An overview on the use of neural networks for data mining tasks

Journal:	*WIREs Data Mining and Knowledge Discovery*
Manuscript ID:	Draft
Wiley - Manuscript type:	Overview
Date Submitted by the Author:	n/a
Complete List of Authors:	Stahl, Frederic; University of Portsmouth, School of Computing Jordanov, Ivan; University of Portsmouth, School of Computing
Keywords:	neural networks, data mining, classification&prediction, cluster analysis
Choose 1-3 topics to categorize your article:	Prediction (DEAH) < Technologies (DEAA), Classification (DEAC) < Technologies (DEAA), Structure Discovery and Clustering (DEAJ) < Technologies (DEAA)

SCHOLARONE™
Manuscripts

Wiley Interdisciplinary Reviews

DATA MINING AND KNOWLEDGE DISCOVERY

Please select the appropriate article type (per your Contributor's Agreement) by double-clicking on the associated check box.

☒	Overview	Overviews will provide a broad and relatively non-technical treatment of important topics at a level suitable for advanced students and for researchers without a strong background in the field.	~8,000 words 16 figures/tables 25-50 references
☐	Advanced Review	Advanced Reviews, aimed at researchers and advanced students with a strong background in the subject, will review key areas of research in a citation-rich format similar to that of leading review journals.	~5,000 words 10 figures/tables 50-75 references
☐	Focus Article	Focus articles will describe specific real-world issues, examples, implementations, etc. These articles will be technical in nature.	~3,000 words 6 figures/tables 15-25 references
☐	Opinion	Opinions provide a forum for thought-leaders, hand-picked by the editors, to provide a more individual perspective on the field in question.	~3,000 words 6 figures/tables 15-25 references

An Overview of the Use of Neural Networks for Data Mining Tasks

First author: Full name and affiliation; plus email address if corresponding author
[*Frederic Stahl; University of Portsmouth; Frederic.Stahl@port.ac.uk]

Second author: Full name and affiliation; plus email address if corresponding author
[Ivan Jordanov; University of Portsmouth]

Abstract

In the recent years the area of data mining has experienced a considerable demand for technologies to extract knowledge from large and complex data sources. There is a substantial commercial interest as well as research activities in the area that aim to develop new and improved approaches for extracting information, relationships, and patterns from datasets. Artificial Neural Networks (NN) are popular biologically inspired intelligent methodologies, whose classification, prediction and pattern recognition capabilities have been utilised successfully in many areas, including science, engineering, medicine, business, banking, telecommunication, and many other areas. This paper highlights the implementation of NN, trained by supervised and unsupervised learning, for pattern recognition, classification, prediction and cluster analysis, from a data mining perspective, and discusses some examples of their usage in bioinformatics and financial data analysis tasks.

Introduction

Currently, ubiquitous computer systems record our habits, decisions, behaviours and lives daily. We use debit and credit cards to pay and club cards to save on our shopping. We are recorded on CCTV cameras and leave our information in the World Wide Web using applications such as email, blogs, twitter or social networks. But it is not only us, the individuals that constantly generate data records in various large data bases every day, it is also scientific applications that record large amounts of data constantly. For example, the NASA's system of earth orbiting satellites which generates a terrabyte of data every day[61]; the human genome project[1] that stores the genetic blueprint of our bodies; molecular simulations that study the dynamic behaviour of molecules and generate large amounts of data stored in distributed data warehouses[53]. Considerable advances in storage technology made it possible to store all this information at relatively low cost. However, just having the data available in databases does not make us more knowledgeable. We may be 'information rich', but still 'knowledge poor'[15] and the gap between recording information and understanding it is growing[7].

Data mining can be used to close the gap between collecting and understanding the information. It can be regarded as the extraction of meaningful, and potentially useful patterns from large data bases[65], or referring to the earlier used expressions, as the 'understanding' of recorded 'information'. For the extraction and discovery of knowledge, data mining employs a variety of intelligent methods, among which, NN are frequently used for different data mining tasks, including classification and cluster analysis. The authors of[28] highlight some of the advantages of using NN for data analysis: they are tolerant to noise; can deliver high accuracy on complex non linear mappings and at the same time can be implemented on parallel hardware; they are highly maintainable (can easily be updated with fresh data) and can easily be automated.

This paper highlights the use of NN in the field of data mining. First, the basic Knowledge Discovery process with data mining is outlined and the basics of NN are introduced. Next, the NN are put into the context of data mining and the Knowledge Discovery process and their usage for data mining tasks is described. This is followed by a summary of actual applications of NN, more specifically in bioinformatics and finance areas and finally, concluding remarks are made.

KNOWLEDGE DISCOVERY AND DATA MINING

Knowledge Discovery from Data

Data Mining is often described with the metaphor of 'gold mining'[15]. In it, the 'gold' is the actual valuable 'knowledge' to be extracted. Thus, according to this metaphor, 'data mining' should be better named 'knowledge mining' as pointed out in[15]. In fact, another popular term used, instead of 'data mining', is '**K**nowledge **D**iscovery from **D**ata' (KDD). In the KDD process, data mining is often regarded as just one, but very important step.

The following steps (Fig.1) describe the main activities involved in the KDD process:

1. *Integration and Cleaning*: Integration of different data sources into one data store and removal of inconsistent data and noise;

2. *Selection and Transformation*: Selection of data that is relevant to the analysis task and its transformation into a standard format. Also, data reduction techniques may be applied in this step;

3. *Data Mining*: Applying intelligent methods to the preprocessed data in order to extract meaningful patterns, such as decision rules;

4. *Evaluation and Interpretation*: Identification of 'interesting' aspects of the extracted patterns and representing them in a meaningful way.

Data Mining

The data mining step in the KDD process aims to extract meaningful patterns from large amounts of data, using different intelligent techniques[15]. There are two typical fields of application that can be automated by the use of data mining of large databases: (a) the prediction of trends and behaviours; and (b) the extraction of unknown patterns and relationships.

With respect to (a), classifiers can be generated from labelled data samples. These classifiers can then be applied to previously unseen data instances in order to classify them. This is one of the most important data mining techniques. Typical applications of classifiers are in the area of marketing[58]. Data mining can be applied on past promotional mailings in order to classify receivers into groups that are likely or unlikely to take advantage of the promotions. This information can be used in the future to save money by only targeting groups that are likely to respond to the promotion. Further applications of classification are risk assessment for credit card loan[50] and mortgage applications[10], as well as bankruptcy predictions. However, prediction comprises not only classification of unlabelled data, but also prediction of future trends. For example, prediction of future trends could be useful for forecasting the future values of stock market papers, stock market indices, etc[69].

With respect to (b), there are applications that are concerned with the discovery of new, hidden knowledge/patterns in large data bases, rather than predictive modelling. For example, drug designers are interested in discovering knowledge from gene expression datasets about influences of the designed drugs on gene expression levels. Drugs may promote or suppress certain genes which in turn may influence further genes and thus create a complex pattern[70]. Also in the business area patterns that indicate which products are likely to be bought together are useful for product placement[71]. Or in banking, discovered patterns may indicate fraudulent credit card transactions[22].

Commonly used methods in data mining comprise:

- *Artificial Neural Networks (NN)*: NN are biologically inspired predictive models (mimicking the functioning of human brain), that can learn and map linear and nonlinear functions[52];

- *Genetic Algorithms (GAs)*: GAs are nature inspired evolutionary approaches, that simulate natural selection and survival of the fittest; they are often used for global optimization tasks, and also to optimise other pattern recognition algorithms[56];

- *Generalised Rule Induction (GRI)*: generates rules about significant relationships, rather than just predicting a class value[51, 2];

- *Top Down Induction of Decision Trees (TDIDT)*: Induces classification rules in the intermediate form of a tree structure[38];

- *K-Nearest Neighbour (KNN)*: It classifies unlabelled data instances with a majority class of the *k* most similar data instances in the training set[7].

ARTIFICIAL NEURAL NETWORKS

This paper focuses on the use of NN in the data mining step of the KDD process (Fig. 1). This section highlights NN in general. The use of NN in data mining is discussed in the 'Neural Networks in Data Mining' Section. NN are computer paradigms inspired from the biological NN that can be used to generate predictive models from historical data.

They have been applied in data mining for prediction, classification and cluster analysis. Initially, the application of neural networks has faced strong opposition due to the fact that they are complex structures, difficult to interpret and need a long training time[33]. Yet NN produce compelling accuracy and are highly adaptable across a large range of applications. There have been also developed technologies that extract explicit rules using NN, hence NN are currently experiencing a renaissance in data mining[33].

Analogy to Biological Neural Networks

The term artificial neuron can be traced back to the 1940s, in an early work of McCulloch and Pitts[32], where it was described as a mathematical simulation of a biological neuron[32]. This paper highlights the most common form of an NN, the Multi Layer Perceptron (MLP) and briefly introduces alternative topologies.

A biological neuron, as illustrated in Fig. 2(a), sends a signal if the collective influence of all its inputs reaches a threshold level (*axon hillock*'s threshold) – the neuron is said to **fire**. The impulse is send from the *cell body* (*soma*), through the *axon*, which influences the *dendrites* of the next neuron over narrow gaps called *synapses*. They translate the pulse in some degree into **excitatory** or **inhibitory** impulse of the next neuron. It is this receiving of signals, summation and transferring procedures that are emulated by the artificial neuron (a perceptron, Fig. 2(b)).

As in elaborate 'Lego' constructions and integrated circuits, which are built from a multitude of simple constituent units that interact with one another, the NN consists of organised in layers, interconnected neurons, as shown in Fig. **Error! Reference source not found.**.

Although the units from Fig. 3 can perform very simple tasks (as summation and transferring signal depending on the sum), when organised in NN, they can solve complex mapping, classification and recognition tasks. In the NN literature these units are also often referred to as neurons, nodes, or perceptrons. Neurons influence each other using 'weighted' connections between them; positive weights have stimulating influence, while the negative ones have inhibiting influence. The number of neurons in the input and output layers depend on the problem's inputs and outputs, and the number of units in the hidden layer depends on the complexity of the problem or concept to be represented and solved. The higher the absolute value of the weight the stronger the influence of one neuron to another. The input layer receives data from the outside world in the form of a vector which is weighted and passed through the connections to the (first) hidden layer. In each hidden unit

the incoming products of the weights and the input values are summed and 'squashed' through an activation/transfer function that generates an output value, which in turn is input for the next layer and so on. Two commonly used transfer functions are the sigmoid and the hyperbolic tangent functions.

The calculation of weighted input for the kth unit of the hidden layer is then:

$$v_k = \sum_{i=1}^{n} w_{ki} x_i \, , \, k=1,...,m;$$

where $x_i, i=1,..., n$ are the inputs, w_{ki} are the weights and m is the number of units in the layer. The output y_k of neuron k is then calculated by a transfer function: $y_k = \phi(v_k)$ and y_k is then input value for the next layer of units.

Training and Testing of Neural Networks

An NN is typically learned using training, validation and testing phases. During the training phase the weights of the NN are adjusted using training data and the validation data subset is used to avoid so called 'overfitting' (when the NN starts learning more to memorise, rather than to generalise), and in the testing phase the NN is checked whether and how well it has learned the task. For predictive data mining, the training and testing in its basic form divides the dataset into two subsets, a train subset and a test subset. The train subset is used to train/learn the NN and the test subset is used to check whether the NN learned the knowledge and can generalise. Generalisation is the ability of a trained NN to correctly classify, fit, map, recognise input instances/samples/examples that have not been used during the training.

In the *training phase* there are several learning rules that can be used and they determine how the weights are being adjusted. However, most learning approaches can be categorised into supervised and unsupervised learning.

- In *supervised learning* the correct output for each input training data instance is a-priory known. The training input instances are fed into the network, propagated, the actual output of the network is compared with the target one, and if there is a difference, then the weights are adjusted in order to minimise this difference.

- In *unsupervised learning* the output is not known, and the network itself adjusts the weights in order to map inputs to outputs without any additional knowledge.

Backpropagation[63, 45] is the most popular training technique for supervised learning of MLPs. It uses gradient based (deterministic) methods for minimising an error function, which represents the difference between the actual and the desired output, through adjusting the weights of the network. During this process there is a danger of overfitting, when the error function becomes zero, but the NN is learned to memorise, rather than to generalise. In this case, an additional validation subset is used in parallel to the training one and when the validation error starts to increase, the training should stop (regardless of further decrease of the training error).

In the *testing phase* there are no changes performed on the networks weights, once trained, the NN weights are fixed. During this phase, a testing (unseen during the training) data subset, is presented to the NN, in order to validate how well the NN can generalise (in other words, its ability to produce the target output when presented with unseen input instances).

Further possible learning rules that can be used for adjusting the weights of a NN are, for example, Hebb's rule[12]; the Delta Rule[64]; Competitive Learning[26]; and global optimisation techniques, such as Genetic Algorithms[72], Simulated Annealing[25], etc.

Artificial Neural Network Types

There are a number of variations of NN topologies and architectures. Most of them are feedforward NN (FFNN). In FFNN the information moves only in one direction from the input nodes through the hidden and to the output nodes. Contrary to the FFNN, recurrent NN have a bi-directional data flow, as they propagate data from later processing stages into earlier ones. MLPs can differ in the number of hidden, input and output units. In Radial Basis Function (RBF) networks[44] each neuron in the hidden layer represents a point in the input space. The activation of a hidden neuron depends on the distance of the input values to the point that the hidden neuron represents in the[65]. However, Radial Basis Function networks will not be discussed further in this article, it is referred to[44] for more information. A special NN architecture for cluster analysis will be highlighted in the 'Cluster Analysis' Section.

NEURAL NETWORKS IN DATA MINING

This section describes the basic data mining process in the context of NN. Fig. **Error! Reference source not found.** takes a closer look into the data mining step of the KDD process illustrated in Fig. **Error! Reference source not found.**. This is done in the context of NN based on Ni's description in[33]. Input to the data mining process is the preprocessed data and the patterns generated by the data mining step are called 'useful rules'.

Below is a description of the steps in the data mining process based on NN (Fig. 4):

- *Data Preparation*: converts the data into a suitable format for the particular neural network being used. This involves replacing missing values or deleting data instances with missing values; filtering out instances that introduce noise; and converting categorical data values into numerical ones as NN process numerical data;

- *Learn NN*: The NN is trained, validated and tested for generalization on the prepared data and rules are generated in the form of network weights. In data mining, these learned NN can be treated as 'black boxes' and already be applied;

- *Rule Generation*: Often rules are extracted from NN in order to be able to assess and refine the generated rule set;

- *Rule Assessment*: Loosely speaking, rule assessment validates the rule sets (the NN quality) and possibly refines certain rules. Possible assessment criteria for a rule set or single rules may be the rules, rule sets or the NN's accuracy; finding the optimal order in which rules are being applied; detecting contradicting rules in order to manipulate them, etc.

The following sections highlight some data mining tasks, in which NN are used.

Classification and Numeric Prediction

Two of the most important tasks in data mining are classification and numeric prediction of variables of data instances. An example for *classification* is the analysis of credit card applications. Credit card institutions wish to assess whether issuing a credit card is 'safe' or 'risky'. Usually, a classifier is generated that can be used to predict a categorical class label such as 'risky' or 'save' for previously unseen data instances. An example of *numeric prediction* is a sale at a mail-order house. A marketing analyst is interested in predicting how much a certain customer or a group of customers is likely to spend. A model or predictor is generated that predicts a numeric value.

Classification and numeric prediction usually happen in a two step process as shown in Fig. **Error! Reference source not found.**. First, the classifier is generated on a training data set, of which the class label or the numeric value of interest is known. This is also known as the *training phase*. In the second step, the model is used for classification or numeric prediction. However, these classifiers tend to overfit on the training data, meaning that they memorise anomalies and noise rather than learning the general trend. Thus, this second step is often used to validate the classifier/predictor using test data with known class labels, in order to verify the classifier's accuracy and ability to generalise. There is a variety of techniques used for training and testing (depending on the size and quality of the available data sets), e.g., *split sample*, *k-fold crossvalidation, bootstrapping*, etc.[7, 15, 65], which discussion is however outside the scope of this article.

Three popular approaches for classification/predictions are *Bayesian Classification*, *Rule Based Classification* and *Classification by Backpropagation*. Bayesian classifiers use statistical methods based on the 'Bayes theorem'[27], which do not predict a particular class value, but the likelihood of a data instance belonging to a particular class. Furthermore, they are fast and thus can be applied to large data volumes. Rule based classifiers are in the form of decision rules, these can be decision trees or IF-THEN rules as illustrated in Fig. **Error! Reference source not found.**. A notable classifier for Decision trees is 'C4.5'[38], and for IF-THEN rules 'Prism'[8]. Decision trees are often converted into IF-THEN rules. Both approaches have the problem that they suffer from noisy training datasets, tend to learn the noise, hence the generated rule sets need to be manipulated in order to gain a high classification accuracy. However, they are often used if insights are needed about the predictions that have been made.

NN, as outlined earlier, can be used for classification purposes. In particular, Backpropagation MLPs are used as classifiers and predictors[15]. They have been initially criticised as less suitable for data mining due to their poor interpretability. This is because their symbolic meaning is hidden in the neurons' connections and thus is difficult to be assimilated by humans, hence NN are often referred to as a 'black box'[15]. Their high tolerance to noisy data and the fact that they can classify patterns they have not been trained on, makes them an excellent technology for data mining. Furthermore, NN are well suited to learn continuous data without prior discretisation and predict numeric values, which is not possible with most rule based and Bayesian classifiers. Their ability to produce non linear models enables them to approximate any function with a reasonable accuracy[20, 11, 19]. One of their disadvantages is that they are computationally not very efficient, for example, Backpropagation can be entrapped in local minima, yet they are inherently parallel and thus can easily be executed on

parallel computer architectures. In order to do so for decision trees and rule based algorithms, parallel versions need to be developed first, such as the discussed in[49].

NN are gaining popularity in data mining due to advances in rule extraction techniques from trained NN, which enhance the NN' suitability for classification and numeric prediction. A fully connected NN is usually too complex to be described in the form of rule sets, hence weighted connections between nodes that have least effect on the trained NN are removed prior to the rule extraction. A weighted link may be removed if it, for example, does not decrease the networks classification accuracy during testing. A popular approach to extract rules from NN is the NeuroRule algorithm that expresses rules as binary input[48, 29]. The basic steps of NeuroRule are outlined below:

1. Train the NN;

2. Prune the network by removing redundant connections whilst maintaining its accuracy;

3. Discretise activation values of the neurons in the hidden layer and the variable values of the input layer;

4. Generate rules that explain the relationship between the output and the hidden layer;

5. Generate rules that explain the discretised activation values of the neurons in the hidden layer in relation of the network input values;

6. Generate a set of rules that relate the inputs and outputs of the NN by merging the two rule sets created in step 4 and 5.

Further rule extraction methods comprise the SUBSET algorithm[46, 13] and the MofN algorithm[59].

Cluster Analysis

Cluster analysis aims to organise a dataset into groups of 'similar' data objects according to some distance function. Cluster analysis is an unsupervised technique, which is exploratory in its nature, thus usually performed when there exist no or very little prior knowledge about the hidden concepts in the dataset. There are many application areas, such as finance, (e.g., an analyst may want to find clusters of companies that have similar financial performances); health and medical applications, (an analyst may want to cluster patients with similar symptoms or reactions of patients to a certain drug); marketing, (building clusters of customers with similar shopping preferences), etc.[7].

The authors of[67] describe the clustering process in four steps as shown in Fig. **Error! Reference source not found.**.

1. Usually, the first step is to select and extract features – a very important part of the process, especially if there is a large amount of features in the dataset. It improves the likelihood of finding useful clusters and also increases the algorithms' speed. Feature selection aims to only select the most relevant characteristics for distinguishing a cluster, whereas feature extraction aims to create novel features from two or more original characteristics.

2. Clustering algorithms usually use proximity metric in order to measure the resemblance of individual data instances to each other. The clustering techniques use this proximity metric in

order to build groups of data instances that are 'similar' (in other words, being close in the feature space).

3. Clustering methods tend to build clusters regardless of whether there is an apparent structure in the data that determines different clusters or not. Also, different clustering algorithms tend to converge to different solutions, regardless of using the same datasets. Hence, clusters need to be validated in order to strengthen the users or analysts confidence in the found solution.

4. Finally, the built clusters need to be interpreted in order to provide insights into the meaning of the original data.

Two popular clustering methods comprise *k-means*[16] and *agglomerative hierarchical clustering*[15, 74]. K-means is one of the simplest clustering techniques, an *exclusive clustering algorithm*, meaning that each data object can only be assigned to one cluster. The analyst has to decide the number of clusters *k* to be built, and usually they are initialised randomly. Next, each data point is assigned to a cluster whose centroid (the centre of the cluster according to the used proximity metric) is nearest. After all data objects are assigned to clusters, the centroids are recomputed, and each data object is reassigned. This is repeated until there are no changes in the clusters. The disadvantage of the *k-means* algorithm is that the analyst needs to have some prior knowledge of the number of clusters in order to generate meaningful ones.

In *agglomerative hierarchical clustering*, each data object is initially considered as its own cluster. Next, the distance between each pair of clusters is calculated and the closest pair of clusters is merged into a single cluster. This is repeated until all data objects are assigned to a single cluster. Agglomerative hierarchical clustering produces a dendogram, a binary tree of all clusters build in each iteration.

Kohonen networks (also known as ***feature maps***, or ***topology-preserving*** maps, or simply SOMs)[26], are very popular methods used for cluster analysis[15]. Each output node will represent one cluster (class) of inputs and is associated with a *weight vector*, which size is the same as the *input vector*. There are also lateral connections between the neighbouring output neurons (Fig. 8).

Unlike the FFNN, there is no hidden layer and no transfer function used. Also, during the unsupervised learning neither the clusters, nor which node corresponds to which cluster is specified. The SOM network should work out the clusters for itself and organize the output nodes accordingly. For any given sample input, the output units compete to represent it, and the node with the closest weight vector becomes the winner.

The essential process in the formation of the SOM comprises the following steps:

1. *Competition*: For each input pattern the neurons in the output layer will compete to represent this pattern. The neuron with the closest weight vector to the input vector will update its weights and will become a winner. The closeness is defined by a distance metric (e.g., *Euclidian*, *Hamming*, or others);

2. *Cooperation*: The winning neuron determines the topological neighbourhood, by exciting its surrounding neurons. This is called cooperation, because it strengthens the neuron that is the winner and to some extend the neurons that are close to it;

3. *Synaptic Adaptation*: It adjusts the weights of the connections in order to enable the excited neurons to update their weight vectors to become 'closer' to the input vector.

Cluster analysis using SOMs does not require prior knowledge about the number of clusters, which represents a clear advantage compared with the *k-means* technique. SOMs can also be used to detect outliers in the data and clusters are built in order to isolate instances that are distinctly different from the remaining instances. Sometimes outliers are of more interest than the actual clustered data instances. For example, let the data instances represent branches of a retailer chain. If there are several clusters that comprise branches with different increases of revenue, then it is likely that a branch whose revenue did not increase is not inside any cluster. A closer look into this branch and comparison with branches in clusters that have had an increase in revenue may reveal the reason for its bad performance. For example, this could be a result of distinctly different advertising strategy (the outlier branch may have advertised in the local newspaper, whereas the remaining (non outlier) branches used radio advertisements, or, there may be a smaller customer parking space, comparing with the others, etc.).

APPLICATIONS OF NEURAL NETWORKS

Artificial neural networks have been used in many areas of bioinformatics[75] and financial forecasting[69]. This section highlights general problems in these two chosen areas of NN application and discusses some particular examples.

Bioinformatics

Bioinformatics is an interdisciplinary science (which encompasses computer science, mathematics and statistics), that aims to refine biological information into biological knowledge using computer based methodologies. It has been coined in 1978 by Paulien Hogeweg[17, 18] and gained popularity with the mapping of the human genome in 2001[1]. Dynamic modelling of complex biosystems, sequence analysis, protein structure analysis are just few problem areas that fall within the scope of bioinformatics, but it could be broadly categorised into genomics and proteomics. Genomics investigates the structure and function of a very large number of genes, including mapping sequencing of genes for the entire genome of organisms, and proteomics studies the protein-protein and protein-DNA interactions, enzymatic and biochemical pathways of an organism[54]. Proteins are the building blocks of an organism; the blueprint for a particular protein is encoded in a particular gene of the organism's genome. Proteins are composed of a chain of amino acids and genes are essentially composed of base pairs, consisting thymine, adenine, cytosine and guanine[73]. A gene is a substring of the DNA that codes for a particular protein. Within a gene, a sequence of three consecutive base pairs codes for a particular amino acid in the protein. Now, loosely speaking, in genomics the sequence of base pairs and expression levels of genes are analysed, whereas in proteomics the protein's chemical structure and thus its function is analysed. The authors of[39, 47, 55, 66] outline approaches of using NN for the analysis of protein structure prediction and gene sequence analysis.

One of the biggest problems associated with data mining algorithms in bioinformatics is the curse of dimensionality[5]. While the number of genes in the *Escherichia coli* bacterium is just 4288, the relevant number of genes in a human genome is estimated about 35000, meaning that there are 35000 genes that code for about 35000 proteins. This way microarray datasets can easily reach up to

35000 features (depending on the microarray chip). Yet patterns in this data are often described by a small subset of the feature space. Also, measurements of gene expression levels are generally very noisy. Genomics data is described by a very large number of features compared to the limited number of samples. This makes achieving high accuracy very difficult for task for the predictive algorithms. Hence, feature selection strategies[31] are used to overcome the curse of dimensionality.

Sequence Alignment

In bioinformatics, a sequence alignment is a way of arranging the sequences of DNA base pairs, or protein amino acids, to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences. Following are two references to particular applications where NN have been successfully used for such sequence alignments[4, 23]. The authors of[4] developed a molecular alignment method using a Hopfield NN. They successfully reproduced the three-dimensional structure of the real molecular alignments, which have been obtained experimentally using X-ray crystallography. The authors of[23] developed a neural network architecture named GenTHREADER that was used to predict the similarity between gene sequences. In two cases GenTHREADER successfully predicted the similarity between two structures.

Promoter Sequence Detection

Promoters are regulatory sequences in the genome that are necessary for the initiation of the processes that read the DNA sequence of a particular gene in order to synthesize proteins. Promoters are regions on the gene that mostly control the biological activation of the gene[76]. An early investigation[30] describes the problem of detecting promoter genes in the context of NN. Because only a small portion of promoter sequences are known, NN can only be trained on a limited subset of promoter sequences versus random ones; yet NN achieve a high accuracy in detecting real promoter sequences. The authors of[24] used a multilayered NN successfully for the prediction of a particular promoter sequence (the mycobacterial sequence). The authors of[78] developed a technique to differentiate promoter regions from the rest of the genome based on an unsupervised SOM, which leads to significantly fewer false predictions than existing promoter prediction techniques[78].

Gene Expression Analysis and Gene Regulatory Networks

The process that converts the gene's base pair sequence into proteins is called gene expression. The more and the faster proteins from a certain gene are synthesised, the higher the expression level of this gene. Gene expression levels can be measured using microarray technology[36]. Gene expression levels are often used to compare regular and cancerous cells, e.g., the authors of[34, 3, 62] have been using NN successfully on classification problems in the context of cancerous cells. Gene expression data is also used to create gene regulatory networks; these are maps that describe how genes influence each other's expression levels. In[41, 43] NN have been utilised successfully to identify gene-gene interactions in order to create gene regulatory networks. The authors of[41] developed clustering technique called adaptive double self-organizing map (ADSOM), which does not require prior knowledge about the number of clusters. The authors of[43] show that machine learning strategies for optimising a NN architecture, for the identification of gene-gene interactions in complex human diseases, may be preferable to traditional trial-and-error approaches.

Protein Structure Prediction

Proteins are composed of a chain of up to twenty different amino acids. A protein chain folds in a particular way due to chemical forces between amino acids. Important is the 3D structure of the protein chain as it determines the protein's function. Hence the prediction of the 3D structure of a protein from only knowing the amino acid sequence is important. However, predicting the 3D structure of a protein solely based on the chemical properties of the amino acid sequence is computationally too expensive; hence research follows alternative approaches[53, 6]. One approach is the alignment of the amino acid sequence of the protein with the unknown structure with the amino acid sequences of homologous proteins with known 3D structures. Yet this method only works if there are homologues proteins in the database, hence alternative methods such as NN are needed. Following are two references to particular applications where NN have been successfully used for protein structure prediction[42, 14]. The authors of[42] use separate NN tailored for predicting three standard structures (alpha-helix, beta-strand, and coil). The authors of[14] created a method based on single hidden layer feed-foreward NN for predicting different types of beta-turn structures from the amino acid sequence of a protein. NN based technologies have also been used to fight diseases such as HIV. For example the authors of[77] propose a framework using NN based technologies in order to predict the specifity of HIV-1 protease cleave cites (enzymes that postprocess proteins).

Finance

NN are of practical interests in financial data mining as it has been shown in[79, 40] that they often perform better than statistical approaches for financial forecasting, fraud detection, bankruptcy prediction and marketing.

Financial Market Forecasting

Data mining for financial data has been particularly interesting for financial time series prediction for stock markets in order to maximise the trading revenue. A recent study used a MLP architecture in order to predict the Shanghai composite index[69] based on historical data of the closing prices of the stock market. In order to overcome the lack of explanatory capabilities of the NN, trading strategies (for example buy & sell & hold) have been incorporated. Furthermore the system predicted possible returns. Another recent application for NN is the forecasting of the next day's electricity price in the Spanish electricity market[35]. Their approach is based on MLPs and historical time series data about the electricity demand and price. For a comprehensive review about neural network technologies their application on forecasting stock markets, it is referred to[80].

Credit Card Fraud Detection and Risk Assessment

Credit card fraud is a crime that is defined as the use of credit cards or credit card information without the knowledge of the credit card's owner. Card issuers and merchants employ techniques that screen credit card transactions and extract customer behaviour patterns in order to detect fraud[57]. Data mining techniques can be used to extract such patterns[22], for example NN can be trained on historical transactions and are able to adjust to new forms of fraud. The authors of[68] use SOMs in order to create a profile of a typical cardholder's behaviour. If there are new transactions, then the deviation of the new transactions to the learned profile is used in order to detect suspicious transactions. Also[37] employs SOMs in order to detect fraud, but in real time without prior knowledge about spending patterns.

The decisions on whether to or not to issue credit cards to certain applicants can be vital for financial institutions. This is because issuing to many credit cards to defaulters will result in considerable financial losses. Recently credit analysts are looking for alternative algorithms to statistical models[81] that can provide a higher accuracy and NN are considered to be an adequate alternative[81]. The authors of[86] use logistic regression and MLPs in order to create a credit scoring model for lending decisions for agricultural loans in Thailand.

Bankruptcy Prediction

A company is bankrupt if it has serious and continuing losses and its debts are beyond its assets[21]. A reliable prediction of bankruptcy is obviously important to many stakeholders around this company such as banks, customers, creditors, governments etc.

The authors of[9] adopted data mining techniques and NN for creating a predictor of financial distress using historic financial data (such as financial statements, balance sheets and income statements) of 68 Taiwanese companies. Most approaches of bankruptcy prediction using NN are based on a single NN, however, the authors of[60] considers the usage of an ensemble of several NN in order to predict financial distress. The authors of[82] derived a method for assessing the corporate financial health of various companies. For this they use a hybrid of Multivariate Discriminate Analysis (MDA) and a SOM. Financial ratios drawn from balance sheets and the the MDA's Z-scores on these ratios are the input to the SOM. The system correctly estimated the corporate financial health in Serbia for 95% of companies in 2008 and for 94% of companies in 2009.

Marketing

The application of NN in marketing is popular due to their ability to model nonlinear relationships between variables[83]. Applications of NN in marketing range from Market Segmentation, Market Response Prediction, New Product Launch, Sales Forecasting, Consumer Choice Prediction, etc.[83]. The authors of[84] used a NN approach to examine consumer behaviour in choosing a hospital in southern Taiwan. The consumer behaviour was categorised in four different consumer styles. Then a MLP with two hidden layers using backpropagation was trained in order to derive a classification model for recognising different consumer styles. Their model achieved a classification accuracy of 85.1%. The authors of[85] propose the hierarchical self-organizing segmentation model (HSOS) appraoch as a system for visualising hierarchical clusters step by step. The system is intended to aid a manager as a visualisation tool for market segmentation. Their experimental evalution shows that HSOS's visualistion capabilities allow a better understanding of market segmentatoin than traditional hierarchical clustering approaches. NN have also been used for analysing customer preferences towards certain kinds of shops. For example the authors of[87] have used an MLP in order to explain customers preferences towards traditional and web stores for certain products.

Conclusion

This paper presents the NN as useful technology for data mining tasks. It first highlights how data mining fits into the knowledge discovery process and then, in turn, how NN fit into the data mining process. Subsequently, the use of NN for two very important data mining tasks, namely classification and clustering is described. This is followed by a summary of actual applications of NN for classification and clustering problems in bioinformatics and financial data analysis.

Although formerly criticised for producing incomprehensible models and requiring too extensive learning approaches, NN have been experiencing a renaissance in the recent decades. Their successful use in the data mining field is a result of their ability to produce models and to extract explicit and meaningful rules from these models. Also, because the NN are in their nature highly parallel, the advances in shared memory parallel processor architectures allow to speed up the learning process considerably. However, the most important factor for NN' recent popularity in data mining and data analysis is due to their compelling classification accuracy, especially in domains with noisy datasets, such as in the discussed finance and bioinformatics areas. Speeding up NN using super computers may be unattractive for modest size organisations as they may not have the financial power to afford parallel machines, yet they may still want to process large datasets using NN (using their local area network to execute distributed NN algorithms concurrently). Hence a possible future direction of research for NN in data mining is the development of distributed NN algorithms that can be executed on loosely coupled systems using 'off the shelf' computer workstations.

References

[1] www.ornl.gov/sci/techresources/Human_Genome/home.shtml (accessed July 2011)

[2] Agrawal R, Srikant R. Fast algorithms for mining association rules in large databases. *In VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases*, San Francisco, CA, USA,. Morgan Kaufmann, 1994:487-499.

[3] Ando T, Suguro M, Hanai T, Kobayashi T, Honda H, and Seto M. Fuzzy neural network applied to gene expression profiling for predicting the prognosis of diffuse large b-cell lymphoma. *Cancer Science 2002*, 93(11):1207–1212.

[4] Arakawa M, Hasegawa K, Funatsu K. Application of the novel molecular alignment method using the hopfield neural network to 3d-qsar. *Journal of Chemical Information and Computer Sciences 2003* , 43(5):1396–1402.

[5] Bellman R. *ADAPTIVE CONTROL PROCESSES: A GUIDED TOUR*. Princeton University Press, Princeton, New Jersey, 1961.

[6] Berrar D, Stahl F, Silva CS, Rodrigues JR, Brito RMM. Towards data warehousing and mining of protein unfolding simulation data. *Journal of Clinical Monitoring and Computing 2005*, 19:307–317.

[7] Bramer MA. *Principles of Data Mining*. Springer, 2007.

[8] Cendrowska J. PRISM: an algorithm for inducing modular rules. *International Journal of Man-Machine Studies 1987*, 27(4):349–370.

[9] Chen WS, Du YK. Using neural networks and data mining techniques for the financial distress prediction model. *Expert Systems with Applications 2009*, 36(2):4075–4086.

[10] Collins E, Ghosh S, Scofield C. An application of a multiple eural network learning system to emulation of mortgage underwriting judgements. In *IEEE International Conference on Neural Networks 1988*, USA:459–466.

[11] Cybenko G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS) 1989*, 2:303–314.

[12] Hebb DO *The Organization of Behavior: A Neuropsychological Theory*. Wiley, New York, 1949.

[13] Fu L. Rule learning by searching on adapted nets. In *National Conference on Artificial Intelligence*, 1991 590–595.

[14] Kaur H,   Raghava GP. A neural network method for prediction of beta-turn types in proteins using evolutionary information. *Bioinformatics 2004*, 20(16):2751–2758.

[15] Han J, Kamber M. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2001.

[16] Hartigan JA, Wong MA. A K-means clustering algorithm. *Applied Statistics 1979*, 28:100–108.

[17] Hogeweg P. Simulating the growth of cellular forms. *Simulation 1978*, 31(3):90–96.

[18] Hogeweg P. The roots of bioinformatics in theoretical biology. *PLoS Computational Biology 2011*, 7(3) 2011.

[19] Hornik K. Approximation capabilities of multilayer feedforward networks. *Neural Networks 1991.*, 4:251–257.

[20] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. *Neural Networks 1989*, 2:359–366.

[21] Hua Z, Wang Y, Xu X, Bin Zhang, and Liang Liang. Predicting corporate financial distress based on integration of support vector machine and logistic regression. *Expert Systems with Applications 2007,* 434–440.

[22] Huang C, Chen M, Wang C. Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications 2007*, 33:847–856.

[23] Jones DT. Genthreader: an efficient and reliable protein fold recognition method for genomic sequences. *Journal of Molecular Biology 1999*, 287(4):797 – 815.

[24] Kalate RN, Tambe SS, Kulkarni BD. Artificial neural networks for prediction of mycobacterial promoter sequences. *Computational Biology and Chemistry 2003*, 27(6):555 – 564.

[25] Kirkpatrick S, Gelatt CD, Vecchi JR, Vecchi MP. Optimization by simulated annealing. *Science 1983*, 220(4598):671–680.

[26] Kohonen T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics 1982*, 43:59–69.

[27] Langley P, Iba W, Thompson K. An analysis of bayesian classifiers. AAAI'92 Proceedings of the tenth national conference on Artificial intelligence. AAAI Press; 1992 223-228.

[28] Lisboa PJG, Vellido A, and Edisbury B. Business Applications of Neural Networks: The State-Of-The-Art of Real-World Applications, 2000.

[29] Lu H, Setiono R, Liu H. Neurorule: A connectionist approach to data mining. In Proceedings of the *21th International Conference on Very Large Data Bases*, Morgan Kaufmann, San Francisco, 1995 478–489.

[30] Lukashin AV, Anshelevich VV, Amirikyan BR, Gragerov AI, Frank-Kamenetskii MD. Neural network models for promoter recognition. *Journal of biomolecular structure & dynamics 1989*, 6(6):1123–1133.

[31] Ma S, Huang J. Penalized feature selection and classification in bioinformatics. <u>*Briefings in Bioinformatics*</u> *2008*, 9(5):392–403.

[32] McCulloch WS, Pitts W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics 1943*, 5(4):115–133.

[33] Ni X. Research of data mining based on neural networks. *World Academy of Science, Engineering and Technology 2008*, 39:381–384.

[34] O'Neill M, Song L. Neural network analysis of lymphoma microarray data: prognosis and diagnosis near-perfect. *BMC Bioinformatics 2003*, 4(1):13.

[35] Pino R, Parreno J, Gomez A, Priore P. Forecasting next-day price of electricity in the spanish energy market using artificial neural networks. *Engineering Applications of Artificial Intelligence 2008*, 21:53–62.

[36] Quackenbush J. Computational genetics: Computational analysis of microarray data. *Nature Reviews Genetics 2001*, 2:418–427.

[37] Quah JTS, Sriganesh M. Real-time credit card fraud detection using computational intelligence. *Expert Systems with Applications 2008*, 35(4):1721–1732.

[38] Quinlan RJ. C4.5: programs for machine learning. Morgan Kaufmann, 1993.

[39] Ray SS, Bandyopadhyay S, Mitra P, Pal SK. Bioinformatics in neurocomputing framework. *IEE Proceedings - Circuits, Devices and Systems*, 152(5):556–564, 2005.

[40] Refenes AN, Zapranis A, Francis G. Stock performance modeling using neural networks: A comparative study with regression models. *Neural Networks 1994*, 7(2):375 – 388.

[41] Ressom H, Wang D, Natarajan P. Clustering gene expression data using adaptive double self-organizing map. *Physiological Genomics 2003*, 14(1):35–46.

[42] Riis SK, Krogh A. Improving prediction of protein secondary structure using structured neural networks and multiple sequence alignments. *Journal of Computational Biology 1996*, 3:163–183.

[43] Ritchie MD, White BC, Parker JS, Hahn LW, Moore JH. Optimization of neural network architecture using genetic programming improves detection and modeling of gene-gene interactions in studies of human diseases. *BMC Bioinformatics 2003*, 4:28.

[44] Howlett RJ and Jain LC. *Radial Basis Function Networks: 1 Recent Developments in Theory and Applications*. Physica-Verlag, Heidelberg, 2001.

[45] Rumelhart DE, Hinton GE, Williams RJ. Learning internal representations by error propagation. *MIT Press 1986*, Cambridge, MA, USA, 1: 318–362.

[46] Saito K, Nakano R. Medical diagnostic expert system based on pdp model. In Proceedings *IEEE International Conference on Neural Networks volume 1*, 1988 255–262.

[47] Draghici S, *Data Analysis Tools for DNA Microarrays*, Chapman&Hall/CRC, 2003.

[48] Setiono R, Liu H. Symbolic representation of neural networks. *IEEE Computer 1996*, 29(3): 71–77.

[49] Shafer J, Agrawal R, Metha M. SPRINT: a scalable parallel classifier for data mining. In Proceedings of the *22nd Int'l Conference on Very Large Databases*, Morgan Kaufmann, 1996 544–555.

[50] Shen A, Tong R, and Deng Y. Application of classification models on credit card fraud detection. In *International Conference on Service Systems and Service Management*, China, IEEE, 2007 1–4.

[51] Smyth P, Goodman RM. An information theoretic approach to rule induction from databases. *IEEE Transactions on Knowledge and Data Engineering 1992*, 4(4):301–316.

[52] Jin song Pei and Eric C. Mai. Neural network initialization for modeling nonlinear functions in engineering mechanics. In Proceedings of the *24rd International Modal Analysis Conference*, St. Louis, 2006.

[53] Stahl F, Berrar D, Silva CSG, Rodrigues JR, Brito RMM. Grid warehousing of molecular dynamics protein unfolding data. In Proceedings of the *Fifth IEEE/ACM Int'l Symposium on Cluster Computing and the Grid*, pages 496–503, Cardiff, IEEE/ACM, 2005 496-503.

[54] Lockhart DJ, Winzeler EA. Genomics, gene expression and DNA arrays, *Nature 2000*, 405(6788):827-836.

[55] Drăghici S, Potter RB. Protecting HIV drug resistance with neural networks. *Bioinformatics 2003*, 19(1):98-107.

[56] Sureka A, Indukuri KV. Using genetic algorithms for parameter optimization in building predictive data mining models. In Proceedings of the *4th international conference on Advanced Data Mining and Applications*, ADMA '08, Springer, 2008 260–271.

[57] Tan YH, Thoen W. An outline of a trust model for electronic commerce. *Applied Artificial Intelligence 2000*, 14(8):849–862.

[58] http://www.thearling.com/text/dmwhite/dmwhite.htm (accessed July 2011).

[59] Geoffrey G. Towell and Jude W. Shavlik. The extraction of refined rules from knowledge-based neural networks. *Machine Learning 1993*, 13(1):71–101.

[60] Tsai CF, Wu JW. Using neural network ensembles for bankruptcy prediction and credit scoring. *Expert Systems with Applications 2008*, 34:2639–2649.

[61] Way J, Smith EA. The evolution of synthetic aperture radar systems and their progression to the eos sar. *IEEE Transactions on Geoscience and Remote Sensing 1991*, 29(6):962–985.

[62] Wei JS, Greer BT, Westermann F, Steinberg SM, Son CG, Chen QR, Whiteford CC, Bilke S, Krasnoselsky AL, Cenacchi N, Catchpoole D, Berthold F, Schwab M, Khan J. Prediction of clinical outcome using gene expression profiling and artificial neural networks for patients with neuroblastoma. *Cancer Research2004*, 64.

[63] Werbos PJ. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, 1974.

[64] Widrow B, Hoff ME. Adaptive switching circuits. *IRE WESCON Convention Record* 1960, 96-104.

[65] Witten IH, Eibe F. *Data Mining: Practical Machine Learning Tools and Techniques* second edition. Morgan Kaufmann, 2005.

[66] Herrero J, Valencia A, Dopazo J, A hierarchical unsupervised growing neural network for clustering gene expression patterns, *Bioinformatics 2001*, 17(2):126-136.

[67] Xu R, Wunsch D. Survey of clustering algorithms. *IEEE Transactions on Neural Networks 2005*, 16(3):645–678.

[68] Zaslavsky B, Strizhak A. Credit card fraud detection using self-organizing maps. *Information and Security 2006*, 18:48–63.

[69] Zhang D, Jiang Q, Li X. Application of neural networks in financial data mining. In *International Conference on Computational Intelligence*, 2004 392–395.

[70] Bate A, Lindquist M, Edwards IR, Olsson S, Orre R, Lansner A, De Freitas RM. A Bayesian neural network method for adverse drug reaction signal generation. *European Journal of Clinical Pharmacology 1998*, 54(4):315-321.

[71] Surjandari I, Seruni AC. Design of Product Placement Layout in Retail Shop Using Market Basket Analysis. *MAKARA Seri Teknologi 2005,* 9(2): 43-47.

[72] Unnikrishanan N, Mahajan A, Chu T. Intelligent system modeling of a three-dimensional ultrasonic positioning system using neural networks and genetic algorithms. In Proceedings of the *Institution for Mechanical Engineers, Part I, Journal of Systems and Control Engineering 2003*, 217: 367-377

[73] Alberts B, Bray D, Johnson A, Lewis J, Raff M, Roberts K, Walter P. *Essential Cell Biology,*Garland Science, New York, 1998.

[74] Kaufmann L, Rousseeuw PJ. *Finding Groups in Data: In Introduction to Cluster Analysis*. John Wiley & Sons, 1990.

[75] Lancashire LJ, Lemetre C, Ball GR. An introduction to artificial neural networks in bioinformatics—application to complex microarray and mass spectrometry datasets in cancer studies. *Briefings in Bioinformatics 2009*, 10(3):315-329.

[76] Pedersen AG, Baldi P, Chauvin Y, Brunak S. The biology of eukaryotic promoter prediction: a review. *Computers and Chemistry 1999*, 23(3-4):191–207, 1999.

[77] Li X, Hu H, Shu L. Predicting human immunodeficiency virus protease cleavage sites in nonlinear projection space, *Molecular and Cellular Biochemistry 2010*, 339(1-2):127-33

[78] Abeel T, Saeys Y, Rouzé P, Van de Peer Y: ProSOM: core promoter prediction based on unsupervised clustering of DNA physical profiles. *Bioinformatics 2008*, 24(13): 24-31.

[79] Huang W, Lai KK, Nakamori Y, Wang S, Yu L. NEURAL NETWORKS IN FINANCE AND ECONOMICS FORECASTING. *International Journal of Information Technology & Decision Making (IJITDM) 2007*, (6)1:113-140.

[80] Atsalakis GS, Valavanis KP, Surveying stock market forecasting techniques - Part II: Soft computing methods, *Expert Systems with Applications 2009*, 36(3):5932-5941.

[81] Islam MS, Zhou L, Li F. *Application of Artificial Intelligence (Artificial Neural Network) to Assess Credit Risk: A Predictive Model For Credit Card Scoring*. MSc Thesis, Blekinge Institute of Technology, 2009.

[82] Simić D, Kovačević I, Simić S. Insolvency prediction for assessing corporate financial health. *Logic Journal of the IGPL, 2011*.

[83] Paliwal M, Kumar UA. Review: Neural networks and statistical techniques: A review of applications. *Expert Systems with Applications 2009.* 36(1):2-17.

[84] Lee WI, Shih BY, Chung YS, The exploration of consumers' behavior in choosing hospital by the application of neural network, *Expert Systems with Applications 2008*, 34(2):806-816.

[85] Hung C, Tsai CF, Market segmentation based on hierarchical self-organizing map for markets of multimedia on demand, *Expert Systems with Application 2008*, 34(1):780-787.

[86] Limsombunchai V, Gan C, Lee M, An analysis of credit scoring for agricultural loans in Thailand, *American Journal of Applied Sciences 2005*, 2(8):1198–1205.

[87] Chiang WK , Zhang D, Zhou L, Predicting and explaining patronage behavior toward web and traditional stores using neural networks: A comparative analysis with logistic regression. *Decision Support Systems 2006,* 41(2):514–531.

Figure captions

Figure 1: The Knowledge Discovery Process, comprising four important steps: data integration and cleaning; data selection and transformation; data mining; and evaluation and interpretation.

Figure 2: (a) A sketch of a biological neuron. The neurons are cells that can be electrically stimulated. Each neuron in the brain can take electrochemical signals as input via its *dendrites* and can process them before sending new signals along the *axon* and via the *dendrites* of the other connected neurons; (b) An artificial neuron (a perceptron) which emulates a biological neuron.

Figure 3: An NN – organised in interconnected neurons. The weighted input is squashed through a transfer function and the output from one layer is the input for the next one, and so on, until the NN output is produced (NN with only one hidden layer is shown).

Figure 4: The Data Mining Step of the KDD process using Neural Networks, comprises four important sub-steps: Prepare data for the NN; Learn the NN; Extract rules from the trained NN and Asses the extracted rules.

Figure 5: Training and testing of a Classifier/Predictor. First the classifier is generated from training data; then the classifier is either applied to new unlabelled data or refined by testing it on labelled test data.

Figure 6: The top half of this figure shows a rule based classifier in the form of a decision tree. The bottom half shows some of the rules, encoded in the tree, in the form of IF-THEN rules.

Figure 7: Cluster Analysis portrayed as a four step process comprising: the extraction and selection of features; application of the clustering technique; validation of the derived clusters; and the interpretation of the clusters.

Figure 8: Self Organising Map with two inputs and 49 outputs. Each output unit will represent one cluster (class) of inputs and is associated with a *weight vector*, which size is the same as the *input vector* size.

Further Reading/Resources

[Please insert any further reading/resources here]

Comment [JW3]: For readers who may want more information on concepts in your article, provide full references here to additional recommended resources (books, articles, websites, links to multimedia, etc.) that are not included in the reference section. Please do not include links to non-academic sites such as Wikipedia or YouTube, or to impermanent websites.

Rel═══ Articles

Article ID	Article title
214	Knowledge discovery
216	Self-organizing maps and data mining
217	Neural based learning classifiers

WIREs DMKD article list.doc

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

Step 1: Data Integration and Cleaning

Step 2: Data Selection and Transformation

Step 3: Data Mining

Step 4: Evaluation and Interpretation

Knowledge

Patterns
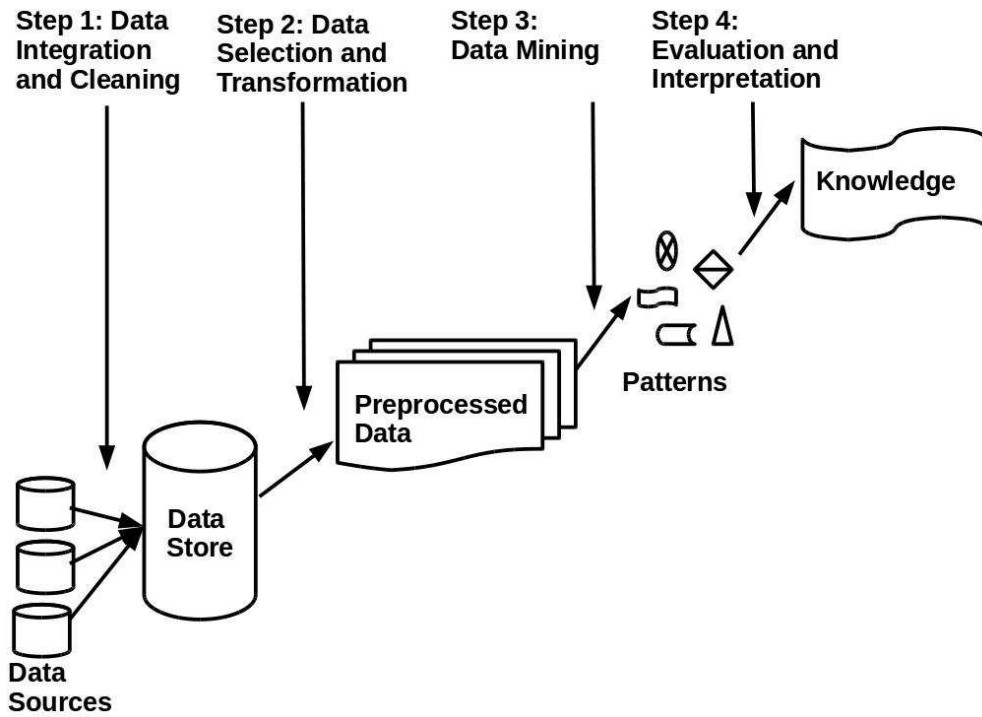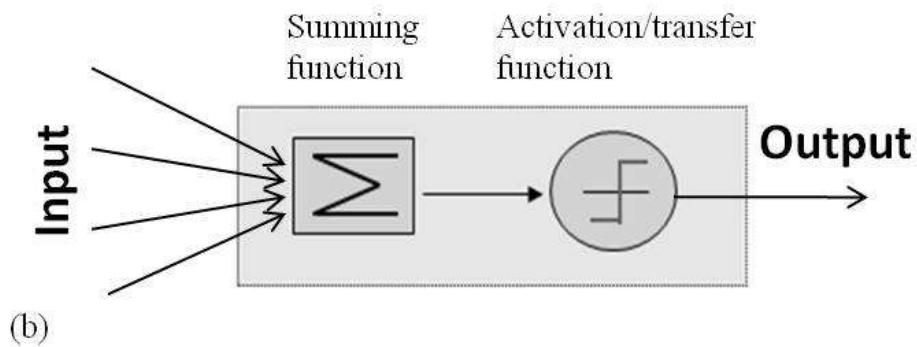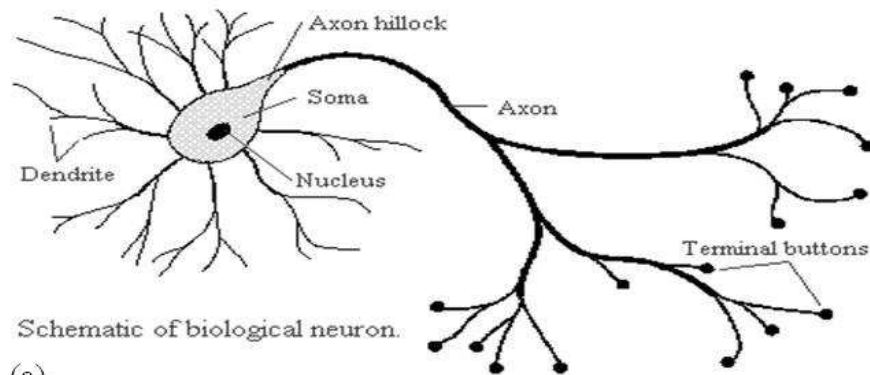
Preprocessed Data

Data Store

Data Sources

Figure 1: The Knowledge Discovery Process, comprising four important steps: data integration and cleaning; data selection and transformation; data mining; and evaluation and interpretation.

31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
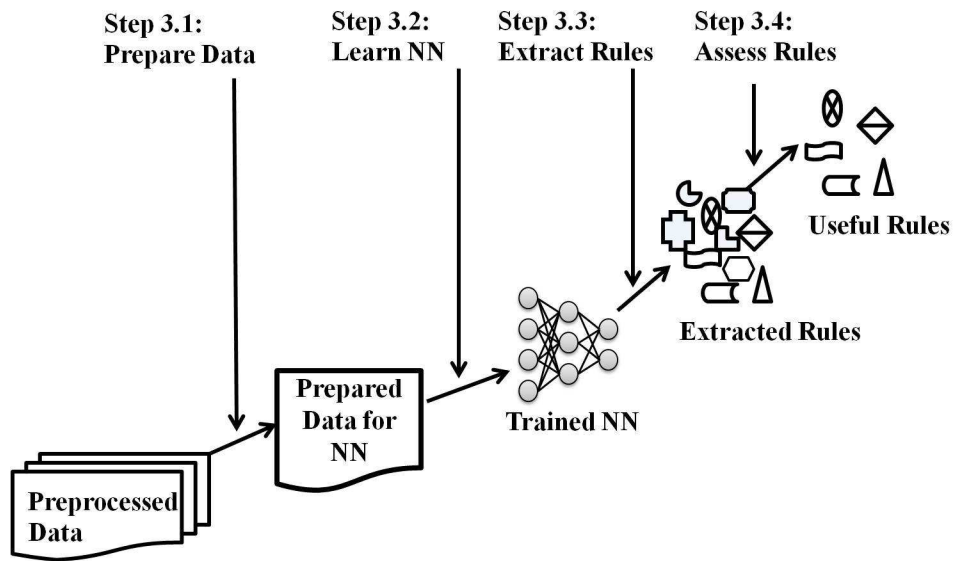41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

Figure 2: (a) A sketch of a biological neuron. The neurons are cells that can be electrically stimulated. Each neuron in the brain can take electrochemical signals as input via its dendrites and can process them before sending new signals along the axon and via the dendrites of the other connected neurons; (b) An artificial neuron (a perceptron) which emulates a biological neuron.
123x96mm (150 x 150 DPI)
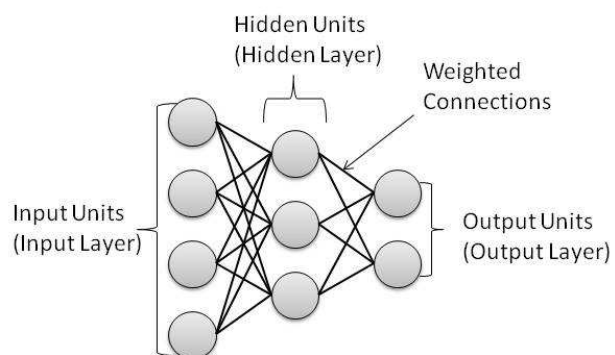
Figure 3: An NN – organised in interconnected neurons. The weighted input is squashed through a transfer function and the output from one layer is the input for the next one, and so on, until the NN output is produced (NN with only one hidden layer is shown).
254x190mm (96 x 96 DPI)

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

Step 3.1:
Prepare Data

Step 3.2:
Learn NN

Step 3.3:
Extract Rules

Step 3.4:
Assess Rules

Useful Rules

Extracted Rules

Trained NN

Prepared Data for NN

Preprocessed Data

Figure 4: The Data Mining Step of the KDD process using Neural Networks, comprises four important sub-steps: Prepare data for the NN; Learn the NN; Extract rules from the trained NN and Asses the extracted rules.
264x150mm (150 x 150 DPI)
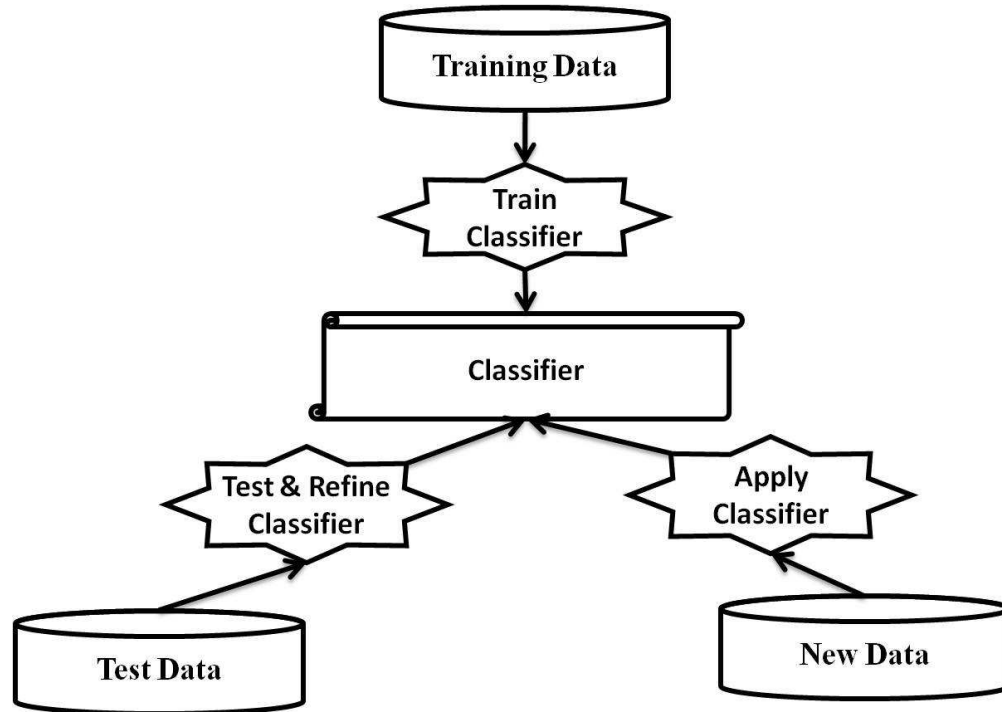
Figure 5: Training and testing of a Classifier/Predictor. First the classifier is generated from training data; then the classifier is either applied to new unlabelled data or refined by testing it on labelled test data.
204x145mm (150 x 150 DPI)

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
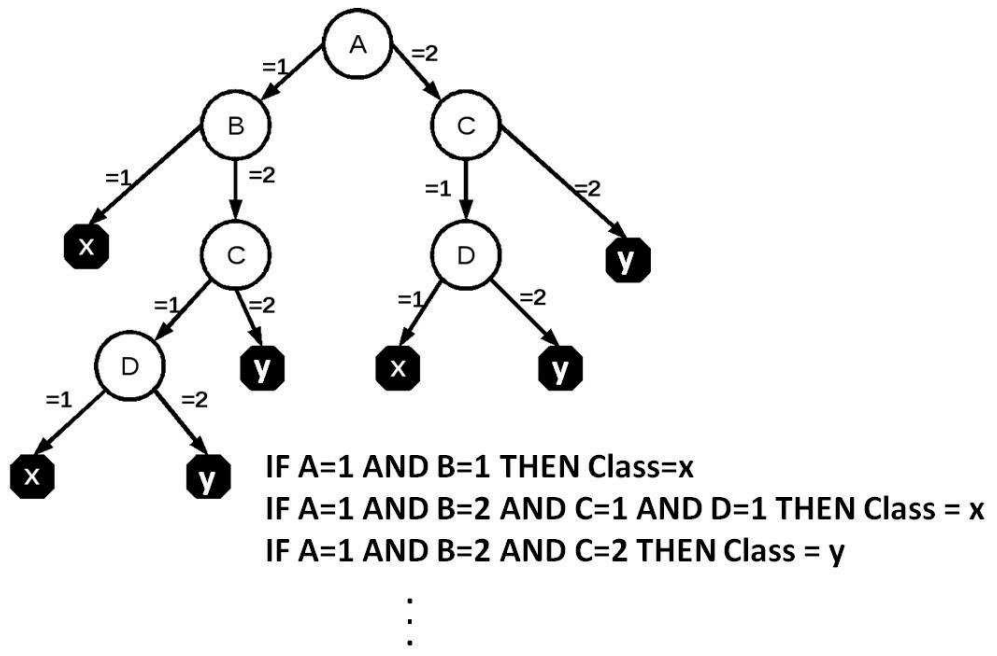
Figure 6: The top half of this figure shows a rule based classifier in the form of a decision tree. The bottom half shows some of the rules, encoded in the tree, in the form of IF-THEN rules.
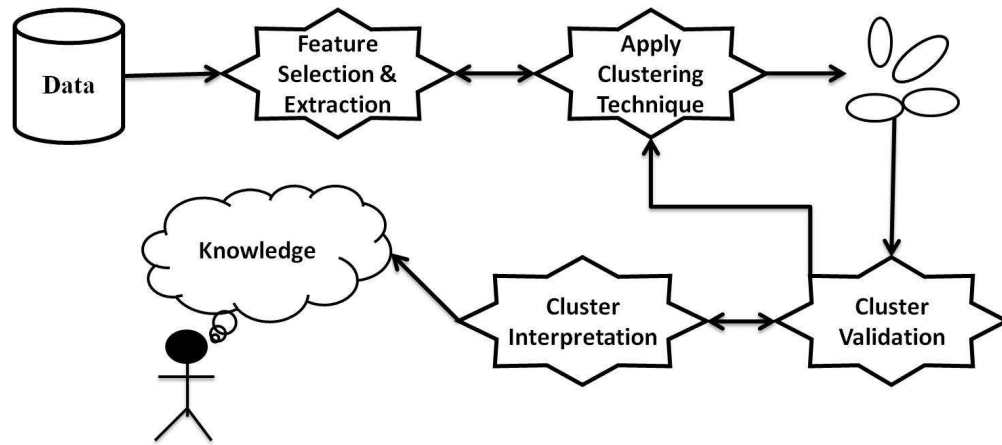283x186mm (96 x 96 DPI)

Figure 7: Cluster Analysis portrayed as a four step process comprising: the extraction and selection of features; application of the clustering technique; validation of the derived clusters; and the interpretation of the clusters.
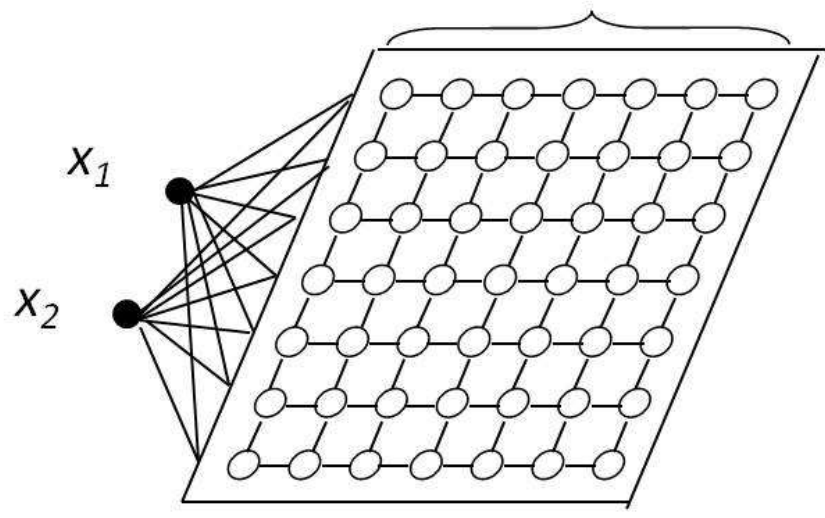249x110mm (150 x 150 DPI)

Figure 8: Self Organising Map with two inputs and 49 outputs. Each output unit will represent one cluster (class) of inputs and is associated with a weight vector, which size is the same as the input vector size.

122x77mm (150 x 150 DPI)