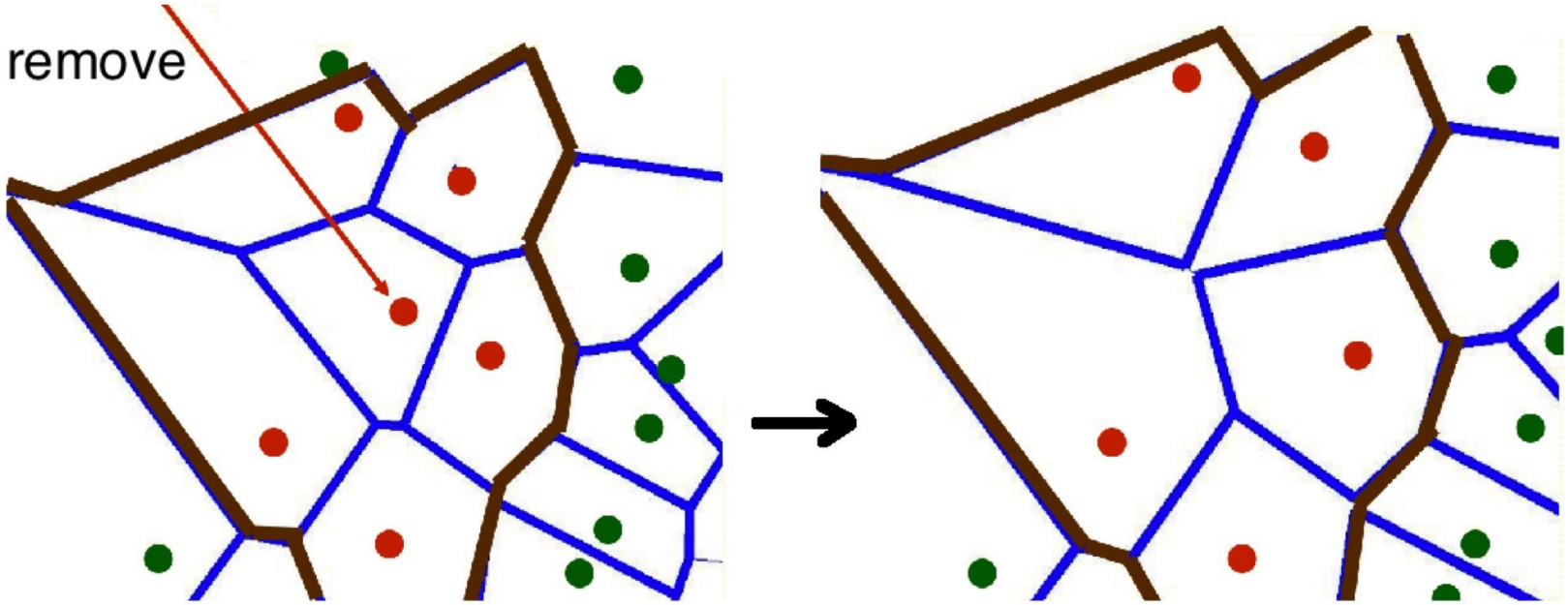
Potential issue:

- The algorithm can be slow, since it involves storing all training data and computing the distance to every training data point each time new data are considered

Improvement for 1NN:

- Can remove redundant training data points (if training data are final) – i.e. remove any data point that is only surrounded by data points of the same type.

Instance-based Learning: KNN classification



For KNN: Can go through points and remove them if doing so does not change the classification regions...

Instance-based Learning: KNN classification

WARWICK
THE UNIVERSITY OF WARWICK

KNN Classification - Summary:

- Naturally forms complex decision boundaries (e.g. no assumption of linear boundary)
- Works well if there are lots of training datapoints

- Potential issues:
  - Can be slow (scales linearly with number of training data points)
  - Can be sensitive to misclassification
  - Sensitive to scale of attributes (i.e. the different variables in $\boldsymbol{t}$)

Instance-based Learning: KNN classification

WARWICK
THE UNIVERSITY OF WARWICK

Exercise 8.1 KNN classification

Download the "iris.xls" dataset. This is a classic dataset collected by Fisher in 1936, and records the attributes of examples of three types of flower

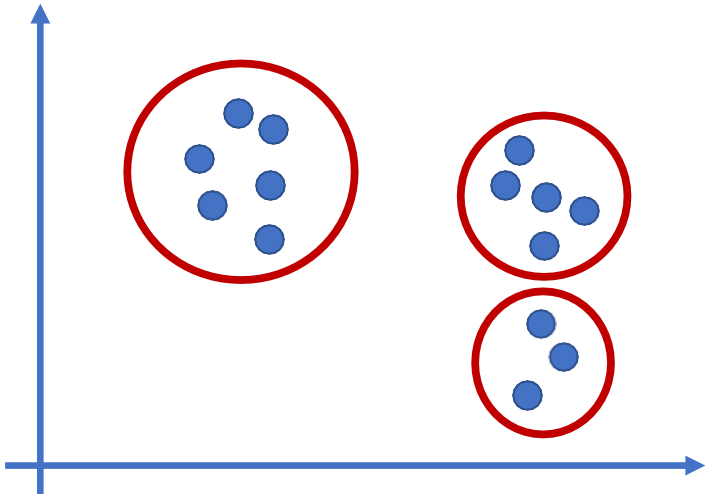
i. Write computing code to pick 80% of the data points at random to use as training data
ii. Classify the other 20% of the data points using KNN, and calculate the accuracy (the proportion of correct classifications)
iii. Repeat this for a range of values of K, and plot the accuracy as a function of K
iv. Repeat the analysis above, but first scaling the attributes to lie exactly in the range [0,1]. Does this appear to make a difference to the accuracy in this case? What happens if you instead use only one of the attributes? Which single attributes provide good predictions?

Clustering

WARWICK
THE UNIVERSITY OF WARWICK

- Clustering: Unsupervised learning
- Requires data but no labels
- Groups together similar data to detect patterns (e.g. customer shopping patterns, first step to group websites for search results, etc.)

E.g.

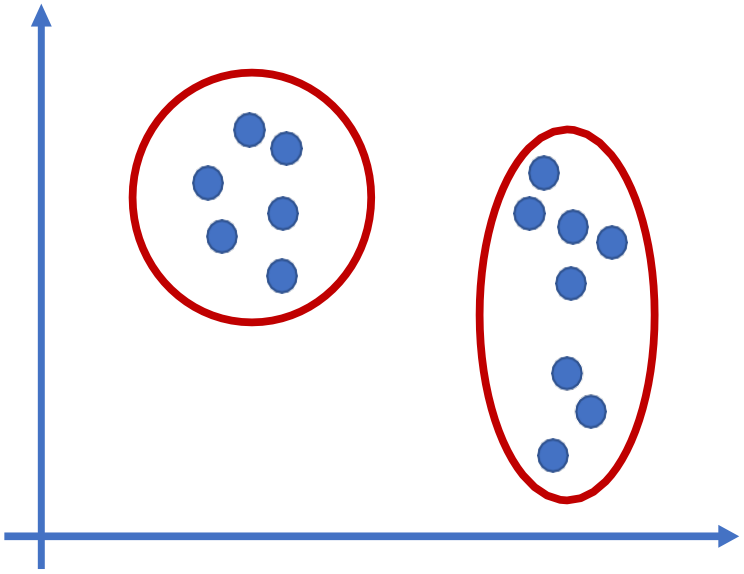


Clustering

WARWICK
THE UNIVERSITY OF WARWICK

- Clustering: Unsupervised learning
- Requires data but no labels
- Groups together similar data to detect patterns (e.g. customer shopping patterns, first step to group websites for search results, etc.)

E.g.



Clustering

WARWICK
THE UNIVERSITY OF WARWICK

- What does “similar” mean?

One measure is again Euclidean distance:

$$distance(\boldsymbol{t}, \boldsymbol{s}) = \|\boldsymbol{t} - \boldsymbol{s}\| = \sqrt{\sum_{i=1}^{d} (t_i - s_i)^2}$$

Clustering

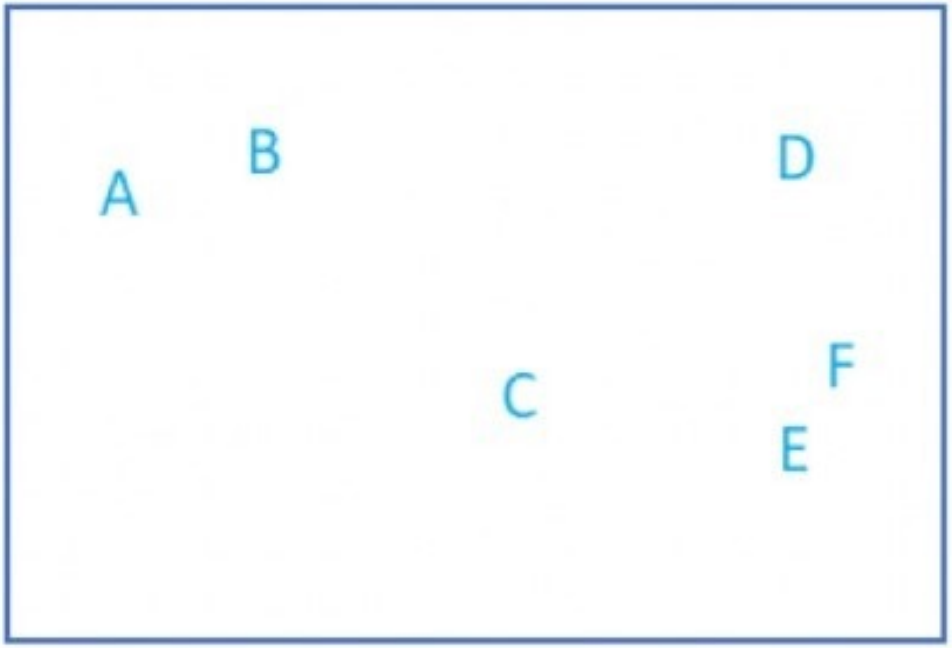
WARWICK
THE UNIVERSITY OF WARWICK

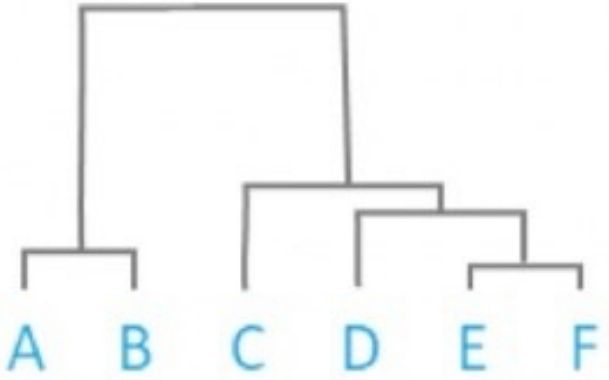
- Generally, there are two types of clustering algorithm:
  hierarchical and partition (n.b. K Means is a partition algorithm).

- Hierarchical involves a set of sequential decisions that assign points to clusters (Decision Tree)

- E.g. Start with each point as a separate cluster. Merge the 2 clusters that are nearest each other. Repeat until you have the target number of clusters.
- By instead continuing until only a single cluster remains, can create a tree showing similarity of different points (where working back up from the bottom of the tree shows similarity of points).

Clustering





Clustering

WARWICK
THE UNIVERSITY OF WARWICK

- Generally, there are two types of clustering algorithm: hierarchical and partition (n.b. K Means is a partition algorithm).

- **Partition** involves partitioning the space (rather than sequentially assigning points to their final cluster)

K Means Clustering

WARWICK
THE UNIVERSITY OF WARWICK

Algorithm (K means clustering):

- Initial step: Pick *K* random co-ordinates as cluster centres

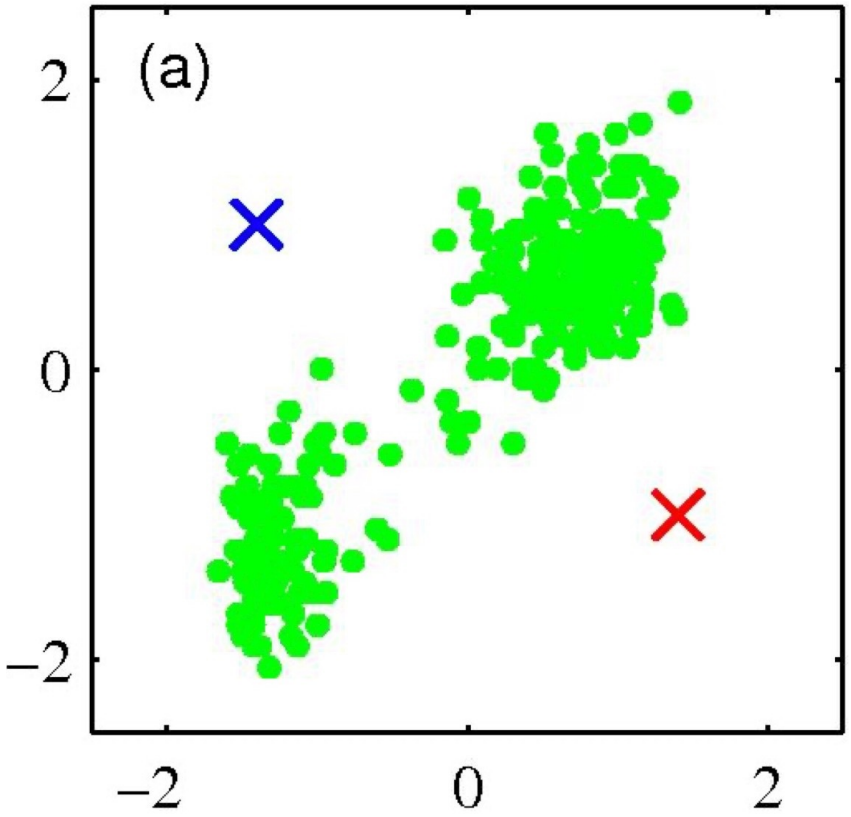
- Repeat: i) Assign data points to closest cluster centre; ii) Change the cluster centre to the average of its assigned points Continue until the cluster centres no longer move.

- This involves sequential partitioning of the space – points are not assigned to their final cluster immediately.

K Means Clustering

WARWICK
THE UNIVERSITY OF WARWICK

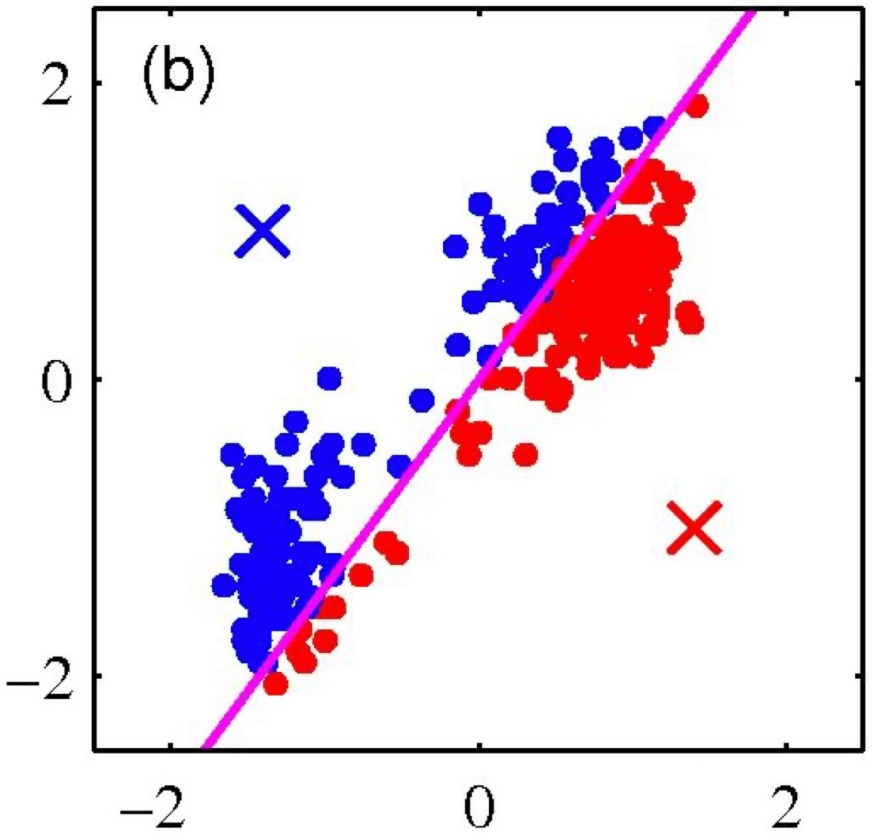
Example (K=2)



K Means Clustering

WARWICK
THE UNIVERSITY OF WARWICK

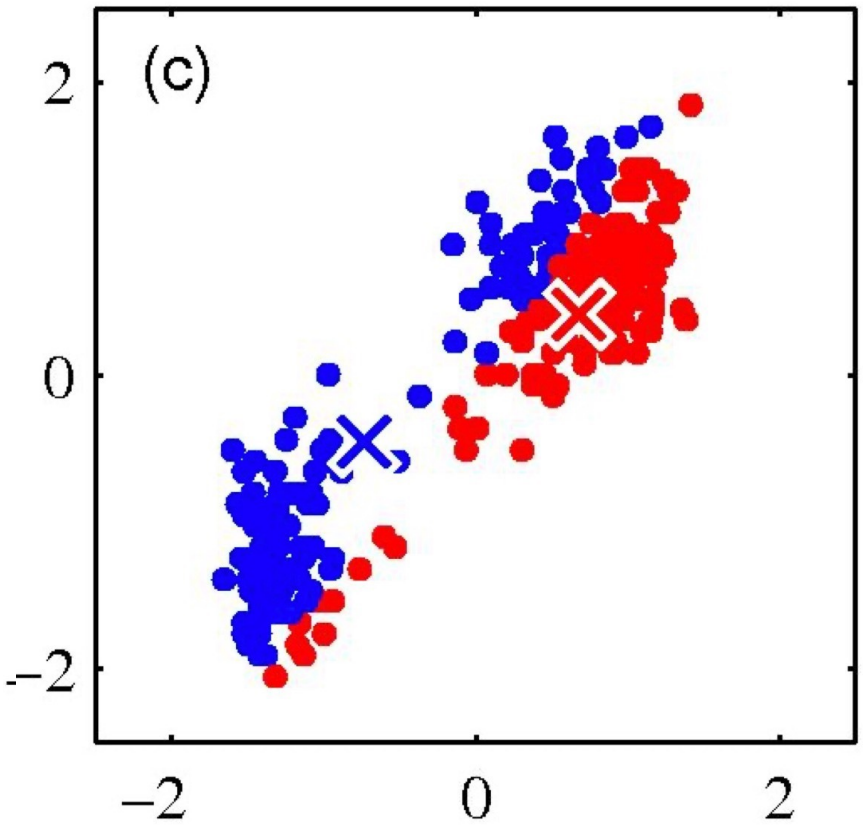
Example (K=2)



K Means Clustering

WARWICK
THE UNIVERSITY OF WARWICK

Example (K=2)

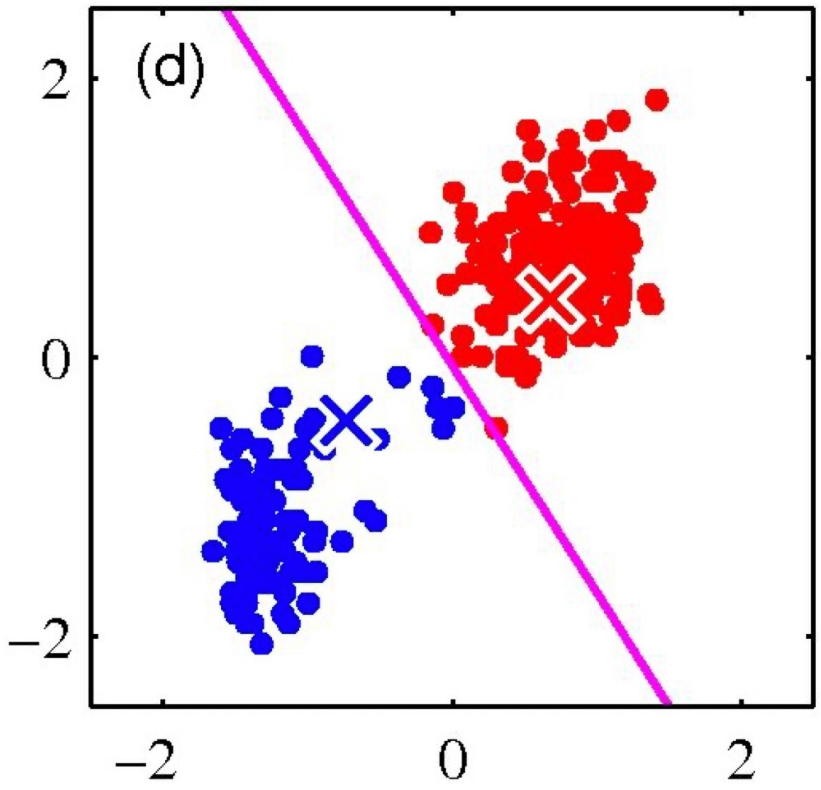


K Means Clustering

WARWICK
THE UNIVERSITY OF WARWICK

Example (K=2)

Repeat until convergence

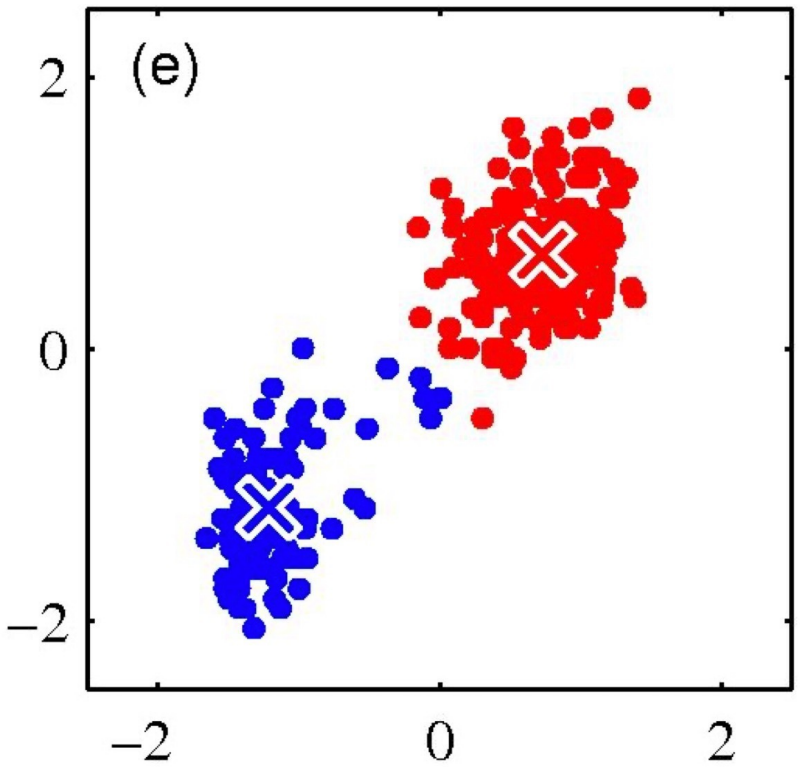


K Means Clustering

WARWICK
THE UNIVERSITY OF WARWICK

Example (K=2)

Repeat until convergence

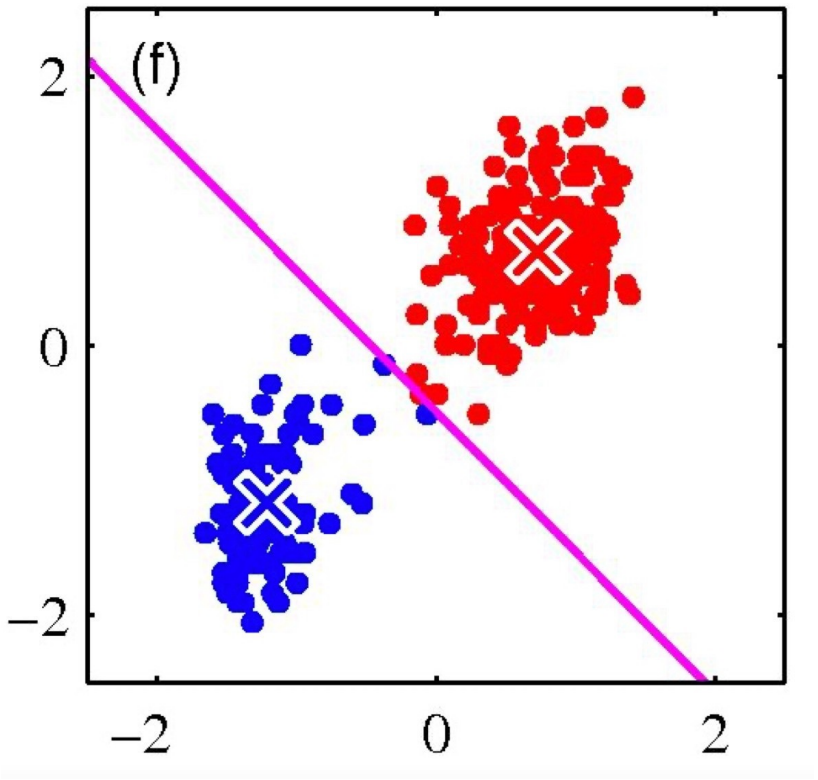


K Means Clustering

WARWICK
THE UNIVERSITY OF WARWICK

Example (K=2)

Repeat until convergence

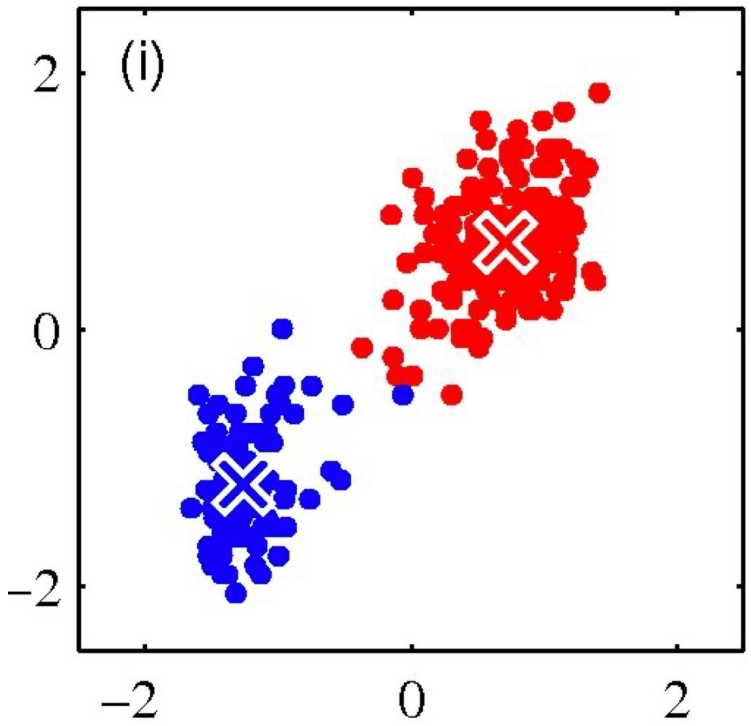


K Means Clustering

WARWICK
THE UNIVERSITY OF WARWICK

Example (K=2)

Repeat until
convergence



K Means Clustering

WARWICK
THE UNIVERSITY OF WARWICK

Exercise 8.2. K Means clustering
Download the Wisconsin cancer dataset. There are samples from 569 breast cancer patients. The first column is the patient ID (can be ignored), the second column is their true diagnosis (benign or malignant), and the other 30 columns are the features (describing the characteristics of the cell nuclei in images).

First, scale the data so each feature column lies exactly in range [0,1]. Write code to perform a K=2 Means clustering.

Perform a classification based on the clusters: Pick one cluster at random to represent “benign” (and the other one malignant). By comparing your result to the diagnoses, assess the accuracy (the proportion of the time that the data point is in the correct cluster), and the probability of type I and type II errors. Repeat this with the clusters in the opposite order.

Neural Networks

- Type of machine learning algorithm, that mimics the way neurons in the brain signal to one another
- Used for standard machine learning tasks of the types discussed so far – e.g. classification
- Also known as artificial neural networks/simulated neural networks

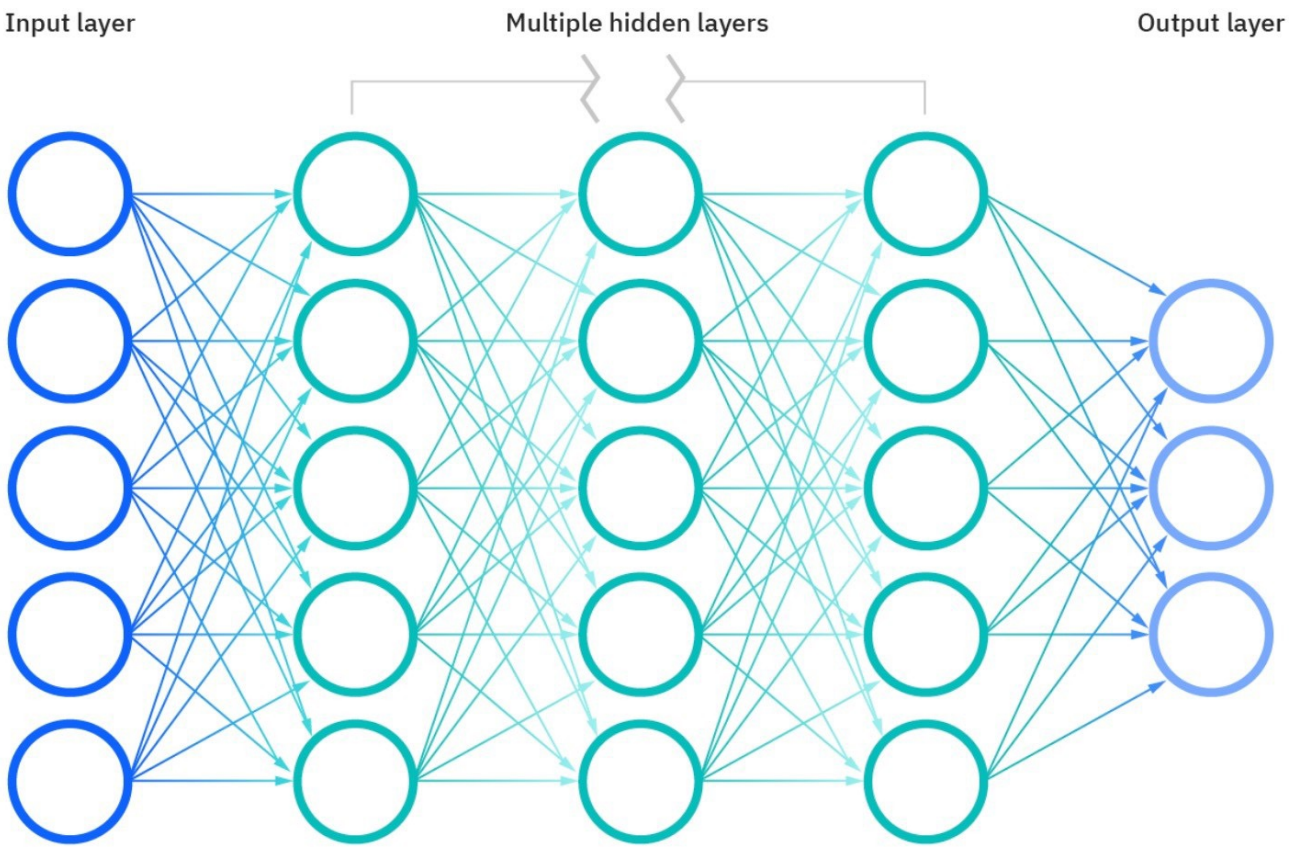
Neural Networks

WARWICK
THE UNIVERSITY OF WARWICK

Idea:

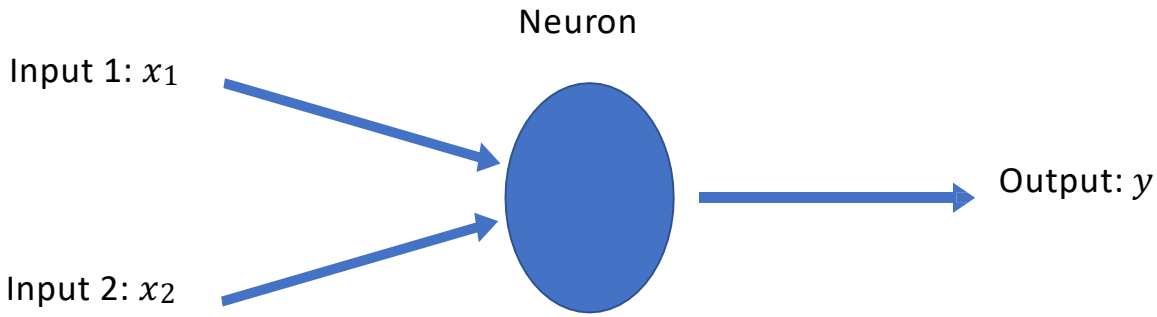
- Multiple layers, made up of neurons (nodes)
- Each node is connected to others in the next layer
- Each node takes inputs from the previous layer, weights them, and then outputs values to the next layer (sometimes only if the total is above a certain threshold/according to an activation function)
- The weights are then optimised using the training data, to generate a predictive model

Neural Networks



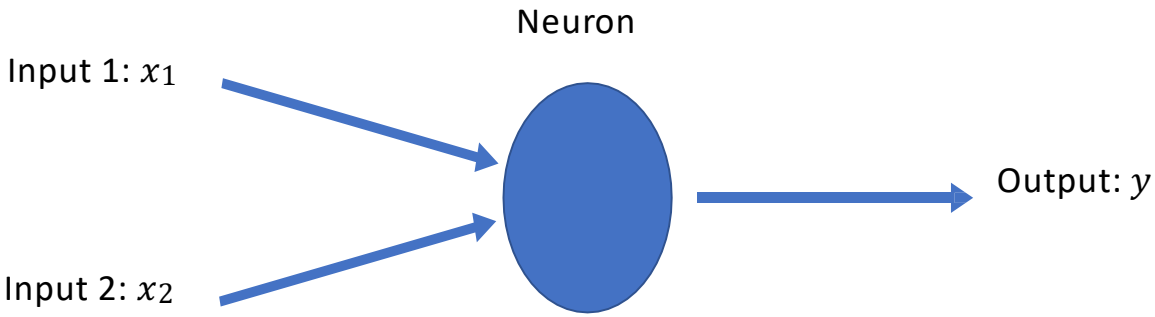
Neural Networks

Main building block – the neuron



Neural Networks

Main building block – the neuron



- Output is a function of the weighted combination of the inputs plus a bias
- $y = f(w_1x_1 + w_2x_2 + \mathrm{b})$

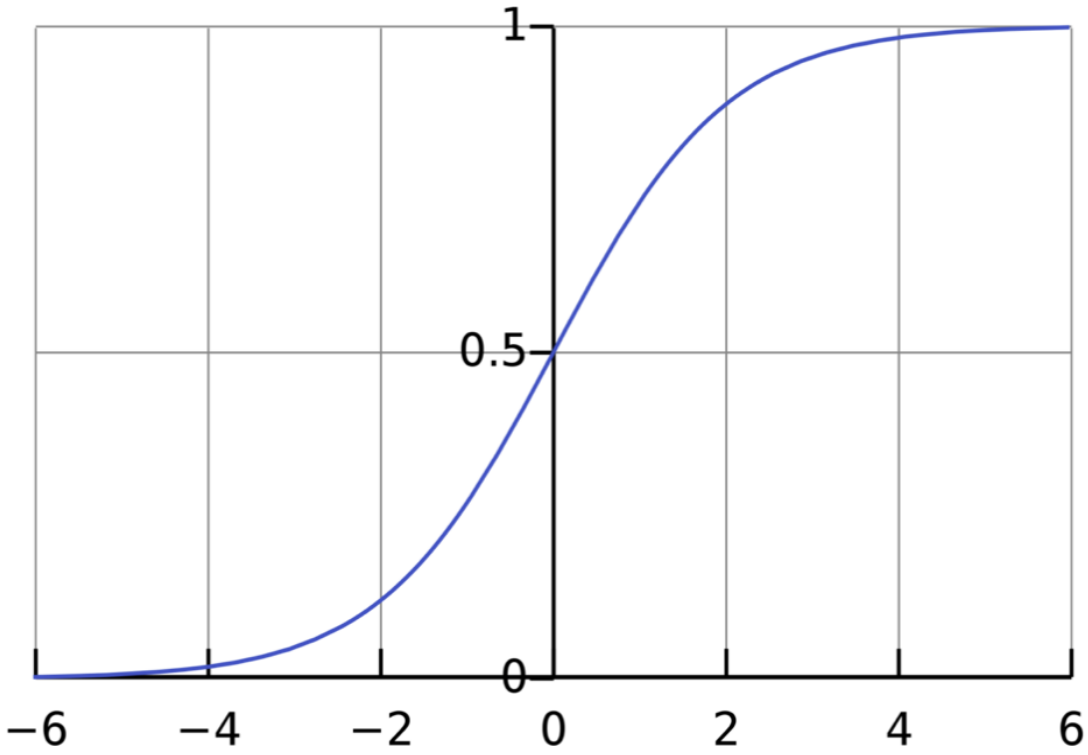
Activation function

Weights

Bias

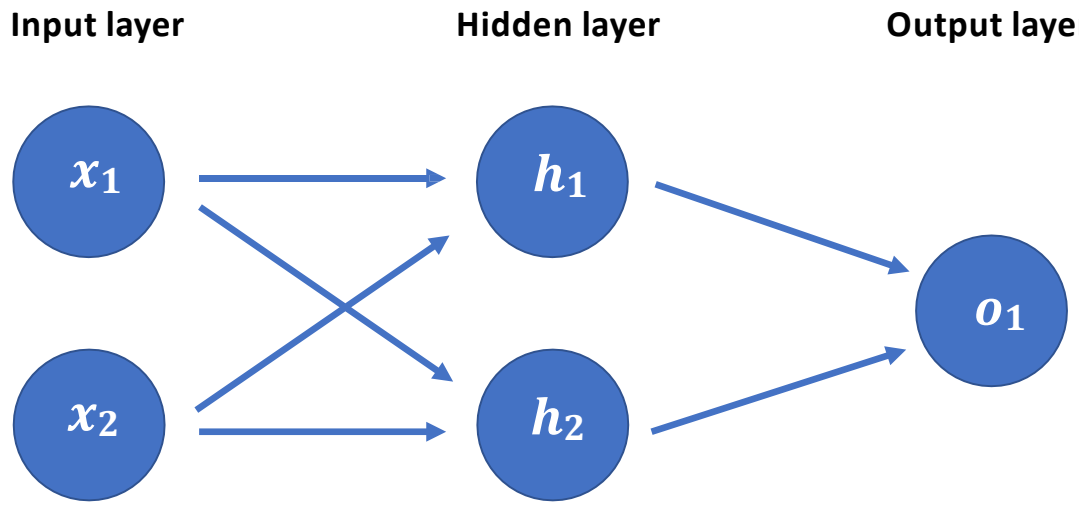
Neural Networks

- Activation function is often a sigmoid. $f: (-\infty, \infty) \to (0, 1)$
- $f(x) = \frac{1}{1+\exp(-x)}$



Neural Networks

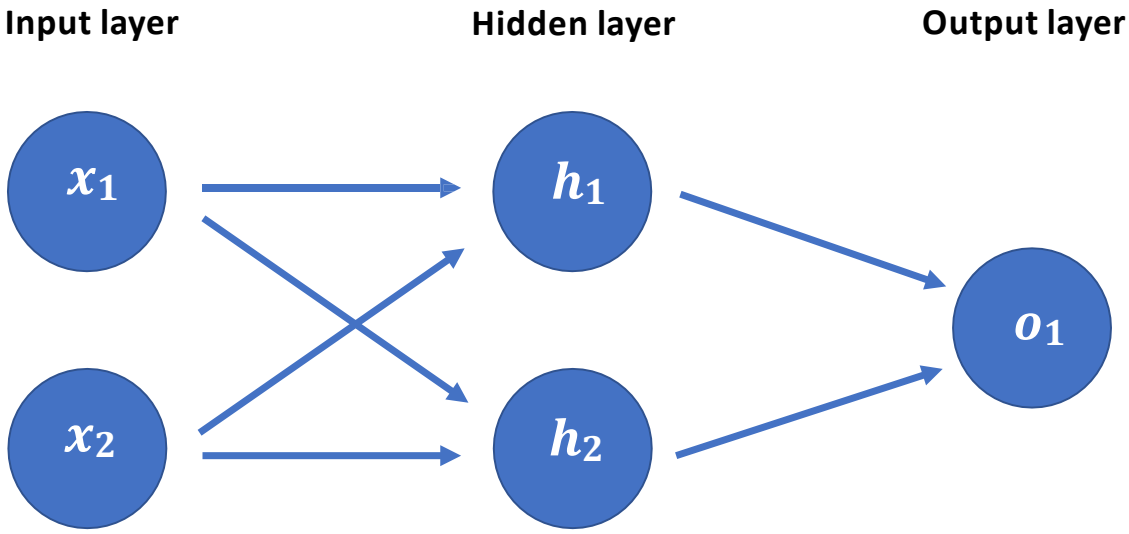
Neurons are then combined into a network



(the output variable name is marked on each node)

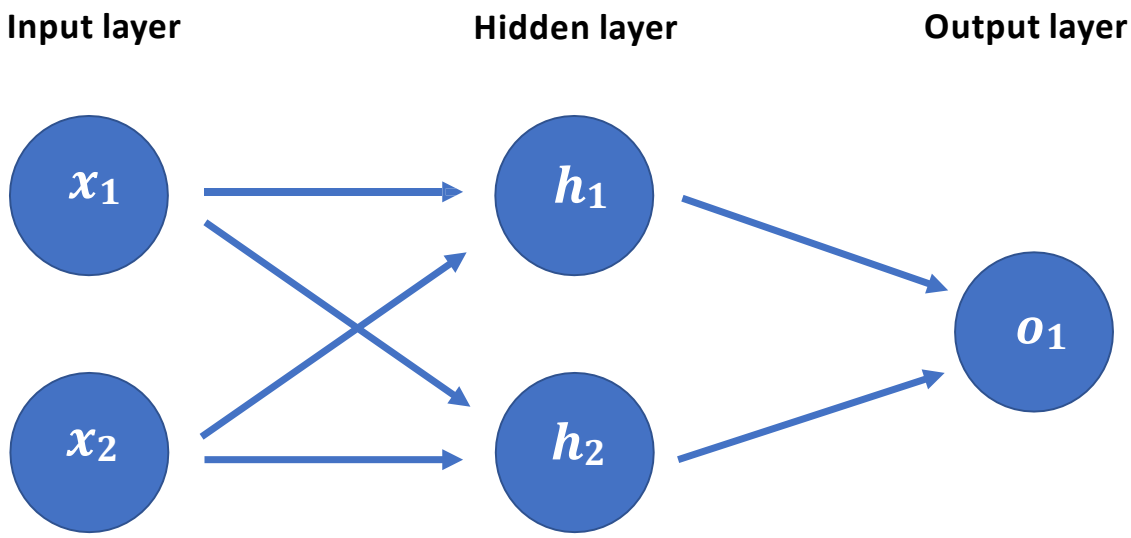
WARWICK
THE UNIVERSITY OF WARWICK

- **Example (board):** For the network below, calculate the "feed forward" output when the input is $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$, if neurons in the hidden and output layers are associated with weights $\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, bias $b = 0$, with previously stated activation function.



Typically, there can be any number of hidden layers

Exercise 8.3. Calculate the output from the previous example if there were instead 2 hidden layers.



Training a Neural Networks

WARWICK
THE UNIVERSITY OF WARWICK

- Now, we will consider how to train a neural network as a predictive model
- Consider developing a classification model, in which the input is someone's height and weight, and the output is whether or not they regularly eat grapefruit
- The output variable lies in (0,1), where a value nearer 1 means they regularly eat grapefruit

Training a Neural Networks

WARWICK
THE UNIVERSITY OF WARWICK

- Training data

Name	Weight	Height	Grapefruit?
A	-2	-1	1
B	25	6	0
C	17	4	0
D	-15	-6	1

Training a Neural Networks

WARWICK
THE UNIVERSITY OF WARWICK

- Training data

Name	Weight	Height	Grapefruit?
A	-2	-1	1
B	25	6	0
C	17	4	0
D	-15	-6	1

- Need a measure of error in model predictions compared to the training data
- We use mean squared error:

$$E = \frac{1}{n}\sum(o_{\text{true}} - o_{\text{predicted}})^2$$

Training a Neural Networks

WARWICK
THE UNIVERSITY OF WARWICK

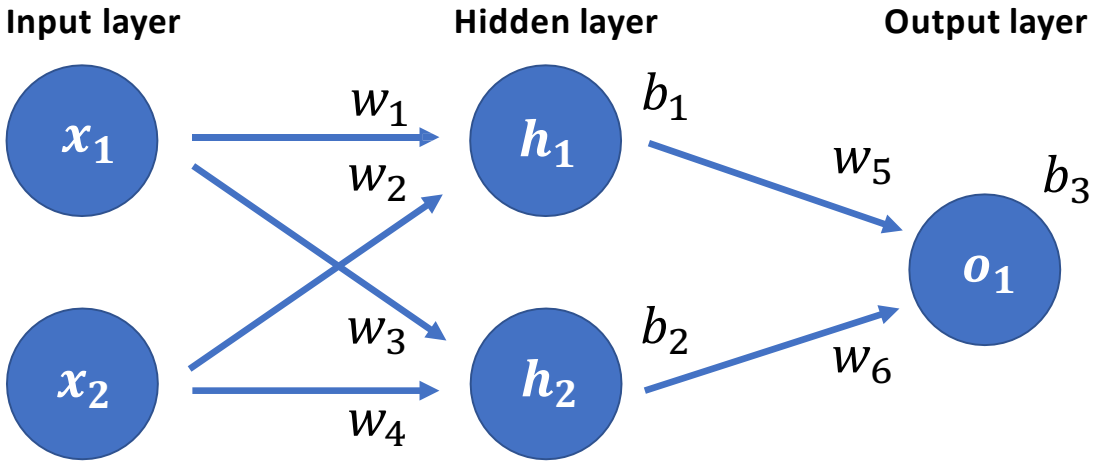
- Training data

Name	Weight	Height	Grapefruit?
A	-2	-1	1
B	25	6	0
C	17	4	0
D	-15	-6	1

- For example, if model is “everyone eats grapefruit regularly”, then

$$E = 0.5$$

Training a Neural Networks



- In principle, there can be different weights and biases for each neuron
- Want to train this network to minimise the error, given the training data
- Error is a function of the weights and biases:

$$E = E(w_1, w_2, w_3, w_4, w_5, w_6, b_1, b_2, b_3)$$

- **Backpropagation** computes the gradient (partial derivatives) in weight space w.r.t. an error/lost/cost function

Training a Neural Networks

- First, consider training the network on data from a single person:

Name	Weight	Height	Grapefruit?
A	-2	-1	1

Training a Neural Networks

- First, consider training the network on data from a single person:

Name	Weight	Height	Grapefruit?
A	-2	-1	1

- Would then use gradient descent to tweak the values of $w_1, w_2, w_3, w_4, w_5, w_6, b_1, b_2, b_3$ to minimise E (backpropogation needed)

Training a Neural Networks

- However, we have training data from 4 people:

Name	Weight	Height	Grapefruit?
A	-2	-1	1
B	25	6	0
C	17	4	0
D	-15	-6	1

Training a Neural Networks

WARWICK
THE UNIVERSITY OF WARWICK

ALGORITHM (Neural network with stochastic gradient descent):

1. Pick person from the training data at random
2. Compute neural network output based on current w/b values (feed forward)
3. Calculate the relevant partial derivatives and update w/b values based on a single step of gradient descent (backpropogation needed)
4. Repeat from step 1

Exercise 8.4. (Optional) Neural network with stochastic gradient descent

We will use the algorithm to optimise the neural network for predicting whether someone eats grapefruit regularly based on their height/weight

i) By hand, calculate the partial derivatives of E for a single individual using activation function $f(x) = \frac{1}{1+\exp(-x)}$. [You do not have to simplify the expressions fully – just enough to do part ii!]

ii) Write code to train the neural network for 1000 steps on the data for 4 people, step size $\alpha = 0.1$, all weights = 1 initially, all biases = 0 initially.

iii) Plot the error function, as a function of the step number

iv) Predict whether someone with (height,weight) = (-7, -3) regularly eats grapefruit. What about someone with (height, weight) = (20,2)

Optional Extra

WARWICK
THE UNIVERSITY OF WARWICK

- Research how to use an in-built Machine Learning toolbox to run ML models using software of your choice

- For example, for MATLAB, one option is:

https://uk.mathworks.com/solutions/machine-learning.html?s_tid=hp_brand_machine

(practical workshop of ML)