Multi-Layered Neural Networks and the Back-propagation Algorithm

Contents

- Remark
- Multi-Layered Neural Network - Recall
- Tasks:

Remark

Working with neural networks we use the representation of vectors and weights same as in the MATLAB, in particular within the toolboxe for neural networks:

Input vectors are always column vectors.

For easy computing potentials on a neuron, the weights of incoming synapses of the neuron are stored as a row vector.

Multi-Layered Neural Network - Recall

Let us take a neural network with the topology [2,2,1], i.e. the network has 2 input neurons, 2 hidden neurons in a single hidden layer, and one output neuron. Let the weights of synapses between the input and the hidden layer be in the following matrix:

```
w_i_h = [ 0.50000 1.50000
         -0.50000 0.50000];
```

`w_i_h(i,j)` is the weight of the synapse from the input `j` into the hidden neuron `i`. I.e. each row of the weight matrix corresponds to weights of synapses leading **to** the neuron!

Let the synaptic weights between the hidden and the output layer be in the matrix:

```
w_h_o = [2.00000 -1.00000];
```

`w_h_o(1,i)` is the weight of the connection from the hidden neuron `i` to the output neuron. thresholds of the hidden neurons are in the matrix:

```
b_h = [ 0.00000
        0.50000];
```

and the threshold of the outout neuron is:

```
b_o = -0.50000;
```

Hence the weights from the input layer into the hidden layer with added virtual neuron with fixed output 1 (for representing thresholds) are:

```
w_i_hb=[w_i_h b_h]
```

```
w_i_hb =

    0.5000    1.5000         0
   -0.5000    0.5000    0.5000
```

The weights from the hidden layer into the output layer with added virtual neuron with output 1 are:

```
w_h_ob=[w_h_o b_o]
```

```
w_h_ob =

    2.0000   -1.0000   -0.5000
```

A sigmoidal transfer function is implemented in MATLAB (in its Neural Networks Toolbox)

$$logsig(x) = \frac{1}{1+e^{-x}}.$$

It is the sigmoid function with the slope equal 1.

Tasks:

1. Compute the output of the network for the input patterns `p1=[-1; 1]` and `p2=[1; -1]`.
2. The input pattern `p1=[-1; 1]` is a training vector with the desired out put `0.9`. Which is the error of the network on this pattern? How will change the weights of the network after one step of the backpropagation learning algorithm with the learning rate $\alpha =$ `0.2`?