Interpretable Neural Networks with BP-SOM

Ton Weijters[1], Antal van den Bosch[2], and Jaap van den Herik[3]

[1] Information Technology, Eindhoven University of Technology, The Netherlands
[2] ILK / Computational Linguistics, Tilburg University, The Netherlands
[3] Department of Computer Science, Universiteit Maastricht, The Netherlands

Abstract. Interpretation of models induced by artificial neural networks is often a difficult task. In this paper we focus on a relatively novel neural network architecture and learning algorithm, BP-SOM, that offers possibilities to overcome this difficulty. It is shown that networks trained with BP-SOM show interesting regularities, in that hidden-unit activations become restricted to discrete values, and that the SOM part can be exploited for automatic rule extraction.

1 Introduction

Nowadays artificial neural networks (ANNs) are successfully used in industry and commerce. However, the interpretation of ANNs is still an obstacle: "For ANNs to gain a even wider degree of user acceptance and to enhance their overall utility as learning and generalization tools, it is highly desirable if not essential that an *explanation capability* becomes an integral part of the functionality of a trained ANN."[ADT1995]. BP-SOM is an relatively novel neural network architecture and learning algorithm which overcomes the obstacle mentioned during the learning of classification tasks.

In earlier publications [Wei95, WVV97, WVVP97] experimental results were reported in which the generalization performances of BP-SOM were compared to two other learning algorithms for multi-layer feed-forward networks (MFNs), viz. BP and BPWD (BP augmented with *weight decay* [Hin86]). In this paper, we concentrate on interpreting two aspects of the typical knowledge representation of BP-SOM: (i) hidden-unit activations tend to end up oscillating between a limited number of discrete values, and (ii) the SOM can be seen as an organizer of the instances of the task at hand, dividing them into a limited number of subsets that are homogeneous with respect to their class labelling. Furthermore, we illustrate how dividing the learning material into a limited number of homogeneous subsets can be exploited for automatic rule extraction.

2 BP-SOM

Below we give a brief characterisation of the functioning of BP-SOM. For details we refer to [Wei95, WVV97, WVVP97]. The aim of the BP-SOM learning algorithm is to establish a cooperation between BP-learning and SOM-learning in order to find adequate hidden-layer representations for learning classification

tasks. To achieve this aim, the traditional MFN architecture [RHW86] is combined with SOMs [Koh89]: each hidden layer of the MFN is associated with one SOM (see Figure 1). During training of the weights in the MFN, the corresponding SOM is trained on the hidden-unit activation patterns.

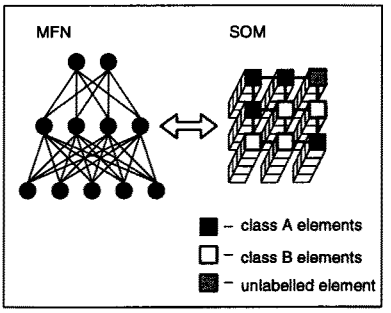

Fig. 1. An example BP-SOM network.

After a number of training cycles of BP-SOM learning, each SOM develops, to a certain extent, self-organisation, and translates this self-organisation into classification information, i.e., each SOM element is provided with a class label (one of the output classes of the task). For example, let the BP-SOM network displayed in Figure 1 be trained on a classification task which maps instances to either output class A or B. We can visually distinguish areas in the SOM: areas containing elements labelled with class A and class B, and areas containing unlabelled elements (no winning class could be found).

The self-organisation of the SOM is used as an addition to the standard BP learning rule [RHW86]. Classification and reliability information from the SOMs is included when updating the connection weights of the MFN (for more details, cf. [Wei95,WVV97, WVVP97]).

3 Knowledge representations in BP-SOM

In this section, knowledge representations of BP-SOM are compared to two related learning algorithms for MFNs, viz. BP [RWH86] and BPWD [Hin86] by training the three algorithms on the parity-12 classification task, i.e., to determine whether a bit string of 0's and 1's of length 12 contains an even number of 1's. The training set contains 1,000 instances selected at random (without replacement) out of the set of 4,096 possible bit strings. The test set and the validation set contain 100 other instances each.

For all experiments reported we have used a fixed set of parameters for the learning algorithms. The BP learning rate is set to 0.15 and the momentum to 0.4. In all SOMs a decreasing interaction strength from 0.15 to 0.05, and a decreasing neighbourhood-updating context from a square with maximally 9 units to only 1 unit (the winner) is used [Koh89].

The hidden layer of the MFN in all three algorithms contains 20 hidden units (the optimal number for a BP trained network), and the SOM in BP-SOM con-

tained 7×7 elements. The algorithms are run with 10 different random weight initialisations. If we compare the average incorrectly-processed test instances of BP, BPWD, and BP-SOM, we see that BP-SOM performs significantly better (6.2%) than BP (27.4%) and BPWD (22.4%).

Clustering of hidden-layer activation patterns To visualise the differences among the representations developed at the hidden layers of the MFNs trained with BP, BPWD, and BP-SOM, respectively, we also trained SOMs with the hidden-layer activations of the trained BP and BPWD networks. Figure 2 visualises the class labelling of the SOMs. The SOM of the BP-SOM network is much more organised and clustered than that of the SOMs corresponding with the BP-trained and BPWD-trained MFNs. It can be seen that the overall reliability of the SOM of the BP-SOM network is considerably higher than that of the SOM of the BP-trained and BPWD-trained MFNs.

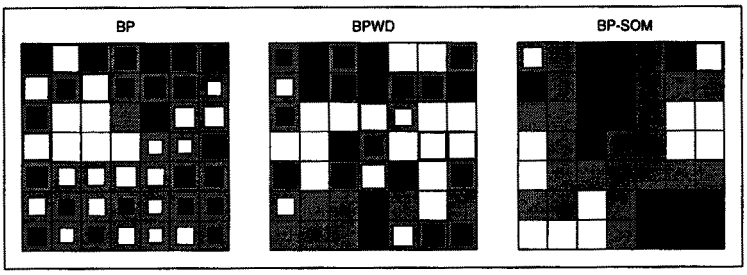

Fig. 2. Graphic representation of a 7×7 SOM: associated with a BP-trained MFN (left), with a BPWD-trained MFN (middle), and with a BP-SOM network (right). White squares represent class 'even'; black squares represent class 'odd'. The width of a square represents the reliability of the element; a square of maximal size represents a reliability of 100%.

Simplified hidden-unit activations When analysing the hidden-unit activations in BP-SOM networks, we observe two effects. Hidden-unit activations tend to culminate either (i) in having one stable activity with a very low variance or (ii) in oscillating between a limited number of approximately discrete values. This clearly contrasts with hidden unit activations in MFNs trained with BP, which usually display a high variance.

To illustrate the first effect, Figure 3 displays the standard deviation of the 20 hidden-unit activations of an MFN trained with BP (left), and MFN trained with BPWD (middle) and a BP-SOM network (right), each of them trained on the parity-12 task (1,000 instances). The standard deviations of ten out of twenty units in the BP-SOM network are equal to 0.01 or lower.

Whenever a unit has a stable activation with a low standard deviation for all training instances, it is redundant in the input-output mapping, and the unit can be pruned from the network. Using a stability threshold parameter s of 0.01 (units with a standard deviation below 0.01 are pruned), we found that BP-SOM

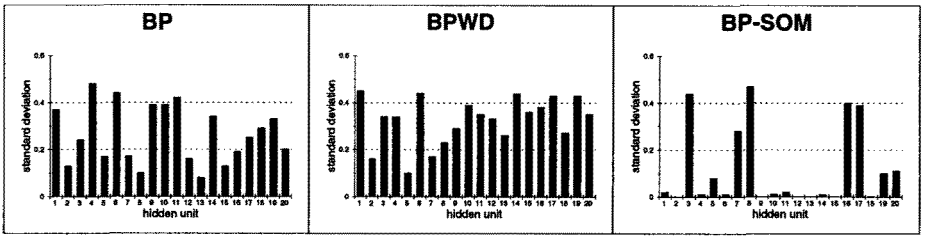

Fig. 3. Standard deviations of the activations of the 20 hidden units of a BP-trained MFN (left), a BPWD-trained MFN (middle), and a BP-SOM network (right), trained on the parity-12 task (1,000 instances).

was able to prune 12 out of 20 hidden units (averaged over 10 experiments), without loss of generalisation accuracy. With the same setting of s, trained on the same tasks, no hidden units could be pruned with BP, nor with BPWD.

To illustrate the second effect, viz. the oscillating of hidden-unit activations between a limited number of discrete values, one typical experiment with an MFN trained with BP-SOM on the parity-12 task is chosen as a sample. In this experiment, 12 out of the 20 hidden units were pruned, while the accuracy of the trained MFN on test material was still acceptable (classification error 0.59%). Figure 4 displays the activations of the first hidden unit of the BP-SOM-trained MFN (displayed on the y-axis), measured for each of the 4096 possible instances (displayed on the x-axis). The instances are grouped on the basis of their respective SOM clustering: we collected for each labelled SOM element all associated instances.

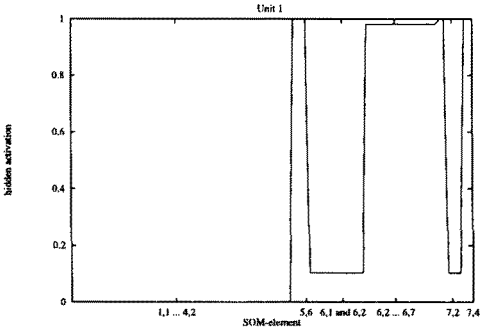

Fig. 4. Activations of the first hidden unit of a BP-SOM network, trained on the parity-12 task, on all 4096 possible instances. The x-axis orders the instances according to their clustering on SOM elements, indicated by the co-ordinates of the elements (e.g., 1,1 indicates SOM element (1,1)).

It can be seen from Figure 4 that the hidden unit ends up oscillating between a discrete number of values, depending on the SOM element on which instances are clustered. The activation values oscillate (approximately) between 0.0, 0.1, and 1.0. The same oscillating phenomenon is present in the activations of the other seven hidden units.

4 Automatic rule extraction based on SOM clustering

In this section we focus on automatic rule extraction based on the clustering of the SOM of a BP-SOM network trained on the monks-1 tasks [Thr91], and give an interpretation of BP-SOM's method of performing this task. Instances are characterised by six attributes $a1 \ldots a6$ which have two, three, or four discrete possible values. An instance is mapped to a class '1' if and only if $(a1 = a2) or (a5 = 1)$. The training set contains 124 instances; all 432 possible instances are used as test material. We used MFNs with one hidden layer of 5 units, 5×5-sized SOMs, and the same experimental settings as described earlier. We found that this network was able to arrive at 100% correct classifications of all instances. After training we collected for each labelled SOM element all associated training instances. Table 1 lists some instances associated with the SOM element at SOM co-ordinates (1,1).

Attributes																		Class
a1			a2			a3		a4			a5					a6		
1	2	3	1	2	3	1	2	1	2	3	1	2	3	4		1	2	
1	0	0	1	0	0	0	1	0	1	0	1	0	0	0		0	1	1
1	0	0	1	0	0	0	1	0	1	0	1	0	0	0		1	0	1
1	0	0	1	0	0	0	1	0	1	0	0	1	0	0		1	0	1
1	0	0	1	0	0	0	1	0	1	0	0	1	0	0		0	1	1
1	0	0	1	0	0	0	1	0	1	0	0	0	1	0		1	0	1
1	0	0	1	0	0	0	1	1	0	0	0	0	0	1		0	1	1
1	0	0	1	0	0	0	1	1	0	0	0	0	0	1		1	0	1
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.		.	.	.
1	0	0	1	0	0	1	0	0	0	1	0	0	0	1		0	1	1
1	0	0	1	0	0	1	0	0	0	1	0	1	0	0		0	1	1
1	0	0	1	0	0	1	0	0	0	1	0	1	0	0		1	0	1
1	0	0	1	0	0	1	0	0	1	0	0	0	0	1		1	0	1
1	0	0	1	0	0	1	0	0	1	0	0	0	1	0		0	1	1
1	0	0	1	0	0	1	0	0	0	1	1	0	0	0		1	0	1
1	0	0	1	0	0	1	0	0	0	1	1	0	0	0		0	1	1

Table 1. List of instances of the monks-1 task associated with SOM element (1,1) of a trained BP-SOM network.

The subset of instances associated with this element are all classified as '1'. More interestingly, only two attributes display a constant value in the subset, viz. attributes $a1$ (always having value 1) and $a2$ (having value 1). This regularity can be exploited to form a rule, which states that if $a1 = 1$ and $a2 = 1$, the corresponding class is '1'. This rule extraction procedure is formalised as follows. For each SOM element, an $IF \ldots THEN$ rule is extracted; it is composed of a conjunction of all attribute values having a constant value throughout the instance subset associated (they are concatenated on the left-hand side of the rule), and of the classification of the instances (on the right-hand side). This procedure, when applied to SOM element (1,1), leads to the rule IF $(a1 = 1)$ & $(a2 = 1)$ $THEN$ class$= 1$. Application of this procedure to all labelled SOM elements results in the rules listed in Table 2. The rules are arranged on the basis of the number of (in)equalities in them. Applying these rules results in a 100% correct classification.

SOM element	rule	coverage	reliability
(1,1)	IF (a1=1 and a2=1) THEN class=1	48	100
(1,5)	IF (a1=2 and a2=3 and a5=1) THEN class=1	12	100
(4,5)	IF (a1=3 and a2=2 and a5=1) THEN class=1	12	100
(5,3)	IF (a1=2 and a2=3 and a5 $\neq$ 1) THEN class=0	36	100
(5,5)	IF (a1=3 and a2=3 and a5 $\neq$ 1) THEN class=0	36	100
(1,4)	IF (a1 $\neq$ 2 and a2 $\neq$ 3 and a5=1) THEN class=1	24	100
(2,5)	IF (a1 $\neq$ 1 and a2 $\neq$ 1 and a5=1) THEN class=1	24	100
(1,3)	IF (a1 $\neq$ 3 and a2 $\neq$ 2 and a5=1) THEN class=1	24	100
(3,1),(3,2)	IF (a1 $\neq$ 2 and a2 $\neq$ 3 and a5 $\neq$ 1) THEN class=0	72	100
(3,4),(3,5)	IF (a1 $\neq$ 1 and a2 $\neq$ 1 and a5 $\neq$ 1) THEN class=1	72	100
(5,1)	IF (a1 $\neq$ 3 and a2 $\neq$ 2 and a5 $\neq$ 1) THEN class=0	72	100
	Totals	433	100

Table 2. The eleven different IF – THEN-rules extracted from the monks-1 training instances matching the same SOM-elements.

5 Conclusions

By letting BP and SOM learning cooperate, BP-SOM can arrive at interpretable MFNs in which both hidden unit activations and SOM clustering display more structure and organisation than with BP. BP-SOM constitutes a basis for automatic rule extraction by means of its ability to structure the data in relevant, task-specific instance subsets. It does so automatically, without the need for postprocessing discretisation or normalisation methods.

References

[ADT95] Andrews, R., Diederich, J., and Tickle, A. B. (1995). A Survey And Critique of Techniques for Extracting Rules from Trained Artificial Neural Networks. *Knowledge Based System*, **8:6**, 373–389.

[Hin86] Hinton, G. E. (1986). Learning distributed representations of concepts. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, 1–12. Hillsdale, NJ: Erlbaum.

[Koh89] Kohonen, T. (1989). *Self-organisation and Associative Memory.* Berlin: Springer Verlag.

[RHW86] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland (Eds.), *Parallel Distributed Processing*, Vol. 1: Foundations (pp. 318–362). Cambridge, MA: The MIT Press.

[Thr91] Thrun, S. B., et. al (1991). *The MONK's Problems: a performance comparison of different learning algorithms.* Technical Report CMU-CS-91-197, Carnegie Mellon University.

[Wei95] Weijters, A. (1995). The BP-SOM architecture and learning rule. *Neural Processing Letters*, **2**, 13–16.

[WVV97] Weijters, A., Van den Bosch, A., Van den Herik, H. J. (1997). Behavioural Aspects of Combining Backpropagation Learning and Self-organizing Maps. *Connection Science*, **9**, 235–252.

[WVVP97] Weijters, A., Van den Herik, H. J., Van den Bosch, A., and Postma, E. O. (1997). Avoiding overfitting with BP-SOM. *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, IJCAI'97, San Francisco, Morgan Kaufmann, 1140–1145.