ASSIGNMENT 2 Solutions

1 (400pts) Neural Networks and Backpropagation

(a) See code on web. Assumed that this is done.

(b) Number of adjustable parameters: $(d+2)N_H + 1$.

Functional Form:

$$g(\mathbf{x}, \mathbf{w}) = S\left(w_{01}^2 + \sum_{i=1}^{N_H} w_{i1}^2 \tanh\left(w_{0i}^1 + \sum_{j=1}^{d} w_{ji}^1 x_j\right)\right)$$

where $S(\cdot)$ could be either tanh or the identity function depending on which output activation function is chosen.

(c)

	(i) tanh backprop	(ii) tanh numerical	(i) identity backprop	(ii) identity numerical
$\frac{\partial E}{\partial w_{01}^1}$	-0.026706	-0.026704	-0.032247	-0.032245
$\frac{\partial E}{\partial w_{02}^1}$	-0.026706	-0.026704	-0.032247	-0.032245
$\frac{\partial E}{\partial w_{11}^1}$	-0.026706	-0.026704	-0.032247	-0.032245
$\frac{\partial E}{\partial w_{12}^1}$	-0.026706	-0.026704	-0.032247	-0.032245
$\frac{\partial E}{\partial w_{21}^1}$	-0.026706	-0.026704	-0.032247	-0.032245
$\frac{\partial E}{\partial w_{22}^1}$	-0.026706	-0.026704	-0.032247	-0.032245
$\frac{\partial E}{\partial w_{01}^2}$	-0.17906	-0.17904	-0.21621	-0.21618
$\frac{\partial E}{\partial w_{11}^2}$	-0.11373	-0.11372	-0.11373	-0.11372
$\frac{\partial E}{\partial w_{21}^2}$	-0.11373	-0.11372	-0.11373	-0.11372

2 (150pts) Training on some Data

(a) Down Load Data. How hard can that be?

(b) See graphs

(c) See graphs

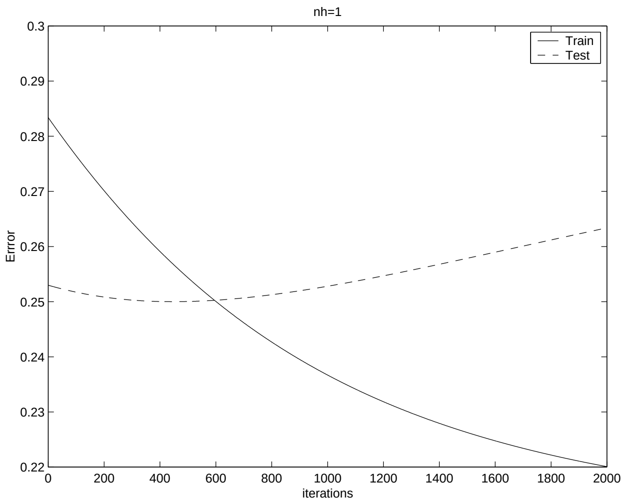
(d) See graphs

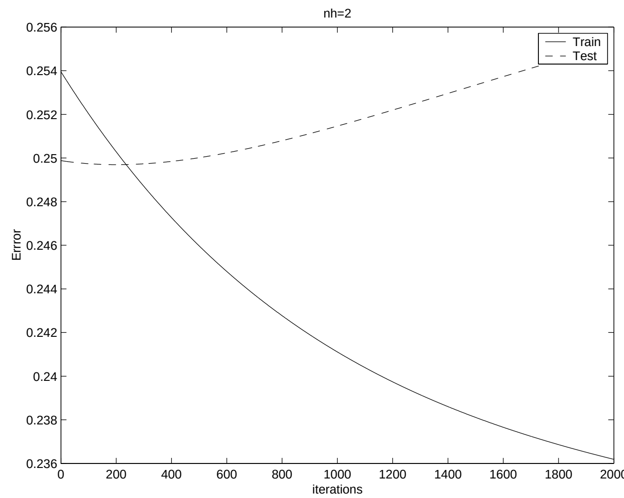
(e) i Training error always decreases.

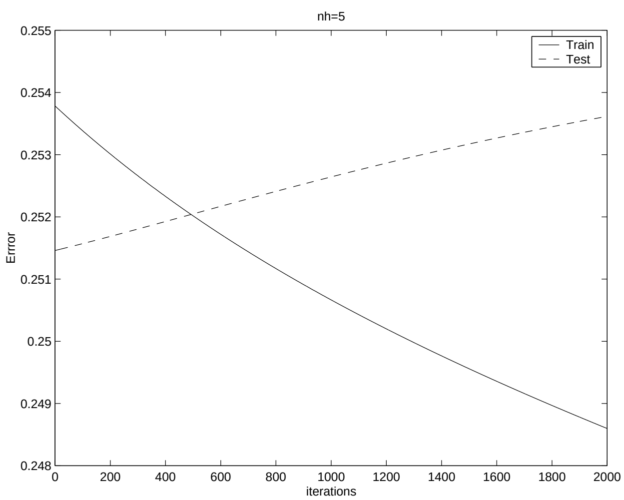
ii The larger the number of hidden units, the lower the training error gets. This is expected begause a more complex learning model will more likely fit the data better.

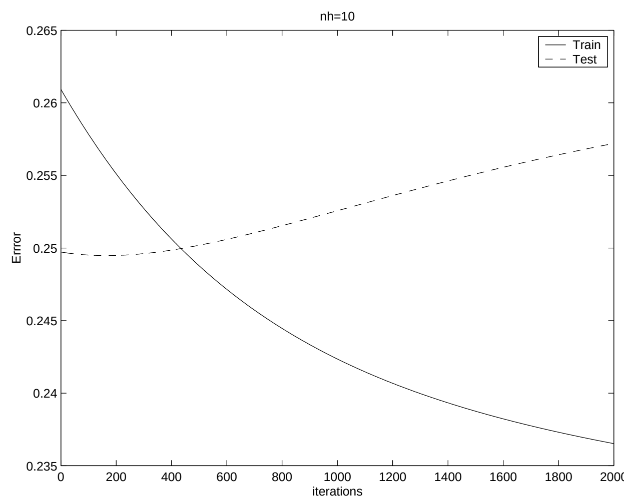
iii The test error becomes less correlated to the train error as the number of hidden units increases.

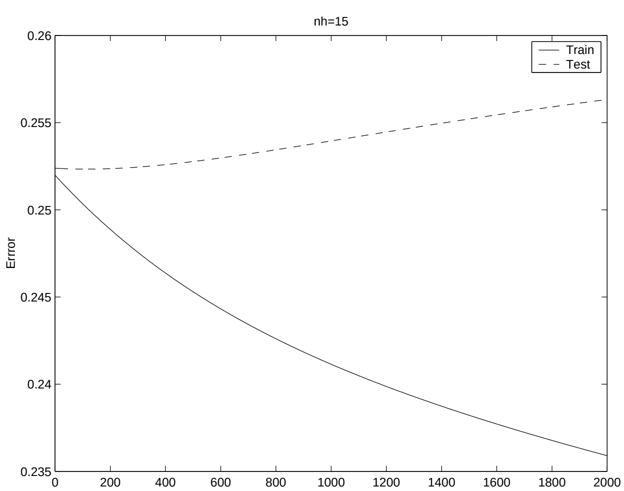
iv Usually, there appears to be an optimum number of iterations where the the test error is a minimum.

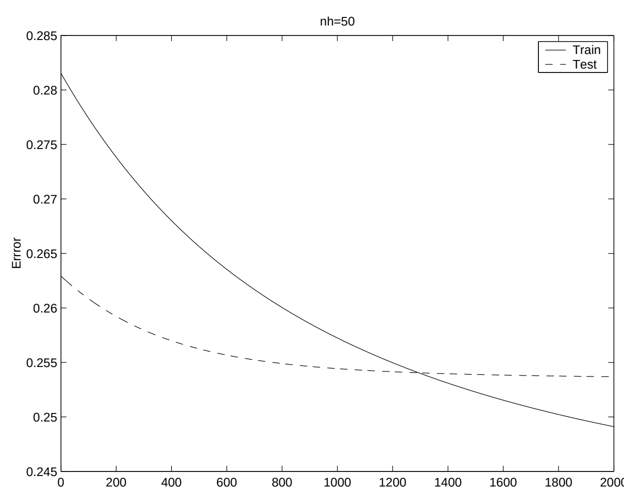






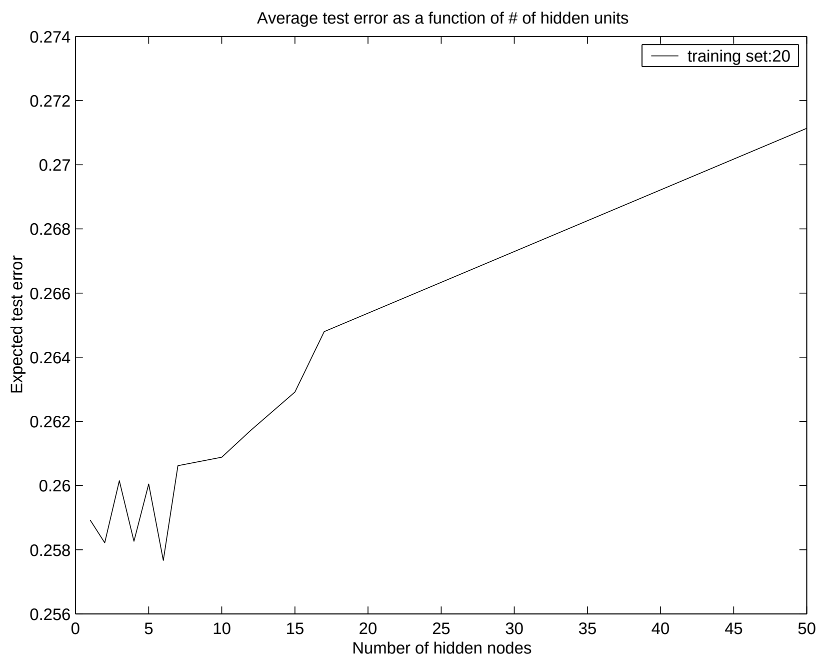






3 (150pts) Expected Test Error

(a)



(b) Each number represents the estimate of the expected/average test performance having trained for 2000 gradient descent iterations on 20 data points for the specified number of hidden units, N_H.

4 (150pts) Regression Problems

(a) For the discrete case, $R(\alpha_i|x) = \sum_j \lambda(\alpha_i, w_j)p(w_j|x)$. As the number of states becomes continuous, the sum converts to an integral,

$$R(y(x)|x) = \int dz\ \lambda(y(x), z)p(z|x)$$

(b) $\lambda(y, z) = (y - z)^2$. Bayes optimal risk tells us to pick the y that minimizes $R(y(x)|x) =$, i.e.,

$$\frac{d}{dy}R(y|x) = \frac{d}{dy}\int dz\ (y-z)^2 p(z|x) = 2\int dz\ (y-z)p(z|x) = 0$$

which gives

$$y\underbrace{\int dz\ p(z|x)}_{1} = \underbrace{\int dz\ zp(z|x)}_{E[z|x]}$$

giving $y(x) = E[z|x]$ as required. To test that this is actually a minimum, the second derivative test gives $\frac{d^2}{dy^2}R(y|x) = 2 > 0$.

5 (150pts) Boolean Univerasal Approximation

(a) We show how to implement them by specifying the weights of the perceptron.

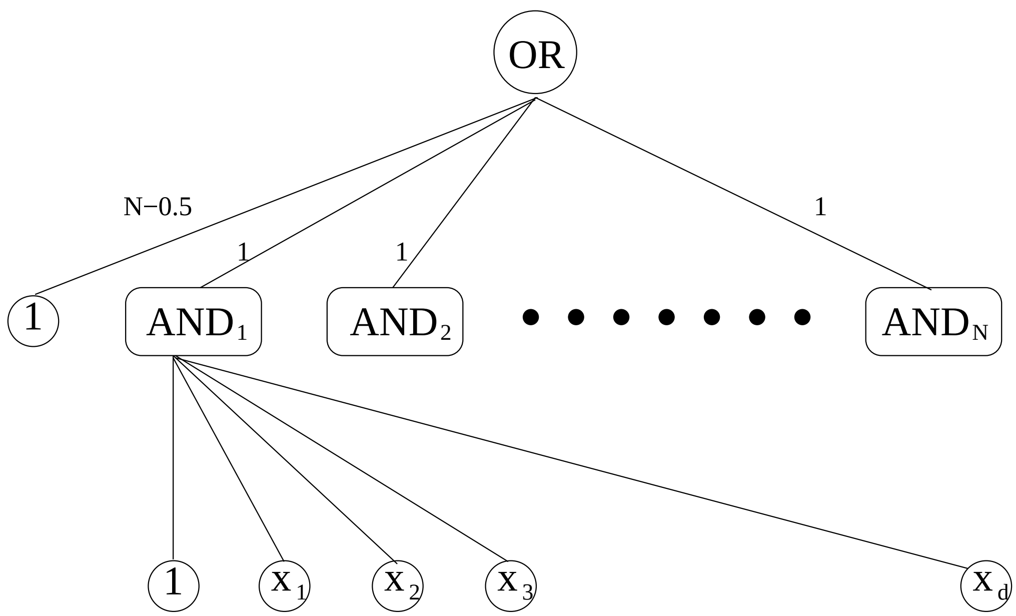
$AND(x_1, \ldots, x_d) : w_0 = -(d - \frac{1}{2}),\ w_i = 1$ for $i = 1, \ldots, d$.
$OR(x_1, \ldots, x_d) : w_0 = d - \frac{1}{2},\ w_i = 1$ for $i = 1, \ldots, d$.
$NOT(x_1) : w_0 = 0,\ w_1 = -1$ for $i = 1, \ldots, d$.

(b) **Theorem** *Any Boolean function can be implemented using a multilayer perceptron with all the activation functions being sign*$(\cdot)$

PROOF: We will use DNF though a similar proof with CNF also follows. Let the $f(\mathbf{x})$ be the boolean function and let its DNF representation be given by

$$f(\mathbf{x}) = OR(AND_1(y_1, \ldots, y_d), AND_2(y_1, \ldots, y_d), \ldots, AND_N(y_1, \ldots, y_d))$$

where each y_i is either x_i, $NOT(x_i)$ or 1. This function can be implemented by the following perceptron



Where the for each AND, if a particular $y_i = x_i$ then the connection weight is 1, if a particular $y_i = NOT(x_i)$, then the connection weight is -1 and if the $y_i = 1$ then the connection weight is 0 and the w_0 weight is incremented by 1 (so the final w_0 weight will be $-(d - \frac{1}{2})$ plus the number of y_i's that are 1). ■