# Applications of Neural Networks
# Computer Systems Lab 2006-2007

Jeff Chen

June 12, 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.

**Keywords:** neural network, function approximation, music categorization

## 1 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.

The major problem with predicting the stock market so far has been that one cannot factor in all the events that may occur. Will there be a scandal?

Will a related company go bankrupt? So far, there has not been a decent solution for this. Furthermore, prediction of the stock market using mathematical formulas are hideously complicated, and often inaccessible to the layman. This project aims to provide a blackboxed neural network implementation that is simple to use and effective in predicting the stock market. There is not a way to predict what events would cause the stock price to fluctuate. Thus, a concrete prediction is impossible. This project therefore will attempt to plot the expected path of the stock, as well as an approximate region where the stock price is expected to be.

Similarly, musical compositions are hard for computers to compose. Neural networks do not have any creativity; they merely emulate and categorize data. Therefore, it is necessary to train the network on the music you want it to compose. Given that there are so many different genres of music, teaching the network all the music will take a prohibitively long time as well as prevent it from making a coherent composition. This project will instead attempt to categorize music based on the notes and the instruments used in the composition. This section will use a competitive network, which groups data by having different networks "compete" for the best fit of the data. The "winning" network will then alter itself to have criteria closer to the data it just won, thus differentiating the networks and allowing classification later.

# 2 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.

Another common type of network is the Competitive Network. This network is actually a series of networks which all compete to be the best fit for the data. As a network wins a set of data, it changes itself to better match similar datasets in the future. The networks will all specialize and be more attractive to different sets of data, thus making it possible to use the network for classification.

The use of neural networks to predict the stock market is not new. Many websites have constantly running neural networks which you can use to help your investments. Though the validity of such predictions is questionable, many websites claim that their predictions have a high accuracy.

The composition of music with neural networks has been done, including projects from many universities. Classification of music has also been done, but to a lesser extent, through several projects from college students. That project (Music Classification through Neural Networks, Scott 2001) extracts features of music from .wav files. My project will take a different tack, using instead the notes and instruments used to classify classical music instead of modern music.

## 3 Development

This project will be coded using Java 1.5, as it uses extensive implementation of the new templates to maximize efficiency in coding and runtime. The neural network has 3 parts, the Neuron, the Network, and the Cases on which they train on.

The Neuron class will store all the parents and children it has. The Neurons store not just the other Neurons but the connections as well, to maximize efficiency during backpropogation and training. Furthermore, it has an array of all the weights for its input. For backpropogation to work, the activation function for each neuron must be a continuous, differentiable function. Thus, a simple threshold function will not work. Instead, the Neurons use the sigmoid function

$$f(x) = \frac{1}{1+e^{-x}} \tag{1}$$

Which passes all requirements, as well as having a bound between 0 and 1 (to simulate a bit switch), and decaying quickly to both poles, making it desirable for this application.

The Case class is used solely for the purpose of testing. The Case class allows the easy association with input and desired output, eliminating any confusion that might be caused using a different system.

The Network class is where it all comes together. The Network has 2 implementations, the BackpropogationNetwork and the CompetitiveNetwork. Each network has its own training algorithm and input, but both have the same setup. Both use the Neuron class to store the Neuron's information, as well as similar write/read methods to files that allow a network to be saved and resumed at a later time. Further optimization is obtained by storing an array of all Connections as well as Neurons. This allows for much faster searching during training and running data.

Previously, when the network was being trained, the backpropogation algorithm requires that neurons pass their errors to its parent neurons so that they can calculate their errors. Each Neuron stored its parents and children, so the error propogation did not take long. However, the problem was that the parents also needed to find the specific weights it assigned to the child neurons, to figure out its part of the error. That took O(N) time, where N was the number of children. The optimization of an edge list improved this twofold. First, it allowed instant search for the weights. Second, it contains all the information needed to train in one class, instead of having to pass data. This sped up the process to taking only O(1) time, as there was no searching involved.

# 4 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.

The main problem of the competitive network is the lack of data for training as well. There were no sites online that contained usable representations of compositions, because of copyright violations as well as the lack of need of that kind of data. As a result, the data had to be created by hand through analysis of sheet music. There is not enough data to train effectively at this time.

# 5 Literature Cited


1. M.C. Mozer, "Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multiscale processing.", *Connection Science 6*, pp. 247-280, 1994.

2. Tradetrek.com, "Neural network 5 day forecast", Retrieved January 8, 2007, from Web Site: http://tradetrek.com/Education/webclassroom/nn5forecast.asp

3. P. Scott, "Music classification using neural networks.", Retrieved January 25, 2007, from Stanford University Web Site: http://www.stanford.edu/class/ee373a/mu
2001, Spring.

4. L. Smith, "An introduction to neural networks.", Retrieved January 9, 2006, from http://www.cs.stir.ac.uk/ lss/NNIntro/InvSlides.html. 2005, October 20.