

A Morphological Neural Network Approach to Information Retrieval

Christian Roberson and Douglas D. Dankel II

University of Florida
Box 116120, E301 CSE, CISE
Gainesville, FL 32611-6120 USA
croberso@cise.ufl.edu, ddd@cise.ufl.edu

Abstract

We investigate the use of a morphological neural network to improve the performance of information retrieval systems. A morphological neural network is a neural network based on lattice algebra that is capable of solving decision boundary problems. The morphological neural network structure is one that theoretically can be easily applied to information retrieval. In this paper we propose a new information retrieval system based on morphological neural networks and present experimental results comparing it against other proven models.

Introduction

Many information retrieval systems use well-known algorithms and techniques but these algorithms were developed for relatively small and coherent collections such as newspaper articles or book catalogs in a (physical) library. The Internet requires new techniques, or extensions to existing methods, to address gathering information, making index structures scalable and efficiently updateable, and improving the ability of search engines to discriminate useful from useless information.

One alternative architecture for information retrieval involves the use of a neural network. Most existing neural network IR systems are based on some type of classical artificial neural network model [1, 4]. There are other variations on the neural network concept that could also be considered for this problem. For example, a modern type of neural network, known as a Morphological Neural Network (MNN), differs from most other neural networks in the way computation occurs at the nodes [2]. Our goal is to show that these networks have a variety of valuable uses in the information retrieval field by creating a query engine that transforms user queries into an MNN capable of filtering document vectors in a latent semantic space.

Morphological Neural Networks

Morphological neural networks (MNNs) are the lattice algebra-based version of classical artificial neural

networks. Just like the classical model, MNNs can be single-layer or multilayer and are capable of solving various types of classification problems [2]. The computation that occurs at each dendrite is based on morphological operations using lattice algebra. We examine an information retrieval system based on this model.

The MNNIR Engine

The Morphological Neural Network Information Retrieval (MNNIR) Model is built on top of a standard Vector Model design for representing the terms and documents in a collection. Unlike the traditional Vector Model [3] where the query is converted into a pseudo-document vector and the cosine angle formula determines the relationship between each document and the query, in the MNNIR engine the query vector is used to dynamically construct a morphological neural network to rank the documents in the collection.

The query network is a single-layer morphological perceptron with a single positive dendrite designed to input a document vector and return a measure of relevance for the examined query. For each non-zero term in the query vector, an excitatory connection is made to the dendrite, and the connection weight is determined by the term weight of the query vector. Because the strength of the dendrite response is important, the neuron's activation function is a linear function rather than the standard hard-limiter function.

Let j represent the j^{th} excitatory connection to the dendrite in the network and let $w_{j,q}$ be the query pseudo-vector weight for this term. For each j let $w_j^l = -w_{j,q}$ be the connection weight to the dendrite, which is negative to provide a minimum threshold for the terms in the network. The dendrite output for any document vector x in the system is defined as

$$\tau(x) = \bigvee_{j=1}^n (x_j + w_j^l)$$

A linear activation function is applied to $\tau(x)$ to obtain the perceived relevance of the document.

At query time, a network is constructed from the provided query and each document in the collection is run through the network. Once the relevance scores for the collection have been obtained, the system then ranks the documents in decreasing order of relevance and returns the results to the user. Because of the speed of the morphological neural network, the IR system can quickly and efficiently determine the relevance of the documents and filter out any unwanted parts of the collection.

Experimental Evaluation

Our experiment used the TIME document collection, which consisted of 424 TIME magazine articles about the Cold War and 82 queries of varying lengths. The indexed collection contains over 15,000 unique terms and just under 300,000 words.

Experimental Procedure

The MNNIR model was compared to two other models: the vector IR model and the three-layered neural network model. To ensure an unbiased comparison, all three models were built using the same code base and executed using the same term-document matrix.

The vector model used the standard Salton-Buckley weights [3] to calculate the term-document matrix and the query pseudo-vectors. For each query a relevance score for all the documents in the collection was calculated using the cosine distance formula to find the angle between the document vector and the query vector. These results were then ranked and used to return the documents to the user.

The neural network model was an implementation of the three-layer model [4]. All of the connection weights between the document layer and the term layer used the weights from the Salton-Buckley term-document matrix. The initial query term weights were set to one. Once the initial activation levels were calculated, spreading activation continued until some minimum threshold was met. Then, the relevance scores were read from the output nodes and used to rank the documents.

Experimental Results

To compare the different models, each of the three models was run with all 82 pre-fabricated queries. The models were examined both for retrieval effectiveness and for speed.

For all three models tested, as recall increased we saw a drop in the level of precision. Overall, the vector model performed best with an average precision of 54% over all the queries. The MNNIR model had an average precision of 42%, while the neural network model had an average precision of 37%. For some individual queries the MNNIR model performed significantly better than the other models, and for most other queries the results were comparable or only slightly lower.

The other metric used to compare the three models was query processing speed. To minimize any anomalous runs, each model processed all the queries 10 times and the average completion time over all the runs was examined. The MNNIR model had an average run time of 35.8 seconds for all 82 queries, which is about 0.46 seconds per query. The vector model took on average 57.5 to process a set of queries, which is just over 0.7 seconds per query. The neural network model typically required around 20 iterations before stabilizing with each iteration taking approximately 1 second, for an average overall processing time per query of approximately 20 seconds. The MNNIR model ran approximately 37% faster than the vector model and about 43 times faster than the neural network model.

Conclusions and Future Work

Overall the simple MNNIR system performed very well when compared to the established IR models. While the model did not perform quite as well as the vector model in terms of its precision, there is potential for improvement and superior performance in a more advanced implementation. In addition, the improved speed of the un-optimized MNNIR model over the traditional models is very promising. It is possible that the shortcoming in precision could be the result of the simple network used in the query engine and a more advanced network could yield better results.

Future work is required to study the possible benefits of the MNNIR system. Additional modifications to the structure of the MNN and the weighting system used by the model could provide further improvement. Further study will include using larger networks in the query engine. We also intend to test our MNNIR engine against some larger and more robust datasets.

References

- [1] Kwok, K. L. (1989). A Neural Network for Probabalistic Information Retrieval. *Proc. of the 12th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 21-30.
- [2] Ritter G. X., Sussner P. (1996). An Introduction to Morphological Neural Networks. . *Proc. of the 13th International Conference on Pattern Recognition*, p. 709-717
- [3] Salton G., Buckley C. (1988). Term-weighting Approaches in Automatic Retrieval. *Information Processing & Management*, v. 24, n. 5, p. 513-523.
- [4] Wilkinson R., Hingston P. (1991). Using the Cosine Measure in a Neural Network for Document Retrieval. *In Proc. of the 14th Annual International SIGIR Conference on Research and Development in Information Retrieval*, p. 202-210.