

Name: _____

Date: October 30, 2002
Time: 5:00 – 7:00 PM

Mid Test

I. Give a brief answer to the following questions. (4pts each)

1. Why is it recommended to normalize/scale data before using the data for training?
2. We know that unsupervised learning involves no target values. What does it learn then?
3. Explain the Bayesian rule. What information do you need for using the Bayes rule in a classification problem?
4. Explain the universal approximation theorem and its application to multilayer perceptrons.
5. Is it possible to solve the logical XOR problem by a multi-layer perceptron that has one hidden layer with the activation functions for the hidden nodes and the output node being linear? Justify your answer. Would changing the activation function of the output node to a logistic sigmoid make a difference?
6. What are the differences between the linear least-squares (LLS) solution and the least mean square (LMS) algorithm?
7. Explain the effect of a momentum term in a gradient descent algorithm.
8. What is the difference between a feed-forward network and a recurrent network?
9. What is the difference between pattern and batch learning styles?
10. Explain the differences among the gradient descent, Newton, Gauss-Newton, and Levenberg-Marquardt methods.
11. What do we mean by generalization? What are the factors that influence generalization?
12. Why do we need to divide data samples into training (estimation), validation, and test datasets prior to training? What is the purpose of each subset?
13. List at least three data preprocessing techniques and explain the reason why we preprocess data using these techniques?
14. What are overfitting and underfitting and how can we prevent them?
15. What is curse of dimensionality?
16. Explain how a competitive learning works.

II. Problems

Problem 1 (10 pts): Consider the four samples shown in Table 1. Let $x(n) = [x_1(n) \ x_2(n)]^T$ be the input vector for the n th sample and $d(n)$ be the corresponding desired output.

n	$x_1(n)$	$x_2(n)$	$d(n)$
1	0	0	0
2	0.25	0.5	0
3	0.75	0.5	1
4	1	1	1

Table 1

- Design a single linear node with a bias to estimate the desired output using the *linear least-squares* (LLS) method. You don't need to solve it, just give the equations.
- Let the output of the node in (a) be y for a given input pattern x . The input vector x is assumed to belong to class "0" if the output $y < 0.5$; otherwise, x belongs to class "1". Draw one possible decision boundary on x_1 - x_2 plane.
- Replace the activation function of the node in (a) with a logistic function. Let the output of this node be y^* . The input vector x is assumed to belong to class "0" if the output $y^* < 0.5$; otherwise, x belongs to class "1". What kind of decision boundary would this result in? Draw one possible decision boundary on x_1 - x_2 plane.

Problem 2 (10pts): Data samples representing two classes ('+' and 'O') are shown in Figure 1.

- Are the two classes linearly separable?
- Sketch a neural network architecture that can be used to separate the two classes. The architecture should show the network's inputs, outputs, and nodes with their corresponding activation functions.
- Construct the decision boundary produced by the k-nearest neighbor rule (assume $k=1$) applied to this data sample.

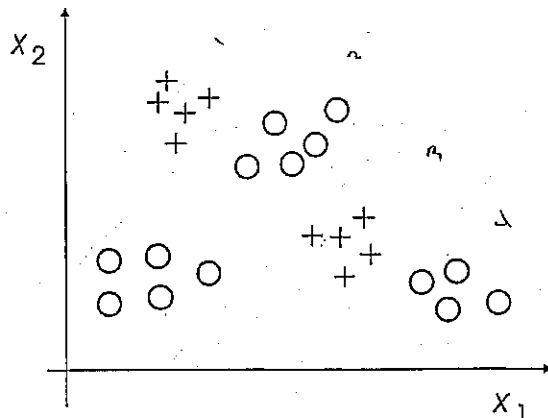


Figure 1

Problem 3 (16 pts): Consider the 2-2-1 neural network architecture shown in Figure 2. Let f and g be the activation functions for the two hidden nodes and the output node, respectively. Using back-propagation, write the weight update equation for each of the 6 weights shown in the figure. Note that $x_1(n)$ and $x_2(n)$ represent the n th sample inputs and $y(n)$ is the n th sample output. The cost function to be minimized is:

$$E(n) = \frac{1}{2} [e(n)]^2,$$

where $e(n) = d(n) - y(n)$, with $d(n)$ denoting the desired output for the n th sample.

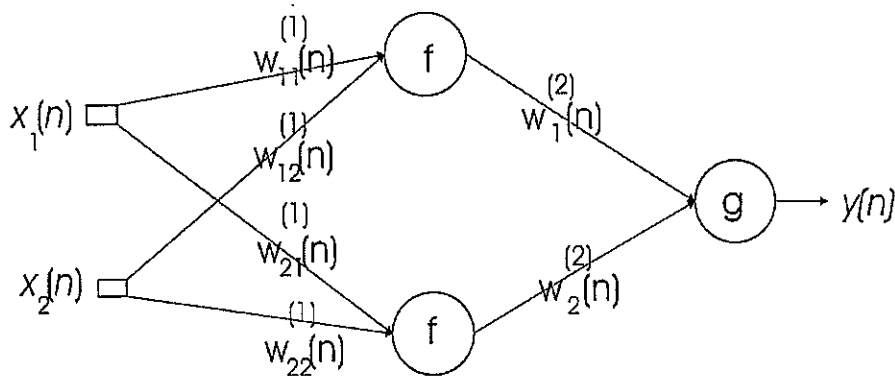


Figure 2

Exam 1

Answer the following questions by “yes” or “no” or short answers. (2 pts each)

1. According to R.P. Lippman, what is the maximum number of layers needed to solve any given problem using multi-layer network (don't include the input as a layer)?
2. Is the solution of a Back Propagation (BP) rule unique?
3. In a network designed to solve the logical XOR problem, activation function of the output node is selected to be linear. Does this selection place any limit on the training MSE?
4. Give two factors that might affect the weights in using a BP learning rule.
5. Give a suggestion for moving a BP process out of a local minimum.
6. Is the relationship between the input and output spaces in an RBF network linear?
7. Can you use a binary function in the BP learning rule? Why?
8. In training a network using BP rule, the weights are oscillating and the network has not achieved the required accuracy. What is the cause and what action do you recommend?
9. How do you know that your BP network is stuck in a local minimum?
10. You are using the Least Mean Square (LMS) algorithm for training of an RBF network. What would happen to the convergence of the network as you increase the variance of the Gaussian functions?
11. What is the difference between the linear least square (LLS) solution and the least mean square (LMS) algorithm?
12. Give two conditions under which the square Φ matrix of the Gaussian nodes in an exact RBF network might approach singularity.
13. You have a classification problem where some regions of the input space are populated with data and some regions are sparse. In selecting your centers for an RBF network, would you recommend a gridded center approach or a clustering algorithm?
14. What is the difference between a feed-forward network and a recurrent network?
15. What is the effect of momentum in the BP rule?
16. You are to approximate a “ $10\sin(x)$ ” function by a multi-layer network. What activation function would you select for the output node of the network?
17. Give three differences between a BP and RBF network.
18. What information do you need for using the Bayes rule in classification problems?
19. In solving for the weight of an RBF network, one encounters a singular matrix. What can be done to eliminate or bypass this problem?
20. Give the conditions under which the classification error rate of a multi-layer network trained with the BP rule approximates the *a posteriori* class probabilities.

Problem 1 (20 pts) – The following four patterns are given. The task is to use a single linear node and find the weights so that the error function defined below is minimized.

- Write the error function as a function of the network weights.
- Use the unconstrained optimization approach and find the matrix equation to be solved for the weights. Don't need to solve for the weights.
- Write the general form of the equation to be used by the Least Mean Square (LMS) algorithm.
- Write the Linear Least Square (LLS) solution. Don't need to solve.
- How would you compare the solutions of parts a), b), and c) above.

x_1	x_2	d
0	0	0
0.25	0.5	0
0.75	0.5	1
1	1	1

$$y = w_0 + w_1x_1 + w_2x_2$$

$$E = 1/2 \sum_{\text{all patterns}} (y - d)^2$$

Problem 2 (15 pts) - You are asked to design a Bayes classifier for a problem with two classes. Assume that there are 50 samples of each class without any bias towards any class, the costs for misclassification are equal, and that the densities of these samples can be identified by two Gaussian functions with means of $\mu_1=[0, 1]$ and $\mu_2=[2, 3]$. Also assume that both covariance matrices are diagonal and the variances are: $\sigma_1^2 = 2$ and $\sigma_2^2 = 4$. Find the Bayes decision boundary for classification of these patterns. Identify the shape of the decision boundary and found its parameters.

Problem 3 (15 pts) - Assume that the activation function of a multi-layer network is sigmoidal and the error function has the given form below.

$$y = \frac{1}{1 + \exp(-\Sigma)}$$

$$E = - \sum_{\text{patterns}} [(1 - d) \ln(1-y) + d \ln(y)]$$

$$\frac{\partial \ln y}{\partial y} = \frac{1}{y}$$

In the error equation d is the target or desired value and y is the actual output. Show that using the above error function, calculation of the derivative term in the delta (δ) function of the BP algorithm is eliminated. Hint: consider using the results of problem 1.1 of the text.

Problem 3 (10 pts) – You have designed a 2-2-1 multi-layer network with all linear nodes to separate the patterns of the XOR problem. Would you be able to provide a set of weights to separate the patterns using the network? What is the best set of weights that you can find? Would changing only the output node to a sigmoid function help?