

Machine learning: Exercises for September 21, 2017

Neural networks and back-propagation

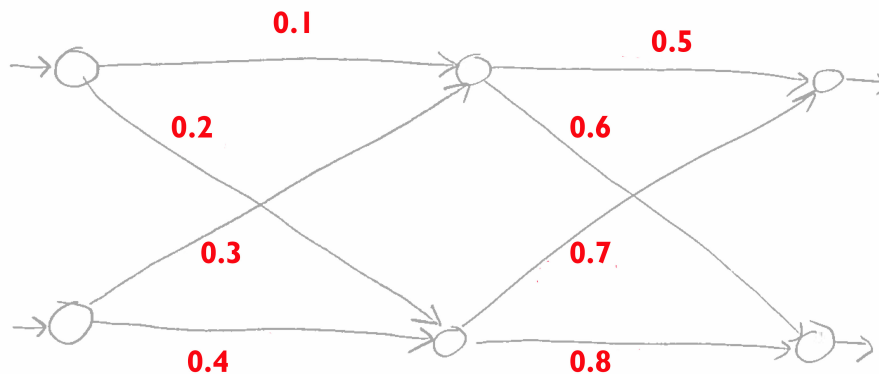
Henning Christiansen

Roskilde University

<http://www.ruc.dk/~henning/>, henning@ruc.dk

1 Hand-simulation of the back-propagation algorithm

In the lecture (see pdf version of slides on moodle), we illustrated one step of the back-propagation algorithm. Here you should do the same, but with different numbers. As before, we consider a network with 2 input nodes, 2 hidden nodes and 2 output nodes. Assume, at a given time during the training of the network, that the weights are shown:



Perform the calculations by hand that adjust the weights for the following tuple (that is part of the training data).

$$\langle \langle 0.5, 0.2 \rangle, \langle 0.7, 0.3 \rangle \rangle$$

Hint: you may or may not find it convenient to use a spreadsheet program for this exercise, at least for calculating the sigmoid function and its derivative.

2 Get familiar with a Processing program that provides data structure for neural networks plus the feed forward algorithm

Find on moodle the two files `neuralNetworks.zip` and `backPropTraining.csv`. Unpack the first one and you will see a folder `neuralNetworks` containing the Processing program; the second one is a file of training data that represents a population of data tuples, each one of the form $\langle \langle x, y \rangle, \langle x + y, x - y \rangle \rangle$.

Be aware that this program shows its output in the lower part of the Processing window, the graphics window is not used (but Processing requires that there is one). First of all, make yourself an overview of the main file and consider the purposes of the other files (no details for now).

2.1 Understand the data structures

The code line

```
SingleHiddenLayerNetwork myNetwork = new SingleHiddenLayerNetwork(2,2,2);
```

is supposed to create a network structure with 2 input nodes, 2 hidden nodes and 2 output nodes. Check that it does so in a correct way, including initializing weight by random numbers; it is suggested that you make drawing the datastructure.

2.2 Check that the feed-forward method works correctly

The code lines

```
float [] input = {0.7,0.4};  
float [] output = myNetwork.run(input);
```

is supposed to call the feed-forward method using the net with its current weights. Go through the code lines and check that it works correctly – if you find any bugs, correct them and report to your teacher immediately ;-)

3 Implement the back-propagation algorithm

Now you should extend the program considered in the previous exercise with an implementation of back-propagation. The file `learning` includes a function

```
void train(SingleHiddenLayerNetwork net, Table trainingData) { ... }
```

It is a stub that does nothing interesting. It contains a proposal for the basic control structure, which you may use (or replace by your own code if you prefer). Your task is now to add the remaining details that implement the back-propagation algorithm and test it.

4 If you have lots of time

Extend the program so it can show a nice picture of the current network structure in the graphics window.