Lessons Learned in Text Document Classification

Dieter Merkl

Institut für Softwaretechnik, Technische Universität Wien
Resselgasse 3/188, A-1040 Wien, Austria
dieter@ifs.tuwien.ac.at

Abstract

Text archives may be regarded as an almost optimal application arena for unsupervised neural networks. This because many of the operations computers have to perform on text documents are classification tasks based on noisy patterns. As a natural result, an ever increasing number of research reports concerned with that type of application appeared in literature. In this paper we argue in favor of paying more attention to the fact that text archives lend themselves naturally to a hierarchical structure. We take advantage of this fact by using a hierarchically organized network built up from independent self-organizing maps in order to enable the true establishment of a document taxonomy.

1 Introduction

The self-organizing map is a neural network model capable of arranging high-dimensional input data within its (usually) two-dimensional output space in such a way that the similarity of the input data is mirrored as faithfully as possible. The utilization of this model is thus especially well suited in application areas where one is highly dependent on a convenient data representation and visualization for subsequent exploration of a high-dimensional input data space. A perfect application of that kind is given by text document classification where the various text documents are usually represented by means of histograms of word occurrences. In such an environment, the words extracted from the text form the feature space for data representation which is high-dimensional by nature.

Concerning the exploration, however, we have to note a deficiency of the self-organizing map. This deficiency is the inability of the standard model to represent cluster boundaries explicitely. A substantial amount of research addressing the representation of cluster boundaries emerged as the natural consequence. Without going into detail, we may distinguish three different approaches to overcome this deficiency. The first one is concerned with improved visualization of the training result of a self-organizing map. Representatives of this line of research are [21, 3, 15]. Second, a number of neural network architectures have been suggested recently that are based on the self-organizing map's principle of unsupervised learning and address cluster boundary detection by means of adaptive network architectures. More precisely, these models rely on incrementally growing and splitting architectures where the final shape of the network is guided by the specific requirements of the input space. The work of [2, 6] represent proponents of research in this direction. The major critique concerning these models is that they require a few more parameters to be specified prior to network training. Conseqently, the utilization of these models is more demanding as far as the necessary experience of the users is concerned. Finally, the cluster boundary detection is addressed by means of individual self-organizing maps assembled into a neural network with layered architecture. We refer to [16, 22] as examples for research in this direction.

In this paper we argue in favor of using a model from the third group as described above. More precisely, we utilize the hierarchical feature map [17], a neural network model with layered architecture, where each layer consists of a number of independent self-organizing maps. Such a model seems to be remarkably well suited for text archive representation because the documents contained in text archives may be described conveniently by means of hierarchies or taxonomies. This is, by the way, the basic form of organization in use by librarians for centuries.

The material contained in the remainder of this paper is organized as follows. In Section 2 we give a brief review of text representation techniques. Section 3 contains a description of self-organizing maps and hierarchical feature maps as they are used in Section 4 for text archive organization. Finally, we give some conclusions in Section 5.

2 Text representation

Generally, the task of text classification aims at uncovering the semantic similarities between various documents. In order to enable further analyses, the documents have, in a first step, to be mapped onto some representation language. Still one of the most widely used representation languages is single-term full-text indexing. Roughly speaking, the documents are represented by feature vectors $x = (\xi_1, \xi_2, \ldots, \xi_n)$. Thereby the ξ_i, $1 \leq i \leq n$, refer to terms extracted from the various documents contained in the archive. The specific value of ξ_i corresponds to the importance of this feature in describing the particular document at hand. One might find a lot of strategies to prescribe the importance of a feature for a particular document [19]. Without loss of generality, we may assume that the importance is represented as a scalar in the range of [0, 1] where zero means that this particular feature is unimportant to describe the document. Any deviation from zero towards one is proportional to the increased importance of the feature in question.

In the perhaps most basic form of text representation, i.e. binary single-term indexing, the "importance" of a specific feature is represented by one of two numerical values. A value of one represents the fact that the corresponding feature was extracted from the document at hand. Contrary to that, a value of zero means that the corresponding feature is not contained in that document.

Such a feature-based document representation is known as the vector-space model of information retrieval. In this model, the similarity between two text documents corresponds to the distance between their vector representations [20].

In the experiments described in this work we will rely on a feature-based document representation. This representation may directly be used as the input to an artificial neural network. However, we do not want to let this section pass without giving hints to other document representation techniques. The utilization of a term-based document representation may lead to a very high-dimensional feature space. This feature space is generally not free from correlations. As a consequence, one might be interested in the transformation of the original document representation into a lower dimensional space. Statistical techniques that can be used to perform such a transformation include multidimensional scaling [4] and principal component analysis [9]. The former is the underlying principle of *Latent Semantic Indexing* [5]. The latter has been described and successfully applied to text corpora analysis in [1]. In [14] the feature space for document representation is reduced by means of an autoassociative multilayer perceptron. In this work the activations of the hidden layer units are used to represent the input data for subsequent self-organizing map training. Only recently, a number of papers have been published on the utilization of the self-organizing map for large scale document representation [8] based on the seminal work of [18] and subsequent document classification [12].

3 SOM and HFM — A short tour

The self-organizing map, *SOM*, as described in [10, 11] is a general unsupervised tool capable of ordering high-dimensional input patterns in such a way that patterns being similar in terms of the input space are arranged within neighboring regions of the neural network's output space. This property is achieved thanks to a learning rule that adapts the weight vector of the winner of the input presentation as well as a number of units in the vicinity of this winner. The spatial range of the adaptation, or in other words the number of units affected by adaptation, are usually defined by means of a neighborhood kernel taking into account the distance between the winner and the unit in question. Commonly, in the beginning of the learning process the neighborhood kernel is selected large enough to cover a wide area of the output space. The spatial width of the kernel is gradually reduced during the learning process. Such a strategy enables rough cluster formation in the beginning and fine-grained input discrimination towards the end of the learning process.

The key idea of the hierarchical feature map, *HFM*, as suggested in [16, 17] is to build a hierarchically organized neural network with layered architecture where each layer consists of a number of independent *SOM*s. More precisely, for each unit in one layer, a *SOM* is added to the next layer. This network is trained sequentially from the first layer downwards along the hierarchy until the maps of the last layer are reached. Hence, as soon as the first layer map has reached its stable state, training continues with the maps in the second layer. In this layer, however, each map is trained only by using the patterns mapped onto the corresponding unit of the layer above. Moreover, on the transition from one layer to the next, the input patterns may be shortened by omitting those components that are equal in the patterns mapped onto the same unit. This feature is of especial importance when dealing with text as the underlying input data because first text documents are represented in a very high dimensional feature space and second, some of these features are common to each text belonging to a particular topic, i.e. cluster. Thus, the pattern reduction inherent to *HFM*s is expected to play an important role during the learning process.

A highly valuable property of *HFM*s is the remarkable speed-up of the training process as compared to *SOM*s. An explanation that goes beyond the obvious dimension reduction of the input data may be revealed by an investigation of the general properties of the self-organizing training process. In *SOM*s, the units that are subject to adaptation are selected by means of the neighborhood kernel. It is common practice that in the beginning of the training process almost the whole map is affected by the presentation of an input vector. With this strategy, the map is forced to establish initial clusters of similar input items in the beginning of learning. By means of decreasing the neighborhood kernel in the course of the training process, the training process is capable of learning ever finer distinctions within the clusters while the overall topology of cluster arrangement is maintained. The back-side of the coin, however, is that units along the boundary between two clusters tend to be occasionally modified as belonging to either one of these clusters. This interference is the reason for the time-consuming self-organizing process. Contrary to *SOM*s, such an interference is dramatically reduced in *HFM*s. This reduction is due to the architecture of this neural network. The topology of the high-level categories is represented in the first layer of the hierarchy. Each of its sub-categories are then independently organized within separate maps at lower levels within the hierarchy. These maps in turn are free from maintaining the overall structure as this structure is already determined by the architecture of the *HFM*. As a summary, much computational effort is saved due to the fact that the overall structure of clusters is determined by the architecture of the neural network rather than by its learning rule.

4 Document space organization

Throughout the remainder of the paper we will use the various manual-pages of the NIH Class Library [7], *NIHCL*, as a sample document archive. The selection of the *NIHCL* is motivated by the fact that software libraries represent a convenient application arena for information retrieval systems. The reason is that much of the information about a particular software component is available as its textual description organized in manual pages. Moreover, the results from the classification process may easily be evaluated because the semantics of the software component, i.e. its functionality, is well-known.

The *NIHCL* is a collection of classes developed in the C++ programming language. The class library covers classes for storing and retrieving arbitrarily complex data structures on disk, like

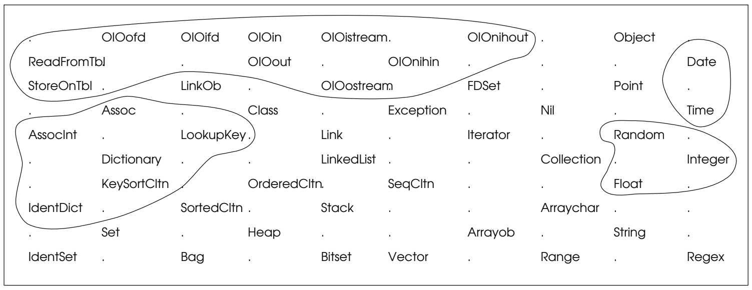

Figure 1: NIHCL presented in a 10×10 self-organizing map

`OIOin` and `OIOnihout`, generally useful data types such as `String`, `Time`, and `Date`, and finally a number of container classes as, for example, `Set`, `Dictionary`, and `OrderedCltn`.

The full-text of the various manual pages describing the classes is accessed and indexed in order to generate a binary vector-space representation of the documents. Just to provide the exact figure, the indexing process identified 489 distinct content terms and thus, each component is represented by a 489-dimensional feature vector. These vectors are subsequently used as the input data to the artificial neural network. Please note that we do not make use of the meta-information contained in the manual-entries, like the reference to base class, derived classes, and related classes. Such an information is not generally available in document archives and thus, for the sake of wide applicability, we omitted the usage of this type of meta-information here.

A typical result from the application of *SOM*s to this type of data is represented in Figure 1. In order to provide the means for convenient discussion we marked some of the highly interesting regions manually. In the upper part of the map we recognize the arrangement of all classes performing file I/O, these classes are designated by the `OIO`-part of their respective class name. The region itself is organized in such a way that a class performing an input operation is mapped neighboring its output counterpart, e.g. `OIOifd` and `OIOofd`. Just below the I/O region we find a large area of the map consisting of the various classes representing data structures. Within this larger region we marked the area comprising classes that allow an access to their elements via a key-attribute, i.e. `Dictionary`, `IdentDict`, and `KeySortCltn`. Within this area we find the classes that actually implement this type of access, too, i.e. `Assoc`, `AssocInt`, and `LookupKey`. On the right-hand side of the map we find the classes representing data structures. Again, their arrangement reflects their mutual similarity. Due to the limited space in this paper we cannot cover these results in more detail, we rather refer to [13].

What is less obvious, however, is the detection of the various cluster boundaries, at least in this basic form of visualization. In order to draw a line separating two clusters one has to have some insight in the underlying text archive. In the basic visualization a novice exploring the text archive might easily come up with some erroneous perceptions of the mutual similarity between the stored text documents. The organization of the same text archive by means of *HFM*s, as presented in Figure 2, is less likely to produce similar misconceptions. Due to the hierarchical arrangement of a number of independent *SOM*s, the inherent hierarchical organization of the text archive is mirrored in almost perfect fashion. In the top-layer map, shown in the center of Figure 2, the four main groups of classes are clearly separated. Each group in its turn is further detailed within its own *SOM* in the second layer of the hierarchy.

The hierarchical organization of text archives is especially appealing because it emerges naturally with text collections. The inherent hierarchical organization of text collections originates from their decomposition into various topics that are dealt with within the text documents. This is, obviously, the underlying organizational principle used by librarians to structure the contents of a library.

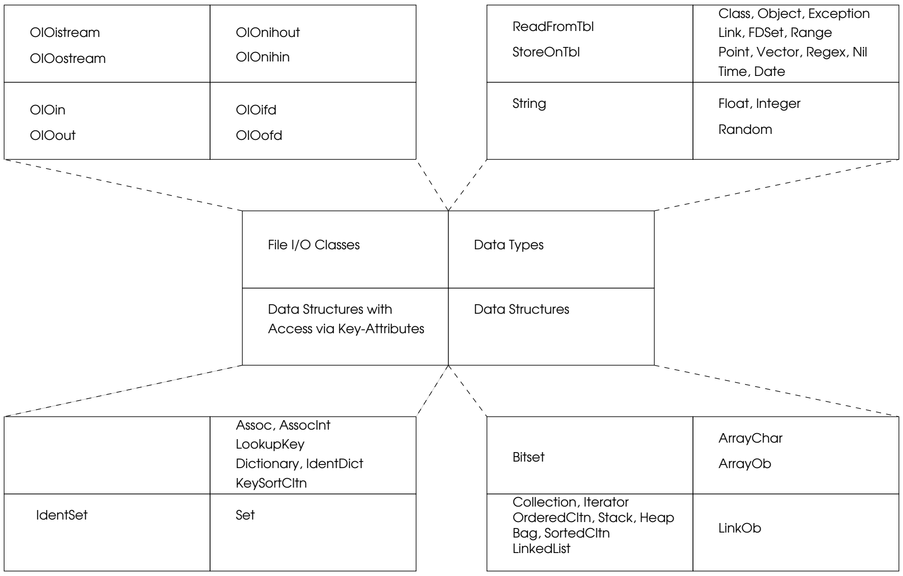

Figure 2: NIHCL presented in a hierarchical feature map

5 Conclusion

In this paper we compared the results from using self-organizing maps and hierarchical feature maps for text document classification. In principle, both models yielded similar results as far as the arrangement of semantically similar text documents in neighboring areas of the maps is concerned. The extra benefit of hierarchical feature maps is to impose an explicit hierarchical organization on the underlying text archive. Such an organization form seems to be well suited for text archives. In fact such an organization is in use by libarians for centuries. With the resulting taxonomy of documents it is less likely that the user of the text archive comes up with erroneous perceptions concerning document similarity, especially along the boundary between clusters. Moreover, the data reduction on the transition from one layer to the next one is another point in favor of the utilization of hierarchical feature maps. This feature of hierarchical feature maps is especially remarkable with the advent of massive digital libraries where commonly very high-dimensional feature vectors are used to represent the various documents.

Acknowledgments

Part of this work was performed while the author was visiting the Department of Computer Science, Royal Melbourne Institute of Technology.

References

[1] T. Bayer, I. Renz, M. Stein, and U. Kressel. Domain and language independent feature extraction for statistical text categorization. In *Proc Workshop Language Engineering for Document Analysis and Recognition*, Sussex, UK, 1996.

[2] J. Blackmore and R. Miikkulainen. Incremental Grid Growing: Encoding high-dimensional structure into a two-dimensional feature map. In *Proc IEEE Int'l Conf on Neural Networks*, San Francisco, CA, 1993.

[3] M. Cottrell and E. de Bodt. A Kohonen map representation to avoid misleading interpretations. In *Proc Europ Symposium on Artificial Neural Networks*, Brugge, Belgium, 1996.

[4] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling.* Chapman & Hall, London, 1994.

[5] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Hashman. Indexing by latent semantic analysis. *Journal American Society for Information Science*, 41(6), 1990.

[6] B. Fritzke. Growing Cell Structures: A self-organizing network for unsupervised and supervised learning. *Neural Networks*, 7(9), 1994.

[7] K. E. Gorlen. *NIH class library reference manual.* National Institutes of Health, Bethesda, MD, 1990.

[8] T. Honkela, V. Pulkki, and T. Kohonen. Contextual relations of words in Grimm tales analyzed by self-organizing maps. In *Proc Int'l Conf on Artificial Neural Networks*, Paris, France, 1995.

[9] I. T. Jolliffe. *Principal Component Analysis.* Springer-Verlag, Berlin, 1986.

[10] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43, 1982.

[11] T. Kohonen. *Self-organizing maps.* Springer-Verlag, Berlin, 1995.

[12] K. Lagus, T. Honkela, S. Kaski, and T. Kohonen. Self-organizing maps of document collections: A new approach to interactive exploration. In *Proc Int'l Conf on Knowledge Discovery and Data Mining*, Portland, OR, 1996.

[13] D. Merkl. A connectionist view on document classification. In *Proc Australasian Database Conf*, Adelaide, Australia, 1995.

[14] D. Merkl. Content-based document classification with highly compressed input data. In *Proc Int'l Conf on Artificial Neural Networks*, Paris, France, 1995.

[15] D. Merkl and A. Rauber. On the similarity of eagles, hawks, and cows – Visualization of similarity in self-organizing maps. In *Proc Int'l Workshop Fuzzy-Neuro-Systems'97*, Soest, Germany, 1997.

[16] R. Miikkulainen. Trace feature map: A model of episodic associative memory. *Biological Cybernetics*, 66, 1992.

[17] R. Miikkulainen. *Subsymbolic Natural Language Processing: An integrated model of scripts, lexicon, and memory.* MIT-Press, Cambridge, MA, 1993.

[18] H. Ritter and T. Kohonen. Self-organizing semantic maps. *Biological Cybernetics*, 61, 1989.

[19] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5), 1988.

[20] H. R. Turtle and W. B. Croft. A comparison of text retrieval models. *Computer Journal*, 35(3), 1992.

[21] A. Ultsch. Self-organizing neural networks for visualization and classification. In *Information and Classification – Concepts, Methods, and Applications.* Springer-Verlag, Berlin, 1993.

[22] W. Wan and D. Fraser. Multiple Kohonen self-organizing maps: Supervised and unsupervised formation with application to remotely sensed image analysis. In *Proc Australian Conference on Neural Networks*, Brisbane, Australia, 1994.