Character Recognition with Neural Networks
TJHSST Senior Research Research Paper
Computer Systems Lab 2009-2010

Patrick Stalcup

January 28, 2010

Abstract

1 Abstract

This paper explores the implementation and testing of neural network libraries in Cython, a subset of the Python programming language. It goes in depth about design and creation of the network, basic methods for training the network based on the error of the output, and was to use these networks to solve problem. It documents the problems encountered while researching and developing this network, and limitations of the network in its current state. It uses the case study of handwriting character recognition to demonstrate principals and methods behind these neural networks, and goes on to attempt to create a general method for recognizing characters.

2 Introduction

Handwriting analysis is one of the most commonly researched topic in computer science. It can be used on many things, from simply recognizing certain characters to identifying the writer of a particular word, sentence, paragraph, or longer body of work. There are many approaches to solving it, some using conventional geometric methods and other using more complex, algebraic

methods. For my research project, I propose that handwriting can be used as a stepping stone for learning neural networks to other, more advanced topics, such as board game heuristics.

3 Background

Neural networks are one of the oldest examples of machine learning, and one of the most popular today. The design mimics actual neurons, and it helps solve many problems. Using neural networks as opposed to complex, multivariate calculus (bordering on thousands of variables) helps the computer work efficiently, and it helps the user to design and test the network more effectively. Using knowledge of neural networks gained from smaller, sample problems, such as the binary operator XOR, neural networks can be used to solve a wide range of problems. Because of the way networks are designed, with multiple inputs, it lends itself well to identifying things based on several variables at one time.

4 Development

The neural network library designed will be used to create networks for a wide variety of things. One of the goals of the designer is to create something where simple commands can be used to generate the long, tedious lines of code that oftentimes come with creating neural networks in an object oriented fashion. After an initial network design is completed, it is the goal of the designer to create several applications of these networks, with the forerunner being handwriting analysis, followed by board game heuristics. Other areas that will be explored are differing methods of calculating error, faster corrections for error, and different ways of inputting and outputting the data presented. After prototyping is done, the network framework will be tested using various aplications

This project will be designed in the Python programming language. This language is being used for its simplicity in regards to object orientation, its duck-typing, and its simple syntax. Duck-typing is a way that the Python interpreter can let the user ignore typing. Python code is very easy to write another program to generate, so much of the code used will be generated by another program. Additionally, because of the varied ports of Python,

it comes with a system compatibility that is rarely matched. The neural networks will be created using a simple yet powerful object library, with much of the calculations related to input to hidden and hidden to output ratios being done by the computer. The network will take several inputs, for example from a picture it will take a value for each pixel, and it will feed the inputs through the network. After these inputs are fed through, they will come out of the output nodes, and be applied to a result. There may be any number of input nodes, output nodes, and hidden nodes used in the network. For example, in the handwriting analysis problem, there will be one input for each pixel and one output for each letter of the character library to be recognized. However, due to limitations with Python and its speed, some comprimises will need to be made. The compromise that is currently in place is the use of Cython. Cython is a small subset of Python that is used mainly for speed. It actually compiles into the C programming language, which can be compiled and optimized significantly easier than python.

This project will be used on any number of problems. The initial problem to be confronted with this project is to identify any number of characters from a given library based on pictures of handwriting and other samples, including ASCII art. After this initial problem is confronted, other topics will be explored, including creating heuristics for board states of board games such as Othello, Checkers, and Tetris. These heuristics will be applied in game to create an artificial intelligence client that will attempt to play either another computer or human player. There are still many topics to be explored in regards to neural networks, and there are many problems that can be solved using neural networks.

The expected results of this project is the successful completion of a simple yet powerful Cython neural network library that will be used to solve a variety of problems. The case study problem of this project will be handwriting recognition. It will be used to identify strengths and weaknesses of the network, and the experience gained there will be applied to other implementations of neural networks. The eventual end result will be a robust and versatile network that is used to solve a wide variety of problems.

5 Conclusion

Neural Networks can be used to solve many problems, but they can also be difficult to implement. When using an object oriented approach and a

slow language such as python, complex problems with a large amount of inputs can take far too long to work for them to be viable. However, if these networks are constructed properly, one single training session can be used to solve a problem hundreds and thousands of times. The weights of these networks can also be reused as initial training weights in other situations to help decrease the amount of time spent in development and training of new neural networks.