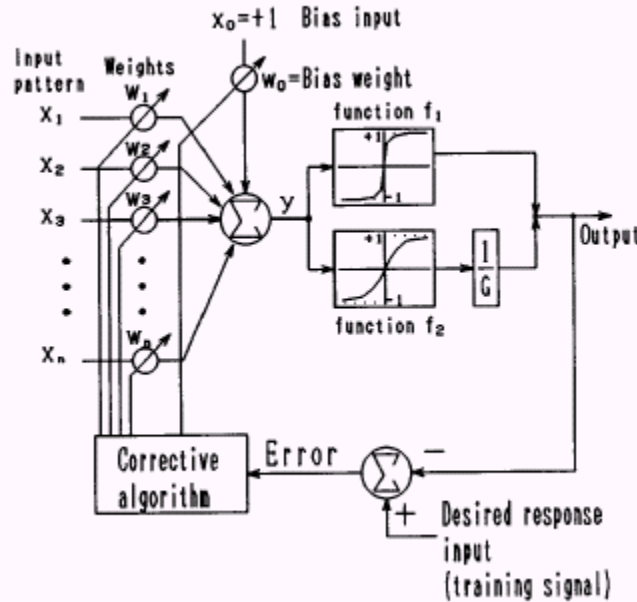


ENEE434 Spring 2003 Final Exam

95 points; 120 minutes, open book, open notes; if stuck go on to the next; good luck!

1. [35 points, 50 minutes]

Refer to “Rotation-invariant neural pattern recognition system with application to coin recognition” presented by A. Foroutan.



Consider the neural model in above figure. Let the input pattern be

$$\mathbf{X} = [x_0, x_1, x_2]^T = [1, 2, 1]^T$$

and the corresponding target is

$$t = 2.$$

The initial weights and biases are chosen to be

$$\mathbf{W}(0) = [w_0, w_1, w_2]^T = [-1, 1, 2]^T$$

where the bias weight w_0 is connected to a constant input $x_0 = 1$. The output of the neuron is

$$y = \mathbf{X}^T \mathbf{W}$$

and the output response of the network is

$$O(y) = f_1(y) + f_2(y)/G$$

The transfer functions adopted in the network are

$$f_i = \frac{2}{1 + \exp(-b_i * y)} - 1, \quad i = 1, 2$$

where the function parameters are chosen to be $G = 100$, $b_1 = 2$ and $b_2 = 0.5$.

The algorithm of adaptation is given by

$$W_{ij}(k+1) = W_{ij}(k) - \eta \delta_i x_j$$

where η is the step size and the quantity δ_i is defined as

$$\delta_i = O'_i(y) \times (t_i - O_i(y))$$

where $O'_i(y)$ is the derivative of $O_i(y)$ with respect to y .

a) Perform one iteration of this algorithm with step size = 1/3; show your calculations

b) What are the updated weights and bias, $\mathbf{W}(1)$?

2. [30 points, 30 minutes]

This problem is based upon ideas of the paper presented by Mei Lee, "Face Recognition with Radial Basis Function (RBF) Neural Networks."

The problem is to design an RBF type of neural network that determines whether a complex scalar, u , broken into real and imaginary parts, $u = u_r + ju_i$, has magnitude = 1 ($=|u| = \text{norm} = \text{distance from origin}$) within an error of 0.02; that is $0.98 \leq |u| \leq 1.02$. Recall the matlab functions:

`hardlim(.)` is the unit step function

$$\text{dist}(x - y) = \sqrt{\sum_i (x_i - y_i)^2} = \|x - y\| = \text{Euclidean distance}$$

$$\text{radbas}(n) = e^{-n^2}$$

a) In the input layer for input U use two inputs, the real and the imaginary parts of u , that is $U = [u_r, u_i]^T$, one `dist(.)` fed by weights of $1/\sigma$; in the hidden (RBF) layer use one `radbas(.)` function, with an input bias $-c/\sigma$, to form the network function n_h similar to that of the paper:

$$R(x, c, \sigma) = \exp(-(\text{dist}(x) - c)^2 / \sigma^2) \quad (=n_h \text{ when } x=U)$$

Evaluate $R(0, \sigma, \sigma)$.

b) Follow by an output layer with one `hardlim(.)` output function with a bias = $-R(0, \sigma, \sigma)$, that is $y = \text{hardlim}(n_h - R(0, \sigma, \sigma))$, for which the output y should be 1 if the magnitude of u is 1 within the prescribed error and 0 if not. Sketch the network using these matlab functions and appropriate labels (node signals, weights and biases).

c) Design the network by giving c and σ .

d) Prove (or disprove) that your network always gives the correct result.

3. [30 points, 30 minutes]

A neural network is constructed from n neurons each of which uses an activation function $f(x)$ which takes the value c for $x < c$ and d for $d < x$ and linearly connects these two regions for $c < x < d$.

a) sketch $f(x)$

b) writing $f(x) = \alpha + \beta \cdot \text{satlin}(\gamma \cdot x + \delta)$

rewrite this $f(x)$ in terms of c and d .

c) Assuming the neural network is described by

$$\frac{dx_i}{dt} = -x_i - \sum_{j=1}^n w_{ij} y_j \quad i = 1, \dots, n$$

$$y_j = f(x_j)$$

c1) Using vector-matrix notation $x = [x_i]$, $y = [y_j]$, $W = [w_{ij}]$ write the describing equations in matrix form.

c2) Using the Hopfield means of obtaining an energy function give a Lyapunov function for this neural network, assuming symmetric W , $W = W^T$.

c3) find the equilibrium points for this network and discuss their relationship to the Lyapunov function and their uniqueness.