EDINBURGH NAPIER UNIVERSITY
CONPUTER SCHOOL

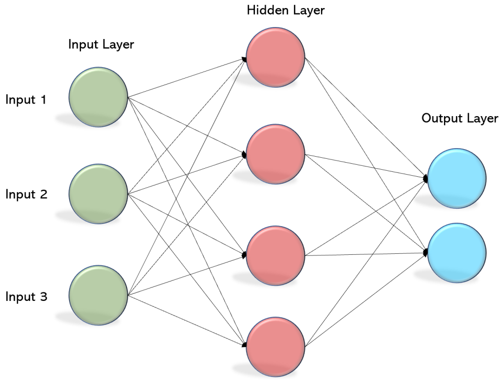
# Computational Intelligence Lecture7

Student name: *weiguang ran*

Professor: *Dr. Kevin Sim*
Due date: *Febuary 28th, 2020*

## Note1

Multi Layer Perceptrons



**Note.**

- Single Perceptrons can be combined into multi-layer networks
- In theory – any function can be approximated by a multi-layer network (here only 2 classes and 2 attributes are shown)

**Example Multilayer perceptron for XOR** A network that solves XOR

1. 3 neurons are used
2. 2 in a 'hidden layer' and 1 for the output
3. The weights are calculated to produce the correct output for all input pairs

## Note 2

Learning XOR

**Note.** Learning XOR

$$\sum w_i x_i = 1 + 1 - 1.5 = 0.5 \tag{1}$$

## Note3

Training multilayer Networks

**Answer.**

- The simple perceptron learning rule can't be applied to a multi- layer network
    1. As there is a knock-on effect with the output of one node being the input of the next
- For any given input pattern, we can only easily calculate the error at the final output node(s)
    1. we don't know what the outputs for the hidden nodes should be
- There are 2 ways to find the required weights:
    1. Backpropagation (for supervised learning, a bit more later)
    2. Evolutionary Algorithms (unsupervised learning, next week)

## Note4

Classification with MLPs

**Answer.**

- What kind of problems can we use NNs for
- Dealing with data
- Designing the neural network
- Training the network (for classification)
- Backpropagation
- Testing the network

## Note5

What kind of problems are NNs good at?

**Answer.**

- Instances can be represented by many input-output pairs (both can be vectors)
- Target function can be discrete-valued, real-valued, or a vector of real/discrete attributes
- Training examples can contain errors

- Long training times are acceptable
- Fast evaluation of the target functions is required once trained
- The ability of humans to understand the learned target function is not important

**Note6**

And when they are not very good

**Answer.**

- If you need to understand the reasons behind output:
  - E.g medical applications – predicting a heart attack
- No relationship between input and output variables
  - E.g day of the week -> weather forecast
- No relationship between the past and the future
  - E.g predicting the lottery numbers for next week
- Training data is very biased:
  - Only have examples from one particular class (e.g trying to predict class of degree but only have examples of students who obtained 2(1) and 3rd.)

**Note7**

Dealing with data

**Answer.**

- Factors to consider
  - categorical, numerical, discrete, ordered
- input encoding
  - Transforming the data
- output encoding

**Note8**

How do we convert the output to a price ?

**Answer.**

- We can do the same for the output:
- Prices vary between £52,500 and £225,000
- Set up output so that £50,000 maps to 0, and £250,000 maps to 1

$$output = \frac{120,920 - 50000}{250,000 - 50000} = 0.3546 \tag{2}$$

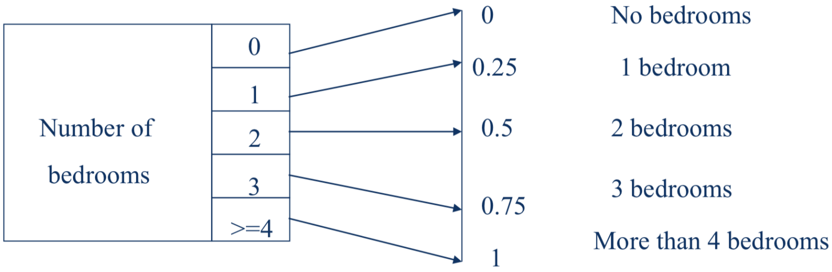
Note that the NN is unlikely to be good on new data that is outside of the boundaries you set so choose them carefully

### Note9

Discrete Data

**Answer.**

- Discrete data such as number of bedrooms, number of bathrooms, also has minimum and maximum values
- Convert to numerical values on a scale:



### Note10

Backpropagation

**Answer.** The Backprop Algorithm:

- For each pattern presented it:
  - Calculates error at output (easy)
  - Changes weights to output layer to reduce this error
  - Calculates error at last-but-one layer (harder)
  - Changes weights to reduce this
  - Repeat....
- The algorithm is applied each iteration until the network error is minimised