ARTIFICIAL NEURAL NETWORK-BASED CLASSIFICATION OF VECTOR SETS FOR SURFACE INSPECTION

Michael Gyimesi[(a)], Felix Breitenecker[(b)], Wolfgang Heidl[(c)], Christian Eitzinger[(d)]

[(a)] [(b)]Vienna Technical University, Wiedner Hauptstr. 8-10, Wien, Austria
[(c)] [(d)]Profactor GmbH, Im Stadtgut A2, 4470 Steyr, Austria

[(a)]mgyimesi@osiris.tuwien.ac.at, [(b)]fbreiten@osiris.tuwien.ac.at, [(c)]Wolfgang.Heidl@profactor.at, [(d)]Christian.Eitzinger@profactor.at

ABSTRACT
In applications of pattern recognition a set of objects - usually represented by feature vectors - is extracted from an image and needs to be classified as a whole set of objects, meaning that some properties of the set of objects are an aggregation of the single feature vectors – and the classification of the set of objects may depend on exactly these properties. If the number of objects, respectively the number of features, is not known or limited a priori standard classification algorithms such as support vector machines or linear classifiers cannot applied in a straightforward way due to the fixed size of the number of features in these methods. Therefore the set of object's "structure" may not be implemented properly.
In this paper we will discuss some issues of these problems and propose recurrent neural networks (RNN) as a promising method to use for such problems.

Keywords: pattern recognition, neural networks, feature aggregation, classification problem

1. INTRODUCTION

In many applications of image processing respectively pattern recognition, a set of objects - usually represented by feature vectors - is extracted from an image and needs to be classified. The classification decision can rest on properties represented by single feature vectors e.g. if the maximum 'length' of a feature vector exceeds a certain threshold. These problems are typically solved in an algorithmic way.

Another type of classification problems deals with properties, constituted by a fixed number of feature vectors. Typical examples for these problems are e.g. signal processing, data mining problems. To tackle such problems, one typically uses classification algorithms like linear classifiers, support vector machines or feedforward neural networks.

Things become more complicated, if the number of feature vectors is not known a priori.

Typical examples are

- Surface Inspection: A number of faults (= "objects") on the surface is extracted and each one is represented by a feature vector containing e.g. it's size, position, or shape. In order to achieve a proper classification e.g. good/bad, possible reasons for a "bad" decision may be the number of faults, their spatial distribution or other, even more complicated, aggregated properties. Therefore, the whole set of objects needs to be taken into consideration.
- Object Recognition: Recognizing objects that are composed of several components - e.g. an object that consists of an unspecified number of rectangles (windows), some significant edges (the walls) that provide a frame for the windows and a roof on top of all the windows can be identified as a house.
- Biomedical Imaging: The crystallization patterns of dried biological fluids, in particular native blood drops (clots) on a common medical slide, seem to contain a lot of information about diseases and other pathological disorders. Geometrical and color features are extracted out of scanned images from these clots, but only the accumulation of all these varying features of different regions of the blood spot decides, whether the patient has some pathological disorder.

The common requirements of these classification tasks:

- The number of objects in the set is different for each image.
- Aggregated information from the whole set of objects is necessary.
- The classifier has to be trainable, since the classification rules are not known beforehand.

Since our main application is surface inspection we cannot guarantee that the low level procedures produce ordered results:

- The result of the classification has to be invariant up to the order in which the objects represented by single feature vectors are aggregated.

Currently, there exist no mathematical structures that fulfill all of these properties to a high degree. In particular, the problem of trainable aggregation functions has not been investigated in great detail and most standard classification methods, such as linear classifiers, feedforward neural networks or support vector machines use a pre-specified size of feature vectors.

In the current work, these problems are not completely solved. We built different scenarios with partially simplified problems and investigated them by applying existing methods like feedforward neural networks to these problems. Furthermore, recurrent neural networks as a well suited method are one proposition to tackle these problems.

The paper is organized as follows. Section 2 explains the used methods and introduces artificial neural networks, in particular recurrent neural networks.

Section 3 provides details about the scenarios, the test data, mostly related to the application in pattern recognition.

While section 4 presents some information about the experiments and results of our work we conclude with an outlook at future work in section 5.

2. METHODS

We will investigate classification methods that take a set of feature vectors

$$X = \{x_1, x_2, ..., x_n\} \quad with\ x_i \in R^m \tag{1}$$

as input and provide a binary classification result {0,1}. The feature vectors x_i are supposed to be of the same dimension and to contain the same attributes, but there is no other relationship or structure in the data and the numbers of vectors *n* may vary for each image.

A possible solution is to arrange the feature vectors the feature vectors in a single vector,

$$\left[x_1^T, x_2^T, ..., x_n^T, 0^T, 0^T, ..., 0^T\right]^T \quad \in R^{m \cdot n_{\max}} \tag{2}$$

to fill it with zeros up to a pre-specified dimension $m \cdot n_{\max}$ and to apply a standard classification method, such as linear classifiers, feedforward neural networks, support vector machines, or any other classification method. The maximal number n_{max} of feature vectors needs to be specified and the dimension of the combined vector is n_{max} times the number of features *m*.

2.1. Linear Classifiers

Linear classifiers provide a good basis for comparison of the results as they are fully understood with respect to their properties. They try to find a hyperplane in the feature space which separates the samples into two classes (Figure 1).

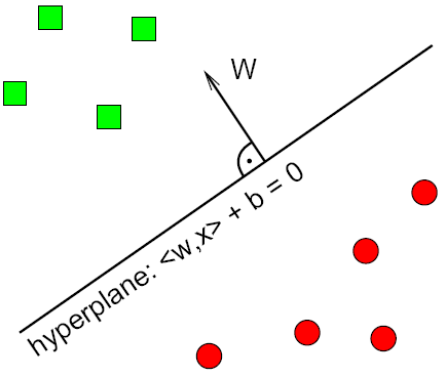

Figure 1: Linear classifier with separating hyperplane

The resulting classification problem is usually prohibitively big and requires training sets of very large size. Moreover, this trivial method does not consider the specific nature of the set of objects.

2.2. Artificial Neural Networks

A modeling method which is widely used is the classification artificial neural networks. The simplest type of artificial neural networks is a feedforward neural network. Equally structured neurons are arranged in layers which are connected by unidirectional connections. There is usually one input layer, one or more hidden layers and one output layer (Figure 2). In our application, classifying images as good or bad, only one output neuron is required to indicate the classification result.

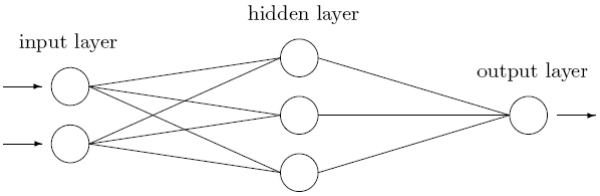

Figure 2: Feedforward neural network

While feedforward neural networks are very successful in classification and are applied in different areas like speech recognition and signal processing they have some undesirably problems. On the one hand they act as a black box and the implemented functions do not really connect to the way humans think about the original problem. On the other hand the original data has to be encoded in a finite dimensional vector space with a prior known dimension which leads to a loss of structural information.

2.3. Recurrent Neural Networks

Recurrent neural networks use, unlike simple feedforward networks, recursive neurons to build feedback structures. These neurons may have connections to all other neurons, meaning that

connections in the opposite direction of the information flow will occur. This composition allows saving information on already given input data up to a certain degree and to learn regularities on structured data. Since the dynamic of recurrent neural networks are linked to the dynamics of finite automata, an interpretation of results via classical formalism is more likely than an interpretation of results of a feedforward neural network (Hammer 2000).

Furthermore, neural networks of this type can deal with a variable number of input vectors and it is not necessary to adapt the network size according to the number of data objects to process. This may be a solution to one of the key problems of the classification task –the number of objects in the set is different for each problem and not known before.

Assuming that networks with a recurrent structure require "no" limits on the number of feature vectors, they can be combined with some added feedforward network layers to generate the desired output. A typical recurrent neural network is shown in Figure 3.

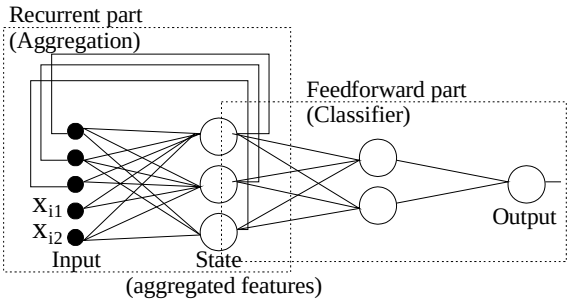

Figure 3: Simple recurrent network combined with feedforward neural network as classifier

Recurrent neural networks have been widely investigated in the field of time series classification/prediction, e.g. (Connor, Martin and Atlas 1994). Various structures have been proposed, e.g. Elman, Jordan, Wiliams-Zipser, recursive auto-associative memory (Pollack 1990) and a number of training methods have been developed; the most common seems backpropagation through time (Werbos 1990).

2.3.1. Related Work

Authors that dealt with the issue of classification of structures (Sperduti and Starita 1997; Frasconi, Gori and Sperduti 1998; Hammer 2000; Hammer 2001; Hammer 2002) propose to use different generalized recursive neurons to represent the structure of a graph. They relate their method to standard neural networks (for the classification of single patterns) and recurrent neural networks (for the classification of sequences).

Recent literature adresses other related problems such as multi-instance learning (Ramon and De Raedt 2000), long-short-term memory (Hochreiter and Schmidhuber 1997), classifying relational data (Uwents, and Blockeel 2005; Perlich and Provost 2003) and aggregation functional (Ovchinnikov and Dukhovny 2002).

3. SCENARIOS AND TEST DATA

We developed different scenarios with various test data and tasks to solve.

3.1. Comparison of different classifiers

A total of 10 test datasets, each one comprising 20.000 pictures with a different number of flaws and features per flaw but with a maximum of 24 features per picture was specified.

The specified tasks to solve where defined as follows:

3.1.1. Comp1 – Sum of a vector

The feature vector is added up and rated as "bad" if the sum fails to exceed a certain threshold value.

3.1.2. Comp2 - Areamax

If the maximal value of the area of one flaw exceeds some arbitrary number, the picture is rated as "bad".

3.1.3. Comp3 - Areasum

One feature of each flaw, which can for instance represent the area and is nonnegative, is added up for all feature vectors. The picture is rated as "bad" if the total area of flaws exceeds a certain threshold value.

3.1.4. Comp4 – Number of flaws

If the number of flaws per picture is greater than some fixed number, the picture is rated as "bad".

3.1.5. Comp5 – Minimal distance

The minimal distance between any two flaws is determined, which demands high computing effort. If it lies beneath a certain threshold, the picture is rated as "bad".

3.1.6. Comp6 – Cluster formation

A rather advanced classification task is to investigate the formation of clusters within the data points. In order to detect a cluster, the number of data points in the neighborhood of every data point was determined.

3.1.7. Comp7 – Number of flaws in a specific region

This criterion rates a picture as "bad" if the number of flaws in a certain region exceeds some arbitrary threshold value.

The according parameters where chosen in a way, that approximately half of the pictures are rated as "good" respectively "bad".

3.2. Evaluation of recurrent artificial networks

In order to evaluate the performance of different network architectures and network sizes, we have generated a total of eight data sets with varying complexity. All but one data set require utilizing aggregated information of the flaws present in one image to solve the associated classification task. One

data set requiring no aggregate information has been added to enable the comparison of the combined aggregation / classification networks with standard classifiers.

A subset of three so-called *simple* data sets aims at measuring a candidate network's principal ability for information aggregation and has little immediate correspondence to practical applications. The five remaining data sets simulate classification tasks being typical for *surface inspection*.

3.2.1. Simple Data Sets

The structure of the feature data is identical for all three simple data sets. The data sets differ in terms of the rules used to generate the (good/bad) labels. The data sets contain 20.000 sets of real-valued 3-dimensional vectors. Each of the sets consists of a minimum of 3 and a maximum of 12 such vectors. The real-valued vector entries are randomly drawn from a uniform distribution in the interval [-0.5, 0.5]. Each vector is interpreted as a point in Euclidean 3-space.

Based on these point sets different kinds of aggregated features are computed for each data set. Image labels are then assigned by applying a threshold to the respective aggregated feature. The thresholds have been chosen to yield an even split of good/bad labels on each of the generated data sets.

- **Simple 1:** The aggregated feature used in the labeling rule is the mean value of the Euclidean norms in the set of feature vectors. Computing the mean value can be done incrementally and requires memory for the running mean value and counting the already processed entries. We consider this to be the base-line task for trainable feature aggregation.
- **Simple 2:** The aggregated feature is the sum of the two largest vector norms in the set. Like in the Simple 1 data set this task can be performed in a single step but it requires memory to store the two largest vector norms.
- **Simple 3:** The labeling rule in this data set is based on the minimum Euclidean distance between two points of the set corresponding to an image. Computing this rule is not possible in a single pass with constant memory. For classification of sets with arbitrary cardinality using finite memory the trainable aggregation function has to learn a heuristic leading to an approximate solution. However, since the cardinality of the vector sets in the test data is limited to 12, a finite network should be able to implement this kind of aggregation in an exact manner.

3.2.2. Surface Inspection Data Sets

Five data sets containing 20.000 images each have been generated. The images and the rules applied for labeling simulate typical tasks occurring in surface inspection applications. One bright spot within the image represents one object (potential fault) that is described by a feature vector. All of the objects in the image make up a set of faults that needs to be classified. Typical example images are displayed in Figure 4 and Figure 5. Both of them show preprocessed images from a surface inspection application. The background is removed and only relevant objects (potential faults) are presented in the image. The gray value corresponds to the degree of deviation from 'normal appearance' of the part of the image.

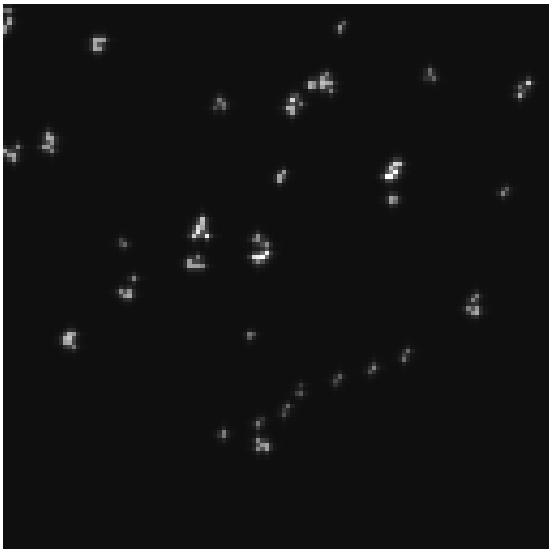

Figure 4: images from a surface inspection application with more evenly distributed flaws and line of flaws in the lower half.

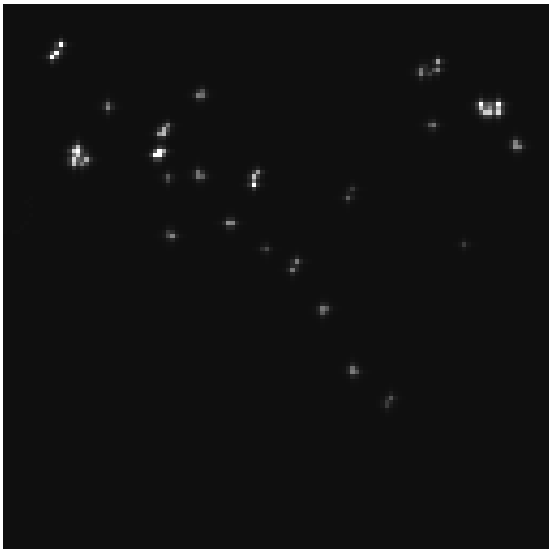

Figure 5: images from a surface inspection application with flaws concentrated in the upper half of the image except a line-like structure from the upper left to the lower left.

Both pictures show flaws of similar type and amount but a completely different distribution can be noticed. The flaws in Figure 1 are more evenly distributed, with some flaw clusters and some flaws apparently aligned. In contrast to that the flaws in

Figure 2 tend to remain in the upper half of the image and some of them build a nearly straight formation running from the upper left to the lower right.

The different composition of flaws leads to different properties of each image, respectively the corresponding feature sets.

The characteristic object features used to define the rules in the surface inspection data sets are:

- Center coordinates
- Area
- Compactness
- Maximum gray value
- Circumference
- Minimal and maximal extension

Based on these features rules to label images as good or bad are built by combining multiple sub rules with a logical OR operation. The sub rules correspond to simple threshold operations on object features and more importantly, on aggregated features. An example sub rule very common to surface inspection tasks combines object size (represented by area) with deviation from normal color (represented by maximum gray value):
An image is labeled bad if there is an object larger than 55 pixels AND maximum gray value is above 25.

All surface inspection data sets contain such sub rules based on object features only. Four surface inspection data sets additionally contain sub rules involving aggregate features, which should be learned by the aggregation part of our model. As already mentioned before, we have included one data set (Surface Inspection 1) not requiring aggregated information at all for correct classification. Below we describe the specific aggregated features used in sub rules of the data sets 2-5:

- **Surface Inspection 2:** The used rules to label the images contain two sub rules involving an aggregate feature, specifically the object count. *An image is labeled bad if there are more than 7 faults OR if there are more than 5 faults in the right half of the image.*
- **Surface Inspection 3:** The rules used to label the images contain two sub rules involving aggregate features based on counting and clustering. A cluster is defined as a number of objects concentrated in a circular area. *An image is labeled bad if there is a cluster of at least 4 objects within a 20 pixel radius OR if there are at least 3 clusters with 3 objects.*
- **Surface Inspection 4**: In this data set we have applied a sub rule based on object features to multiple objects, thus yielding a rule requiring aggregate features. *An image is labeled bad if there is an object larger than 10 pixels with maximum gray value above 150 AND there is another object larger than 15 pixels with maximum gray value above 100.*
- **Surface Inspection 5:** We consider this data set to contain the most difficult learnable aggregation rule in all given test data sets. By some means, the aggregation function has to test for compliance of a number of objects with a geometrical model: *An image is labeled bad if there are at least 6 objects aligned on a line, with the distance between object centers being smaller than 70 pixels and the orthogonal distance to the line being smaller than 15 pixels.*

4. EXPERIMENTS AND RESULTS

Due to the known principal limitations of standard classification methods like linear classifiers and feedforward neural networks in the field of classifying sets of objects with a priori not known number of objects and the known advantages of recurrent neural networks, we investigated different problems to compare recurrent neural networks to the standard classification methods. We therefore developed the settings Comp1 to Comp7 (Section 3).

4.1. Linear Classifier

The tests with linear classifiers where obtained as a benchmark for tests with feedforward neural networks as well as with recurrent neural networks. Due to the specific method, linear classifiers use, we arranged all feature vectors in one vector.
The percentage of falsely classified pictures is displayed in Table 1.

Table 1: Percentage of falsely classified pictures by linear classifiers

DS	crit. / feat.	'sum of vector'	'areamax'	'areasum'	'number of flaws'	'minimal distance'	'cluster formation'	'number in region'
1	1	0	30.220	0	0	24.005	41.165	18.460
2	1	0	31.290	0	0	27.855	40.855	18.940
3	2	0	37.200	0	0.050	22.230	31.785	26.735
4	2	0	29.980	0.100	0.025	24.790	36.785	29.470
5	2	0	29.820	0	0.055	27.650	36.720	27.475
6	2	0	28.925	0	0.165	22.560	32.405	23.885
7	3	0.065	28.945	0.075	0.050	31.285	41.155	32.205
8	3	0.085	28.84	0	0.050	27.205	29.385	36.975
9	4	0	23.540	0.065	0.050	27.110	27.110	41.405
10	4	0.030	26.495	0	0.055	27.935	27.935	33.750

4.2. Feedforward neural networks

The architecture was chosen according to the test data. We used a network with 24 input neurons, one hidden layer with 6 neurons and one output neuron. As a training method a backpropagation method was used.

Results of classification are shown in Table 2.

Table 2: Percentage of falsely classified pictures by feedforward neural networks

DS	crit. / feat.	'sum of vector'	'areamax'	'areasum'	'number of flaws'	'minimal distance'	'cluster formation'	'number in region'
1	1	0.30	17.36	0.12	0.01	19.71	32.01	11.01
2	1	0.12	15.31	0.12	0.20	20.78	31.66	13.95
3	2	0.15	8.58	0.11	0.08	20.52	25.77	6.95
4	2	0.11	6.54	0.15	0.09	19.48	26.50	8.26
5	2	0.08	10.52	0.13	0.03	20.08	26.35	7.60
6	2	0.13	5.56	0.20	0.09	19.93	26.20	7.56
7	3	0.12	3.32	0.41	0.93	18.73	18.17	2.90
8	3	0.09	3.42	0.20	0.11	17.62	17.16	2.88
9	4	0.09	6.25	0.30	0.07	14.86	13.30	0.65
10	4	0.22	2.06	0.11	0.03	16.24	15.82	1.09

4.3. Recurrent neural networks

According to the structure of the networks, much smaller recurrent neural networks compared to the tests with feedforward neural networks have been used. Results are given in Table 3.

Table 3: Percentage of falsely classified pictures by recurrent neural networks

DS	crit. / feat.	'sum of vector'	'areamax'	'areasum'	'number of flaws'	'minimal distance'	'cluster formation'	'number in region'
1	1	8.37	35.05	29.49	27.77	22.74	40.30	13.88
2	1	23.86	33.53	43.37	37.41	20.84	36.90	27.81
3	2	25.82	40.89	26.96	6.95	33.67	29.27	34.92
4	2	20.06	44.63	18.93	10.88	21.91	28.89	22.84
5	2	14.99	45.51	17.61	44.11	23.34	27.90	16.52
6	2	19.96	36.27	27.28	4.20	20.95	30.92	10.16
7	3	16.68	45.18	28.12	6.76	21.21	29.99	19.26
8	3	15.61	39.94	25.29	14.72	33.78	23.57	14.76
9	4	12.95	44.38	24.32	8.54	29.62	31.88	9.56
10	4	13.98	43.61	15.29	10.03	29.71	20.99	10.10

Since some tasks where known as linearly separable, the good performance of the linear classifier was expected. For these problems nearly as good results were obtained with the feedforward neural network but it outperformed the linear classifier in the more complex problems partially dramatically.

The effort of the recurrent neural networks is twofold. On the one hand the results of classification are not that outstanding, especially for the linearly separable tasks. On the other hand, the effort in solving complex problems is comparable with results from the other methods, in particular if we consider the small number of neurons and therefore number of weights compared to the number of weights for feedforward neural networks.

Since these test scenarios did not have nearly the number of possible features of real surface recognition problems, the structural limitations of feedforward neural networks will probably eliminate them as a possible solution for such problems

5. OUTLOOK AND FUTURE WORK

Even though recurrent neural networks seem to be a good method to solve classification problems of complex type, significant and thorough research has to be done.

It is still unclear which training algorithm will give best results, as our backpropagation still has problems if the number of iterations is larger than 10. This poses the question of the best architecture in respect of the number of layers, neurons and iterations needed. The application of evolutionary computation methods like genetic algorithms and their combination with artificial neural networks gives a promising direction for future research.

To get more insight in the behavior and properties of recurrent neural networks, scenarios Simple 1 – Simple 3 and Surface Inspection 1 – Surface Inspection 5 will be investigated.

REFERENCES

Connor, J.T., Martin, D., Atlas, L.E. 1994. Recurrent neural networks and robust time series prediction, *IEEE Trans. on neural networks*. 5 (2): 240-254

Frasconi, P., Gori ,M., Sperduti, A. 1998. A general framework for adaptive processing of data structures. *IEEE Trans. On Neural Networks*, 9 (5)

Hammer, B., 2000. Hammer, B., 2001. Neural Networks Classifying Symbolic Data. *Proceedings of the ICML-2000 Workshop on Attribute-Value and Relational Learing: Crossing the Boundaries, 61-65*

Hammer, B., 2001. Generalization Ability of Folding Networks. *Knowledge and Data Engineering*, 13 (2): 196-206

Hammer, B., 2002. Recurrent networks for structured data – a unifying approach and its properties. *Cognitive Systems Research*, 3: 145-165

Hochreiter, S., Schmidhuber, J. 1997. Long short-term memory. *Neural Comp*. 9: 1735-1780

Ovchinnikov, S., Dukhovny, A. 2002. On order invariant aggregation functionals, *J. of Math. Psychology*, 46: 12-18

Perlich, C. Provost, F. 2003. Aggregation-based feature invention and relational concept classes, *Proc. 9th ACM SIGKDD international conference on knowledge discovery and data mining*. 167-176

Pollack, J.B. 1990. Recursive distributed representations, *Artificial Intelligence*, 46: 77-105

Ramon, J., De Raedt, L. 2000. Multi instance neural networks, *Proc. of the ICML workshop on attribute-value and relational learning*

Seichter S. 2006, Model-based Classifiers for Surface Inspection Problems, Diploma Thesis, *Vienna University of Technology*

Sperduti, A., Starita A., 1997. Supervised neural networks for the classification of structures. *IEEE Trans. On Neural Networks*, 8 (3): 714-735

Uwents, W., Blockeel, H. 2005. Classifying relational data with neural networks. *Proceedings (Kramer, S. and Pfahringer, B., eds.). Lecture Notes in Computer Science*. 384-396, Inductive Logic Programming, 15th International Conference, ILP 2005

Werbos, J. 1990. Backpropagation through time: What it does and how to do it, *Proc. IEEE*. 78 (10): 1550-1560