CS3002: Introduction to Neural Networks - ASSESSED

This worksheet must be assessed by a GTA. All assessed exercises must be completed and checked by a GTA in the lab. This is PASS / FAIL and all assessed sheets must be passed in order to pass the coursework for this module. You can make multiple attempts but do not ask to be assessed until you are ready.

Datasets:

- Wine Data

Functions:

- `neuralnet`,
- `compute`,

Perceptrons

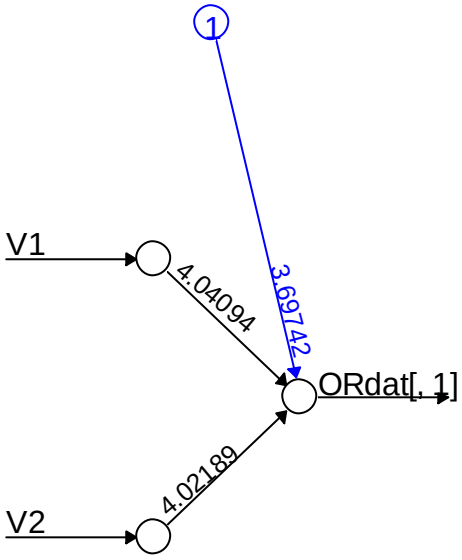
Let's start by defining some input and output data for an OR gate (note the output is only zeros when both inputs are -1).

```
install.packages("neuralnet")
library(neuralnet)

#OR gate input data
trainin = rbind(c(1,1), c(1,-1), c(-1,1), c(-1,-1));
#OR gate output data
trainout = rbind(1, 1, 1, 0);
#Combined OR gate data
ORdat=cbind(trainout,trainin)
```

And then setting up a single binary perceptron with two inputs and one output using `neuralnet` to train it:

```
# fit neural network with no hidden layers
set.seed(2)
NN = neuralnet(ORdat[,1]~., ORdat[,-1], hidden = 0 , threshold = 0.001,
stepmax = 1e+05, linear.output = FALSE)
#visualise the NN
plot(NN)
```



Error: 0.000664 Steps: 47

This automatically gives us random weights and biases which we can check with:

```
NN$weights
```

Now let us use `compute` to see if the network responds to an input signal (1,1).

```
testin= rbind(c(1,1))
predict_testNN = compute(NN, testin)
```

The activation of the output neuron is here
```
predict_testNN$net.result
```

To calculate the discrete class we threshold it at 0.5:
```
predict_out = as.numeric(predict_testNN$net.result>0.5)
print(predict_out)
```

so you can see it correctly predicts a 1 as output

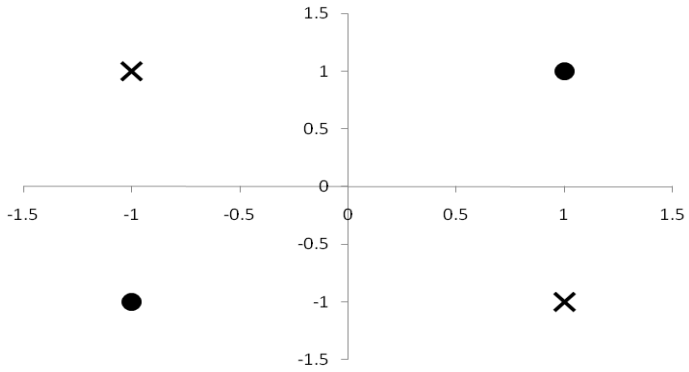
Note that we could present the a number of inputs in a sequence and get the outputs in a sequence as well.

```
#set up the input sequence
testin=rbind(c(1,1),c(1,-1),c(-1,1), c(-1,-1))

predict_testNN = compute(NN, testin)
predict_testNN$neurons
predict_testNN$net.result
predict_out = as.numeric(predict_testNN$net.result>0.5)
predict_out
```

ASSESSED EXERCISE:

Now try some other inputs and targets are the test set. See if you can set up the XOR problem (below) and see what happens when you try and learn the weights.



Input: (-1,-1); Output: 0
Input: (1,-1); Output: 1
Input: (-1,1); Output: 1
Input: (1,1); Output: 0

Does it learn to correctly classify all the inputs?

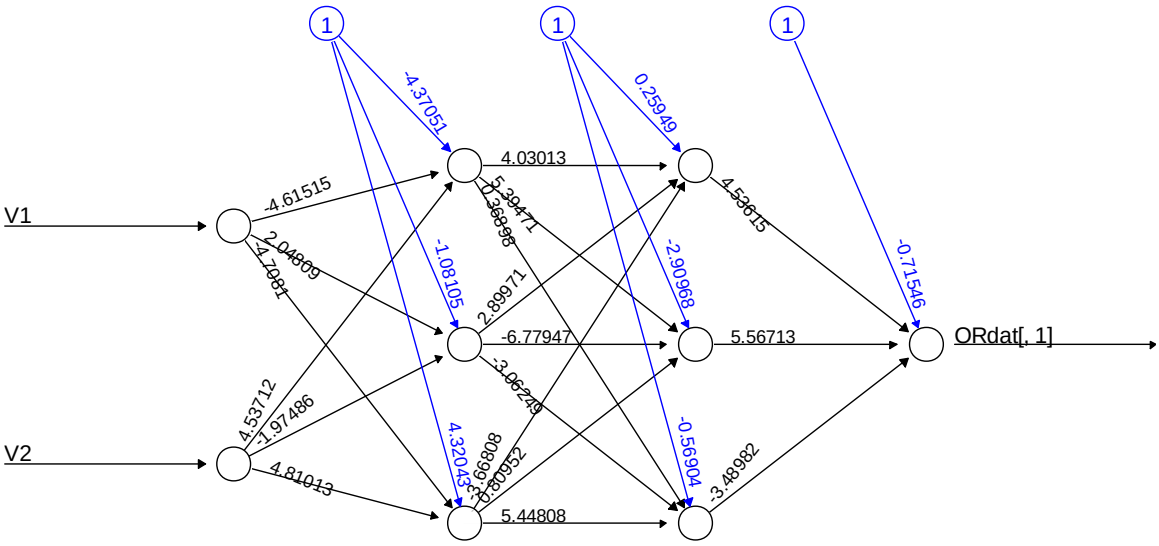
Multilayer NNs

Set up a 2 layer Neural network to solve the XOR problem with three hidden layers, each with 2 neurons in, and one output layer using `neuralnet`:

```
#XOR gate input data
trainin = rbind(c(1,1), c(1,-1), c(-1,1), c(-1,-1));
#XOR gate output data
trainout = rbind(0, 1, 1, 0);
#Combined XOR gate data
XORdat=cbind(trainout,trainin)

#train a neural network on the XOR data
set.seed(2)
NN = neuralnet(XORdat[,1]~., XORdat[,-1], hidden = c(3,3) , threshold =
0.001, stepmax = 1e+05, linear.output = FALSE)
```

This will train a neural network on the XOR data using the following architecture:



Error: 0.001151 Steps: 49

Now simulate using the same inputs used to train it:

```
testin = rbind(c(1,1), c(1,-1), c(-1,1), c(-1,-1));
testout=rbind(0,1,1,0)

predict_testNN = compute(NN, testin)
predict_testNN$neurons
predict_testNN$net.result
predict_out = as.numeric(predict_testNN$net.result>0.5)
predict_out
```

Repeat the training a few times (remember to initialize the networks beforehand). Try changing the number of epochs and the goal parameter to see what effect it has.

For three or more class problems we need more output neurons (one per class). Each neuron then represents a class and the class node with the highest output activation when testing is the most likely class.

ASSESSED EXERCISE:

Build a Neural network classifier of the wine or face data.

1. Read in “winedata2.csv” from last week’s labs
2. Build the architecture of your neural network. The output must be between one and zero.
3. Using any two variables from the wine data, set up the data as you did for the linear classifier with a train and test set
4. Train the neural network on half of the data and test it on the remaining.
5. Calculate the accuracy

Deep Learning using TensorFlow in R

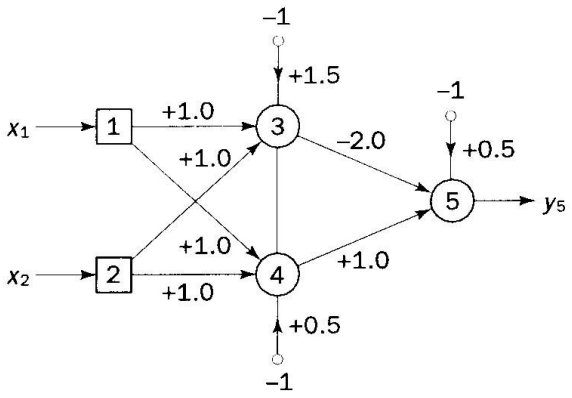
Please see: https://tensorflow.rstudio.com/

SAMPLE EXAM QUESTIONS:

Given the truth table of the Logical OR Operator
below, where binary variable x_1 *v* x_2 are input and *y* is output.

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

Q1) Describe, using a diagram, the architecture of a neural network to solve the problem.
Q2) Describe a suitable training algorithm for your network.
Q3) Given the following network:



What would the output be for the inputs in the Logic OR truth table?