

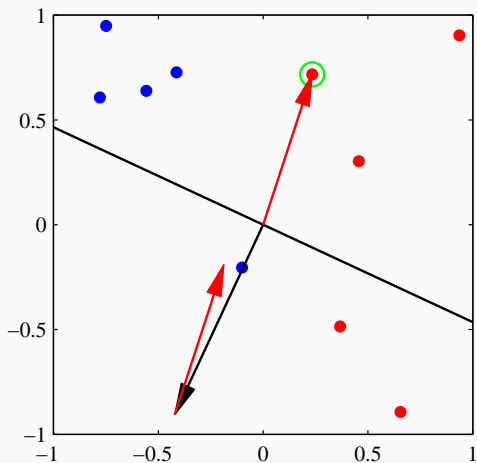
# Introduction to neural networks

---

Dr Gianluca Campanella

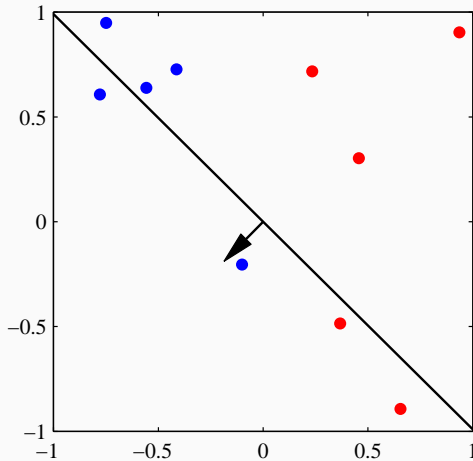
9<sup>th</sup> June 2016

# Perceptrons



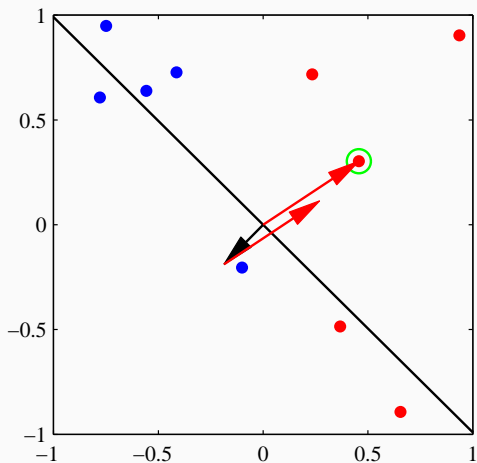
(From *Pattern Recognition and Machine Learning*)

# Perceptrons



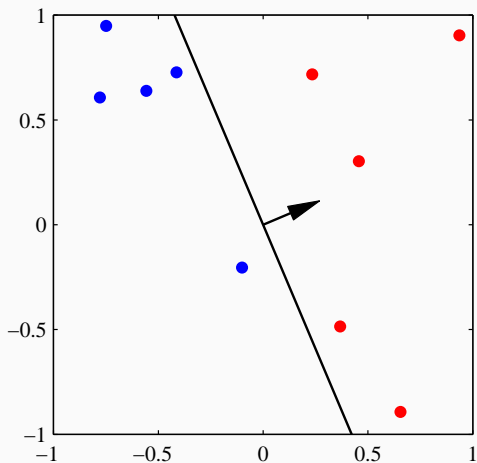
(From *Pattern Recognition and Machine Learning*)

# Perceptrons



(From *Pattern Recognition and Machine Learning*)

# Perceptrons



(From *Pattern Recognition and Machine Learning*)

# Perceptrons

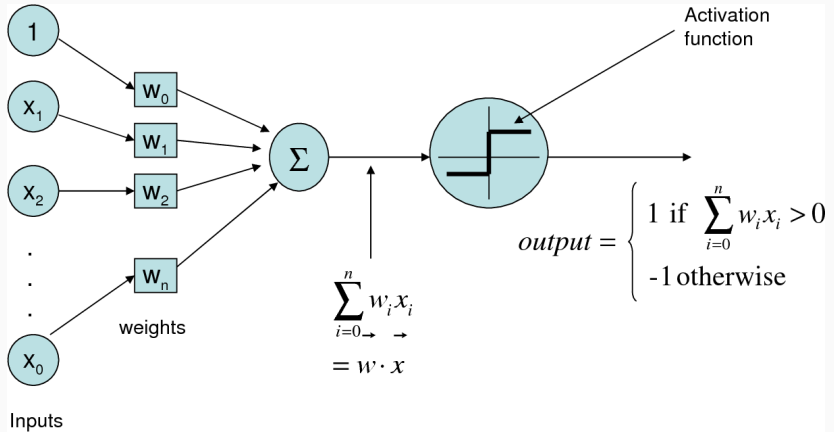
## Support vector machines

- Are trained on the entire dataset at once
- Try to find the largest possible margin

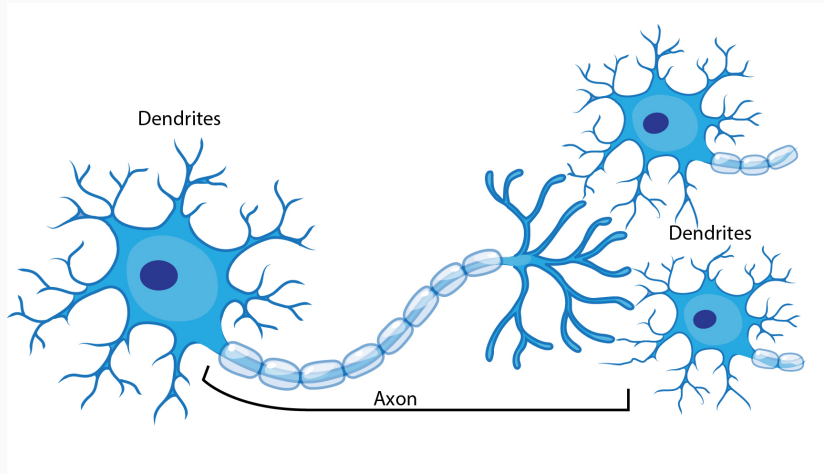
## Perceptrons

- Can be trained online (as the data arrives)
- Do not necessarily maximise the margin

# Perceptrons and neurons

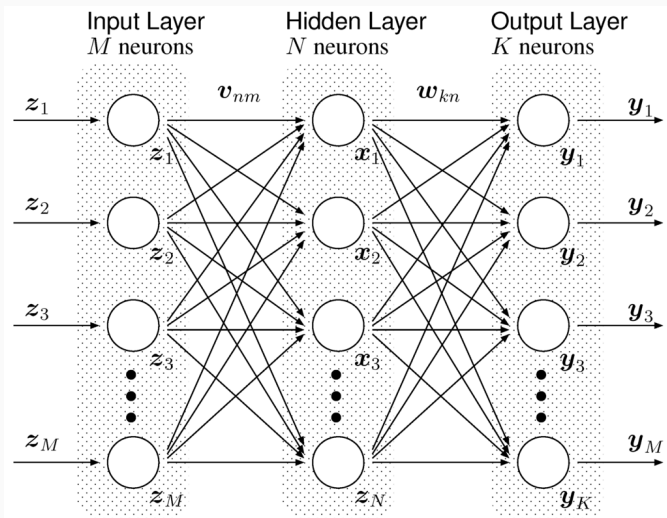


# Perceptrons and neurons





# Multi-layer perceptrons



# Multi-layer perceptrons

## Feed-forward of information

- Receive a new sample  $X$  with outcome  $y$
- Compute value for each unit in each layer
- Compute prediction  $\hat{y}$  and error  $\hat{\epsilon}$

## Back-propagation of error

- Compute 'blame'...
  - For output units:  $y - \hat{y}$
  - For all other layers, as weighted contribution to blame of next layer's units
- Adjust weights and biases

# Multi-layer perceptrons

## Questions

- How many hidden layers?
- How many units in each layer?
- Which activation function?
- How do we initialise weights?
- How do we minimise error?

# Pros and cons

## Pros

- Handle large datasets
- Effective in high-dimensional spaces ( $p > n$ )
- Predictions are fast

## Cons

- Can require considerable parameter tuning
- Training is somewhat cumbersome
- New data can cause 'forgetfulness'