Engineering Tripos Part IIB FOURTH YEAR

Paper 4F10: Statistical Pattern Processing

STATISTICAL PATTERN RECOGNITION

Examples Paper 2

Straightforward questions are marked †
Tripos standard (but not necessarily Tripos length) questions are marked ∗

Deep Learning

1. † A multi-layer perceptron (feed-forward, fully connected, neural network) consists of d inputs, L hidden layers with M hidden units in each hidden layer, and K output nodes. Write down an expression for the total number of weights (including biases) in the network. Describe the factors that influence the number of hidden layers, the activation functions on the output layer, and the number of hidden units.

2. † For the logistic regression function, $\phi(z)$, show that

$$\frac{\partial}{\partial z}\phi(z) = \phi(z)(1 - \phi(z))$$

How does the nature of the activation function affect the computational cost of the error-back propagation algorithm?

3. † A *leaky ReLU* activation function is to be used in a multi-layer perceptron. This activation function has the form

$$\phi(z_i) = \left\{ \begin{array}{ll} z_i; & z_i \geq 0; \\ \alpha z_i & z_i < 0 \end{array} \right.$$

A large number of samples, generated from a Gaussian distribution with zero mean and a variance of σ^2, are passed through this activation function. What is the variance of the data at the output of the activation function?

How could this information be used when initialising the network with N nodes per layer?

4. Consider the optimisation of a set of weights where the magnitude of the gradient of the error function with respect to the weight space is approximately constant. The following update rule is used

$$\mathbf{w}[\tau + 1] = \mathbf{w}[\tau] + \boldsymbol{\Delta}\mathbf{w}[\tau]$$

where

$$\boldsymbol{\Delta}\mathbf{w}[\tau] = -\eta \left. \boldsymbol{\nabla} E(\mathbf{w}) \right|_{\mathbf{w}[\tau]} + \alpha \boldsymbol{\Delta}\mathbf{w}[\tau - 1]$$

(a) If the sign as well as the magnitude of the gradient is approximately constant, show that the effect of the momentum term is to increase the effective learning rate from η to $\frac{\eta}{1-\alpha}$.

(b) What is the effective learning rate for a region where the gradient descent scheme is oscillating about the real solution?

5. $*$ The Hessian is a useful matrix for use in the optimisation of the weights of multi-layer perceptrons.

(a) Describe how the Hessian may be used for optimising the weights of a multi-layer perceptron. Discuss the limitations for the practical implementation of such schemes.

(b) For the least squares error function

$$E = \sum_{p=1}^{n} E^{(p)} = \frac{1}{2}\sum_{p=1}^{n}(y(x_p) - t(x_p))^2$$

show that the elements of the Hessian matrix can be expressed as

$$\frac{\partial^2 E}{\partial w_{ij}\partial w_{lk}} = \sum_{p=1}^{n} \frac{\partial y(x_p)}{\partial w_{ij}}\frac{\partial y(x_p)}{\partial w_{lk}} + \sum_{p=1}^{n}(y(x_p) - t(x_p))\frac{\partial^2 y(x_p)}{\partial w_{ij}\partial w_{lk}}$$

For the case of well trained, sufficiently powerful, network, with an infinitely large training set, show that at the minimum the second term may be ignored. This is called the *outer-product* approximation.

(c) The Hessian after the N^{th} data point is approximated by

$$\mathbf{H}_N = \sum_{p=1}^{N} \mathbf{g}^{(p)}(\mathbf{g}^{(p)})'$$

where

$$\mathbf{g}^{(p)} = \boldsymbol{\nabla} y(x_p)|_{\mathbf{w}[\tau]}$$

By using the equality

$$(\mathbf{A} + \mathbf{BC})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{I} + \mathbf{CA}^{-1}\mathbf{B})^{-1}\mathbf{CA}^{-1}$$

where $\mathbf{I}$ is the identity matrix, show that

$$\mathbf{H}_{N+1}^{-1} = \mathbf{H}_N^{-1} - \frac{\mathbf{H}_N^{-1}\mathbf{g}^{(N+1)}(\mathbf{g}^{(N+1)})'\mathbf{H}_N^{-1}}{1 + (\mathbf{g}^{(N+1)})'\mathbf{H}_N^{-1}\mathbf{g}^{(N+1)}}$$

Why is this a useful approximation to estimate the inverse Hessian during multi-layer perceptron training.

Support Vector Machines

6. † A binary classifier is to be trained. What are the limitations of linear decision classifiers and why do non-linear mappings of the feature space allow improved discrimination? Under what conditions is it guaranteed that a non-linear mapping will allow perfect classification of the data?

7. For the `XOR` problem described in lecture notes show that the solution given satisfies the training conditions given. What is the equation of the final decision boundary?

8. The following data is to be used for training an SVM

$$\begin{array}{lccc} \omega_1: & \begin{bmatrix}1\\1\end{bmatrix} & \begin{bmatrix}2\\2\end{bmatrix} & \begin{bmatrix}2\\0\end{bmatrix} \\ \omega_2: & \begin{bmatrix}0\\0\end{bmatrix} & \begin{bmatrix}1\\0\end{bmatrix} & \begin{bmatrix}0\\1\end{bmatrix} \end{array}$$

   (a) Plot the training points and, by inspection, determine the position of the optimal, maximum margin, decision boundary.

   (b) What are the support vectors?

   (c) Express the decision boundary in terms of the Lagrange multipliers, α_i and show that this satisfies the KKT conditions.

Classification and Regression Trees

9. A tree classifier is to be built for a one-dimensional two category problem. A large number of training samples are available. These samples are drawn from two classes with equal priors. The class-conditional probability distributions for the two classes are Gaussian distributed with

$$\begin{aligned} p(x|\omega_1) &= \mathcal{N}(x; 0, 1) \\ p(x|\omega_2) &= \mathcal{N}(x; 1, 1) \end{aligned}$$

All nodes will have decisions of the form "`Is` $x \leq x_s$" where x_s is some threshold. At the top the level the value of the split threshold is x_1. The size of the tree is limited. It is a binary tree with a root node and two non-terminal nodes yielding a total of four leaf nodes. The binary split cost is given by

$$\Delta\mathcal{I}(N) = \mathcal{I}(N) - f_{\mathtt{L}}\mathcal{I}(N_L) - (1 - f_{\mathtt{L}})\,\mathcal{I}(N_R)$$

where $f_{\mathtt{L}}$ is the fraction of the data from the current node assigned to the left descendant. The entropy cost function is to be used. For the non-terminal node that satisfies the root node question find an expression, in terms of the cumulative density function for a Gaussian, for the binary split cost.

Non-Parameteric Techniques

10. $*$ n samples are drawn from a Gaussian distribution with mean, μ, and variance, σ^2. Consider a Gaussian window function of the form

$$\phi(x) = \mathcal{N}(x; 0, 1)$$

Show that the Parzen window estimate of the true distribution, $p(x) = \mathcal{N}(x, \mu, \sigma^2)$,

$$\tilde{p}(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h_n}\phi\left(\frac{x - x_i}{h_n}\right)$$

has the following properties (for small h_n):

(a) $\mathcal{E}\{\tilde{p}(x)\} = \mathcal{N}(x; \mu, \sigma^2 + h_n^2)$.

(b) $\text{var}[\tilde{p}(x)] \approx \frac{1}{2nh_n\sqrt{\pi}}p(x)$

(c) $p(x) - \mathcal{E}\{\tilde{p}(x)\} \approx \frac{1}{2}\left(\frac{h_n}{\sigma}\right)^2\left(1 - \left(\frac{x-\mu}{\sigma}\right)^2\right)p(x)$

Note the following equality may be used

$$\int_{-\infty}^{\infty} \mathcal{N}(x; v, \sigma_1^2)\mathcal{N}(v, \mu, \sigma_2^2)dv = \mathcal{N}(x, \mu, \sigma_1^2 + \sigma_2^2)$$

Speaker verification

11. $*$ A Support Vector Machine (SVM) is to be used for speaker verification. A 1-dimensional feature-vector is used top represent each frame of data. The feature-space to be used for with the SVM with observations $\mathbf{X}_{1:T} = \{x_1, \ldots, x_T\}$ is defined as

$$\boldsymbol{\Phi}(\mathbf{X}_{1:T}) = \left[\begin{array}{c} \frac{\partial}{\partial\mu_1}\log(p(\mathbf{X}_{1:T})) \\ \vdots \\ \frac{\partial}{\partial\mu_M}\log(p(\mathbf{X}_{1:T})) \\ \frac{\partial^2}{\partial\mu_1^2}\log(p(\mathbf{X}_{1:T})) \\ \vdots \\ \frac{\partial^2}{\partial\mu_1\partial\mu_M}\log(p(\mathbf{X}_{1:T})) \\ \vdots \\ \frac{\partial^2}{\partial\mu_M^2}\log(p(\mathbf{X}_{1:T})) \end{array}\right]$$

where the generative model is an M-component Gaussian Mixture Model (GMM), so

$$p(\mathbf{X}_{1:T}) = \prod_{t=1}^{T}\sum_{m=1}^{M} c_m\mathcal{N}(x_t; \mu_m, \sigma_m^2)$$

(a) Why is this form of feature-space suitable for use with SVMs when classifying variable-length data-sequences, such as in speaker verification? Why is an SVM a suitable form of classifier as M (the number of components) gets large? What is the dimensionality of the feature-space in this case?

(b) Derive an expression for $\frac{\partial}{\partial \mu_i} \log(p(\mathbf{X}_{1:T}))$. This should be expressed in terms of the $P(i|x_t)$, the posterior probability that component i generated the observation.

(c) Hence show that

$$\frac{\partial^2}{\partial \mu_j \partial \mu_i} \log(p(\mathbf{X}_{1:T})) = -\sum_{t=1}^{T} P(i|x_t)P(j|x_t)\frac{(x_t - \mu_j)(x_t - \mu_i)}{\sigma_i^2 \sigma_j^2}$$

Do you expect these second-order derivative terms to help in classification?

M.J.F. Gales
November 2003,2004,2007