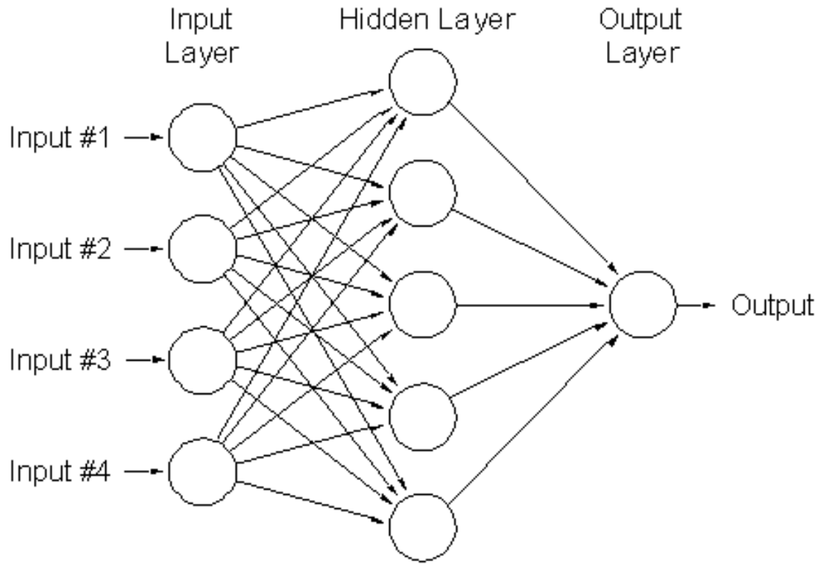
Advanced Applications of Neural Networks

By Jeff Chen
Computer Systems Lab, 2006-2007

Abstract

Neural networks are a powerful way of finding patterns and functions. Traditional methods and algorithms often have trouble finding patterns in data when there are noisy data and imperfections. Neural networks are designed to handle noise and be able to find complex patterns in noisy data. In that way, they are ideal for applications like predicting the stock market, compressing data and analyzing musical compositions. Furthermore, different types of neural networks have different strong points; this project will attempt each of the applications with various types of neural networks.



Introduction

The purpose of this project is to present a viable solution using neural networks to applications including predicting the stock market, efficient compression of data, and the analysis and composition of music. The stock market has frustrated investors for years, and attempts to find patterns in stocks have mostly failed. Neural networks may be a great solution to this problem, since neural networks don't have the restrictions of continuity of functions, and even several neurons in a network may simulate a function of numerous variables.

Background

An artificial neural network is a series of "neurons" which are connected to each other, making a neural network. Each connection between neurons has a weight to it, which is modified to alter the behavior of the network; this is how a network learns. The most common type of neural network is the backpropogation network. In this network, neurons are arranged in layers, with each layer connecting to the next and previous layers, but to neurons in their layer. The first layer is the input layer, and the last layer is the output layer. The intermediate layers are hidden layers. This type of network learns when input and desired output is given. The network then figures out the error between actual output and the desired output, then propogates the error backwards and modifies the weights accordingly.

Results

Both the Backpropogation network and the Competitive network have been coded so that they will be able to be adapted for a variety of applications. The purpose of this project was not just to find good applications for neural networks and use them, but also to create a flexible set of networks and utility classes for the Java platform that anyone may use for their own purposes.

On test data, the Backpropogation network ran fast and was trained effectively. However, the training data for testing on stocks was hard to find and parse, and though the network was able to accurately model the past performance of a certain stock, the prediction for future stock prices was sometimes accurate and sometimes way off the mark, as the network has to keep learning and altering its model.