ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 253-258

Neural Networks for Invariant Pattern Recognition *

Jeffrey Wood and John Shawe-Taylor

Dept. of Computer Science, Royal Holloway, Un. of London

Abstract

In this paper, we discuss a methodology for applying feedforward networks to problems of invariant pattern recognition. We present the Group Representation Network (GRN), a type of feedforward network with the property that its output is invariant under a group of transformations of its input. Since the invariance of such a network is inbuilt, it does not need to be learned. Consequently it is capable of a better generalization performance than a conventional network for solving the same symmetric problem. In addition, the GRN has fewer free parameters than connections and we can hence expect it to train faster than an ordinary network of the same connectivity.

1 Introduction

Let us consider a pattern classification problem in which the classification of any given pattern is invariant under certain linear transformations of the pattern. These transformations form a group.

Since we have prior knowledge of the classification problem, we should be able to improve the generalization ability of any given pattern classifier (in our case, a feedforward neural network) by incorporating this knowledge into the classification system. We achieve this by enforcing constraints on the weights of the connections in our network in order to make the output automatically invariant under the desired group.

The remainder of the paper is arranged as follows. In Section 2 we define our new class of invariant networks. In the following two sections, we present results regarding the mathematical structure of these networks. In Section 5 we describe some experiments performed using these networks, before we conclude.

We will refer often to terms of group representation theory. These can be found in the standard literature, see for example [2], [4], [5].

*This work has been carried out with the support of the Engineering and Physical Sciences Research Council and the Defence Research Agency under CASE award no. 92567042

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 253-258

2 The Group Representation Network

Neural networks invariant or partially invariant under certain transformations have been constructed in earlier work; see for example [1], [3], [8]. The networks which we are about to describe are based on Symmetry Networks (see [6], [7]), which are invariant under input node permutations.

Before we introduce our new class of networks, we wish to make some preliminary definitions. Definition 2.1 is standard representation theory; definition 2.2 is our own.

Definition 2.1 *Given any two representations A and B of a group $\mathcal{G}$, a* homomorphism *from A to B is a linear transformation T such that $TA(g) = B(g)T$ for all $g \in \mathcal{G}$. The space of all homomorphisms from A to B is called the* intertwining space *of (A, B).*

Definition 2.2 *Let A be a linear representation of the group $\mathcal{G}$ acting on the real vector space V. Then the function $f : I\!R \mapsto I\!R$ is said to* preserve the representation *A if*

$$\underline{f}(A(g)v) = A(g)\underline{f}(v) \qquad \forall\, v \in V,\ g \in \mathcal{G},$$

where $\underline{f}$ denotes the extension of f to componentwise action on a vector.

We are now able to define our new model.

Definition 2.3 *A* Group Representation Network (GRN) *over a group $\mathcal{G}$ is a feedforward network for which the following laws hold :*

1. *The nodes are partitioned into layers, and with each layer is associated a representation of $\mathcal{G}$. There are no intra-layer connections. Each output node is regarded as a separate layer associated with the trivial representation of $\mathcal{G}$. When a given representation is finite, its dimension will be equal to the number of nodes in the corresponding layer.*

2. *The linear transformation defined by the weights of connections between two given layers is a homomorphism from the representation associated with the lower layer to that associated with the higher.*

3. *The activation function f is the same for all nodes in a given layer, and preserves the representation A associated with this layer.*

The reader should note that we treat the operation of subtraction of a threshold as being part of the activation function; thus in particular thresholds must be the same for all nodes in a given layer.

The inductive proof that a GRN is indeed invariant under the action of the group on the input layer is given in [9].

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 253-258

3 Choice of Activation Functions

The choice of group representations in a GRN determines and is determined by the choice of activation functions, as dictated by the constraint that the activation function of a given layer should preserve the corresponding representation. We define two classes of representation which will prove to be particularly useful.

A *permutation representation* of $\mathcal{G}$ acting on a vector space V is one which acts by permuting the components of any $v \in V$. An *inversion representation* is one which permutes the components and also multiplies certain components by -1.

It is easy to prove that all functions preserve any permutation representation, and also that all odd functions (but only odd functions) preserve any inversion representation (see [9]). Other classes of representation are preserved only by small classes of activation functions (such as linear functions) and are of little use in a GRN.

4 Weight Matrix Structure

We now proceed to analyse the structure of the weight matrices of connections within a GRN. At this stage we will need to assume that $\mathcal{G}$ is finite, an assumption which we have not made so far.

Consider two connected layers of a GRN, the lower and higher layers corresponding to representations A and B respectively. The weight matrix of the connections can be any homomorphism from A to B.

In [9] we prove that any homomorphism from A to B has the form

$$W = \sum_{g \in \mathcal{G}} A(g^{-1}) X B(g)$$

for some matrix X, and that furthermore any matrix of this form is such a homomorphism. By making X a completely general matrix of independent variables, we obtain an expression for a completely general homomorphism $W_{A,B}$, which is parameterized.

This *generalized weight matrix* $W_{A,B}$ characterizes the intertwining space of (A, B) and we take it to be our weight matrix. The parameters of the matrix X become variable parameters of the neural network, rather than the weights themselves. However, these parameters occur in certain linear combinations, and so the effective number of free parameters, which we call the *parameter dimension* of A and B, is typically considerably less than the number of connections. Since the weights are linear combinations of the new parameters (as seen by the formula above), it becomes easy to adapt standard learning algorithms, such as backpropagation, to GRNs.

We now consider the special case when the representation B is a permutation representation. This is the most useful case, since permutation repre-

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 253-258

sentations are the easiest representations to construct and analyse. The case when B is an inversion representation is covered in [9].

4.1 Permutation Representations

Let A be an orthogonal representation of the group $\mathcal{G}$, and P a permutation representation, associated with two layers of a GRN (and with P corresponding to the higher layer). We also assume P is transitive; if this is not the case, then it can be written as the direct sum of transitive permutation representations, and each of these can be considered separately.

Note that P acts by permutation on the nodes of its associated layer. Let the stabilizing subgroup of one of these nodes be denoted by $\mathcal{H}$.

Furthermore, let $h_1, h_2, \ldots h_m$ denote a set of coset representatives of $\mathcal{H}$, and define the *characteristic matrix* $\mathcal{A}(\mathcal{H})$ by the formula

$$\mathcal{A}(\mathcal{H}) = \frac{1}{|\mathcal{H}|} \sum_{h \in \mathcal{H}} A(h)$$

Now we have the following result, proved in [9].

Theorem 4.1 *The parameter dimension of A and P is equal to the trace of the characteristic matrix $\mathcal{A}(\mathcal{H})$, and the generalized weight matrix is given by :*

$$W_{A,P} = \begin{pmatrix} \rho\mathcal{A}(\mathcal{H})A(h_1) \\ \rho\mathcal{A}(\mathcal{H})A(h_2) \\ \vdots \\ \rho\mathcal{A}(\mathcal{H})A(h_m) \end{pmatrix},$$

where ρ denotes an n-dimensional row vector of independent parameters.

This result allows us to see with relative ease what the number of parameters resulting from a given choice of permutation representation P would be, and also provides a much simpler formula for the weight matrix itself.

5 Simulations

Symmetry Networks are a subclass of GRNs. In [6], some experiments were carried out using Symmetry Networks on the graph isomorphism problem. In these experiments, the Symmetry Networks trained much faster than conventional networks.

Some experiments have also been done on the n-bit parity problem. With ± 1-valued inputs, this problem becomes one which is invariant under the inversion of an even number of bits. In this problem, the group invariance specifies the problem completely.

We trained a GRN on the n-bit parity problem (with $n = 2 \ldots 5$) to compare its learning speed with that of an ordinary network with the same connectivity. The GRN structure used was a single hidden layer, with a permutation

ESANN'1995 proceedings - European Symposium on Artificial Neural Networks
Brussels (Belgium), 19-20-21 April 1995, D-Facto public., ISBN 2-9600049-3-0, pp. 253-258

representation associated with the trivial subgroup. The learning algorithm used was a variant of backpropagation, and the results obtained are listed in Table 5.

n	Network Type	Frequency of convergence	Mean # iterations when converged
2	std	80%	52
2	GRN	100%	30
3	std	100%	69
3	GRN	100%	45
4	std	100%	201
4	GRN	95%	63
5	std	100%	229
5	GRN	90%	100

Table 1: Convergence rates on the n-bit parity problem. For each n, the first row gives the results for a standard network, and the second those for a GRN.

The GRN converged significantly faster in all cases than the other network, though for higher values of n it occasionally failed to converge at all.

Another experiment was carried out on a simple character recognition problem, with invariance under 30° rotations and reflection in a vertical axis. The 72 training patterns were letters A, M and X, each specified by the coordinates of their five key points. Thus each pattern was specified by a ten-dimensional real vector. The networks used had three output nodes and two hidden layers each of twelve nodes.

The GRN took an average of about 2800 iterations of backpropagation to converge on this problem, and having converged generalized perfectly on a set of 72 test patterns. The conventional network trained in only 700 iterations, but seemed to have learned the problem imperfectly, since it achieved a poorer recognition rate of 92% on the test set. We suspect that this loss of generalization ability accounts for the faster training of the ordinary feedforward network.

Practical Note

Another advantage in the use of GRNs is that if all elements of an input set are images of each other under the group action, then only one member of that set needs to be presented to the GRN during training. The classification of all the other members is given automatically by the group invariance.

6 Summary

We have described the Group Representation Network, which is a new type of feedforward network suitable for learning problems of invariant pattern classification. The structure of a GRN is determined by a choice of representation of the invariance group for each layer in the network. This choice may restrict the class of activation functions which are usable.

In the case where the invariance group is finite, we have presented formulae for the matrices of weights in the GRN. These weights are expressed as linear combinations of variable parameters. Standard learning algorithms can easily be modified to adapt these new parameters, rather than the weights themselves.

We have reasoned, and supported with some experimental evidence, that a GRN will learn faster and/or generalize better than a comparable standard feedforward network.

References

[1] K. Fukushima and S. Miyake. Neocognitron : A new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern Recognition*, Vol 15(6), 1982.

[2] W. Fulton and J. Harris. *Representation Theory (A First Course)*. Springer-Verlag, New York, 1991.

[3] Y. Le Cun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, Vol 1:541–551, 1989.

[4] W. Ledermann. *Introduction to Group Characters*. Cambridge University Press, 1977.

[5] W. Scott. *Group Theory*. Prentice-Hall, Englewood Cliffs, 1964.

[6] J. Shawe-Taylor. Building symmetries into feedforward networks. In *Proceedings of First IEE Conference on Artificial Neural Networks*, pages 158–162, 1989.

[7] J. Shawe-Taylor. Symmetries and discriminability in feedforward network architectures. *IEEE Transactions on Neural Networks*, Vol 4(5):816–826, Sep 1993.

[8] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol 37(3), Mar 1989.

[9] J. Wood and J. Shawe-Taylor. Representation theory and invariant neural networks. Submitted to Discrete Applied Mathematics, 1993.