MONASH University

Information Technology

FIT5186 Intelligent Systems

Lecture 3

Multilayered Networks

Learning Objectives

- Understand
  - how a multi-layered perceptron can classify linearly non-separable data
  - the credit assignment problem with multi-layered perceptrons using a discrete activation function
  - the structure of multilayered feedforward neural networks
  - some classification applications of multilayered feedforward neural networks
  - specific requirements of the assignments

Review of Last Lecture

- A single layer discrete Perceptron can dichotomise linearly separable data.
- R single layer discrete Perceptrons can categorise linearly separable data into R categories.
- If the data is not linearly separable (like the XOR problem), then a single layer Perceptron cannot categorise it.

Multi-Layered Perceptrons

- Minsky and Papert's book in 1969 showed the constraint of linear separability for Perceptrons.
  - Most research ceased.
  - Reputation of neural networks damaged.

- Rumelhart and McClelland proposed a solution in 1986:

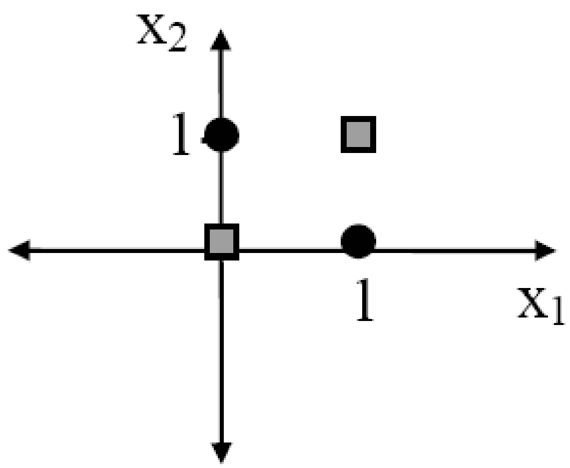
  The Multi-Layered Perceptron.

Multi-Layered Perceptrons (Continued)

- The idea was to use more than one Perceptron, each set up to identify small, linearly separable sections of the input space, then combine their outputs into another Perceptron, which produces the final classification.

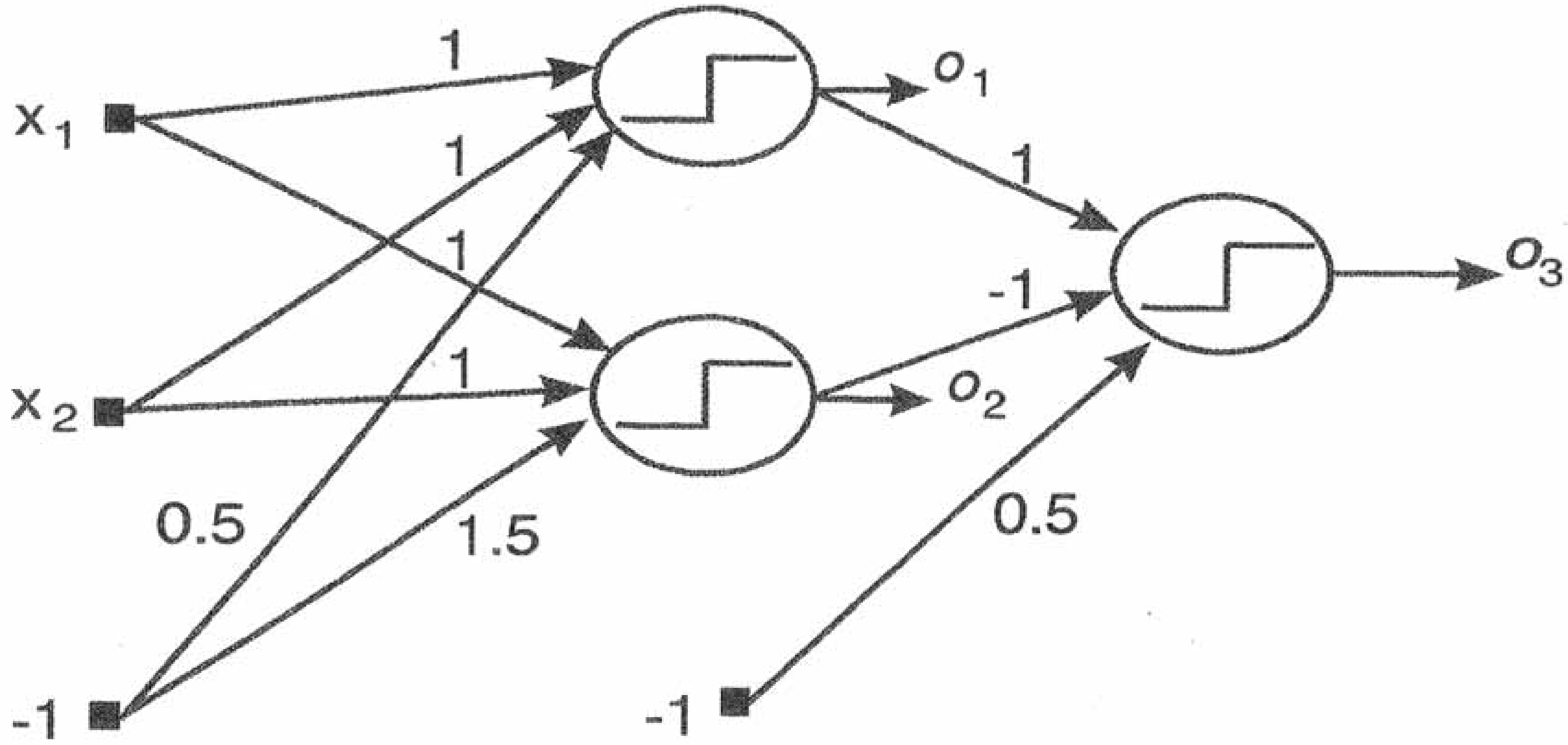
- As such, a multi-layered Perceptron can classify linearly non-separable data (if the weights are known) such as

  the XOR (exclusive OR) problem.



Multi-Layered Perceptrons for the XOR Problem

A multilayered Perceptron for classifying the XOR problem



- See **additional material for Lecture 3**.
- Show the role of hidden neurons in separating data.

Multilayered Perceptron Applications

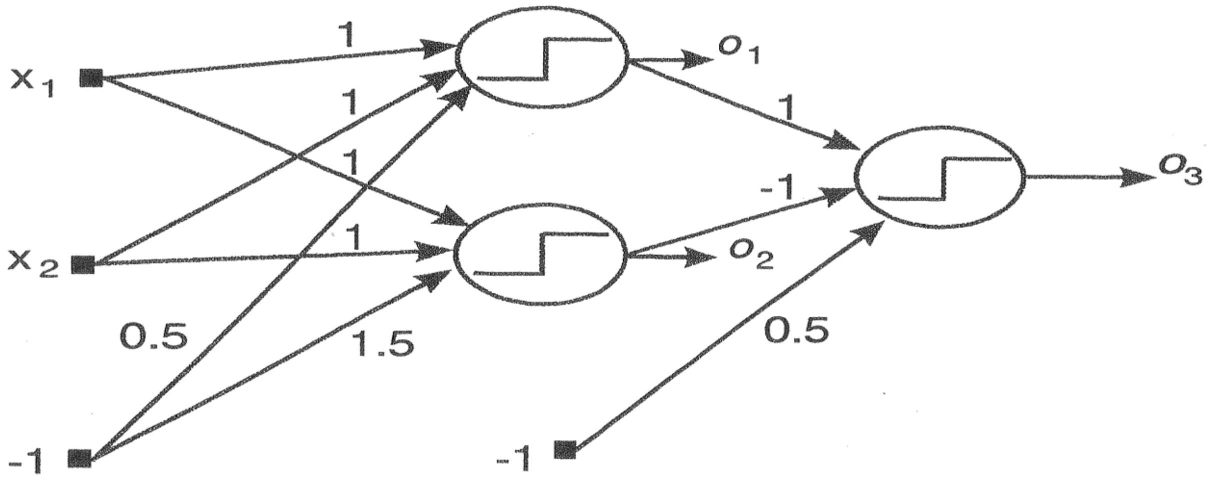
- There are many examples of classification (and prediction) using multilayered perceptrons.
- Choice of inputs and outputs is crucial to the success of classification or prediction.
- Multilayered Perceptron can find relationships between inputs and outputs, through (supervised) training based on existing data.
- Hidden layer outputs provide us with extra (combined) information.
- Let’s talk now about the training ...

Multilayered Perceptron Learning

- How do we determine the weights which will allow a multilayered Perceptron to classify linearly non-separable data?
- We need a learning rule, and to train the network like before.
- **Major Problem:** Remember that learning is all about reinforcing good behaviour and punishing bad behaviour.
  - We need to know which weights are affecting the correct or incorrect classification.
  - We cannot work this out because of the discrete thresholding functions.

Credit Assignment Problem

- Consider again the XOR classification shown below:



  – For the Perceptrons in the 1st layer, the inputs come from the actual network inputs.
  – For the Perceptron in the 2nd layer though, the inputs come from the outputs of the 1st layer.
  – The Perceptron in the 2nd layer doesn’t know what the effect of the 1st layer weights are, and so cannot assign “credit” or “blame”.

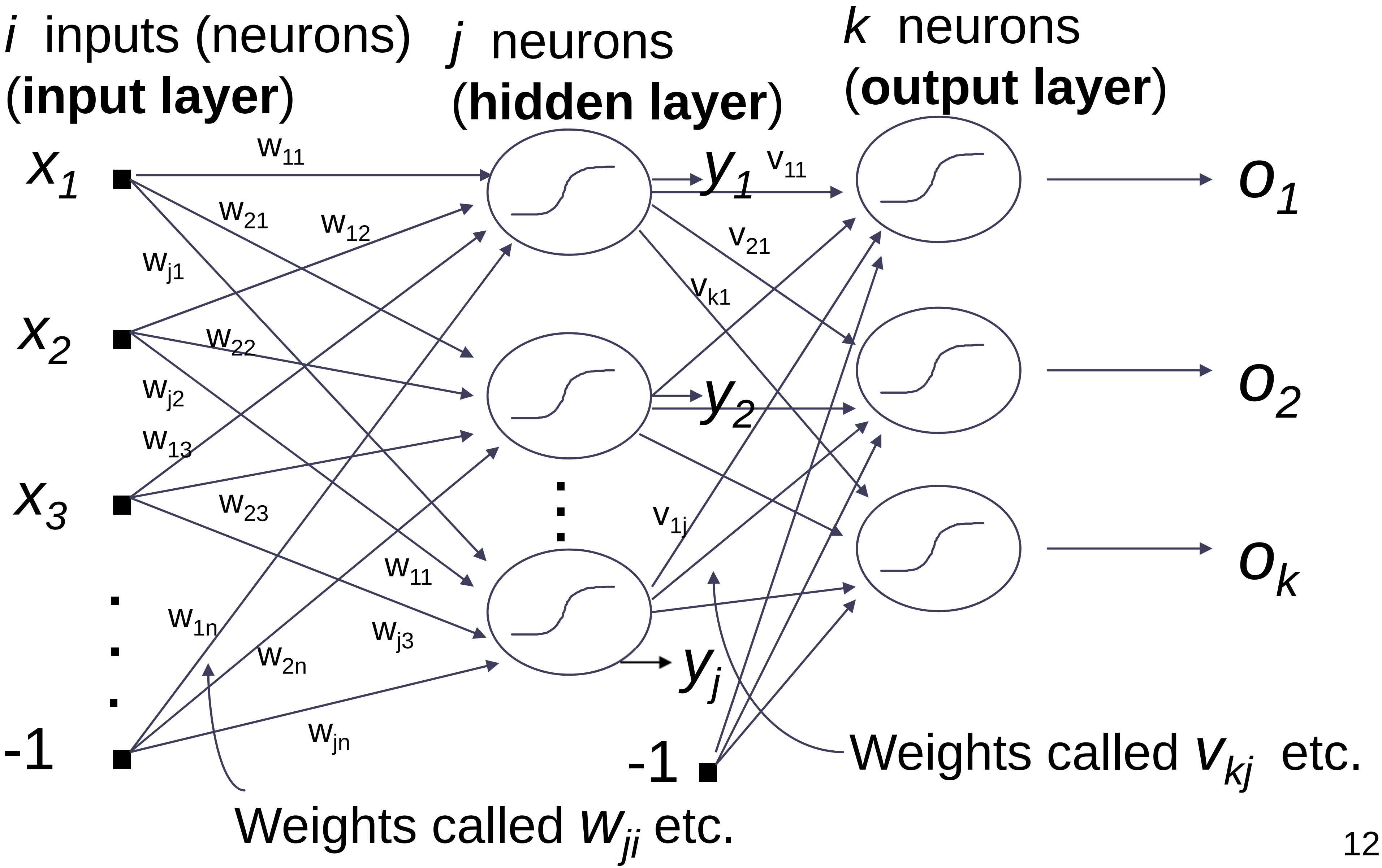
Credit Assignment Problem (Continued)

- Since learning corresponds to strengthening the connections between active inputs and active neurons (recall the Pavlov’s dog example), it is impossible to strengthen the correct parts of the network.
  – The actual net inputs are effectively masked off from the 2nd layer neuron(s) by the intermediate layer(s).
  – The use of the binary neurons (on or off) gives us no indication of the scale by which we need to adjust the weights (whether the net input exceeds the threshold by a lot or a little we don’t know).

The Solution

- If we replace the discrete activation function with a continuous (e.g. sigmoidal) activation function, we retain this information.
  - If the net input exceeds the threshold by a lot, the output is 1, if it exceeds it by a little, the output is positive but smaller.
  - We can determine when we need to strengthen or weaken weights (and by how much).
  - So the network can ***learn***!
  - The network is now called **Multilayered Feedforward Neural Network** (MFNN).

Multilayered Feedforward Neural Network (MFNN)



Structure of MFNN

- The MFNN model has three layers:
  - Input layer
  - Hidden layer (1st column of neurons, maybe more?)
  - Output layer (final column of neurons)
- Each neuron uses a sigmoidal activation function (after summing weighted inputs).
- Thresholding of the activation function is still handled using an extra -1 input and corresponding weights.

Learning Rule

- The learning rule for MFNN’s is called the “generalised delta rule” or “backpropagation rule”.
- Attributed to Rumelhart and McClelland (1986), but really discovered earlier by Werbos (1974) unknown to them.
- A bit more complicated than the previous rule because it involves a derivative.

Learning Rule (Continued)

- All you need to know is that the derivative of the activation function

$$o = f(net) = \frac{1}{1 + e^{-\lambda\, net}}$$

is equal to:

$$\frac{df(net)}{d(net)} = \lambda f(net)\,(1 - f(net))$$

output (not zero)

$$= \lambda\, o\,(1 - o)$$

- We will discuss how this activation function is used in the backpropagation learning algorithm in the next lecture.
- Now let’s look at some classification applications of MFNN.

Classification Application: Marketing

- Applied to determine ATM popularity in Wales
  - Inputs:
    - Consumer expectations of ATMs
    - Degree of functional risk involved
    - Extent of customer confidence
    - Degree of financial risk
  - Outputs:
    - Recommendation of ATM to others
    - Degree of satisfaction
    - Extent of usage
    - Intention to continue using ATMs

Classification Application: Marketing

(Continued)

- Data collection:
  - 200 people given questionnaire
- 4 hidden neurons, appear to correspond to:
  - Consumer motivation
  - Consumer opportunity
  - Consumer ability
  - Associated beliefs
- Trained network showed the strongest weight was between the degree of consumer exposure/attention and opportunity (input 1 to hidden 2).
- Neural network can be used to classify data, as well as determine the relationships among the inputs (and the impact of each input).

Classification Application: Law

- Australia’s LaTrobe University team developed a neural network to mimic the thinking of a Family Law Court judge deciding how to divide assets in a divorce (program called *Split Up*).
- Inputs: financial and personal information (age, kids, investments, assets, etc.).
- Trained on 400 family law court cases over past 3 years (desired output is actual decision of judge).

Classification Application: Education

- Classifying students as likely to pass or fail.
  - 39 inputs: year, sex, visa, VCE marks, accommodation, school, socio-economic group, etc.
  - Hidden layer: varied number of neurons.
  - Output: 1 neuron (pass or fail).
  - Trained network to learn predictions for various Business Systems 1st year subjects.
  - Compared to traditional techniques, NN good.
  - Very useful in identifying potential failures to give them extra help.

Classification Application: Insurance

- 17,000 records of policy holders
  - Age, sex, rating, car type, premium, suburb, etc.
  - Costs of accidents (car repair, hospital, etc.)
- Can we learn to classify which types of policy holders are more likely to have an expensive accident?
- Insurance premiums can then be adjusted.
- Two goals:
  - Learn the classifications and segmentations.
  - Have a predictive tool to estimate how much a new person is likely to cost.

Classification Application: Telecommunications

- Competitive industry (winback and churn)
- Sales people visit businesses and try to sell new products (call waiting, etc.)
  - If existing customer, try to get them spending more.
  - If competitor's customer try to winback.
  - Need to try to reduce churn (customers coming in for a short time and then leaving).
- Can we use the experiences of salespeople (and questionnaires) to determine
  - Which types of businesses are likely to be wonback?
  - Which types of customers are likely to buy more?
  - Which businesses are likely to let you in the door on a given day? ... etc.

Week 3 Tutorial

- Classifying linearly separable data into two groups using
  - Dichotomiser (dichotomiser.exe).
  - R category classifier (with R=2) (rclass classifier.exe).
  - Understand indecision regions.
  - Try moving decision boundaries by adding new data points.
  - As soon as the data becomes linearly inseparable, these Perceptrons will not work.

Selected Reading

- Rosenblatt, F. (1958).The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386-408.
  - Read introduction (first 7 pages) and conclusion (middle part very mathematical and hard to follow).
  - Just appreciate historical significance.
- Flitman, A.M. (1997). Towards analysing student failures: neural networks compared with regression analysis and multiple discriminant analysis. *Computers & Operations Research,* 26(4), 367-377.

FIT5186 Assignment

- TASK:

  Choose a problem of classification, prediction or forecasting which interests you, and train a neural network to solve the problem using NeuroShell 2 and the backpropagation learning rule.

- Worth 40% of the final mark.
- Assignment proposal (10%): Due Week 9
- Assignment report (paper)(30%): Due Week 13

Data Sets

- Try to find your own data (maybe there is a classification, prediction or forecasting problem related to your study, job or hobby?).
- Other ideas: sales figures, rainfall, sport results, character recognition, temperature.
- You may try to search the Internet for some data, for example:
  - The UC Irvine Machine Learning Repository

    http://archive.ics.uci.edu/ml/
  - The neural networks *Frequently Asked Questions* (FAQ) has links to data: ftp://ftp.sas.com/pub/neural/FAQ.html
  - http://www.markosweb.com/www/globalfindata.com/

Data Sets – An Example

- 15 other stocks from Aust. Stock Exchange (supplied by Colin Foster from Dial & Chart Pty. Ltd in Mt. Waverley)
  - ANI Australian National Industries

BML	Bank of Melbourne	BPC	Burns Philp Co.
BUG	Buderim Ginger	CRA	CRA
DGD	Delta Gold	PDP	Pacific Dunlop
FBG	Fosters Brewing	WMC	Western Mining
QAN	Qantas	HLX	Helix Resources
SGP	Stockland Trust	HAH	James Hardie
WBC	WestPac	GIO	GIO Australia

- Each records opening, closing, high, low prices and volumes of trade over the last few years.
  - You can try to predict closing price, or high over a two week period, or volume, etc.

A Direct Marketing Problem

- Direct Marketing Database
  - Using demographic information and previous purchase patterns, which customers are likely to respond to an advertising campaign?
  - Train on 150 buyers and 150 non-buyers.
  - Evaluate on a set of 770 unknowns.

Data Files: A Direct Marketing Problem

- training.xls
  - An Excel file containing the 300 training examples (150 buyers and 150 non-buyers).
  - 11 columns of demographic information.
  - 16 columns of purchase information.
  - 1 final column for buyer “1” or non-buyer “0”.
- product.xls
  - “production set” of 770 examples.
  - * used instead of buyer status.

Data Files: A Direct Marketing Problem

- After training the neural network (i.e. training and test errors are roughly equal and low), evaluate product.xls (surname.pro)
  - Use Apply to File module in NeuroShell 2 to change the data file to surname.pro
- Post-process the output file to produce a tab delimited text file results.txt
  - One column for pattern # (1 to 770)
  - One column with answer (1 for buyer, 0 for non-buyer)

Assignment Proposal (10%)

- Due date: **Friday 27 April 2018 (Week 9)**
- 400-800 words (2-3 pages).
- When you have identified your problem, you need to write a proposal which outlines your problem

  (the exact thing you want to try to predict or classify)
  - Where will you get your data from?
  - What will be the inputs and outputs?
  - What kind of experiments or analysis will you do?

Format of Assignment Proposal

- Title page – including the (tentative) title of your assignment proposal
- **Problem** (the exact thing you try to predict or classify, a brief description of the context or motivation for the problem)
- **Data** (where will you get your data set? what is the data set? what will be the inputs and outputs?)
- **Method** (what neural network architecture or other techniques/models will be used or investigated? What software or system will be used if not NeuroShell 2?)
- **Analysis** (what kind of experiments or analysis will be conducted?)

Final Assignment Report (Paper) 30%

- Due date: **Friday 25 May 2018 (Week 13)**
- Write the report like a short conference paper or journal paper.
  - 1,500-2,500 words (5-8 pages).
  - You may read some papers related to your topic to get an idea of the style and format.
  - You may use any commonly used referencing style (e.g. Author-Date or Numbered) as long as it is consistently applied.
  - This is a valuable experience if you intend to do a research degree, especially a PhD.

Format of Assignment Paper

- Title page – including the title of your assignment paper
- **Abstract** (using no more than 150 words to summarise what the paper is about, including main findings)
- **Introduction**: outline of the problem; brief literature review of related work; why NNs are suited?
- **Data Sets**: where to you find the data? how are you using it? e.g. what are the inputs and outputs? preprocessing?
- **Training Issues**: choice of architecture, parameters, etc.
- **Results**: analysis and comparisons
- **Limitations**: e.g. data or conditions required;
  how useful are your results?
- **Conclusion**: your assessment of NNs for the problem
- **References**