Implementation of Large Backpropagation Networks for Text Retrieval

Thomas Mandl
Information Science - University of Hildesheim
Marienburger Platz 22 - 31141 Hildesheim - Germany
Phone.: ++49-5121-883-837
e-mail: mandl@rz.uni-hildesheim.de

ABSTRACT: Soft Computing techniques like neural networks are well suited to process texts due to their vague processing capabilities. However, indexing in Information Retrieval (IR) produces large and sparse patterns. In order to implement IR Systems based on neural networks, the data needs to undergo appropriate pre-processing. This paper shows how Singular Value Decomposition can be used to reduce the dimensionality of the data and how a standard neural network software package can implement a IR System.

INFORMATION RETRIEVAL SYSTEMS

Internet search engines are a typical example for Information Retrieval Systems. Their increasing importance also demonstrates, that the search for texts is crucial facing the ever growing amount of information available.
IR Systems try to solve users' information problems. The principle challenge is to gather information on documents from large collections and extract appropriate representations. The most common models use a document-term-matrix over all terms and documents where the value of each cell expresses the importance of a term for a document. These document representations are then compared to users information needs expressed in queries usually comprising of several words. The documents with the highest similarity to the query are returned to the user as result (figure 1).

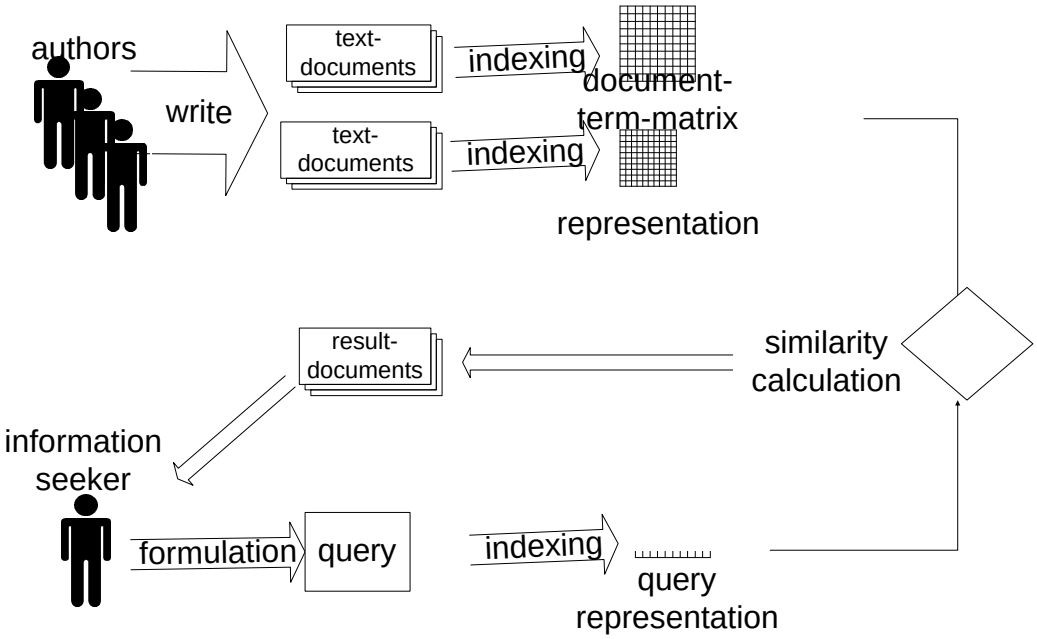


Figure 1: The Information Retrieval Process

Another challenge facing IR systems are heterogeneous representations resulting from different indexing vocabulary. Specialised information providers often rely on intellectual indexing, which requires human judgement on each document in the collection. To guarantee consistency over several human indexers, controlled vocabulary in the form of a thesaurus is introduced. A thesaurus is usually constructed for a specific area of interest, e.g. the social sciences or mathematics. Still, searching on different collections is often of great interest for users. For example, many market

researchers would like to use a single query to search within a special database and the internet. In order to allow such queries, heterogeneous data collections indexed by different methods and according to different vocabulary need to be combined or one needs to be transformed into the other.
A document-term-matrix in IR tends to be very large. For controlled vocabulary, it may be around 20.000 terms whereas for free text indexing it may comprise more than 50.000 terms. As most terms are not relevant for a document, the document-term-matrix is sparsely populated. That means that many of its cells have the value zero.
During the last decade, many IR systems based on neural networks have been developed. An overview can be found in Mandl 1998a. To enhance their learning capabilities new architectures seem to be necessary (Mandl 1998b). However, sparse patterns present a considerable problem for IR Systems applying neural networks.

DIMENSIONALITY REDUCTION DURING PRE-PROCESSING

Pre-processing is known to be a crucial factor for the success of neural networks. For IR in general and for IR systems based on neural networks, dimensionality reduction techniques have been applied.
Latent Semantic Indexing (LSI) has proved to be an efficient method to reduce the dimensions of IR data (cf. Dumais 1994). LSI is based on Singular Value Decomposition (SVD), a mathematical algorithm comparable to factor analysis (Berry 1992). SVD extracts between 50 to 300 dimensions or factors from all terms and aligns the documents in the new and much smaller term space. Syu et al. 1996 have used LSI in combination with a two-layer neural network.
Merkl 1995 has applied backpropagation to dimensionality reduction. In his model, input and output are identical and after convergence, the hidden layer contains the compressed representation. In this case however, a neural network still has to deal with the sparse patterns. A compression without connectionist models seems to be more promising.
In the experiment described here, a two stage pre-processing approach was pursued:

- Dimensionality reduction using LSI
- Standard normalization

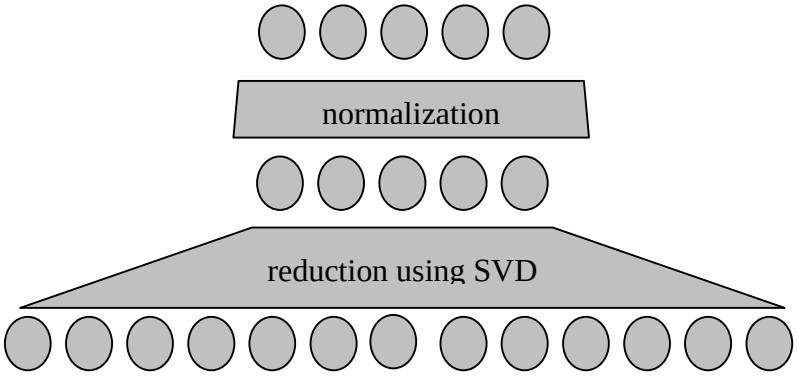


Figure 2: Two Step Pre-Processing

Experimental software from Bellcore has been used for the Latent Semantic Indexing. It runs on UNIX workstations. Input is a regular term-document-matrix containing all terms and documents. The output after the reduction is a term-document-matrix which still represents all the documents, but uses much fewer terms or factors. These factors cannot be interpreted as the terms. They represent complex combinations of many original terms.
The second step of pre-processing can be carried out within DataEngine. The data needs to be imported into DataEngine where it is normalized easily and quickly.

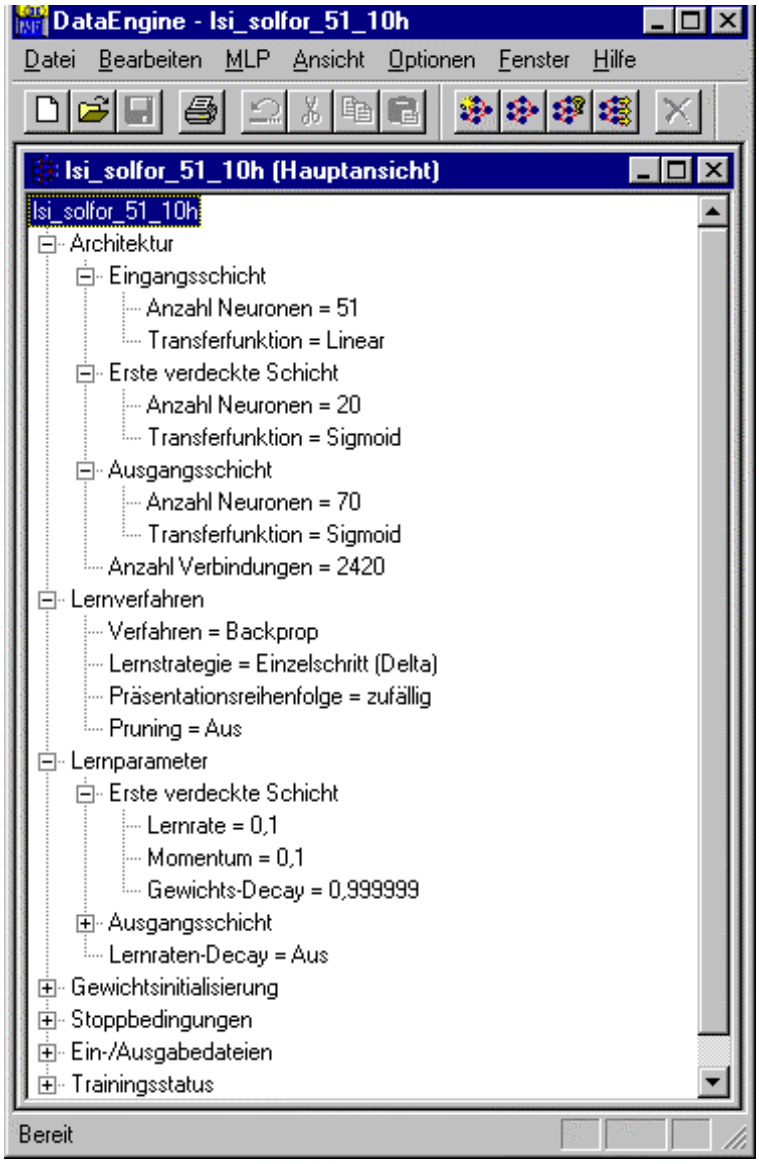
EXPERIMENT AND RESULTS

The experiment reported is further described in Mandl 1999. It attempts to create a mapping between two intellectual indexing schemes, which are used for different purposes for the same document collection. One is a thesaurus of some 6.000 terms and the other is a classification of some 150 categories. For the trained network only 70 categories occurring more frequently were chosen.
The thesaurus was reduced to 50 factors by LSI. The number 50 was chosen heuristically. Normally, LSI is used to extract between 100 and 300 factors. However, the number of original terms is higher then in the case here addressed.

These factors were then mapped onto the classification. The results were promising. A recognition rate of 96% could be reached.

IMPLEMENTATION WITH DATA-ENGINE

The experiment was implemented in DataEngine version 2.1. The data was imported using the ASCII filter. The second step of pre-processing was carried out in DataEngine. For all 50 input factors normalization between 0 and 1 was chosen within the Data Editor. Other methods like standardization were also tested but led to a higher error rate.



After the pre-processing, a neural backpropagation network can easily be implemented. The network has 50 input and 70 output units and one hidden layer. Considering the size of the network, training in DataEngine is rather fast. Each of the tested configurations was trained in less then one hour. Other networks with ten and 30 hidden units were also implemented and tested. Further parameters can be seen in figure 3.

Some small problems occurred while implementing the project. For a backpropagation neural network, normalization needs to be applied consistently to the whole data set including test and training patterns. After that, DataEngine requires a separation of training and test set into different files that are referred to from a network project. In order to test the different scaling methods provided by DataEngine, these steps need to be taken several times, resulting in several files.

Testing different distributions of patterns over the test and the training set to assure the validity of the results also leads to different files. Therefore, a stronger integration of on one hand training and test set and on the other of the data and the model would be of advantage. For example, there may be one pattern file where the user can choose normalization method and distribution of patterns over training and test set by determining the pattern numbers.

Figure 3: Network Model in DataEngine

The results of the network can be seen in figure 4. The network converged after some 100 epochs. Although the RMS as the typical measure for network quality is high compared to other problems, the quality measure for the mapping task reached a satisfactory level as stated above. This may be due to the fact that result patterns only contained 0 and 1, as the network implemented a classification task. This result underlines the importance of domain dependent measures which can yield results different from the standard backpropagation error rates.

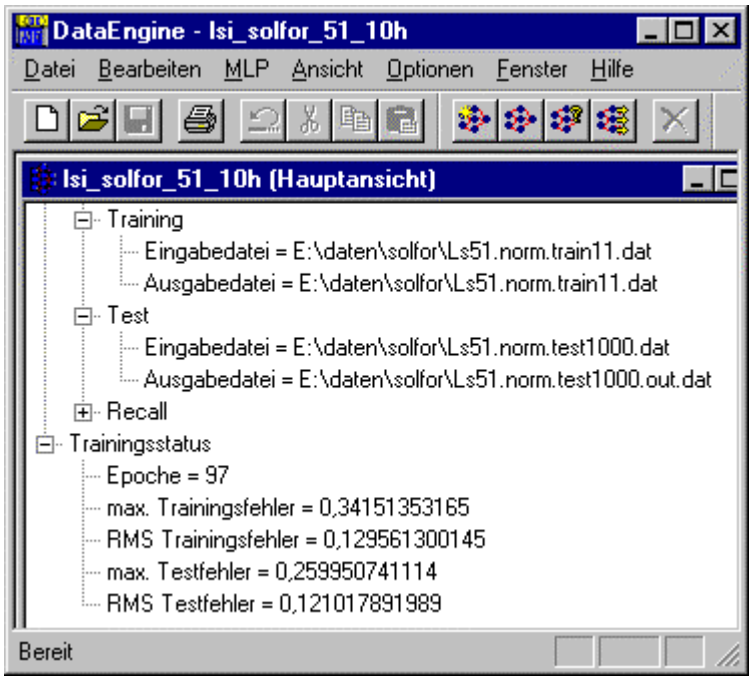


Figure 4: Results in DataEngine

CONCLUSIONS

IR is faced with large and sparse patterns and SVD could be applied successfully as a solution for this problem. Therefore, SVD may be a useful way of pre-processing for other domains with sparse vectors. This paper shows further, how a large mapping problem in IR could be easily implemented in a backpropagation network using DataEngine after the external pre-processing. Although the network is of considerable size, handling does not pose serious problems.

REFERENCES:

Berry, Michael (1992): Large Scale Sparse Singular Value Computations. In: International Journal of Supercomputer Applications. 1992. pp.13-49.

Dumais, Susan T. (1994): Latent Semantic Indexing (LSI) and TREC-2. In: Harman, Donna K. (ed.) (1994): The Second Text REtrieval Conference (TREC-2). Washington. pp. 105-115. URL: http://trec.nist.gov/pubs/trec2/t2_proceedings.html

Mandl, Thomas (1998a): Das COSIMIR-Modell: Information Retrieval mit Neuronalen Netzen. Informationszentrum Sozialwissenschaften Bonn, Arbeitsbericht, Februar. 1998. http.//www.uni-hildesheim.de/~mandl/cosimir

Mandl, Thomas (1998b): Vague Transformations in Information Retrieval. In: Zimmermann, H.; Schramm, V. (Eds.): Knowledge Management und Kommunikationssysteme: Workflow Management, Multimedia, Knowledge Transfer. Proc. 6. International Symposium für Informationswissenschaft. (ISI ´98). 3.-7.11.98. Karlsuniversität Prag, UVK: Konstanz [Schriften zur Informationswissenschaft vol. 34]. pp. 312-325.

Mandl, Thomas (1999): Efficient Preprocessing for Information Retrieval with Neural Networks. In: Zimmermann, Hans-Jürgen (ed.): EUFIT ‘99. 7th European Congress on Intelligent Techniques and Soft Computing. Aachen, Germany.

Merkl, Dieter (1995): Content-Based Document Classification with Highly Compressed Input Data. In: Proceedings of the International Conference On Artificial Neural Networks (ICANN ’95). Paris. October 9-13 1995. vol. 2. pp. 239-244.

Syu, Inien; Lang, S. D.; Deo, Narsingh (1996): Incorporating Latent Semantic Indexing into a Neural Network Model for Information Retrieval. In: ACM Conference on Information and Knowledge Management (CIKM.’96). Rockville MD. pp. 145-153.