Exam # 1, EE5353, Fall 2012

1. Here we consider MLPs with binary-valued inputs (0 or 1).
   (a) If the MLP has N inputs, what is the maximum degree D of its PBF model?
   (b) If the MLP has N inputs, what is the maximum value of L' in its PBF model?
   (c) If the MLP has N inputs, what is the maximum number of hidden units the network requires (to be complete, as in reference material)?
   (d) For a 6-input parity check network, how many hidden units are required, at most?

2. Here we model product networks for two and four inputs. Use only the necessary terms!
   (a) Give the exhaustive model for a two-input product network, which has two hidden units with the activation, net$^2$.
   (b) Give the exhaustive model for a four-input product network having two hidden layers. Assume that the activation function is net$^2$, and that the first hidden layer has four units, and the second hidden layer has three units.

3. Some compilers produce faster executables when matrix operations are used in the source code. In a FLN, let the rows of the data matrices $D_X$, $D_t$, and $D_y$ store $(X_p)^T$, $(t_p)^T$ and $(y_p)^T$ respectively, so that the data matrices’ dimensions are $N_v$ by $L$, $N_v$ by $M$, and $N_v$ by $M$. Assume that the MSE $E$ to be minimized is defined, as usual, as

$$E = \sum_{i=1}^M E(i),$$

$$E(i) = \frac{1}{N_v} \sum_{p=1}^{N_v} [(t_p(i) - y_p(i))^2]$$

   (a) If $y_p = W \cdot X_p$, write $D_y$ in terms of $D_X$ and $W$. (Hint: use the transpose operation)
   (b) We want to convert the $D_y$ equation of part (a) into our familiar equations $C = R \cdot W^T$. In order to do this, what do we pre-multiply the $D_y$ equation by?
   (c) Let $D_y(i)$ and $D_t(i)$ denote the ith columns respectively of $D_y$ and $D_t$. Write $E(i)$ in terms of $D_y(i)$, $D_t(i)$ and any other necessary symbols.
   (d) Replacing $D_y(i)$ and $D_t(i)$ in your $E(i)$ expression by $D_y$ and $D_t$ and using the trace operator ($\text{tr}(A) = a(1,1)+a(2,2)\ldots$), generate the expression for $E$. Is this calculation efficient?
4. Consider the Schmidt procedure, as applied to a neural network. Assume that the \( A \) (See appendix), \( R \), and \( C \) matrices are available for the training data. \( R \) and \( C \) are the usual correlation matrices.

(a) Give the orthonormal system’s weights \( w_o'(i,k) \) in terms of elements of \( A \), \( R \), and \( C \).

(b) Now give the original system’s output weights \( w_o(i,k) \) in terms of \( w_o'(i,k) \) and elements of \( A \).

(c) In the orthonormal system, suppose that \( X_1' \) through \( X_{N+1}' \) came from the inputs and the constant, 1, and that the remaining basis functions came from hidden units. Let \( y_i(k) \) denote \( y_i \) calculated from all inputs, the constant 1, and \( k \) of the hidden units. Give an expression for \( y_i(k) \) in terms of \( w_o'(i,m) \) and \( X_m' \).

(d) Given an orthonormal basis vector \( X' \), how many multiplies are needed to construct outputs \( y_i(k) \) for \( 0 \leq k \leq N_h \) ?

5. MLP number 1 has input vectors \( x_p \) of dimension \( (N+1) \), where \( x_p(N+1)=1 \). Additional parameters are \( w(k,n), w_{ob}(i,k), \) and \( w_{oi}(i,n) \). Let \( x_p \) be transformed as modeled as \( z_p = Ax_p \) where \( A \) is \( (N+1) \) by \( (N+1) \). For MLP no. 2, \( z_p \) is the input vector. MLP no. 2 has parameters \( w'(k,n), w_{ob}(i,k), \) and \( w_{oi}'(i,n) \).

(a) Given \( W \), find the \( N_h \) by 1 net function vector \( n_p \) in terms of \( W \) and \( x_p \).

(b) Assume that MLP no. 2 is equivalent to MLP no. 1 and has the same net function vectors (so \( n_p' = n_p \)). Find its input weight matrix \( W' \) in terms of \( W \) and \( A \).

(c) Let \( G \) be a negative gradient matrix whose elements are

\[
g(k,n) = \frac{-\partial E}{\partial w(k,n)} = \frac{1}{N_v} \sum_{p=1}^{N_v} \delta_p(k) x_p(n)
\]

so the \( N_h \) by \( (N+1) \) negative gradient matrix for \( W \) is

\[
G = \frac{1}{N_v} \sum_{p=1}^{N_v} \delta_p(x_p)^T
\]

where \( \delta_p \) is the \( N_h \) by 1 vector of hidden unit delta functions. \( G' \) is the negative gradient matrix for MLP no. 2, with elements \( g'(k,n) = -\partial E/\partial w'(k,n) \). Find \( G' \) in terms of \( G \) and \( A \). Remember that \( n_p' = n_p \) so \( \delta_p' = \delta_p \).

(d) Using your results from part (b), \( G' \) can be mapped back to MLP no. 1 and the resulting negative gradient matrix is \( G'' \), which can be used to train MLP 1. Express \( G'' \) in terms of \( G \) and \( A \).

(e) If \( G = G'' \), what condition should \( A \) satisfy?

(f) If \( E \) is minimized with respect to \( W \) in MLP no. 1, is \( E \) also minimized with respect to \( W' \) in MLP no. 2?
6. In two-stage OWO-BP training, we use the negative gradient matrix $G$ and the optimal learning factor (OLF) $z$ to modify $W$. In the multiple optimal learning factors (MOLF) algorithm, we use a different OLF $z_k$ for each hidden unit. After we’ve found $G$, our error function in terms of $z$ and $G$ is

$$E(z) = \frac{1}{N_v} \sum_{p=1}^{N_v} \sum_{i=1}^{M} [t_p(i) - y_p(i)]^2,$$

where the output, in terms of the OLFs is

$$y_p(i) = \sum_{n=1}^{N+1} w_{oi}(i,n)x_p(n) + \sum_{k=1}^{N_h} w_{oh}(i,k) f \left( \sum_{n=1}^{N+1} [w(k,n) + z_k \cdot g(k,n)] x_p(n) \right)$$

(a) Give $\frac{\partial y_p(i)}{\partial z_m}$ where the partial is evaluated for $z_k$s equal to 0.

Remember, $f(n_p(k)) = O_p(k)$.

(b) Give an expression for $g(m) = -\frac{\partial E(z)}{\partial z_m}$ in terms of the symbols $\frac{\partial y_p(i)}{\partial z_m}, t_p(i), y_p(i)$, etc. Note that $g(m)$ is an element of $g$.

(c) If Newton’s algorithm is used to find $z$ in a given iteration, the Hessian matrix elements are $h(u,v) = \frac{\partial^2 E}{\partial z_u \partial z_v}$. Give the Gauss-Newton expression for $h(u,v)$. What are the dimensions of $H$?

(d) Give the equations to be solved for $z$, in matrix vector form. What method can be used to solve these linear equations?
The variance of $X$ is

$$\sigma^2_X = E[X^2] - E[X]^2 = E[(X - E[X])^2]$$

An N-input complete network of degree D is one that has enough hidden units so that any D-degree polynomial function of N inputs can be formed by adding one output node and connecting weights to the existing hidden units. The weights for the new output can be found by solving linear equations. If a complete network's inputs and hidden units are linearly independent, the exhaustive PBF model of the network has a square $C$ matrix.

In a MLP with one hidden layer and linear output activations, the output and hidden unit deltas for the $p$th pattern are respectively

$$\delta_{po(i)} = -\frac{\partial E_p}{\partial net_{o}(i)}$$

$$= 2 \bullet (t_{pi} - y_{pi})$$

$$\delta_{p}(k) = f'(net_{pk}) \sum_{i=1}^{M} \delta_{po(i)w_{oh}(i,k)}$$

Then, $-\partial E_p/\partial w_{oh}(i,k)$ and $-\partial E_p/\partial w(k,n)$ are found as

$$-\frac{\partial E_p}{\partial w_{oh}(i,k)} = \delta_{po(i)} \bullet O_{pk}$$

$$-\frac{\partial E_p}{\partial w(k,n)} = \delta_{p}(k) \bullet x_{pn}$$

where the kth unit is an input or hidden unit and the nth unit is an input unit.
Some Equations from Schmidt Procedure

\[ y_i = \sum_{k=1}^{N_w} w'_o(i,k)X'_k \]

\[ y_i = \sum_{m=1}^{N_o} w_o(i,m)X_m \]

\[ X'_k = \sum_{m=1}^{k} a_{km}X_m \]

Newton’s Method

Assume all the weights you’re interested in training are stored in the vector \( w \), of dimension \( N_w \). Let \( g \) be the negative gradient vector (negative Jacobian) for the training error \( E \). For the Hessian matrix \( H \), the mth row, nth column element is

\[ h(m,n) = \frac{\partial^2 E}{\partial w(m) \partial w(n)} \]

Let \( e = w' - w \) be the unknown weight change vector, where \( w' \) is the new version of \( w \) that we’re trying to find. Then,

\[ H \cdot e = g \quad (3) \]

and we see that \( e = H^{-1} \cdot g \). The weight vector is then updated as

\[ w' = w + e \]