

Training a simple perceptron to solve the stock market prediction problem introduced in the lecture:

Age	Previous Movement	Buy/Sell
10	-10	Buy
10	+30	Buy
8	-50	Sell
6	-20	Sell
3	+20	Buy
2	+20	Sell
2	+60	Buy
7	+15	?

In OL you'll find a Excel worksheet that defines the neural network. Each row defines the state of the network as it is trained. The columns are:

- 1. W1 and W2** The weights for the two inputs. The initial weights are chosen at random
- 2. Bias** The weight of the bias input. This will act as a threshold.
- 3. Age and Movement** The first two inputs, scaled to be in the range (-1,1).
- 4. Target** The output that we would like the network to produce.
- 5. Sum** The weighted sum of the inputs.
- 6. Output** The output of the threshold function.
- 7. Error** The difference between the actual output and the target.
- 8. Rate** How quickly the network should learn.
- 9. New W1, W2, Bias** The new weights once the learning rule has been applied

Exercises

1. Look at the contents of each cell in the first row and check you understand what they all mean.
(Some of the cells have equations defined in them, so you'll have to click on the cell to see the contents.)
2. Fill in the next row of the spreadsheet. Copy the contents by reference where appropriate. (ie use '=C1' to make the contents of a new cell equal to C1, rather than just cutting and pasting the value.)
3. Create the remaining rows of the training set. This is your first training cycle.
4. Calculate the *total error* for the first training cycle.
5. Use cut and paste to repeat the training cycle. Has the error improved?
6. Repeat the training process until the network has learned the entire training set.
7. Use the trained network to make predictions about the following unknown stocks.

Age	Previous Movement	Buy/Sell
7	-15	?
4	+20	?
2	+10	?
10	-10	?

8. Try the following to see what effect they have on the learning process:
 - Try altering the learning rate. Can you make the network learn more quickly? What happens if the learning rate is too high?
 - Try altering the initial random weights. Can the network learn from any starting state?
 - Try unnormalising the data (ie using the original input and output values). Can the network still learn reliably?