

Instructions:

You may spend as much time working the problems as you require, but the exam is due at the beginning of class (3:35 p.m.) on Friday, April 29, 2005.

This take-home exam is open book and open notes. Any sources used, excluding course lectures, must be cited in your solutions. You may not consult any human being other than your instructor. Failure to comply with these guidelines will be viewed as violations of academic honesty, as described in *The Cat's Tail*.

Good Luck!

Problem 1

(For this problem you are encouraged to use Mathematica, Matlab, or a more general computer program. Please turn in your source code with your solutions.)

- (a) Consider the following dichotomy of 20 patterns in \mathbb{R}^3 .

$$X_{20} = \{((61, 4, 6)^T, 1), ((-83, 14, -60)^T, 1), ((40, 65, -95)^T, 1), ((-85, 55, 95)^T, -1), \\ ((-74, -99, -25)^T, 1), ((-43, 9, 22)^T, -1), ((70, 22, 34)^T, -1), ((59, -86, -34)^T, 1), \\ ((59, -37, -74)^T, 1), ((24, 23, 9)^T, -1), ((-52, -41, -17)^T, 1), ((-11, -57, 49)^T, -1), \\ ((-82, -68, 35)^T, 1), ((1, -51, -49)^T, 1), ((-58, -99, 88)^T, -1), ((73, -89, 72)^T, 1), \\ ((-39, 98, -40)^T, -1), ((-42, -91, -85)^T, 1), ((28, -100, 53)^T, 1), ((-38, -9, -96)^T, 1)\}$$

Is this dichotomy linearly separable. If your answer is *yes*, then construct an LTU that implements it. If your answer is *no*, please provide a proof.

- (b) Apply the pseudoinverse method to obtain the weight vector that minimizes the LMS error

$$E(\mathbf{w}) = \sum_{i=1}^m (\hat{\mathbf{x}}_i^T \hat{\mathbf{w}} - b_i)^2,$$

for the above training data. Please comment on how well this weight vector classifies the training data.

Problem 2

Let $d(m, n)$ denote the number of dichotomies of m patterns in general position in \mathbb{R}^n that are linearly separable, and let $P(m, n) \triangleq d(m, n)/2^m$.

Let $\epsilon > 0$. Prove the limit,

$$\lim_{n \rightarrow \infty} P\left(m = \frac{2(n+1)}{1-\epsilon}, n\right) = 0.$$

(Hint: Invoke the Lemma in the notes "Capacity of an LTU.")

Problem 3

A *quadratic threshold unit* (QTU) contains n inputs $\mathbf{x} = (x_1, x_2, \dots, x_n)$, and $n(n+1)/2 + n + 1$ weights ($n(n+1)/2$ weights of the form $w_{i,j}$, for $1 \leq i \leq j \leq n$; and $n + 1$, of the form w_i for $0 \leq i \leq n$), with an output y , given by

$$y = \text{sgn} \left(\sum_{i=1}^n \sum_{j=i}^n w_{i,j} x_i x_j + \sum_{i=1}^n w_i x_i + w_0 \right).$$

We will now explore how well this unit can implement a dichotomy of m patterns distributed at random in \mathbb{R}^n .

- Extend the definition of “general position” to this case.
- Assume that m points are in general position in \mathbb{R}^n , according to Part (a). Derive an expression for $q(m, n)$, the number of dichotomies (defined on these points) that can be implemented using a quadratic threshold unit. (*Hint*: This exercise is easier if you assume the previously derived expression for $d(m, n)$, the number of dichotomies of m points in \mathbb{R}^n that are linearly separable.)

Problem 4

Consider a feed-forward neural network with contains n_0 input units, $L-1$ hidden layers, with n_1, n_2, \dots, n_{L-1} units in each, and n_L output units. Assume that $L > 1$ and that the sigmoidal threshold function for every hidden and output unit is given by $\sigma(u) = u$. Show that this network is functionally equivalent to a feed-forward network without any hidden units, i.e., just n_0 inputs and n_L output units. (This exercise demonstrates the importance of nonlinear computational elements.)

Problem 5

A certain neural network contains n_0 real-valued inputs $\mathbf{x} = (x_1, x_2, \dots, x_{n_0})^T$, and n_2 outputs $\mathbf{y} = (y_1, y_2, \dots, y_{n_2})$ such that, for $1 \leq i \leq n_2$,

$$y_i = \sum_{j=1}^{n_1} w_{i,j} \phi \left(\beta_j^2 \|\mathbf{x} - \boldsymbol{\mu}_j\|^2 \right) + w_{i,0},$$

where $\boldsymbol{\mu}_j \in \mathbb{R}^{n_0}$ and $\beta_j \in \mathbb{R}$, for $j = 1 \dots, n_1$, play the role of first-layer weights, and

$$\phi(\rho) = \exp \left(-\frac{\rho}{2} \right),$$

plays the role of the hidden layer threshold function. Similarly $w_{i,j}$, for $1 \leq i \leq n_2$ and $0 \leq j \leq n_1$, play the role of second-layer weights.

Given a training set of m patterns,

$$\mathcal{X}_m = \{ (\mathbf{x}^{(1)}, \mathbf{t}^{(1)}), \dots, (\mathbf{x}^{(m)}, \mathbf{t}^{(m)}) \},$$

with $\mathbf{x}^{(p)} \in \mathbb{R}^{n_0}$ and $\mathbf{t}^{(p)} \in \mathbb{R}^{n_2}$, use the principle of gradient descent to obtain an *update rule* for the first and second layer weights, that enables the network to learn the training patterns. For what kinds of problems might this network architecture be suited?