1. Four PLNs are to be fit to some training patterns \((x_p, t_p)\). The forward (F) network maps \(x\) to \(t\), and has a mean-squared error (MSE) \(E_F(i)\) for \(1 \leq i \leq M\). The reverse (R) network maps \(t\) to \(x\), and has a MSE of \(E_R(n)\) for \(1 \leq n \leq N\). Now assume that concatenated \((N+M)\)-dimensional vectors \((x_p : t_p)\) are formed. For each segment of the resulting space, we form a linear mapping from \(t\) to \(x\). The resulting RJ network \((J\) meaning joint\) has a MSE of \(E_{RJ}(n)\) for \(1 \leq n \leq N\) and uses knowledge of the correct input segment to improve the estimate of \(x\). Similarly, for each segment of the \((N+M)\)-dimensional space, we form a linear mapping from \(x\) to \(t\). The resulting FJ network has a MSE of \(E_{FJ}(i)\) for \(1 \leq i \leq M\).

(a) How do we determine if the input \(x(n)\) is ill-posed (poorly estimated using the desired outputs) ?

(b) How do we determine if the output \(y(i)\) is ill-posed (poorly estimated using the inputs) ?

(c) Suppose that we have \(N_1\) ill-posed inputs and \(M_1\) ill-posed outputs, where \(0 \leq N_1 \leq N\) and \(0 \leq M_1 \leq M\). What can we say about \(N_1\) and \(M_1\) if the overall mapping is \(OTO\) ?

(d) Continuing part (c), what can we say about \(N_1\) and \(M_1\) if the overall mapping is \(OTM\) ?

(e) Continuing part (c), what can we say about \(N_1\) and \(M_1\) if the overall mapping is \(MTO\) ?

(f) Continuing part (c), what can we say about \(N_1\) and \(M_1\) if the overall mapping is \(MTM\) ?

2. Consider the case of a nearest neighbor classifier (NNC) and PLN classifier (PLNC) that have the same number of clusters \(K\), the same cluster center vectors, and the same distance measure. The NNC satisfies the inequality: \(P_{eB} \leq P_{eNNC} \leq 2 \cdot P_{eB}\) where \(P_{eB}\) is the Bayes probability of classification error.

(a) Suppose that the \(k\)th NNC cluster belongs to class \(i_c\). If we want the PLNC’s \(k\)th cluster to always map patterns to class \(i_c\), how should the elements of the \(M\) by \((N+1)\) \(A_k\) matrix be chosen? Call this special \(A_k\) matrix \(A'_k\). Remember that in general, the PLNC’s output vector is \(y_p = A_k \cdot x_{ap}\) where \(x_{ap} = [x_p^T : 1]^T\).

(b) Suppose that a PLNC’s \(A_k\) matrices are designed normally by minimizing the MSE over the training data. For some clusters, the PLNC classifier could initially perform worse than the NNC. How do we guarantee that the PLNC is at least as good as the NNC, for the training data?

(c) Assuming that your design method in part (b) holds true as \(N_v\) approaches infinity, rewrite the inequality in the problem statement to include \(P_{ePLNC}\).
3. Given training patterns \((x_p, t_p)\), the piecewise linear network (PLN) is designed by (1) clustering the vectors \(x_p\) to produce \(K\) mean vectors \(m_k\) and (2) designing a different linear mapping for each cluster. The PLN’s output vector is \(y_p' = A_k \cdot x_{ap}\) where the distance \(d(x_p, m_k)\) is the smallest. Let \(d(x_p, m_k)\) denote the weighted distance between \(x_p\) and \(m_k\), expressed as

\[
d(x_p, m_k) = \sum_{n=1}^{N} w(n)[x_{p(n)} - m_k(n)]^2
\]

(a) What is the storage capacity of the PLN if we assume that each cluster has exactly the maximum number of patterns that can be memorized?
(b) Find \(P_{ab}/M\) in terms of \(N, K\), and \(M\).
(c) Suppose that we want our training algorithm to maximize the actual number of memorized patterns. If \(N_v >> C_{PLN}\), what is the actual number of memorized patterns if the patterns are evenly distributed among the \(K\) clusters?
(d) Continuing part (c), how should the clustering algorithm distribute the patterns among the clusters to maximize the actual pattern memorization?

4. A fully connected MLP has one hidden layer. Given the PLN from the previous problem, which has parameters \(N, K\), and \(M\), we want to find the MLP structure which is equivalent.
(a) Give the upper bound on \(C_{MLP}\), for a fully-connected MLP having \(N\) inputs, \(N_h\) hidden units, and \(M\) outputs. Equating \(C_{MLP}\) to \(C_{PLN}\) from problem 3(a), solve for \(N_h\), the number of MLP hidden units, in terms of \(K\) and other parameters.
(b) Give an expression for \(M_{MLP}\). Equating \(M_{PLN}\) for the network of problem 3 to \(M_{MLP}\), solve for the number of MLP hidden units.

5. Consider the forward counterpropagation network (CPN) in interpolative mode, with \(N\) inputs, \(K\) Kohonen units, \(M\) outputs, and normal cascade connectivity.
(a) If \(N=1\) and \(x\) can be positive or negative, but not 0, what is the storage capacity \(C_{CPN}\) for this network?
(b) What change can we make to the input normalization and distance measure to improve \(C_{CPN}\) for the CPN of part (a)?
(c) Besides increasing \(K\), what changes can we make to the network structure that are guaranteed to increase \(C_{CPN}\) for the CPN of part (b)?
(d) For what values of \(K_{on}\) does the CPN of part (c) produce a continuous mapping from input to output?
(e) Give the name of a training algorithm that, when applied to the CPN of part (d), can produce a network with \(C_{CPN}\) close to its upper bound.
6. We want to use BP to train a radial basis function (RBF) network with standard cascade connectivity and no thresholds. Its weighted distance measure, activation output, and error function are

\[ d(x_p, m_k) = \sum_{n=1}^{N} w(n) \left[ x_p(n) - m_k(n) \right]^2, \]

\[ O_p(k) = e^{-\beta_d(x_p, m_k)}, \]

\[ y_p(i) = \sum_{k=1}^{N_h} w_o(i, k) O_p(k) \]

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

(a) Give a two-line expression for \( \frac{\partial E}{\partial \beta_m} \) in the form

\[ \frac{\partial E}{\partial \beta_m} = \frac{?}{N_v} \sum_{p=1}^{N_v} \sum_{i=1}^{M} [t_p(i) - y_p(i)] \frac{\partial ?}{\partial ?}, \]

\[ \frac{\partial ?}{\partial ?} = ? \]

(b) Suppose that Newton’s algorithm is to be used to train the \( N_h \) by 1 vector \( \beta \).

Give the Hessian matrix element \( h(m, k) \) in Gauss-Newton form where

\[ h(m, k) = \frac{\partial^2 E}{\partial \beta_m \partial \beta_k} \]

Remember, the Gauss-Newton approximation is

\[ \frac{\partial^2 f^2(x, y)}{\partial x \partial y} = 2 \frac{\partial f(x, y)}{\partial x} \cdot \frac{\partial f(x, y)}{\partial y} \]