

Document Classification using Neural Networks

Nitin N. Pise, Maharashtra Institute of Technology, Pune, India

Abstract

The paper starts with the need for classification. Then the reasons why neural networks are suitable for document classification are explained. The paper continues with the details of the most commonly used topologically organized network model proposed by Kohonen (1982), referred to as the self-organizing map (SOM). The general idea proposed is to display the contents of a document library by representing similar documents in similar regions of the map. Without knowledge of the type of and the organization of the documents it is difficult to get satisfying results without multiple training runs. So the paper discusses the possibility to use a hierarchical structure of independent SOMS, referred to as GHSOM, where for every unit of a map, a SOM is added to the next layer. The essential steps in the classification system are given. The paper concludes with applications of document classification using neural networks.

Keywords : neural, network, document, classification, self, organizing, map, hierarchical, layer, units, learning

Classification is a method of describing resources and means of organizing libraries. It collects the similar types of records together and stores the records in hierarchical structure. The classification system facilitates the browsing and searching for internet environments as well as for the company environments. The number of classification systems like subject-specific or homegrown schemes can be used. Subject-specific schemes are designed for use by a particular (international or national) subject community, for example the National Library of Medicine (NLM) Classification, the Engineering Information (EI) Classification codes etc. Homegrown schemes are devised for use in a particular device, e.g. that used by Yahoo.

The main advantage of creating a completely new classification scheme is that a gateway is able to create a customized scheme, adapted to its specific content and user groups, which should be able to meet all of its specific requirements. This should allow for easier and more consistent browsing of a gateway, for example, there should be no unnecessary parts of the structure that would end up being unused. The best reason for inventing a classification system for a new service is when there is absolutely no suitable or adaptable system available or many different small ones only not providing the necessary coverage.

Need for Classification Systems [2]:

The completeness and timeliness of information are vital elements for modern organizations, whether they are large international agencies or small enterprises. The amount of information now available to them has become huge, and the growth trend is nearly exponential. This is actually a drawback since traditional search systems are starting to reach their limits. The use of the older systems is getting more and more difficult for the users who want to retrieve relevant information, as well as for the maintainers who have to carry out such activities as indexing, document classification and thesaurus maintenance. The two key requirements for solving this problem are :

1. Simplification of the search activities carried out by non-expert users.
2. Reduction in maintenance costs.

Artificial Neural Networks have qualities that can be exploited successfully in order to fulfill these requirements. Document handling tasks are usually characterized by a lack of pre-defined rules ; moreover, they can often be reduced to classification tasks. Research in the last ten years has shown that artificial neural networks are particularly good at dealing with such ill-structured classification tasks.

Libraries have long experience of applying classification schemes to resources mainly books. The idea of classification is to make it easier for users to find and retrieve resources. By building a hierarchical structure, the classification schemes enable users to look for related items that have not previously identified as relevant. This facilitates browsing within a physical library.

The use of classification systems offers one solution to providing improved access to web resources. The web is full of websites that have been created to act as guides to other web sites selected according to some pre-specified criteria, e.g. they are judged to be good quality resources or relevant to a particular subject-area.

Use of Neural Network in the Classification System

Artificial neural networks [1] refer to computing systems whose central theme is borrowed from the analogy of biological neurons. In principle, neural networks can compute any computable function, i.e. they can do everything a normal digital computer can do. Especially anything that can be represented as mapping between vector spaces can be approximated to arbitrary precision by feed forward neural networks which are the most often used type.

The reasons for choosing the neural network implementation are :

1. Neural networks appear well suited to pattern recognition roles where the matching required is inexact. These flexible matching properties are expected to improve retrieval, particularly for inexperienced end users.
2. Neural networks learning allow matching and recognition software to be crafted using the structure of the data itself. This paper adapts an approach whereby only training data require constructing a version of the system for a document collection is the document collection itself. This will allow the cataloging of the new document collections without time consuming and costly manual work.
3. While the training phase of the neural network development can be computationally intensive and require considerable periods of time to complete, once trained neural network algorithms, if suitably organized can prove both fast and efficient. The training for a collection can be performed off-line, and the trained networks can then offer a good user response time.

The Growing Hierarchical Self-Organizing Map (GHSOM) [3]

The Self-Organizing Map (SOM), [1] proposed by Kohonen, combines a competitive learning principle with a topological structuring of nodes such that adjacent nodes tend to have similar weight vectors. The SOM is an artificial neural network model that is well suited for mapping high-dimensional data into a two-dimensional representation space. The training process is based on the weight vector adoption with respect to the input vectors. The SOM has shown to be a highly effective tool for data visualization in a broad spectrum of application domains. Especially the utilization of the SOM for information retrieval purposes in large free form document collections has gained wide interest in the last few years. The general idea is to display the contents of a document library by representing similar documents in similar regions of the map. Without knowledge of the type of and the organization of the documents it is difficult to get satisfying results without multiple training runs using different parameter settings, which obviously is extremely time consuming given the high-dimensional data representation.

One possibility is to use a hierarchical structure of independent SOMS, where for every unit of a map, a SOM is added to the next layer. This means that on the first layer of Hierarchical Self-Organizing Map (HSOM), we obtain a rather rough representation of the input space but with descending the hierarchy, the granularity increases. Such an approach is especially well suited for the representation of the contents of a document collection. However, like with the original SOM, HSOM uses a fixed architecture with a specified depth of the hierarchy and predefined size of the various SOMs on each layer. Again, we need profound knowledge of the data in order to define a suitable architecture. So the Growing Hierarchical Self-Organizing Map (GHSOM) model consists of a hierarchical architecture where each layer is composed of independent SOMS that adjust their size according to the requirements of the input data.

Steps in the Classification System [2]:

The following are the steps which are required for the document to classify using the system where two major modules are used which are as follows:

1. Preprocessing
2. Self- Organizing Map (SOM)

1. Preprocessing [2]:

- a) Lexing
- b) Excluding stop words
- c) Stemming
- d) Feature Set Reduction
- e) Normalization

f) Preparing Document Term Matrix

2. Self- Organizing Map (SOM):

- a) Apply the document to map.
- b) Find the correct position of the document in the map. i.e. search for the winning code in the map.
- c) Train the nodes around the winning node.

GHSOM (Growing Hierarchical Self Organizing Map) [3]:

The main aim of this paper is to classify the documents using the Self- Organizing Map (SOM) and GHSOM (Growing Hierarchical Self Organizing Map). The key idea of the GHSOM [3] is to use a hierarchical structure of multiple layers where each layer consists of a number of independent SOMs. One SOM is used at the first layer of the hierarchy. For every unit in this map a self-organizing map might be added to the next layer of the hierarchy. This principle is repeated with the third and any further layers of the hierarchical feature map.

Since one of the shortcomings of SOM usage is its fixed network architecture, the scheme uses an incrementally growing version of the SOM. This relieves users from the burden of predefining the network's size which is rather determined during the unsupervised training process. The users start with a "virtual" layer 0, which only consists of one single unit. The weight vector of this unit is initialized as the average of all input data. The training process basically starts with a small map of say, 2×2 units in layer 1, which are self-organized according to the standard SOM training algorithm.

Just to summarize the training algorithm, an input pattern is selected randomly and presented to the neural network. Each unit determines its activation according to the distance between its weight vector and the input vector. The unit showing the smallest distance, i.e. the winner, as well as a number of units in the vicinity of the winner is adapted. Adaptation is performed as a gradual reduction of the difference between the vector's components. Hence, after the adaptation the winner will be more similar to the input pattern.

This training process is repeated for a fixed number of training iterations. Even after training iterations the unit with the largest deviation between its weight vector and the input vectors represented by this very unit is selected as the error unit. In between the error unit and its most dissimilar neighbor in terms of the input space either a new row or a new column of units is inserted. The weight vectors of these new units are initialized as the average of their neighbors. This training process is highly similar to the Growing Grid model. The difference so far is that a decreasing learning rate and a decreasing neighborhood range instead of fixed values is used. Especially the fixed neighborhood range is problematic when the network grows to be larger after a series of insertions.

An obvious criterion to guide the training process is the quantization error q_i . It is calculated as the sum of the distances between the weight vector of a unit and the input vectors mapped onto this unit and may be used to evaluate the mapping quality of a SOM based on the mean quantization error (MQE) of all units in the map. The lower the value of the MQE, the better the map is trained. A map grows until its MQE is reduced to a certain fraction t_1 of the q_i of unit i in the preceding layer of the hierarchy. Thus, the map now represents the data mapped onto the higher layer unit i in more detail.

However, the most important difference with the Growing Grid is that Growing Grid is designed to build a single SOM to represent the input data. In case of a large number of input data, the resulting map will be large, too. Consider a geographical map of Europe containing all the information that we expect a map of Belgium should contain. This hypothetical map of Europe will be of a size making it difficult to find an orientation. A similar situation occurs if a single map represents the contents of a large document library. Thus, it is interesting in building small maps where each unit represents a number of input data that are further expanded in separate maps further down the hierarchy.

The initial architecture of the GHSOM [3] consists of one self-organizing map. Another layer in case of dissimilar input data being mapped on a particular unit expands this architecture. These units are identified by a rather high quantization error q_i above a threshold t_2 . This threshold basically indicates the desired granularity level of data representation as a fraction of the initial quantization error at layer 0. In such a case, new map will be added to the hierarchy and the input data mapped on the respective higher layer unit are self-organized in this new map, which again grows until its MQE is reduced to a fraction t_1 of the respective higher layer unit's quantization error q_i . The depth of the hierarchy will rather reflect the un-uniformity that should be expected in real world data collections.

Depending on the desired fraction t_1 of MQE, the users may end up with either a very deep hierarchy with small maps, a flat structure with large maps, or in the most extreme case – only one large map, which is similar to the Growing Grid. The growth of the hierarchy is terminated when no

further units require expansion, i.e. all units represent the respective data with a quantization error q_i below t_2 .

Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task or tasks drawn from the same population more efficiently and more effectively next time.

The scheme uses the unsupervised learning method for the network. The supervised learning assumes the availability of teacher or supervisor who classifies the training examples into classes, whereas unsupervised learning must identify the pattern class information as a part of the learning process. In general, the task of the unsupervised learning is more abstract and less defined.

Architecture of the system:

Tasks to Perform [2]:

- 1) Preprocessing and Normalization
- 2) Document Classification using GHSOM

Preprocessing [2]:

The preprocessing part is the process which is applied to the document before it is fed to the next part of the system which is the document classification using the GHSOM.

Following are the steps, which are carried out when the document is fed to the preprocessing part of the system:

1) Lexing :

Lexing is the process of separating the words from the input document and store these words in the file (.wrđ file). This is used to maintain the dictionary of the input document in the next phase.

2) Stop words Elimination:

Stop words are the words which don't have meaning with respect to the classification. So these words are extracted when the dictionary is created for the classification purpose. In short the words are extracted from the document which are not required for the next phase (SOM).

3) Stemming:

The stemming process is nothing but removal of prefixes and suffixes. The objective is to eliminate the variation that arises from the occurrence of different grammatical forms of the same word. The stemming process helps to reduce the size of the dictionary file.

4) Weighting scheme:

In reality, the elements in the document term matrix can't measure the occurrence of the terms in the document. But the proposed scheme uses neat weighting scheme that will provide better results. There are two types of the weighting schemes that can be applied to the system are:

a) Local weighting scheme

b) Global weighting scheme

In the local weighting scheme, a term that occurs more frequently in a particular document tends to indicate its significance in conveying the meaning of the document. A term with a greater frequency is given more weightage than that occurs less frequently.

In the global weighting scheme, a term that occurs more frequently in all documents of the corpus is less important than one that occurs only in certain documents. The global weight of the term should decrease with increase in global frequency.

By combining the local and global weighting scheme, the better results can be obtained for the purpose of the classification.

5. Feature set reduction:

In text classification applications, important term selection is critical task for the classifier performance. With increasing number of documents, number of features also increase. To reduce the size of the dictionary, the threshold feature set reduction methods is used.

In this type, the upper and lower thresholds are decided according to the number of words in the dictionary. After that the term which exceeds the upper threshold and the terms below lower threshold are extracted from the document. This helps to reduce the size of the dictionary.

6. Document term matrix and Normalization:

The main purpose of the preprocessing phase is to prepare the document term matrix [2] that will feed to the next phase of the system. i.e. SOM. The document term matrix contains the fields as shown in the following table:

Terms in the document					
Doc id					
	Term1	Term2	Term3	Term n
A					
B					
.					
.					
F					

Table 1: Document term matrix

Normalization is nothing but calculating the normalized weight of each word by using obtained term frequencies. The statistical weight of each term is calculated as a relation of word frequency to the total number of meaningful words in the document.

$$w_i = \text{freq}_i / (\text{sum of freq } k), k=1,2,\dots,n.$$

The starting point for the growth process in Growing Hierarchical Self-Organizing Map is the overall deviation of the input data as measured with the single unit SOM at layer 0. This unit is assigned a weight vector m_0 which is computed as the average of all input data. m_0 is as follows:

$$m_0 = [\mu_{01}, \mu_{02}, \dots, \mu_{0n}]^T$$

The deviation of the input data, i.e. the mean quantization error of this single unit is computed as:

$$mqe_0 = 1/d \parallel m_0 - x \parallel$$

where d represents the number of input data x.

After the computation of mqe_0 , training of the GHSOM starts with its first layer SOM. This first layer map initially consists of a rather small number of units, e.g. a grid of 2*2 units. Each of these units i is assigned an n-dimensional weight vector m_i where

$$m_i = [\mu_{i1}, \mu_{i2}, \dots, \mu_{in}]^T, m_i \in R^n,$$

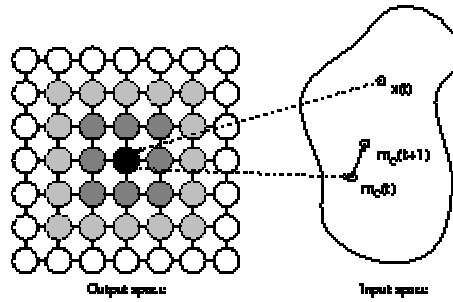
which is initialized with random values. The weight vectors have the same dimensionality as the input patterns.

GHSOM Training [3]:

The learning process of SOMS may be described as a competition among the units to represent the input patterns. The unit with the weight vector being closest to the presented input pattern in terms of the input space wins the competition. The weight vectors of the winner as well as units in the vicinity of the winner are adapted in such a way as to resemble more closely the input pattern.

Consider Figure 1 for a graphical representation of self-organizing maps. The map consists of a square arrangement of neural processing elements, i.e. units, shown as circles on the left-hand side of the figure. The black circle indicates the unit that was selected as the winner for the presentation of the input pattern $x(t)$. The weight vector of the winner, $m_c(t)$, is moved towards the input pattern, and thus, $m_c(t+1)$ is nearer to $x(t)$ than was $m_c(t)$. Similar, yet less strong adaptation is performed with a number of units in the vicinity of the winner. These units are marked as shaded circles in Figure 1. The degree of shading corresponds to the strength of adaptation. Thus, the weight vectors of units shown with darker shading are moved closer to x than units shown with lighter shading.

Figure 1: Architecture of a 7 * 7 self-organizing map



As a result of the training process, similar input data are mapped onto neighboring regions of the map. In the case of text document classification, documents on similar topics as indicated by their feature vector representations are grouped accordingly.

Algorithm for Training process [3]:

The degree of adaptation is guided by means of a learning –rate parameter α , decreasing in time. The numbers of units that are subject to adaptation also decreases in time such that at the beginning of the learning process a large number of units around the winner is adapted, whereas towards the end only the winner is adapted. These units are chosen by means of a neighborhood function hc_i which is based on the units' distances to the winner as measured in the two-dimensional grid formed by the neural network. By combining these principles SOM training, users may write learning rule as below:

$$m_i(t+1) = m_i(t) + \alpha(t) \cdot hc_i(t) \cdot [x(t) - m_i(t)]$$

where x represents the current input pattern, and c refers to the winner at iteration t .

In order to adapt the size of this first layer SOM, the mean quantization error of the map is computed over a fixed number x of training iterations as below:

$$MQE_m = 1/u \cdot \sum_i mqe_i$$

where u refers to the number of units i contained in the SOM m . mqe_i is computed as the average distance between weight vector m_i and the input patterns mapped onto unit i . The mean quantization error of a map is referred as MQE.

The basic idea is that each layer of the GHSOM is responsible for explaining some portion of the deviation of the input data as present in its preceding layer. Adding units to the SOMs on each layer is continued until a suitable size of the map is reached. The SOMs on each layer are allowed to grow until the deviation present in the unit of its preceding layer is reduced to at least a fixed percentage T_m . Obviously, the smaller the parameter T_m is chosen, the larger will be the size of the emerging SOM. Thus as long as $MQE_m \geq T_m \cdot mqe_0$ holds true for the first layer map m , either a new row or a new column of units is added to this SOM. This insertion is performed neighboring the unit e with the highest mean quantization error, mqe_e , after λ training iterations. This unit is referred as the error unit. The distinction whether a new row or a new column is inserted is guided by the location of the most dissimilar neighboring unit to the error unit. Similarity is measured in the input space. Hence, users insert a new row or a new column depending on the position of the neighbor with the most dissimilar weight vector. The initialization of the weight vectors of the new units is simply performed as the average of the weight vectors of the existing neighbors. After the insertion, the learning –rate parameter α and the neighborhood function hc_i are reset to their initial values and training continues according to the standard training process of SOMs. The same value of the parameter T_m is used for each map in each layer of the GHSOM.

Consider Figure 2 for a graphical representation of the insertion of units [3]. In this figure, the architecture of the SOM [3] prior to insertion is shown on the left-hand side where we find a map of 2*3 units with the error unit labeled e and its most dissimilar neighbor signified by d . Since the most dissimilar neighbor belongs to another row within the grid, a new row is inserted between unit's e and d . The resulting architecture is shown on the right-hand side of the figure as a map of now 3*3 units.

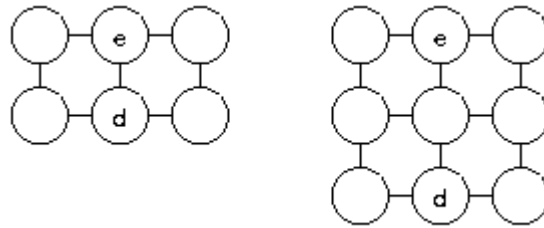


Figure 2: Insertion of units to a self-organizing map

As soon as the growth process of the first layer map is finished, i.e. $MQE_m < T_m \cdot mqe_0$, the units of this map are examined for expansion on the second layer. In particular, those units that have a large mean quantization error will add a new SOM to the second layer of the GHSOM. The selection of these units is based on the mean quantization error of layer 0. A parameter T_u is used to describe the desired level of granularity in input data discrimination in the final maps. More precisely, each unit i fulfilling the criterion given in the following expression will be subject to hierarchical expansion:

$$Mqe_i > T_u \cdot Mqe_0$$

Algorithm for row and column insertion [3]:

The training process and unit insertion procedure [3] now continues with these newly established SOMs. The major difference to the training process of the second layer map is that now only that fraction of the input data is selected for training which is represented by the corresponding first layer unit. The strategy for row or column insertion as well as the termination criterion is essentially the same as used for the first layer map. The same procedure is applied for any subsequent layers of the GHSOM[3].

The training process of the GHSOM [3] is terminated when no more units require further expansion. This training process does not necessarily lead to a balanced hierarchy, i.e. a hierarchy with equal depth in each branch. Rather, the specific requirements of the input data are mirrored in that clusters might exist that are more structured than others and thus need deeper branching. Consider Figure 3 for a graphical representation of a trained GHSOM. In particular, the neural network depicted in this figure consists of a single unit SOM at layer 0, a SOM of 2*2 units in layer 1, three SOMs in layer 2 i.e. one for each unit in the layer 1 map. Each of these maps might have a different number and different arrangements of units as shown in the figure. Finally, there are several SOMs in layer 3, which were expanded from one of the layer 2 units, just indicated by dotted arrows [3].

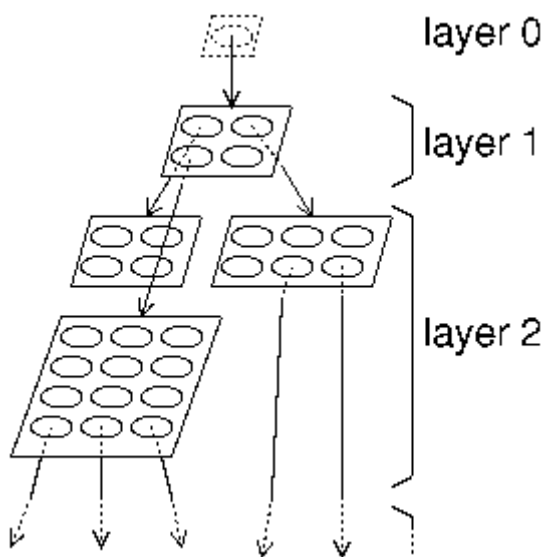


Figure 3. Architecture of a trained GHSOM [3]

To summarize, the growth process of the GHSOM [3] is guided by two parameters T_u and T_m . The parameter T_u specifies the desired quality of input data representation at the end of the

training process. Each unit i with $m_{qe_i} > T_m \cdot m_{qe_0}$ will be expanded, i.e. a map is added to the next layer of the hierarchy, in order to explain the input data in more detail. Contrary to that, the parameter T_m specifies the desired level of detail that is to be shown in a particular SOM. In other words, new units are added to a SOM until the MQE of the map is a certain fraction, T_m , of the m_{qe} of its preceding unit. Hence the smaller T_m , the larger will be the emerging maps. Conversely, the larger T_m , the deeper will be the hierarchy.

Users can traverse the hierarchy of GHSOM using the down and up link in the each GHSOM map. With the help of the down link, the document classification can be viewed at finer level. The trained GHSOM [3] have labels given to each cell. The various labels can then be used to identify clusters within the map by identifying regions, which are labeled with identical keywords. Having a set of 10 labels automatically to the single nodes leaves users with a somewhat clearer picture of the underlying text archive. It allows users to understand the reasons for a certain cluster assignment as well as identify overlapping topics and areas of interest within the document collection.

Applications of the classification system [2]:

1) Library System:

The library system uses the traditional systems classification that is not very much flexible and accurate as compared to document classification for the library system using neural networks.

2) Automatic subject indexing and automatic classification:

The automatic subject indexing has become very much easier when the schemes like the document classification with the neural network that gives very efficient results in the environments like Internet.

3) Document clustering:

Document clustering is the act of collecting similar documents into bins, where similarity is some function on a document. With the exception of Probabilistic Latent Semantic Analysis (PLSA), all use cosine similarity in the vector space model as their metric.

References:

[1]Kishan Mehrotra, Chilukuri K. Mohan, Sanjay Ranka, 'Elements of Artificial Neural Networks', Penram International Publishing, 1997, pp. 187-199

[2]Swapnil Satam, Sadashiv N.N. Khaunte, Kaustubh Thakur, 'A Project Report on Document Classification Using Neural Networks', 2005, pp.1-12

[3] The GHSOM Architecture and Training Process, <http://www.ifs.tuwien.ac.at/~andi/ghsom/description.html>