Massachusetts Institute of Technology
Department of Mechanical Engineering

2.160 Identification, Estimation, and Learning
Spring 2006

Problem Set No. 6
Out: April 12, 2006 Due: April 19, 2006

Problem 1

A multi-layer neural network with two hidden layers is shown below. All the output functions in the hidden layers are logistic functions, while the one in the output layer is a linear function, i.e. $o_5 = net_5$. During the training of the network using the Error Back Propagation Algorithm, the following data were obtained when one of the sample pairs, input $x = 2$ and target $t = 2$, were presented:

$$W_{21} = 2, \quad W_{32} = 3, \quad W_{42} = -2,$$

$$W_{52} = 1, \quad W_{53} = 5, \quad W_{54} = 2,$$

$$f[net_2] = 0.5, \quad f[net_3] = 0.2, \quad f[net_4] = 0.75$$

Compute the weight change ΔW_{21} to be made for this sample presentation. Exclude momentum terms and use a learning rate of $\eta = 0.2$. Note that there is a direct connection between units 2 and 5 in the figure. Consider all the connections through which the weight change ΔW_{21} influences the squared error at the output.

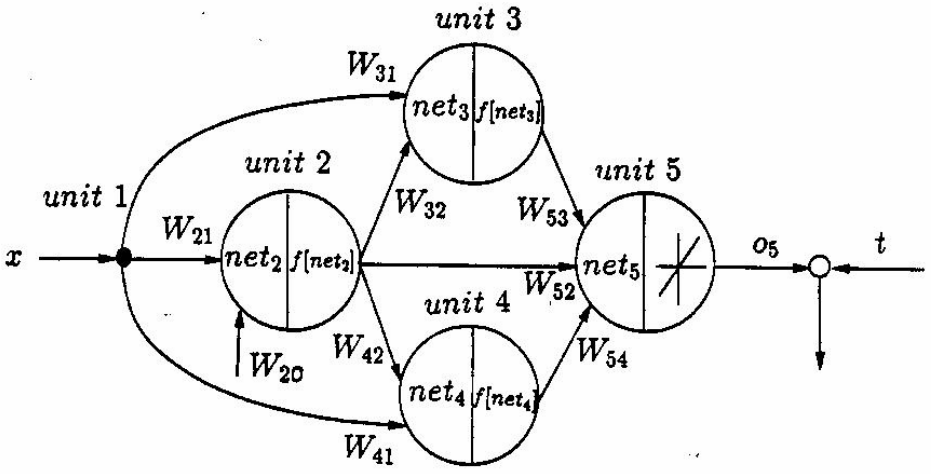


Figure 1: Multi-layer neural network with two hidden layers

Problem 2

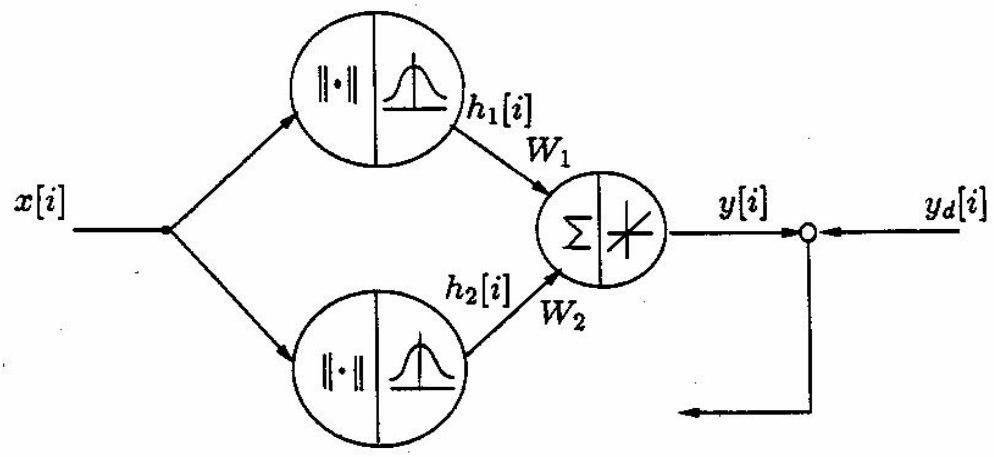
A simple radial basis function network with Gaussian functions is shown below. The center and width of each Gaussian function have been determined by using exemplary data. We want to train the network with respect to weights W_1 and W_2, based on the Recursive Least Square Algorithm. At the $(k-1)$-st iteration during the recursive training, the inverse gain matrix, $\mathbf{R}[k-1]^{-1}$, took the following values:

$$\mathbf{R}[k-1]^{-1} = \sum_{i=1}^{k-1} \mathbf{h}[i]\mathbf{h}[i]^T = \begin{bmatrix} 1.04, & -0.08 \\ -0.08, & 0.96 \end{bmatrix} \tag{1}$$

where $\mathbf{h}[i]$ is the 2×1 vector consisting of the outputs from the two radial basis functions given by

$$\mathbf{h}[i] = \begin{bmatrix} h_1[i] \\ h_2[i] \end{bmatrix} = \begin{bmatrix} exp(-\frac{|x[i]-\mu_1|^2}{\sigma_1^2}) \\ exp(-\frac{|x[i]-\mu_2|^2}{\sigma_2^2}) \end{bmatrix} \tag{2}$$

where $x[i]$ is the input presented at the i-th iteration, and μ_j and σ_j are, respectively, the center and width of the j-th radial basis function.



Figre 2: Radial-basis function network with Gaussian functions

Problem 3

An important step in training a radial-basis-function (RBF) network is to determine the center location and the dilation parameter of each radial basis function so that a limited number of RBF functions may effectively approximate a nonlinear map.

Shown below is an example of optimal allocation of RBF functions for voice data processing. Twenty RBF functions are placed optimally for covering approximately 300 data points in 2-dimensional input space. The dilation parameter, shown by the radius of each circle, is determined based on the variance of the data classified into the same RBF function.

A similar data set has been uploaded to the course site. You are requested to classify these data for the purpose of tuning a RBF network.

Image removed due to copyright reasons.

a). Implement the Generalized Lloyd Algorithm discussed in class for classifying N points of 2-dimensional input data into m clusters, i.e. m RBF functions. Download the web site data and test your program with the data. Set $m = 9$, create an equally-spaced 3-by-3 grid in the 2-dimensional space, and place the center points of the nine RBF functions initially at those grid points. After optimizing the center locations, compute the dilation parameter for each cluster. Plot the results in the same way as the above example.

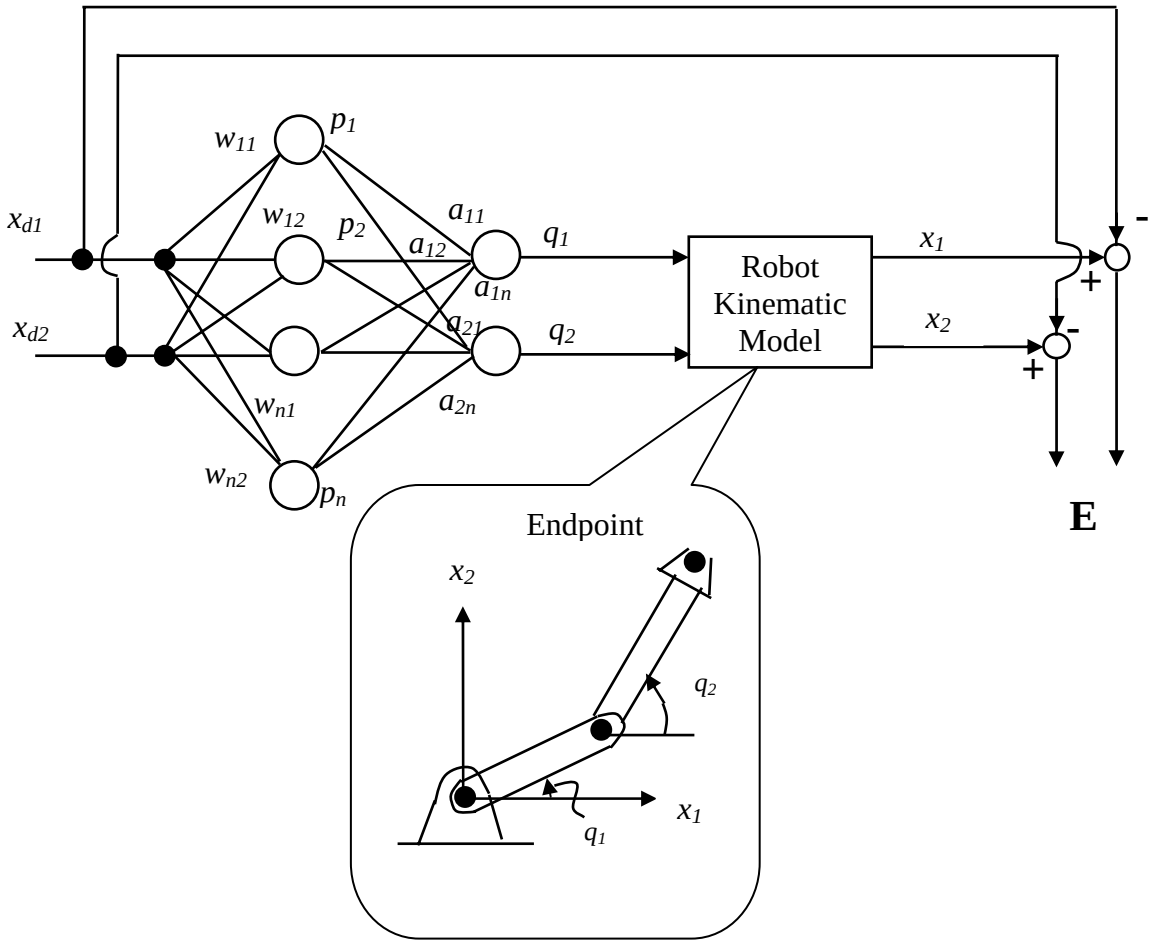
b). Using the Least Square Estimate algorithm, obtain the coordinates of the RBF functions to approximate the downloaded training data.

c). To evaluate the validity of the tuned RBF network, another set of data has been uploaded to the course site. These data are not used for training the RBF network but are used for evaluating the accuracy of the trained network. Using this data file, evaluate the mean squared error of the RBF network.

d). Repeat Parts a) through c) for $m = 25$. If time permits, try out a much larger number of RBF functions, say $m = 100$. Discuss pros and cons of using many RBF functions.

Problem 4

Consider a two degree-of-freedom robot, as shown in the figure below.



The joint angles are denoted q_1 and q_2, and the endpoint coordinates are x_1 and x_2. If the length of each link is 1, the endpoint coordinates are given in the following kinematic model.

$$x_1 = \cos q_1 + \cos q_2 \quad (1)$$

$$x_2 = \sin q_1 + \sin q_2 \quad (2)$$

Let us solve the above equations q_1 and q_2 by using a neutral network, instead of directly solving the algebraic equations. As shown in the figure below, desired endpoint coordinates x_{d1} and x_{d2} are presented to the network, which produced joint angles q_1 and q_2 as outputs. The joint angles q_1 and q_2 are inputted to the robot arm, resulted in actual endpoint coordinates x_1 and x_2. The actual position is then compared with the desired one, and the square error given by:

$$E = \frac{1}{2}\left[(x_1 - x_{d1})^2 + (x_2 - x_{d2})^2\right] \quad (3)$$

is fed back to the network to correct the weights by using the gradient descent method.

The final layer of the network consists of linear units, whose input output relationships are given by:

$$q_1 = a_{11}p_1 + a_{12}p_2 + \cdots + a_{1n}p_n + a_{10} \quad (4)$$

$$q_2 = a_{21}p_1 + a_{22}p_2 + \cdots + a_{2n}p_n + a_{20} \quad (5)$$

where a_{ij}'s are weights and p1 through pn are outputs from hidden layer, as shown in the figure. The network has one hidden layer consisting of all nonlinear units with logistic output functions. The input-output relationship described by:

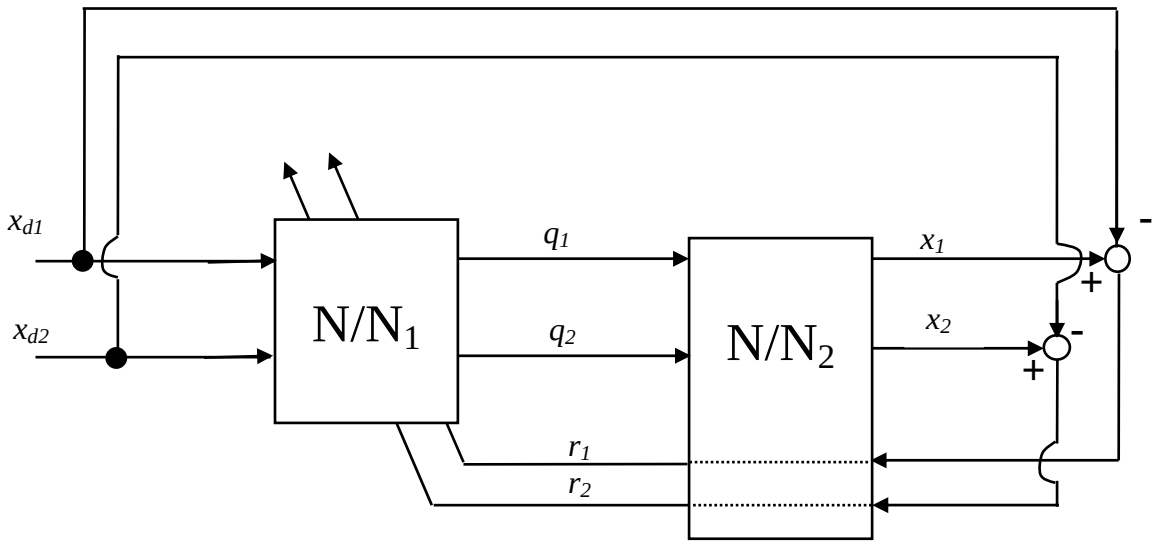
$$p_j = f[net_j] \quad (6)$$

$$net_j = W_{j1}x_1 + W_{j2}x_2 + W_{j3} \quad (7)$$

$$j=1,2,\ldots n.$$

When x_{d1}=1.2 and x_{d2}=0.9 were presented to the network, the network produced $q_1 = 0$ and $q_2 = \pi/2$. For this data presentation, we want to obtain the correction of weights, assuming that learning rate $\eta = 0.1$. Answer the following questions.

1. Compute the square error E.
2. Suppose that the outputs of the hidden units were $p_i = \bar{p}_i$, i=1,…,n. Obtain the weight changes in the final layer, Δa_{ij} by using eqs. (1) and (2).
3. The output of the j-th unit in the hidden layer was p_j=0.5, and the weights associated with the connections between the j-th hidden unit and the two linear units in the final layer were a_{1j}=2 and a_{2j}=3, respectively. Obtain the weight change to be made for W_{ji} of the hidden unit.
4. Instead of using the mathematical model given by eqs. (1) and (2) let us use another neutral network that represents the relationship between endpoint angles and endpoint coordinates. Namely, we first train the second network by presenting various joint angles as inputs and the corresponding endpoint coordinates as desired outputs. After training the second network, we then train the first network by back propagating square error *E* through the second network, as shown in the figure below. What is the physical meaning of the resultant signals, r_1 and r_2 obtained by back propagating the error through the second network, as shown in the figure? Explain how r_1 and r_2 are used for correcting the weights in the first network. Also, compute r_1 and r_2 for the case where x_{d1}=1.2 and x_{d2}=0.9 are presented under the same conditions as the previous questions.



5. Suppose the second network has an insufficient accuracy due to limited sample data for the off-line training. Let is consider a way of improving the second network together with the first network by using a sensor that measures the actual position of the robot endpoint, x_1 and x_2. Design an on-line learning system using the endpoint sensor that allows the second network to correct its imput-output relationship while the first network is learning the inverse relationship. Show a block diagram of the on-line learning system and explain clearly how the two networks are connected based on which signals. (Hint: Input the outputs of the first network to the real robot system, and measure the resultant endpoint position with the sensor).