Multilayer perceptron

From Wikipedia, the free encyclopedia

A **multilayer perceptron** is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate output. It is a modification of the standard linear perceptron in that it uses three or more layers of neurons (nodes) with nonlinear activation functions, and is more powerful than the perceptron in that it can distinguish data that is not linearly separable, or separable by a hyperplane.

Contents

Theory

Activation function

If a multilayer perceptron consists of a linear activation function in all neurons, that is, a simple on-off mechanism to determine whether or not a neuron fires, then it is easily proved with linear algebra that any number of layers can be reduced to the standard two-layer input-output model (see perceptron). What makes a multilayer perceptron different is that each neuron uses a *nonlinear* activation function which was developed to model the frequency of action potentials, or firing, of biological neurons in the brain. This function is modeled in several ways, but must always be normalizable and differentiable.

The two main activation functions used in current applications are both sigmoids, and are described by

$$\phi(v_i) = \tanh(v_i) \quad \text{and} \quad \phi(v_i) = (1 + e^{-v_i})^{-1},$$

in which the former function is a hyperbolic tangent which ranges from -1 to 1, and the latter is equivalent in shape but ranges from 0 to 1. Here y_i is the output of the ith node (neuron) and v_i is the weighted sum of the input synapses. More specialized activation functions include radial basis functions which are used in another class of supervised neural network models.

Layers

The multilayer perceptron consists of an input and an output layer with one or more *hidden layers* of nonlinearly-activating nodes. Each node in one layer connects with a certain weight w_{ij} to every other node in the following layer.

Learning through backpropagation

Learning occurs in the perceptron by changing connection weights (or synaptic weights) after each piece of data is processed, based on the amount of error in the output compared to the expected result. This is an example of supervised learning, and is carried out through backpropagation, a generalization of the least mean squares algorithm in the linear perceptron.

We represent the error in output node j in the nth data point by $e_j(n) = d_j(n) - y_j(n)$, where d is the target value and y is the value produced by the perceptron. We then make corrections to the weights of the nodes based on those corrections which minimize the energy of error in the entire output, given by

$$\mathcal{E}(n) = \frac{1}{2}\sum_j e_j^2(n).$$

By the theory of differentials, we find our change in each weight to be

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial v_j(n)} y_i(n)$$

where y_i is the output of the previous neuron and η is the *learning rate*, which is carefully selected to ensure that the weights converge to a response that is neither too specific nor too general. In programming applications, typically ranges from 0.2 to 0.8.

The derivative to be calculated depends on the input synapse sum v_j, which itself varies. It is easy to prove that for an output node this derivative can be simplified to

$$-\frac{\partial \mathcal{E}(n)}{\partial v_j(n)} = e_j(n)\phi'(v_j(n))$$

where ϕ' is the derivative of the activation function described above, which itself does not vary. The analysis is more difficult for the change in weights to a hidden node, but it can be shown that the relevant derivative is

$$-\frac{\partial \mathcal{E}(n)}{\partial v_j(n)} = \phi'(v_j(n))\sum_k -\frac{\partial \mathcal{E}(n)}{\partial v_k(n)} w_{kj}(n).$$

Note that this depends on the change in weights of the kth nodes, which represent the output layer. So to change the hidden layer weights, we must first change the output layer weights according to the derivative of the activation function, and so this algorithm represents a *backpropagation of the*

activation function.[1]

Applications

Multilayer perceptrons using a backpropagation algorithm are the standard algorithm for any supervised-learning pattern recognition process and the subject of ongoing research in computational neuroscience and parallel distributed processing. They are useful in research in terms of their ability to solve problems stochastically, which often allows one to get approximate solutions for extremely complex problems like fitness approximation.

Currently, they are most commonly seen in speech recognition, image recognition, and machine translation software, but they have also seen applications in other fields such as cyber security. In general, their most important use has been in the growing field of artificial intelligence, where the multilayer perceptron's power comes from its similarity to certain biological neural networks in the human brain.

References

1. ^ Haykin, Simon (1998). *Neural Networks: A Comprehensive Foundation* (2 ed.). Prentice Hall. ISBN 0132733501.