Neural Networks & Pattern Recognition

Problems

Problems 1

1. **(a)** State Bayes' theorem for the belief in classification into class $\mathcal{C}_i$ when information vector $\mathbf{x}$ is observed and explain the significance of each of the terms in the expression.

   (b) If $\mathbf{x}$ represents a variable drawn from a data set whose probability density function (PDF) is $\mathcal{P}(\mathbf{x})$, show that an estimate of this PDF may be written as

   $$\hat{\mathcal{P}}(\mathbf{x}) = \frac{k}{NV(\mathbf{x})}$$

   where N is the total number of data points in the data set, $V(\mathbf{x})$ is a volume centred on $\mathbf{x}$ and k is the number of data points located within $V(\mathbf{x})$. Discuss the validity of any assumptions made.

   (c) **(i)** Using the result of part (b), show that

   $$\hat{P}(\mathcal{C}_i \mid \mathbf{x}) = \frac{k_i}{k}$$

   where k_i is the number of data points of class $\mathcal{C}_i$ within $V(\mathbf{x})$ and k is the total number of data points within $V(\mathbf{x})$.

   (ii) Using pseudo-code if appropriate, describe the operation of the k-nearest-neighbour classifier.

   (d) What advantages and disadvantages might such a classifier have over parametric approaches, such as a Gaussian mixture classifier?

2. **(a)** Show that a classifier which operates by assigning an unknown input, $\mathbf{x}$, to the class with the largest *a posteriori* probability represents a minimum risk system if the penalty for misclassification is equal for each class.

(b) A more complex risk (loss) matrix, $\mathbf{L}$, is introduced such that L_{jk} represents the penalty for misclassification to class C_j when the pattern in fact belongs to class C_k. Under what condition should classification of $\mathbf{x}$ to class C_j be made if the total loss is to be minimised?

(c) What is meant by the *reject option* of a classifier and why is it important?

(e) Discuss briefly the advantages and disadvantages of *logistic* over *linear* regression.

3. **(a)** An automated pattern classification system is to be routinely used in the safety-critical application of aircraft engine checking. If the system is to provide a classification into two classes, airworthy and non-airworthy, detail the important aspects of the classifier design. You should pay particular attention to the issues of reject (doubt) and outlier rejection and the receiver-operator characteristic (ROC) curve.

(b) The output of a classifier designed to assign an unknown pattern x to classes C_1 or C_2 is denoted y. The target coding of the labelled training set consists of $t = 1$ if $x \in C_1$ and $t = 0$ if $x \in C_2$. If the target probability density is Bernoulli of the form

$$P(t \mid x) = y^t(1 - y)^{1-t}$$

prove that, if the training set is large, the output $y(x)$ which minimises the error function $E = -\ln P(t \mid x)$ is $y(x) = P(C_1 \mid x)$, the posterior probability of C_1 given x.

(c) What relative advantages and disadvantages might a K-nearest neighbour classifier and a logistic discriminator have in an analysis based on training sets that are (i) very large and (ii) very small?

4. **(a)** Briefly discuss the advantages and disadvantages of *gradient-descent* and *quasi-Newton* optimisation methods.

(b) The *Expectation-Maximisation* (EM) algorithm is often used to fit a simple Gaussian mixture model to a data distribution for use in a Radial Basis Function (RBF) neural network. The data is $\{\mathbf{x}[n]\}, n = 1...N$

(i) Show that an estimator for the mean of the k-th Gaussian, using the entire data set, may be given by:

$$\hat{\mathbf{m}}_k = \frac{\sum_{n=1}^{N} \mathbf{x}[n] P(k \mid \mathbf{x}[n])}{\sum_{n=1}^{N} P(k \mid \mathbf{x}[n])}$$

(ii) It is proposed that, instead of re-estimating this mean using the entire data set, a *stochastic* (sample-by-sample) algorithm be used. Show that this takes the form of an iterative update equation given by

$$\hat{\mathbf{m}}_k[n] = \hat{\mathbf{m}}_k[n-1] + \alpha[n](\mathbf{x}[n] - \hat{\mathbf{m}}[n-1])$$

where the adaption rate, $\alpha[n]$, is a function of posterior probabilities which you should determine.

(c) Explain the difference between the *K-means* and EM algorithms for estimation of the centre locations of a set of Gaussians.

5. (2000 paper) A two-class classification problem is to be tackled using a *linear classifier* such that the class discriminant measure, y, to datum $\mathbf{x}$ is given as

$$y(\mathbf{x}) = g(\mathbf{w}^{\mathsf{T}}\mathbf{x} + w_{bias})$$

in which $\mathbf{w}$ is a vector of weights, w_{bias} is a bias parameter and

$$g(a) = \frac{1}{1 + \exp(-a)}$$

is the *logistic sigmoid* function. A training set is made available.

(a) The measure $y(\mathbf{x})$ is to be an estimate of the posterior probability of class 1 given datum $\mathbf{x}$. Assuming the two data classes to be modelled by Gaussian (normal) distributions with different means ($\mathbf{m}_1$ and $\mathbf{m}_2$) but a common covariance matrix, $\mathbf{C}$, determine expressions for $\mathbf{w}$ and w_{bias} in terms of the means, covariance and the numbers of examples of each class (n_1 and n_2) present in the training set.

(b) **(i)** If $\mathbf{C} = \sigma^2\mathbf{I}$ where $\mathbf{I}$ is the identity matrix, show that the variance parameter σ^2 acts to *moderate* the probabilities estimated by the classifier. Of what use might this be?

(ii) How might σ^2 be set to an appropriate value given a large training data set?

(c) Explain briefly how regularisation arise naturally as part of a Bayesian approach to data analysis.

(d) What are the major limitations of the linear classifier? How may linear classifiers be used in *hierarchical* classifiers to overcome these limitations?

Problems 2

1. **(a)** **(i)** Explain briefly the concept of *generalisation* and why it is of importance in neural network analysis.

   (ii) Detail the factors which may affect the generalisation of a neural network system.

   (b) **(i)** Explain what is meant by *regularisation* and why is it important to the issue of generalisation.

   (ii) A neural network with parameters (weights) $\mathbf{w} = \{w_i\}$ is optimised (trained) so as to minimise a regularised error functional of the form

   $$E_{reg} = E_{data} + \frac{\alpha}{2}\mathbf{w}^{\mathsf{T}}\mathbf{w}.$$

   If $\mathbf{w}^*$ is the location of the global minimum of E_{data} and $\mathbf{w}^+$ the location of the global minimum of E_{reg} show that

   $$w_i^+ = \frac{\lambda_i}{\lambda_i + \alpha} w_i^*$$

   where λ_i is the i-th eigenvalue of the *Hessian* matrix of E_{data} evaluated at $\mathbf{w}^*$. You may assume that at minima the error functional is well approximated by a quadratic expansion.

   (c) In an on-line application using the regularised system of part (b), a simple gradient-descent scheme is used to iteratively update $\mathbf{w}$ at each time step t according to

   $$\mathbf{w}[t+1] = \mathbf{w}[t] - \eta \nabla E[t]$$

   where ∇E is the gradient of E_{reg} with respect to $\mathbf{w}$. Show that, for convergence in the mean, the adaption parameter η must satisfy

   $$0 < \eta < \frac{2}{\lambda_{max} + \alpha}$$

   where λ_{max} is the largest eigenvalue of the Hessian matrix of the *unregularised* error function and α is the regularisation constant. You may assume that in the region of convergence a quadratic approximation to the error surface is valid.

2. **(a)** **(i)** Explain briefly how regularisation arises naturally in a Bayesian framework via the concept of a prior distribution over parameters (weights) in an analysis system.

(ii) Show that the choice of a Gaussian prior over parameters leads to a *weight-decay* regulariser in which the error functional is of the form

$$E(\mathbf{w}, \alpha) = E_{data} + \frac{\alpha}{2}\mathbf{w}^{\mathsf{T}}\mathbf{w}$$

where $\mathbf{w}$ is the vector of parameters and α is a scalar constant.

(b) An analyser with parameters $\mathbf{w}$ is trained using a sum-of-squares error functional on a data regression problem. The probability density over the output target, t, conditioned on the training data set D, is given in a Bayesian approach by the marginal integral

$$P(t \mid D) = \int P(t \mid D, \mathbf{w}) P(\mathbf{w} \mid D) d\mathbf{w}.$$

(i) Show that, if y is the analyser output associated with target t, then

$$P(t \mid D) \propto \int \exp\left\{-\frac{\beta}{2}(y - t)^2 - E(\mathbf{w})\right\} d\mathbf{w}$$

where β is a scalar constant.

(ii) By expanding the error function about the point of minimum error, $\mathbf{w} = \mathbf{w}^*$, as a second-order Taylor series and expanding y to first order, show that

$$P(t \mid D) \propto \exp\left\{-\frac{\beta}{2}(y(\mathbf{w}^*) + \mathbf{g}^{\mathsf{T}}\Delta\mathbf{w} - t)^2 - \frac{1}{2}\Delta\mathbf{w}^{\mathsf{T}}\mathbf{H}\Delta\mathbf{w}\right\} d\mathbf{w}$$

in which $\mathbf{g} = \frac{\partial y}{\partial \mathbf{w}}$ evaluated at $\mathbf{w}^*$, $\mathbf{H}$ is the Hessian matrix of the error function and $\Delta\mathbf{w} = \mathbf{w} - \mathbf{w}^*$. Verify that this gives rise to a solution in which $P(t \mid D)$ is normally distributed with mean $y(\mathbf{w}^*)$. You may take the variance of $P(t \mid D)$ to be

$$\sigma^2 = \frac{1}{\beta} + \mathbf{g}^{\mathsf{T}}\mathbf{H}^{-1}\mathbf{g}.$$

(iii) A linear system, $y = \mathbf{w}^{\mathsf{T}}\mathbf{x}$ is trained using an unregularised least-squares error function. Using the results from part (ii), show that the variance in output y associated with pattern $\mathbf{x}$ is proportional to

$$\frac{1}{N}\mathbf{x}^{\mathsf{T}}\mathbf{C}^{-1}\mathbf{x}$$

where $\mathbf{C}$ is the sample covariance matrix of the patterns in the data set $\{\mathbf{x}_n\}, n = 1 \ldots N$.

3. **(a)** Explain briefly the differences and similarities between Radial Basis Function (RBF) and Multi-layer Perceptron (MLP) pattern analysers.

(b) A Gaussian mixture model (GMM) with K kernels is to be used to estimate the probability density function (pdf) of a data set $X = \{x_n\}, n = 1...N$.

(i) The Expectation-Maximisation (EM) algorithm is an iterative procedure for efficiently providing maximum-likelihood estimators for the free parameters of the Gaussian kernels in the mixture. By considering a change in parameters from 'old' to 'new' values, show that at each such change a functional Q must be minimised where Q is given as

$$Q = -\sum_{n=1}^{N}\sum_{k=1}^{K} P_{old}(k \mid x_n) \ln \{P_{new}(k) P_{new}(x_n \mid k)\} .$$

Hint: You may require the result of *Jensen's inequality*:

$$\ln \sum_k \lambda_k v_k \geq \sum_k \lambda_k \ln v_k$$

for any v_k given that $\lambda_k \geq 0$ and $\sum_k \lambda_k = 1$.

(ii) By considering the differential of Q with respect to μ_k, the mean of the k-th Gaussian in the mixture, show that the 'new' estimate for μ_k is given by

$$\hat{\mu}_k = \frac{\sum_{n=1}^{N} P_{old}(k \mid x_n) x_n}{\sum_{n=1}^{N} P_{old}(k \mid x_n)}.$$

(iii) Confirm that the same form of equation is obtained by direct differentiation of the data log-likelihood function with respect to μ_k.

(c) Discuss briefly the problems associated with using the EM algorithm to adapt a Gaussian mixture model. How might these problems be overcome?

4. (2000 paper - with slight modification)

(a) Describe the major approaches to training *radial basis function* (RBF) neural networks.

(b) **(i)** A radial basis function network is to be used for a regression problem. A training data set of input-output pairs, $(\mathbf{x}[n], t[n]), \quad n = 1...N$ is available. If the hidden-layer functions are chosen to be Gaussian, what are the appropriate error functions for *supervised* and *unsupervised* training of the Gaussian basis parameters?

(ii) How may matrix *pseudo-inverse* methods be used to determine the weights coupling the hidden-layer Gaussian functions to the output node of the network?

(c) Discuss briefly the similarities and differences between a Gaussian-basis RBF network and the *self-organising map* (SOM)

(d) The output nodes (grid, or map, locations) on the SOM are to be allocated class labels using a labelled training data set so that the data can be simultaneously visualised and classified. If each node is uniquely associated with only one class show that the resultant decision boundaries in the data ($\mathbf{x}$) space are piece-wise linear. How might you obtain smoother decision boundaries?