EXPLANATION OF NEURAL NETWORK CLASSIFICATION

INFORMATICA 4/92

Matija Drobnič,
Viljem Križman
and Borut Korenjak
Institut Jožef Stefan

Keywords: machine learning, neural networks, decision trees

We introduced explanation in human-readable form into a neural network classiffier. The neural network was upgraded by an inductive learning system, which generated the decision tree to explain the way neural network classified new examples. The decision tree learned was compared to the neural network itself and to the inductive learning system regarding both transparrency and classification accuracy.

Razlaga klasifikacije z nevronsko mrežo

V delu predstavljamo metodo, ki pri klasifikaciji z nevronsko mrežo omogoča razlago klasifikacije v človeku razumljivi obliki. Nevronsko mrežo smo nadgradili s sistemom za induktivno učenje, ki generira odločitveno drevo kot razlago delovanja le-te. Tako dobljena odločitvena drevesa smo primerjali z izvirnimi nevronskimi mrežami in s sistemom za induktivno učenje tako glede razumljivosti kot tudi s stališča klasifikacijske točnosti.

1. Introduction

Artificial neural network models were introduced as an attempt to describe the way human brain copes with data, especially in cases of pattern recognition [1]. They are, in principle, based on our understanding of the human brain structure. Their computational power is based on the massive parallelism of simple elements and their dense interconnection. Many different types of neural networks (NN) were introduced during last years. In the field of digital pattern recognition, single-layer networks are mostly used [2], whereas three-layer feed-forward networks can be used as general classifying systems for the data described in an attribute-value laguage [3]. In this field, their adaptability and classification accuracy makes them a very useful tool. Their main drawback is the lack of transparrency to the human user, who cannot figure much out from the values of the NN weights.

Induction learning (IL) is another approach to the classification task (e.g. [4], [5]). Given the examples, the IL system tries to generate a classification function in the form of DT or in the form of **IF - THEN** rules. The main advantage is, the acquisition of knowledge in the form suitable for expert systems, where the transparrency of the results is strictly required. Introduction of statistical methods into the knowledge acquisition process also provides classification accuracy comparable to the one of the classification methods of classical statistics.

In this paper, we try to combine the advantages of both approaches. The main idea is to use IL methods to extract the information hidden in NN weights. The DT obtained in this way, provides an insight into the process of the classification of new examples.

2. Explanation in the neural network

As an example of NN classifier, the growing neural network (GNN) has been chosen. It is a single-layer NN, where neurons are vectors of weights with attached classes, belonging to the same space as the learning examples. Its classification is based on the nearest-neighbour method. To generate the GNN classifier, slightly modified unsupervised learning algorithm proposed by Kohonen [6] was used. It can be put as follows

```
normalise_vectors;
net = first_example_vector;
repeat
   x = next_example_vector;
   y = nearest_vector_from_network(x);
   if class(x) == class(y) then begin
      y = y + α(x — y);
      update_vector(y, net);
   end
   else
      add_vector(x, net);
until no_more_examples;
```

For each new learning example x, we find the nearest vector y from the network according to the $\| \ \|_2$ norm. If their classes match, y is slightly rotated into direction of x (in our experiments, we set $\alpha = 0.2$). Otherwise, x is added to the network as a new vector. When the network is built, the classification process is simple: given an example, find the nearest vector in network according to the $\| \ \|_2$ norm and use its class to classify the example. The implementation of the upper algorithm in C language is given in [7].

As an IL system, ASSISTANT Professional [8] was chosen. It is a tool for the induction of decision trees (DTs) from examples in the attribute-value language, based on ID3 algorithm [4], improved by the binarisation of the attributes, the mechanism for dealing with incomplete data and the tree pruning features.

Several ways of combining the NN and IL methods have been proposed recently [9]. One can classify all the learning examples with the NN learned from them, obtaining their new classes, and then feed them to the IL system as an input. Another possibility is, to generate artificial examples, classify them with NN, and use them as an IL system input again. We have chosen another way: we took the original weight vectors from the GNN and use them as learning examples. In the case of GNN, this simple schema makes sense, since the GNN uses its weight vectors as examples for the nearest-neighbour classification.

3. Experimental results

3.1. Experimental Setup

In our experiments, the medical domain, describing the condition of coronary arteries after the bypass operation has been used. Domain contains 112 examples. Each of them belongs to one of the following classes: deteriorated, unchanged or improved condition. The data is described with 30 parameters, 14 numerical and 16 logical. The numerical attribute values were normalised using the $\| \ \|_\infty$ norm. The logical ones were coded as 0 and 1. Before loaded into ASSISTANT Professional, the numerical values were discretized into 5 equal intervals.

For the cross-validation, the examples were 10 times randomly divided into a training set (70% or 80 examples) and a testing set (30% or 32 examples). For every distribution, GNN was built on learning examples (GNN). Then, DT was learned from the neural network (IL_GNN). As a reference, another DT was learned from the original learning examples (IL). All three methods were compared regarding the transparrency of the classification process.

In the next step, we have used the trees, learned from neural networks as standalone classifiers. All three methods were then compared also regarding the classification accuracy.

3.2. Transparrency of the classification

In this section, we compare three different methods with respect to their transparrency of the classification process to the human user. First, let us examine the GNN classifiers. The algorithm described in Section 2 generates networks containing about 15 neurons. The results are shown in Table 1. Every neuron contains 30 real-valued weights and attached class. A typical network (distribution 2) is presented in Figure 1.

	$Class_1$	$Class_2$	$Class_3$	$\sum$
0	1	0	12	13
1	3	0	13	16
2	2	1	13	16
3	0	1	8	9
4	1	1	13	15
5	2	0	12	14
6	3	1	13	17
7	3	0	16	19
8	2	1	13	16
9	4	1	13	18
$\langle x \rangle$	2.1	0.6	12.6	15.3
σ_x	1.1	0.5	1.9	2.7

Table 1: Number of neurons in GNNs.

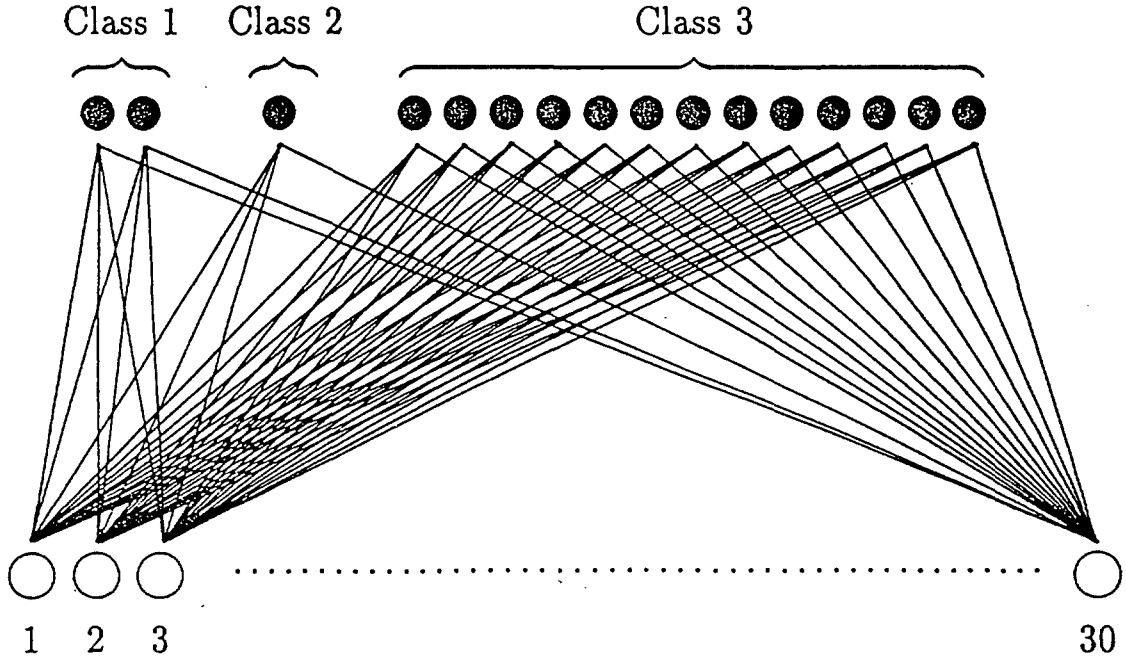

Figure 1: An example of neural network.

Every classification of the new example requires $16 \times 30 = 480$ subtractions, multiplications and additions to calculate Euclidean distance from all the vectors in neural network. Additionally, it also requires 16 comparisons. Even if a human user is capable of using GNN, it is a black box, returning the result without any explanation.

In the next stage, the neurons of GNNs were used as learning examples for the IL system. The DTs learned from the GNNs contained about 4–5 nodes and about 2–3 leaves. An example of such decision tree (distribution 2) is shown in Figure 2.

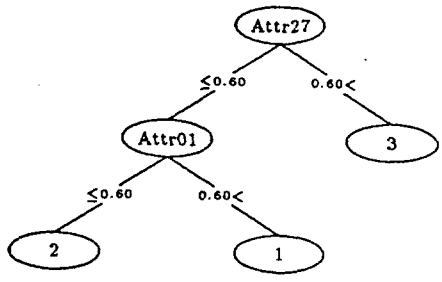

Figure 2: An example of DT, learned from network.

The difference between the classifiers from Figures 1 and 2 is obvious: in the second case, only two comparisons are required during the classification in the worst case. Furthermore, the DT is simple enough to be understood by humans, and can be easily used even without a computer. This is certainly not true with the neural network, where knowledge is hidden into 480 real-valued weights.

For further comparison, we also ran the ASSISTANT Professional with the original examples as an input. The DTs, learned in this way were much bigger than the ones, learned from the neural networks: they typically contained about 18 nodes and about 9 leaves. For comparison with the DTs learned from GNN neurons, the number of nodes and leaves for both methods are presented in Table 2. An example of DT, learned from the learning examples (distribution 2), is shown in Figure 3.

	DT learned from examples			DT learned from GNN		
	nodes	leaves	NULL	nodes	leaves	NULL
0	21	11	2	3	2	0
1	21	11	4	3	2	0
2	17	9	2	5	3	0
3	19	10	3	3	2	1
4	17	9	4	5	3	0
5	15	8	2	3	2	0
6	13	7	3	5	3	1
7	19	10	3	5	3	0
8	19	10	3	5	3	0
9	17	9	3	5	3	0
$\langle x \rangle$	17.8	9.4	2.9	4.2	2.6	0.2
σ_x	2.4	1.2	0.7	1.0	0.5	0.4

Table 2: The sizes of DTs, learned from the GNN neurons and from examples.

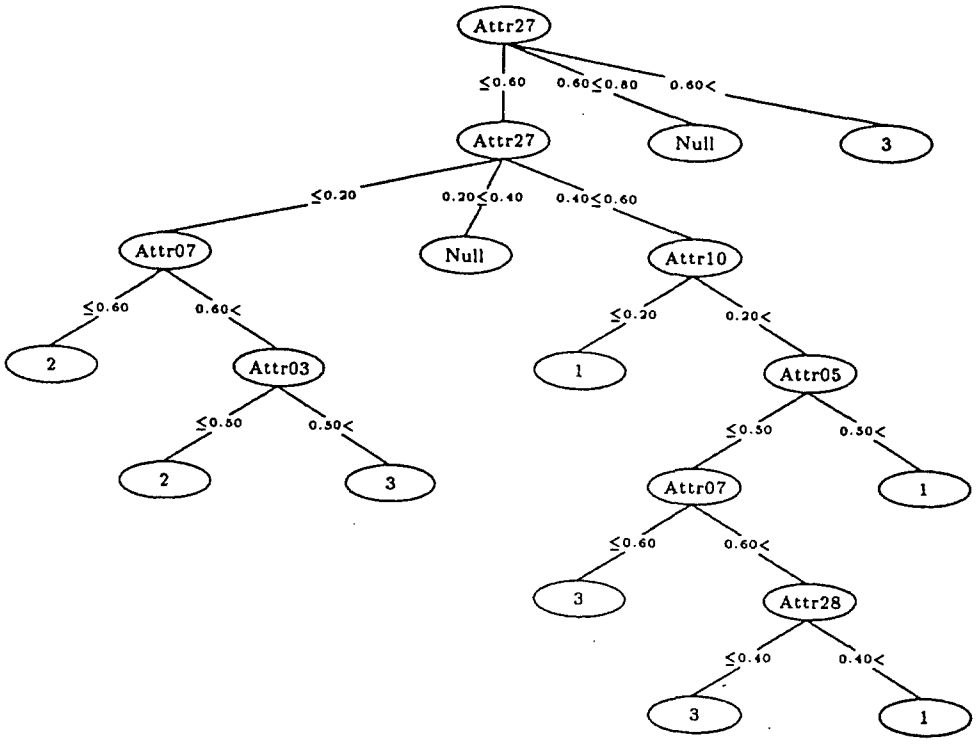

Figure 3: An example of DT, learned from examples.

The transparrency of the upper DT to a human user is also much better than the one of the neural network. Comparing to the trees, learned from neural network, it is slightly less handy for use. However, as it will be shown in the next section, its classification accuracy is much better.

3.3. Classification accuracy

In the next step of our experiments, the DTs learned from neural networks were used as standalone classifiers to validate their quality. their classification accurracy (IL_GNN) was then compared to the one of the GNNs alone (GNN) and to the classification accuracy of DTs learned from the original learning examples (IL). The classification accuracy of all three classifiers was estimated using testing examples. The results are shown in Table 3.

First, the inferiority of the GNN classifiers is obvious. Also the standard deviation of the classification accuracy is very high. The GNN learning algorithm is very sensitive to the ordering of the examples.

	GNN	IL_GNN	IL
0	0.688	0.844	0.906
1	0.813	0.875	0.875
2	0.875	0.875	0.906
3	0.688	0.813	0.906
4	0.813	0.781	0.813
5	0.781	0.844	0.813
6	0.844	0.813	0.844
7	0.844	0.844	0.938
8	0.906	0.938	0.906
9	0.719	0.813	0.906
$\langle x \rangle$	0.797	0.844	0.881
σ_x	0.073	0.042	0.041

Table 3: The classification accuracy of GNN, IL_GNN, and IL systems

However, with some improvements (randomly chosen learning examples, dissmission of weak neurons), the classification accuracy of the GNNs is improved and reaches about 82% [10]. Surprisingly, the classification accuracy of the DTs learned from the GNNs, is greater than the one of the networks themselves.

It seems to us that this happens due to the nature of the IL mechanism, which tries to extract the useful information and to suppress noise in data. In this way, it can use the knowledge hidden in GNN weights, which cannot be used by the nearest-neighbour mechanism of GNN. The DTs, learned directly from learning examples, were significantly more accurate than the ones learned from GNNs This was expected, since their learning sets were larger (the number of learning examples was typically much greater than the number of neurons in the corresponding GNNs).

4. Conclusions

In our attempt to introduce the explanation into a NN classifier, the latter was upgraded by the IL system ASSISTANT Professional. We showed that this system successfully extracted knowledge from the network and presented it in the form of decision tree, so that it could be directly used by human users.

The classification accuracy of the decision trees learned from the neural networks was significantly better than the one of the NNs themselves. This might be caused by badly chosen algorithm for the construction of neural networks, but it seems to us, that IL systems together with their incorporated statistical methods can significantly improve not only the transparrency of the neural network classifiers, but also their classification accuracy.

Acknowledgements

We would like to thank prof. dr. Andrej Dobnikar, who supervised our work on this project and gave us many useful advices. We would also like to thank dr. Matjaž Gams, who also gave us many suggestions during our work. Uroš Rezar implemented algorithms for the construction of the neural network clasifiers, calculated the network weights and provided us domain data. Dr. Bojan Cestnik kindly allowed us to use ASSISTANT 86. Our work was supported by the Ministry of Science, Research and Technology of Republic of Slovenia.

References

[1] T. Kohonen, An Introduction to Neural Computing, "Neural Networks", Vol.1, pp. 3 - 16, 1988.

[2] R. O. Duda and P. E. Hart, "Pattern Classification and Scene Analysis", John Wiley & Sons, New York, 1973.

[3] D. E. Rumelhart, G. E. Hinton and R. J. Williams, Learning Internal Representation by Error Propagation. In D. E. Rumelhart and J. L. McClelland, editors, "Parallel Distributed Processing Explorations in the Microstructure of Cognition. Vol. 1 Foundations", MIT Press, 1986.

[4] J. R. Quinlan, Discovering Rules by Induction from Large Collections of Examples. In D. Michie, editor, "Expert Systems in the Microelectronic Age", Edinbourgh University Press, 1979.

[5] L. Breiman, J. H. Friedman, R. A. Olsen and C. J. Stone, "Classification and Regression Trees", Belmont, California Wadsworth Int. Group, 1984.

[6] T. Kohonen, "Self-Organisation and Associative Memory", Springer-Verlag, Berlin, 1984.

[7] Y. Pao, "Adaptive Pattern Recognition and Neural Network", pp. 291 - 299, Addison Wesley Publishing, 1989.

[8] B. Cestnik, I. Kononenko and I. Bratko, ASSISTANT 86 A Knowledge-Elicitation Tool for Sophisticated Users. In I. Bratko and N. Lavrač, editors, "Proc. 2nd European Working Session on Learning", pp. 31 - 45, Sigma Press, Wilmslow, 1987.

[9] J. L. Shavlik, Constructive Induction in Knowledge-based Neural Networks. In "Proc. 8th International Workshop on Machine Learning", pp. 213 - 217, Evanston, Ilinois Morgan Kaufman, 1991.

[10] U. Rezar and A. Dobnikar, "Prediction of Coronary Disease Expiration", Proc. ERK, Portorož, 1992.