Chapter V

Unsupervised Learning in Artificial Neural Networks

Unsupervised Learning

With the artificial neural networks we have met so far, we must have a training set on which we already have the answers to the questions we are going to pose to the network. Yet humans appear to be able to learn (indeed some would say can only learn) without explicit supervision. The aim of **unsupervised learning** is to mimic this aspect of human capabilities and hence this type of learning tends to use more biologically plausible methods than those using the error descent methods of the last two chapters. The network must self-organise and to do so, it must react to some aspect of the input data, typically either **redundancy** in the input data or **clusters** in the data; for example, there must be some structure in the data to which it can respond. There are two major methods used,

1. **Hebbian learning**
2. **Competitive learning**

We shall examine each of these in turn.

Hebbian Learning

Hebbian learning is so-called after Donald Hebb who in 1949 conjectured:

When an axon of a cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency as one of the cells firing B, is increased. (Hebb, 1949)

In the sort of feedforward neural networks we have been considering, this would be interpreted as the weight between an input neuron and an output neuron is very much strengthened when the input neuron's activation passes forward to the output neuron and causes the output neuron to fire strongly. We can see that the rule favours the strong: if the weights between inputs and outputs are already large (and so an input will have a strong effect on the outputs) the chances of the weights growing is large.

More formally, consider the simplest feedforward neural network which has a set of input neurons with associated input vector, **x**, and a set of output neurons with associated output vector, **y**. Then we have, as before, $y_i = \Sigma_j w_{ij} x_j$, where now the (Hebbian) learning rule is defined by $\Delta w_{ij} = \eta x_j y_i$. That is, the weight between each input and output neuron is increased proportional to the magnitude of the simultaneous firing of these neurons.

Now we can substitute into the learning rule the value of **y** calculated by the feeding forward of the activity to get,

$$\Delta w_{ij} = \eta x_j \Sigma_k w_{ki} x_k = \eta \Sigma_k w_{ki} x_k x_j.$$

Writing the learning rule in this way emphasises the statistical properties of the learning rule, for example, that the learning rule depends on the correlation between different parts of the input data's vector components. It does however also show that we have difficulty with the basic rule as it stands which is that we have a positive feedback rule which has an associated difficulty with lack of stability. If the input and output neurons are tending to fire strongly together, the weight between them will tend to grow strongly; if the weight grows strongly, the output neuron will fire more strongly the next time the input neuron fires and this will cause an increased value in the rate of change of the weights which will.... If we do not take some preventative measure, the weights will grow without bound. Such preventative measures include:

Related Content

Deep Learning for Trilingual Character Recognition
M. Yashodha, SK Niranjanand V. N. Manjunath Aradhya (2019). *International Journal of Natural Computing Research (pp. 52-58).*
www.irma-international.org/article/deep-learning-for-trilingual-character-recognition/219801

Gene Expression Programming and the Evolution of Computer Programs
Cândida Ferreira (2005). *Recent Developments in Biologically Inspired Computing (pp. 82-103).*
www.irma-international.org/chapter/gene-expression-programming-evolution-computer/28325

Symmetric Encryption Algorithm Inspired by Randomness and Non-Linearity of Immune Systems
Suriyani Ariffin, Ramlan Mahmod, Azmi Jaafarand Muhammad Rezal Kamel Ariffin (2012). *International Journal of Natural Computing Research (pp. 56-72).*
www.irma-international.org/article/symmetric-encryption-algorithm-inspired-randomness/72872

Guiding Self-Organization in Systems of Cooperative Mobile Agents
Alejandro Rodríguez, Alexander Grushinand James A. Reggia (2009). *Advancing Artificial Intelligence through Biological Process Applications (pp. 268-290).*
www.irma-international.org/chapter/guiding-self-organization-systems-cooperative/4983

Recent Methodology in Connectionist Systems
Ana B. Porto Pazos, Alberto Alvarellos Gonzálezand Alejandro Pazos Sierra (2009). *Advancing Artificial Intelligence through Biological Process Applications (pp. 94-106).*
www.irma-international.org/chapter/recent-methodology-connectionist-systems/4974