ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 117-122

Application of a Neural Net in Classification and Knowledge Discovery

Kristina Schädler, Fritz Wysotzki

Technical University of Berlin

Abstract. This work demonstrates that a special neural net approach to graph matching can be applied successfully in the domain of classification, learning and knowledge discovery. This neural net which performs quantitative similarity estimation and common subgraph computation is included as a component in different learning and classification algorithms. It is shown that compared with other approaches high classification accuracy is obtained and plausible, comprehensive generalized descriptions of interesting classes of objects are produced.

1. Introduction

In Machine Learning and Knowledge Discovery, Classification and Pattern Matching, more and more complex objects have been considered during the last years. So, objects often are not represented by feature vectors as usual, but as complex aggregations of partial objects and their relations. In general, operations like the comparison or the generalization of such relational or structured objects lead to NP-complete problems. A second difficulty is the quantitative estimation of the similarity or the distance between such objects. This paper shows how a neural net approach can be used to solve these problems. In the first section a family of metrics in the space of labeled graphs is introduced. Then it is shown how an approximation of the distance between graphs and the corresponding matching is computed using an artificial neural net. After that an outline of some applications of the method in classification and learning is given. At last, the results obtained for a benchmark dataset are compared with the results of other learning algorithms and classifiers.

2. The Distance between Graphs

Labeled graphs are described by a 6-tuple $G(N, V, l, e, L, E)$, where N and $V = N \times N$ are the nodes and the edges of G, respectively[1], L and E arbitrary sets of node and edge labels and $l : N \to L$ and $e : V \to E$ are mappings which assign a label to every node or edge of the graph.

During the last years, different measures of the distance or the similarity between graphs have been introduced. The majority of them are based on the computation of a best mapping between the graphs, see for instance [25, 8, 22, 9, 3, 5, 24]. Only a few of them show metric properties. We suggest the following family of distance measures for graphs with metric properties. Let $G = (N_G, V_G, l_G, e_G, L_G, E_G)$ and $H = (N_H, V_H, l_H, e_H, L_H, E_H)$ be two graphs and h a one-to-one mapping from $N_G[h] \subset N_G$ to $N_H[h] \subset N_H$. Assume

[1] Without loss of generality, we consider complete graphs.

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 117-122

that $s_V : E_G \times E_H \to [0,1]$ and $s_N : L_G \times L_H \to [0,1]$ are given similarity measures for the sets of node and edge labels. $\beta \geq 0$ is a problem dependent parameter that weighs the node matchings with respects to the edge matchings. Then the following distance between G and H can be defined:

$$V_{GH|s_V}(h) = \sum_{(x_i,x_j)\in N_G[h]^2} s_V(e_G((x_i,x_j)), e_H((h(x_i),h(x_j))))$$

$$d(G,H) = 1 - \max_h \frac{V_{GH|s_V}(h) + \beta \sum_{x\in N_G[h]} s_N(l_G(x), l_H(h(x)))}{\max(|N_G|,|N_H|)(\max(|N_G|,|N_H|) - 1 + \beta)} \quad (1)$$

This measure is an extension of the measure used and explained in [21]. It can be shown that Eq.(1) describes a family of metrics, if the similarity measures s_N and s_V are obtained by $s_N(l_1,l_2) = 1 - d_N(l_1,l_2)$ and $s_V(e_1,e_2) = 1 - d_V(e_1,e_2)$ from two metrics d_N and d_V for nodes and edges, respectively.

3. A Neural Net for Graph Matching

Eq.(1) can be transformed in the following objective functions of a quadratic programming problem:

$$f^*(o) = \max_{o\in\{0,1\}^{|N_G|\times|N_H|}} \left(\sum_{i,j=1}^{|N_G|} \sum_{k,l=1}^{|N_H|} w_{ij,kl} o_{ik} o_{jl} + \beta \sum_{i=1}^{|N_G|} \sum_{k=1}^{|N_H|} w_{i,k} o_{ik} \right) \quad (2)$$

$$w_{ij,kl} = \left\{ \begin{array}{c} -penalty \text{ for } i = j \vee k = l, penalty > 0 \\ 0 \text{ for } s_V(e_G((x_i,x_j)), e_H((y_k,y_l))) < \theta_V \\ s_V(e_G((x_i,x_j)), e_H((y_k,y_l))) \text{ else} \end{array} \right\} \quad (3)$$

$$w_{i,k} = \left\{ \begin{array}{cc} 0 & \text{for } s_N(l_G(x_i), l_H(y_k)) < \theta_N \\ s_N(l_G(x_i), l_H(y_k)) & \text{else} \end{array} \right\} \quad (4)$$

$x_i, x_j \in N_G, y_k, y_l \in N_V$ are nodes of the two graphs, $\theta_N, \theta_V \in [0,1]$ additional constraints on the minimum similarity of pairs of labels of nodes, edges or node-edge-node-triples which are accepted as a part of a match. So $f^*(o)$ describes an optimal mappig between G and H.

It is well-known that such programming tasks can be solved using artificial neural nets from the family of Hopfield nets ([14, 16, 6, 11, 26, 1]). Despite of the fundamental criticism of these approaches, for instance in [4, 6], we hold the view that the quality of the solutions which can be produced in a smaller time (as compared to exact algorithms) by such neural nets is sufficient for many problems, provided the structure and the parameters of the net are theoretically founded (see [20] for the approach described in this paper). This view is confirmed by the results from Section 4.1. and 4.2.. In our applications, a neural net approach for graph matching developed in [10, 28, 20] has been used. A pair of graphs can be transformed into a neural net whose stable states describe both a good mapping between the graphs and, using the energy of a state corresponding to a low energy, an upper bound for the distance between the graphs. The structure of the net is derived from the compatibility graph of the two graphs. Therefore the definition of the compatibility graph of two graphs (see [2]) is extended as follows:

Definition 1 (extended compatibility graph (CG)) *The compatibility graph* $(\mathbf{N}, \mathbf{V}, \mathbf{l}, \mathbf{e}, \mathbf{M} \subseteq N_G \times N_H \times [0,1], \mathbf{W} = [-penalty, 1])$ *of two graphs* $G = (N_G, V_G, l_G, e_G, E_G, L_G)$ *and* $H = (N_H, V_H, l_H, e_H, L_H, E_H)$ *is constructed as follows:*

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 117-122

- $\mathbf{N} = \{(x, y, s_N(e_G(x), e_H(y))) : x \in N_G \wedge y \in N_H \wedge s_N(l_G(x), l_H(y)) \geq \theta_N\}$
- *Edges* $(\mathbf{n}, \mathbf{m}) = ((x_i, y_k), (x_j, y_l))$ *of nodes* $\mathbf{n}, \mathbf{m} \in \mathbf{N}$ *where* $x_i \neq x_j \wedge y_k \neq y_l \wedge s_V(e_G((x_i, x_j)), e_H((y_k, y_l)) \geq \theta_V$ *are labeled* $s_V(e_G((x_i, x_j)), e_H((y_k, y_l))$.
- *Edges* $\mathbf{n}, \mathbf{m} \in \mathbf{N}, \mathbf{n} \neq \mathbf{m}$ *where* $i = j \vee k = l$ *are labelled* $-penalty$.

This CG is transformed into a neural net (U_{KG}, C) using the following rules:

- For every node of the CG, create a unit of the net.
- For every edge $((x_i, y_k), (x_j, y_l))$ of the CG with a positive label $w_{ij,kl}$ create a (symmetric) connection (u, u') with a positive weight $w((u, u')) \sim w_{ij,kl}$ (an excitatory connection) between the corresponding units of the net.
- For every edge $((x_i, y_k), (x_j, y_l))$ of the CG with the label $-penalty$ create a (symmetric) connection (u, u') with a negative weight $w((u, u')) \sim -penalty$ (an inhibitory connection) between the corresponding units of the net.

Every unit gets a bias input $I(u, t)$ which depends on the weight $w_{i,k}$ of the corresponding node in the CG. The state of a unit is given by its potential $p(u, t) \in (-\infty, +\infty)$. A unit is called active, if $p(u, t) > 0$. The units are updated synchronously, following the rule:

$$p(u, t+1) = (1-d)p(u, t) + \sum_{\{u' : \exists (u,u') \in C\}} w((u, u'))o(u', t) + I(u, t) \quad (5)$$

$$o(u, t+1) = \left\{ \begin{array}{c} 0 \\ p(u, t+1) \\ 1 \end{array} \right\}, \text{ if } p(u, t+1) \left\{ \begin{array}{c} < 0 \\ \in [0, 1] \\ > 1 \end{array} \right\} \quad (6)$$

The exact weights $w((u, u'))$ of the connections and the bias are chosen according to the labels in the CG and the selected distance measure. The parameter d causes the potential of a unit to decrease if the input of the unit does not exceed a certain threshold. In [20] it is shown that the net reaches a stable state which corresponds to a good solution of the problem if certain conditions are fulfilled for the weights and parameters of the net.

The stable state defines a mapping between similar subgraphs of the two original graphs. On the basis of this mapping and some definition of a generalization of labels of nodes and edges, a generalized common subgraph of the graphs can be produced. The program MATCHBOX is a tool for approximate graph matching and generalization of graphs on the basis of the approach described in this section. MATCHBOX is able to process directed and undirected graphs with complex node and edge labels. Labels can be feature vectors containing components like numerical values, members of ordered and unordered sets or concepts in a conceptual hierarchy. Additional knowledge can be included by weighting the features, using MATCHBOX's estimation of the distribution of the feature values, choosing an appropriate distance from the families of metrics given in Section 2. or modifying the net.

4. Applications

4.1. Distance-Based Classification

This section describes how the approach from Section 3. has been applied to the problem of classification and the discovery of classification relevant features in a dataset of organic chemical compounds. The dataset consists of two parts containing 42 and 188 compounds which occur in automobile exhaust fumes and

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 117-122

also in many industrial chemical processes (see [18]). Many of these structures are mutagenic compounds, i.e. can cause cancer. So it is interesting to find substructures which distinguish mutagenic and non-mutagenic substances or occur frequently in mutagenic substances. The compounds are represented by their bond graphs whose nodes are labeled by atom type and charge.

Using the definition in Eq.(1) of a distance between graphs and the neural net for its approximate computation a distance-based classifier can be constructed which predicts the mutagenicity of a compound from its structural formula. Therefore some nearest neighbor classifiers have been implemented. Nearest neighbor classifiers show high accuracy and stability, for instance when noise occurs in the data. On the other hand, the accuracy of nearest neighbor algorithms depends strongly on the selection of an appropriate distance measure. Thus, the good classification results of a NN-classifier in Table 1 are an indication for an appropriate choice and sufficiently correct computation of the distance measure. The best results for the mutagenesis dataset are obtained by using the Variable Kernel Method described in [23, 7] and its extension from [17] where the number of instances used for classification is reduced.

4.2. Learning of Prototypes

Sometimes we want not only to distinguish between mutagenic and non-mutagenic compounds but also to find common structural properties of mutagenic compounds or substructures which may cause the mutagenicity of substances. So generalized prototypes (see [27]) can be determined. A generalized prototype of a group of graphs is a common subgraph of these graphs where the labels of the nodes and edges can vary within certain ranges. An example of a generalized prototype produced by MATCHBOX is shown in Fig.1.

Often a class of objects cannot be described by a single prototype because the class consists of several subclasses. A prototype has to be determined for every subclass. There are several possibilities to find such subclasses. Two methods for finding prototypes for subclasses are sketched in this section. A detailed description can be found in [21].

The distance-guided generalization produces prototypes by finding similar subgraphs of similar graphs of the same class. This process stops when subgraphs are produced which occur in graphs of another class, too. Thus, some kind of clustering of the dataset is performed where the number of clusters is determined by the number of subclasses. The prototypes created by this method have been used for classification in a 1-NN-classifier. The results shown in Table 1 demonstrate that the prototypes are good representatives of the subclasses. Another method uses the subclass decomposition of the datasets given by a decision tree which has been learned in advance. Geibel's TRITOP ([13]) is a special decision tree which uses a graph representation and decomposes the object space according to structural properties of the objects. TRITOP constructed decision trees for the mutagenesis datasets which show very high classification accuracy. TRITOP's decomposition of the dataset can be used as a basis for a prototype construction by MATCHBOX by generalizing all objects corresponding to a leaf in TRITOP's decision trees. In Fig.1 the prototype produced by MATCHBOX for a subset of 8 mutagenic substances is shown, which is consistent with the result obtained by the logical learning method Progol ([18]). Compared with TRITOP's description of the subset `one(A1,A2),type21(A2),n(A1)` or Progol's rule `active(A) :- atm(A,B,c,21,C), bond(A,D,E,2), bond(A,B,D,1)`, the prototype description gives an understandable, comprehensive description of the common properties of the compounds contained in the subset.

Together with the corresponding decision tree, the prototypes are a means of visualization and explanation of the decision tree's classification criteria. In addition, they provide hypotheses about structural properties that cause

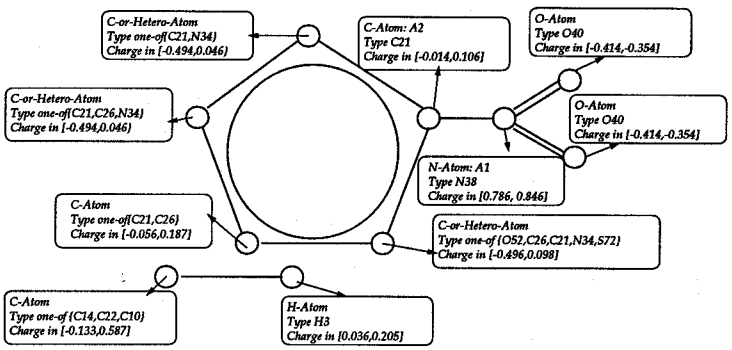

Figure 1: A generalized prototype of mutagenic substances

	188	42		188	42
Linear Regression	0.85	0.67	Prototype 1-NN	0.80	0.81
Neural Net (Backprop)	0.86	0.64	Variable Kernel k-NN, k=3	0.91	0.83
CART	0.83	0.83	Variable Kernel k-NN		
Progol	0.81	0.86	reduced set of instances		
Progol-S2	0.88	0.83	k=6	0.88	0.86
INDIGO	0.86	0.89	k=10	0.88	0.88

Table 1: Classification accuracy for the mutagenesis datasets

other properties like mutagenicity. Moreover, the decision tree classifier can be replaced or extended by a prototype classifier.

5. Results

The right part of Table 1 gives a summary of the classification accuracy (mean cross-validation accuracy) using the classifiers described in Section 4. For comparison, the left part of the table shows the results obtained by other classifiers on the same dataset taken from [12] and [15]. Obviously, the neural net approach from Section 3. is a good basis for the distance-based classification and prototype generation of structured objects. An advantage of this approach is that it processes structures in a neural net without converting them into other representations which often causes a loss of structural information. Graphs of different sizes can be processed without preprocessing. Further, the parameters of the net need not be tuned by hand for different datasets but are set automatically on the basis of a theoretical investigation. Only parameters which have to do with the selection of an appropriate distance measure must be fitted to the problem. Dependent on the chosen measure, not only isomorphic, but also similar subgraphs are found.

The implementation MATCHBOX can be included easily in programs of Machine Learning and in classification algorithms. Other applications of the approach, for instance in case-based reasoning (see [19]) and logic programming have been investigated successfully.

References

[1] D. S. Johnson and M. A. Trick (eds.) *Cliques, Coloring and Satisfiability.* American Mathematical Society, 1996.

[2] Barrow and Burstall. Subgraph isomorphism, matching relational structures and maximal cliques. *Information Processing Letters*, 4(4):83–84, 1976.

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 117-122

[3] G. Bisson. Learning in FOL with a similarity measure. In *Proc. of the 10th AAAI-92*, pages 82–87. AAAI Press/The MIT Press, Menlo Park Cambridge London, 1992.

[4] Jehoshua Bruck and Joseph W. Goodman. On the power of neural networks for solving hard problems. *Journal of Complexity*, 6:129–135, 1990.

[5] H. Bunke and B.T. Messmer. Similarity measures for structured representations. In S. Wess, K. Althoff, and Richter M.M., eds., *Topics in Case-Based Reasoning. EWCBR-93*, no. 837 in LNAI, pages 106–118. Heidelberg, 1993.

[6] L.I. Burke and J.P. Ignizio. Neural networks and operations research: An overview. *Computers Ops.Res.*, 19(3/4):179–189, 1992.

[7] A. W. Moore C. G. Atkeson and S.Schaal. Locally weighted learning. *AI review*, 11, 1996.

[8] L. Cinque, D. Yasuda, L.G. Shapiro, S. Tanimoto, and B. Allen. An improved algorithm for relational distance graph matching. *Pattern Recognition*, 29(2):349–359, feb 1996.

[9] W. Emde and D. Wettschereck. Relational instance-based learning. In L. Saitta, editor, *Proc. of the 13th ICML*, pages 122–130. Morgan Kaufmann, 1996.

[10] J. A. Feldman, M. A. Fanty, N. Goddard, and K. Lynne. Computing with Structured Connectionist Networks. TR 213, CS Dept, University of Rochester, April 1987.

[11] N. Funabiki, Y. Takefuji, and K.-C. Lee. A neural network model for finding a near-maximal clique. *J. of Parallel and Distributed Computing*, 14(3):340–344, 1992.

[12] Peter Geibel and Fritz Wysotzki. Learning relational concepts with decision trees. In Lorenza Saitta, editor, *Proc. of the 13th ICML*, pages 166–174. Morgan Kaufmann, 1996.

[13] Peter Geibel and Fritz Wysotzki. A logical framework for graph theoretical decision tree learning. In N. Lavrac and S. Dzeroski, editors, *Proceedings of the ILP-97*, 1997.

[14] J. Hopfield and D. Tank. Neural computations of decisions in optimization problems. *Biological Cybernetics*, 52:141–152, 1986.

[15] R. D. King, M. J. E. Sternberg, A. Srinivasan, and S. H. Muggleton. Knowledge Discovery in a Database of Mutagenic Chemicals. In *Proceedings of the Workshop "Statistics, Machine Learning and Discovery in Databases" at the ECML-95*, 1995.

[16] C. Looi. Neural network methods in combinatorial optimization. *Computers and Operations Research*, 19(3/4):191–208, 1992.

[17] David G. Lowe. Similarity metric learning for a variable-kernel classifier. Technical Report UBC-TR-93-43, University of British Columbia, 1993.

[18] S. Muggleton. Inverse entailment and progol. *New Generation Computing*, May 1995.

[19] K. Schädler, U. Schmid, B. Machenschalk, and H. Lübben. A neural net for determining structural similarity of recursive programs. In R. Bergmann and W. Wilke, eds., *Proc. of the GWCBR*, pages 199–206, TR University of Kaiserslautern LSA-97-01E, Kaiserslautern, 1997.

[20] Kristina Schädler and Fritz Wysotzki. Theoretical foundations of a special neural net approach for graphmatching. Technical Report 96-26, TU Berlin, CS Dept., 1996.

[21] K. Schädler and F. Wysotzki. A connectionist approach to distance-based analysis of relational data. In X. Liu, P. Cohen, and M. Berthold, eds., *Advances in Intelligent Data Analysis. Proc. of the IDA-97*, pages 137–148, Berlin Heidelberg New York, 1997.

[22] L.G. Shapiro and R.M. Haralick. A metric for comparing relational descriptions. *IEEE Trans.Pattern Anal. Mach.Intell.*, 7(1):90–94, 1985.

[23] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London New York, 1986.

[24] Petko Valtchev and Jerome Euzenat. Dissimilarity measure for collections of objects and values. In P. Cohen X. Liu and M. Berthold, eds., *Advances in Intelligent Data Analysis. Proc. of the IDA-97*, no. 1280 in LNCS, pages 259–272, Berlin Heidelberg New York, 1997.

[25] A. Voß. Similarity concepts and retrieval methods. FABEL Report 13, GMD, Sankt Augustin, 1994.

[26] Jun Wang. *Progress in Neural Networks*, volume 3, chapter 11: Deterministic Neural Networks for Combinatorial Optimization, pages 319–340. Norwood, New Jersey, 1995.

[27] Christel Wisotzki and Fritz Wysotzki. Prototype, nearest neighbor and hybrid algorithms for time series classification. In N. Lavrac and S.Wrobel, editors, *Machine Learning: ECML-95*, number 912 in LNAI, pages 364–367. Springer, 1995.

[28] F. Wysotzki. Artificial Intelligence and Artificial Neural Nets. In *Proc. 1st Workshop on AI*, Shanghai, September 1990. TU Berlin and Jiao Tong University Shanghai.