Chapter III

Supervised Learning with Artificial Neural Networks

Introduction

In this chapter we will look at supervised learning in more detail, beginning with one of the simplest (and earliest) **supervised neural learning** algorithms: the Delta Rule. The objectives of this chapter are to provide a solid grounding in the theory and practice of problem solving with artificial neural networks, and an appreciation of some of the challenges and practicalities involved in their use.

The Delta Rule

An important early network was the Adaline (ADAptive LINear Element) (Widrow & Hoff, 1960). The **Adaline** calculates its output as $o = \Sigma_j w_j x_j + \theta$, with the same notation as before. You will immediately note the difference between this network and the perceptron is the lack of thresholding. The interest in the network was partly due to the fact that it has an easily implementation as a set of resistors and switches.

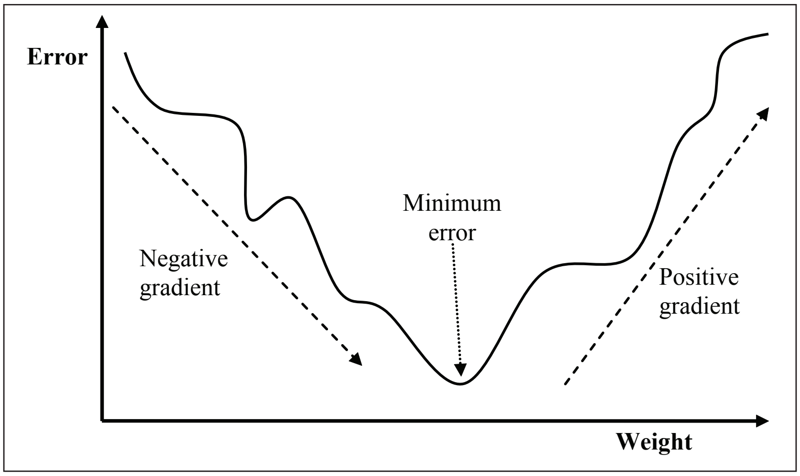
The Learning Rule: Error Descent

For a particular input pattern, x^P, we have an output o^P and target t^P. Then the sum squared error from using the Adaline on all training patterns is given by,

$$E = \sum_P E^P = \frac{1}{2}\sum_P (t^P - o^P)^2$$

where the fraction is included due to inspired hindsight. Now, if our Adaline is to be as accurate as possible, we wish to minimise the **squared error**. To minimise the error, we can find the **gradient** of the error with respect to the weights and move the weights in the opposite direction. If the gradient is positive, the error would be increased by changing the weights in a positive direction and therefore we change the weights in a negative direction. If the gradient is negative, in order to decrease the error we must change the weights in a positive direction. This is shown diagrammatically in Figure 1. Formally:

Figure 1. A schematic diagram showing error descent



Note. In the negative gradient section, we wish to increase the weight; in the positive gradient section, we wish to decrease the weight