TOPIC - MULTILAYER NEURAL NETWORKS

Name-Md Ayan
Roll No- 28630522008
Branch-CSE/DS
Subject - Neural network and deep learning

Multilayer Neural Networks

We learned in Chapter 5 that clever choices of nonlinear arbitrary decision boundaries that lead to minimum error.

Identifying the appropriate nonlinear functions to be used can be difficult and incredibly expensive.

We need a way to *learn* the non-linearity at the same time as the linear discriminant.

Multilayer Neural Networks, in principle, do exactly this in order to provide the optimal solution to arbitrary classification problems.

Multilayer Neural Networks implement linear discriminants in a space where the inputs have been mapped non-linearly.

The form of the non-linearity can be learned from simple algorithms on training data.

Note that neural networks require continuous functions to allow for gradient descent

Multilayer Neural Networks

Training multilayer neural networks can involve a number of different algorithms, but the most popular is *the back propagation algorithm* or generalized delta rule.

Back propagation is a natural extension of the LMS algorithm.

The back propagation method is simple for models of arbitrary complexity. This makes the method very flexible.

One of the largest difficulties with developing neural networks is regularization, or adjusting the complexity of the network.

Too many parameters = poor generalization

Too few parameters = poor learning

Example: Exclusive OR (XOR)

x – The feature vector
y – The vector of hidden layer outputs
k – The output of the network
– weight of connection between input unit i and hidden unit j
– weight of connection between input hidden unit j and output unit k
bias – A numerical bias used to make calculation easier

Example: Exclusive OR (XOR)

XOR is a Boolean function that is true for two variables if and only if one of the variables is true and the other is false.

This classification can not be solved with linear separation, but is very easy for a neural network to generate a non-linear solution to.

The hidden unit computing acts like a two-layer Perceptron.

It computes the boundary + + = 0. If + + > 0, then the hidden unit sets = 1, otherwise is set equal to –1. Analogous to the OR function.

The other hidden unit computes the boundary for + + = 0, setting = 1 if + + > 0. Analogous to the negation of the AND function.

The final output node emits a positive value if and only if both and equal 1.

Note that the symbols within the nodes graph the nodes activation function.

This is a 2-2-1 fully connected topology.

Feedforward Operation and Classification

Figure 6.1 is an example of a simple three layer neural network

The neural network consists of:

- An input layer
- A hidden layer
- An output layer

Each of the layers are interconnected by modifiable weights, which are represented by the links between layers

Each layer consists of a number of units (neurons) that loosely mimic the properties of biological neurons

The hidden layers are mappings from one space to another. The goal is to map to a space where the problem is linearly separable.

Activation Function

Each hidden unit emits an output that is a nonlinear function of its activation, f(net), that is:

)2(

One possible activation function is simply the sign function:

The activation function represents the *nonlinearity* of a unit. The activation function is sometimes referred to as a *sigmoid* function, a *squashing* function, since its primary purpose is to limit the output of the neuron to some reasonable range like a range of -1 to +1, and thereby inject some degree of non-linearity into the network.

Training Protocols

Training consists of presenting to the network the collection of patters whose category we know, collectively known as the training set. Then we go about finding the output of the network and adjusting the weights to make the next output more like the desired target values.

The three most useful training protocols are:

- Stochastic
- On-Line
- Batch

In the stochastic and batch training methods, we will usually make several passes through the training data. For on-line training, we will use each pattern once and only once.

We use the term *epoch* to describe the overall number of training set passes. The number of epochs is an indication of the relative amount of learning.

Structural Adaptability

One of the biggest issues with neural networks is the selection of a suitable structure for the network. This is especially true in unknown environments.

Lee Tsu-Chang developed a method for altering the structure of the neural network during the learning process.

If, during training, the error has stabilized but is larger than our target error, we will generate a new hidden layer neuron.

A neuron can be annihilated when it is no longer a functioning element of the network.

This occurs if the neuron is a redundant element, has a constant output, or is entirely dependent on another neuron.

These criteria can be checked by monitoring the input weight vectors of neurons in the same layer. If two of them are linear dependent, then the represent the same hyper plane in the data space spanned by their receptive field and are totally dependent on each other. If the output is a constant value, then the output entropy is approximately zero.

This methodology is incredibly useful for finding the optimal design for a neural network without requiring extensive domain knowledge.

References

Duda, R., Hart, P., Stork, D. Pattern Classification, 2nd ed. John Wiley & Sons, 2001.

Y. LeCun, L. Bottou, G.B. Orr, and K.-R. Muller. Efficient Backprop. Neural Networks: Tricks of the Trade, Springer Lecture Notes in Computer Sciences, No. 1524, pp. 5-50, 1998.

P. Simard, D. Steinkraus, and J. Platt. Best Practice for Convolutional Neural Networks Applied to Visual Document Analysis. International Conference on Document Analysis and Recognition, ICDAR 2003, IEEE Computer Society, Vol. 2, pp. 958-962, 2003.

L. Tsu-Chang, *Structure Level Adaptation for Artificial Neural Networks, Kluwer Academic* Publishers, Boston, 1991.

Ramirez, J.M.; Gomez-Gil, P.; Baez-Lopez, D., "On structural adaptability of neural networks in character recognition," *Signal Processing, 1996., 3rd International Conference on* , vol.2, no., pp.1481-1483 vol.2, 14-18 Oct 1996