

## **Building a Neural Network to Differentiate Document Types**

Ryan Smith

Class of 2010

#C2691

570-577-4064

rms057@bucknell.edu

10019742

Faculty Advisor: Professor Xiannong Meng, Computer Science

X \_\_\_\_\_  
Ryan Smith

X \_\_\_\_\_  
Xiannong Meng

**Purpose of the Project:**

The purpose of this project is to design and implement a neural network that, after initial training, can differentiate documents that contain a particular type of information. For example, a trained neural network can recognize a document as a memorandum or a lab report based on the format and key words within the document. Besides being a useful tool for various web and desktop search applications, the results of this project will be a useful measure of the viability of training a neural network on relatively small amounts of data, since we will not have nearly the same amount of resources or time that were used by others who have performed experiments involving neural networks.

**Project Details:**

A neural network is, essentially, a network of links and nodes that models a human brain in order to take input from an outside source and, from that input, come to some conclusion about the data that was given to it after being sufficiently trained. It is made up of several layers of nodes, each connected to every node in its adjacent layers by links. Each link has a weight associated with it that is applied to the data that is transmitted through it. When input data is passed to the first layer of nodes, or the input layer, the data is broadcasted from every node through each link that is attached to it, multiplied by the weight of whichever link it is being sent through, and then collected by the nodes of the next layer, called the hidden layer. Once all the data has been collected by the hidden layer, each node performs a calculation and forwards its result to the next layer of nodes. This process continues until the final layer, or output layer, is reached. At this point, another calculation is done by the node or nodes in the output layer, and an answer is reached. Each node in the output layer also contains the desired answer from the given input, and an error gradient is calculated using the desired answer and the actual answer. This error is then propagated backwards through the rest of the neural network, and calculations are done to update the weight of each link. After propagating the error backwards a sufficient number of times, every link is strengthened or weakened appropriately until eventually the network is properly trained to obtain the correct answer from most combinations of input [1].

The original idea for this project came from a paper entitled “Beyond PageRank: Machine Learning for Static Ranking”[2]. In this paper, the authors demonstrated a program called RankNet, which uses a neural network to rank web pages based on various static characteristics, such as the frequency with which they are visited, domain characteristics, and others. They trained the neural network to take the characteristics of two web pages and decide which was the higher quality webpage. Using this model, they achieved an accuracy of 67.3%, versus 56.7% accuracy for Google’s PageRank algorithm. This sparked our interest in using neural networks for other applications.

A neural network that can be trained to differentiate between different types of data or documents could be a useful tool. In order to test this idea, we plan to use the files containing course websites in the computer science department and train a neural network that can differentiate between different types of pages on each website, such as the course description, labs, and assignments, among others. Since there are many years of course

websites on the school file system, there will be enough data to train and test the neural network for our purposes.

In order to adapt the data in the course websites for use by the neural network, we will be using a preexisting program written by Professor Xiannong Meng. This program will take each document, and create a posting list for each term found in the entire document collection. On the list is an item for every document that the term appears in, and contained in each item is the number of times that the term appears in that particular document. From this list, a tf-idf value is computed for each term-document pair. The tf-idf value is calculated by the product of the term frequency (tf), which is the number of times the given term appears in a document, and the inverse document frequency (idf), which is the inverse of the number of documents in which the given term appears [3]. So, for example, a term that appears six times in only one document would have a high tf-idf value for that document, while a term that appears only twice in a document, and is also in seven other documents, would have a lower value. The tf-idf measure can be used as input to a neural network. The output of the trained neural network can be used to differentiate different types of text document.

The result of this project will be a tool that is able to recognize different types of documents such as a course description or lab assignment from a course webpage. Also, I am a Presidential Fellow currently working on a long-term research project with Professor Xiannong Meng, and we both believe that this will be an important component to develop for uses in future web search projects. In the future, we hope to be able to build on this tool and move from differentiating only documents on the school's file system to being able to classifying full web pages from anywhere on the web.

### **Methodology:**

This research project will take place during June and July of 2008. A majority of the work will most likely take place in the computer labs in Dana and Breakiron. During the first two weeks, I will do additional research into how neural networks function and into previous implementations of neural networks in the area of web search. I will also continue to meet with Professor Meng several times a week during this time in order to discuss my findings and ideas for how to implement the project.

During the next two weeks, I will begin to meet with Professor Meng more often and start actually designing the program to both implement the neural network portion of the program and make it work with the preexisting program to create the posting lists for each term. After we design an algorithm that we feel is sufficient, we will organize our data into a large training and a smaller testing set. Then, we will begin running the training data through the network to train, debug, and optimize the network. This stage of the project will take approximately three weeks. For the final week, we will use the testing set of data to collect data and determine whether the network is correctly designed and sufficiently trained to complete its task.

This research project will give me an excellent opportunity to grow academically and intellectually. By going through the entire research process and finishing with a product with practical applications in the field of electronic search applications, Not only will I gain important experience in designing and implementing computer programs, but I will have a better understanding of how to successfully undertake a research project. I am extremely excited by the chance to be a part of this project.

## References:

- [1] Negnevitsky, Michael. *Artificial Intelligence: A Guide to Intelligent Systems*. New York: Addison-Wesley, 2005.
- [2] E. Brill, A. Prakash, and M. Richardson. "Beyond PageRank: Machine Learning for Static Ranking." Proceedings of the 15<sup>th</sup> International World Wide Web Conference, Edinburgh, Scotland, May 2006.
- [3] J. Patel and S. Tata. "Estimating the Selectivity of *tf-idf* based Cosine Similarity Predicates." SIGMOD Record, Vol. 36: June 2007.

## Faculty Endorsement

I really look forward to this wonderful opportunity to work with Ryan, a very capable sophomore Presidential Fellow, who has been working with me in the past year and a half. Ryan is self-motivated, well-organized, and very capable student. During the past year and a half, we met weekly on the subject of web information retrieval (web search), my core research area. I was very impressed by the progress he has been making. Since he was a first year student when we met, I suggested he built some foundations on computer networking, a basic component in web information retrieval. He read and has a very good grasp of a entire textbook on computer network that is meant for a 300 or 400 level computer network course in a semester. We then continued on more advanced topics in web information retrieval. He experimented with programming for web applications which will be needed when we actually implement the project we are currently pursuing. We then move on to the current topics – using neural networks to classify different types of document. I plan to use neural network to classify web pages, as a part of the intelligent web search project. Again Ryan read the literature and came up a basic implementation of a neural network in Java. Our plan for the summer is for him to use this neural network to first link with the parsing programs that I developed early, then actually train and validate the neural network using the existing file systems in the CS course websites. If the project is successful, we will have a very solid foundation on which we can build an intelligent web search engine that can classify and search different types of document based on a small set of trained document.

Because Ryan is a Presidential Fellow, we have been and will be meeting weekly during the semester. I will meet with him more often during the summer. If I am going away for conferences or professional trips, we will keep in touch through email or video conferences.

In summary, I support Ryan Smith's proposal fully without any reservation. I know he will gain tremendous confidence and invaluable experiences through the project while I can advance my research with his help.