

ENEE 434 Homework #5

Work problems 1, 2, 4 for grading, due Tu, May 7, 2002. Others for practice for final exam and represent possible topics for the final exam.

1. (35 points) (WS)

A neuron perceptron shown below can be defined and trained for the given input vectors and targets by the following code. Assume that the initial weights and bias values are zeros. Also, assume the perceptron uses the hard limit transfer function, *hardlim*, where $\text{hardlim}(0) = 1$.

```
net = newp([-5 5; -5 5], 1);
net.trainParam.passes = 1;
p = { [-4 ; 3] [2 ; -3] [1 ; 5] [-2 ; 0] };
t = { 1 0 1 1 };
[net, a, e] = adapt(net, p, t)
```

Find the output, *a*, and error, *e*, that are returned.

Next, we can simulate the trained network for each of the inputs using the function *sim*. Then, to check whether the network has converged and produces the correct outputs for the four input vectors, we can find the error for each of the inputs. The MatLab script is as follow:

```
a1 = sim(net, p)
error = {a1{1}-t{1} a1{2}-t{2} a1{3}-t{3} a1{4}-t{4}}
```

Find the error that is returned. Has the network converged?

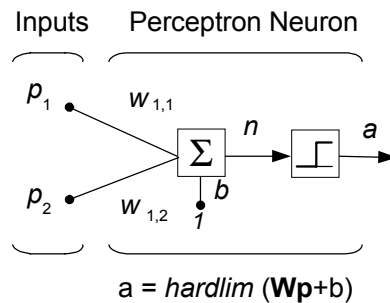


Figure 1: One neuron perceptron

2. (35 points) (TH)

The electrochemical process of zinc (EPZ) can be described by the back propagation neural networks. The following architecture of BP3L-V describes the relationship between the cell-voltage *V* and C_s , C_{Zn} , and D_c . Where C_s is the concentration of sulphuric acid, C_{Zn} is the concentration of zinc sulphate and D_c is current density.

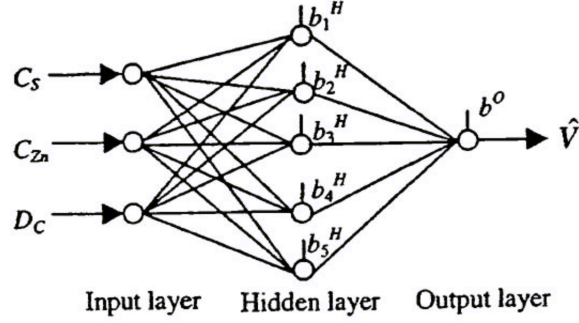


Figure 2: The architecture of BP3L-V

Let $p_1^I = C_s$, $p_2^I = C_{Z_n}$ and $p_3^I = D_c$. The input and output of the i^{th} neuron of the hidden layer of BP3L-V are defined to be

$$p_i^H = \sum_{j=1}^3 w_{ij}^H p_j^I + b_i^H$$

$$y_i^H = \tan \text{sig}(p_i^H)$$

where $i = 1, 2, 3, 4, 5$, w_{ij}^H denotes the weight of the signal from the j^{th} neuron of the input layer to the i^{th} neuron of the hidden layer, b_i^H denotes the bias of the i^{th} neuron of the hidden layer, and $\text{tansig}(\cdot)$ function has the form

$$\tan \text{sig}(x) = \frac{2}{1 + \exp(-2x)} - 1$$

The input and output of the neuron of the output layer of BPL3-V are defined to be

$$p^O = \sum_{j=1}^5 w_j^O y_j^H + b^O$$

$$\hat{V} = p^O$$

where w_j^O denotes the weight of the signal from the j^{th} neuron of the hidden layer to the neuron of the output layer, b^O denotes the bias of the neuron of the output layer.

Let the constant weights and biases of the network be as follow:

$$W^H = \begin{bmatrix} w_{11}^H & w_{12}^H & w_{13}^H \\ w_{21}^H & w_{22}^H & w_{23}^H \\ \vdots & \vdots & \vdots \\ w_{51}^H & w_{52}^H & w_{53}^H \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 2 & 1 & 2 \\ 1 & 2 & 2 \\ 1 & 1 & 1 \end{bmatrix}, \quad B^H = \begin{bmatrix} b_1^H \\ b_2^H \\ \vdots \\ b_5^H \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \\ -2 \end{bmatrix}$$

$$W^O = [w_1^O \quad w_2^O \quad \dots \quad w_5^O] = [1 \quad 1 \quad 2 \quad 2 \quad -3] \text{ and } b^O = 1$$

Suppose a set of 2 network inputs, $C_s(k)$, $C_{Z_n}(k)$, $D_c(k)$, and the targets $V(k)$, $k=1, 2$ are

$$C_s(1) = 1.5, C_{Z_n}(1) = 1.0, D_c(1) = 0.4, V(1) = 3$$

$$C_s(2) = 1.0, C_{Z_n}(2) = 1.5, D_c(2) = 0.6, V(2) = 5$$

The network performance function is defined as the mean square error between the network outputs and the targets, that is

$$E_v = \frac{1}{N} \sum_{k=1}^N [V(k) - \hat{V}(k)]^2$$

where N is the total number of samples used in training.

Find the set of network outputs, $\hat{V}(1)$, $\hat{V}(2)$ and the network performance function, E_v .

3. (RWN)

- a) Design an ANN that has as its output the maximum of n inputs.
- b) Design an ANN that has as its output the minimum of n inputs

4. (30 points) (RWN)

Create a network which takes a 7-vector input, u , where the components are the Boolean bits of a 7-bit Boolean number, and the output, y , is the parity of the sum of the bits, that is whether the sum is even or odd. This is a generalized XOR problem.

Show that this can be done by implementing

$$a_1(u) = \text{hard lim}((\sum_{i=1}^7 u_i) - \frac{7}{2})$$

$$a_2(u) = \text{hard lim}((\sum_{i=1}^7 u_i) - 4a_1(u) - \frac{3}{2})$$

$$y = a_3(u) = \text{hard lim}((\sum_{i=1}^7 u_i) - 4a_1(u) - 2a_2(u) - \frac{1}{2})$$

Using the ANN of these formulas draw the ANN and discuss what are the biases, weights, layers, and meaning of the binary number $a_1 a_2 a_3$. Test it on some sample inputs. What changes if the input vector is an n-vector of binary bits and the parity is desired?