Chapter 8

Conclusion

The central tenet of this thesis is that the application of generic, problem-independent neural networks to invariant pattern recognition is both inefficient and inadequate. Such "black-box" neural networks have no inherent ability to perform invariant pattern recognition, and the only available information concerning their desired properties is that implicit in the data sets used to train them. This is a wasteful approach, since very large training sets, expensive in acquisition, storage and training time, are necessary to specify global invariance under transformations such as shift, rotation and scaling. This approach ignores the fact that the network designer usually knows a great deal about the desired performance of network, and can frequently express this knowledge in the form of a succinct, high-level rule. Something better is required.

Despite this, traditional neural networks have some very attractive properties. A network consists of a set of identical simple computational elements, joined by weighted connections. Computation is usually inherently parallel. Such characteristics make networks of this nature ideal candidates for implementation in hardware, such as specifically-fabricated VLSI circuits. With this in mind, we have sought means of constructing networks which retain these desirable properties, but are constrained to model the particular problem to which they are to be applied. We have called these Model-Based Neural Networks.

The Model-Based Neural Network approach may be thought of as a means of compiling an expert's high-level knowledge into the architecture of a neural network, and constraints on its weight-space. We have proposed a variety of different forms of Model-Based Neural Network in this thesis, but they share the characteristic that they consist of functionally-distinct modules. These modules perform component sub-tasks of the invariant pattern recognition problem, and their functionality is specified, either partially or completely, by the network designer. In this way, the designer can use expert *a priori* knowledge of the desired properties of the resultant network to constrain it so that it is guaranteed to have these properties. Invariance under transformations such as shift, rotation and scaling, often desired in two-dimensional pattern recognition systems, can be made an *inherent* property of the network.

8.1 Using Weighting Functions to Constrain Networks

In Chapter 4 we introduced a form of Model-Based Neural Network in which, rather than being independent, the weights of the connections between nodes in a pair of layers are specified by a function of the nodes' coordinates and a vector of parameters. These parameters can be shared in various ways between sets of connections and nodes, allowing the dimensionality of the space which must be searched during training to be reduced. Moreover, the form of the weighting function can be used to constrain nodes and layers of nodes to implement specific classes of functions, such as spatial filtering, or integration of a filter's response. Once such a network is trained, knowledge of the weighting functions and parameter sharing is no longer required. All that remains is a set of nodes linked by weighted connections. The expert's knowledge, used to specify the architecture and constrain the weight-space during training, has been compiled into the network. Importantly, this allows properties to be introduced into the network completely independently of the training set.

A preliminary demonstration of the advantages of this approach was given by showing that it could be used to construct simple MBNNs which classified textures of differing absolute spatial frequencies correctly, independent of their orientations and positions. The MBNNs achieved 97 ± 7.0 percent correct classification of the two-class test set, compared with 53 ± 12 percent correct for the traditional neural networks. These networks were trained with textures of only one orientation, since rotation and scale invariance are inherent properties of their architectures. Moreover, these MBNNs were specified by only 18 parameters, compared with 908 parameters for the minimal TNNs for this problem. Weight Decay (see §3.3.1) failed to improve the performance of the TNNs, which is unsurprising considering the form of the training set. This simple example demonstrates the three main advantages of the MBNN approach: guaranteed invariant performance, greatly-reduced training set size and far fewer parameters. The last two properties result in much faster training.

In order to emphasize the improvements possible using this particular MBNN approach, a comparison was done with a study [Plaut and Hinton, 1987] which used a TNN to classify synthetic speech spectrograms as risers or non-risers (see §4.3). Expert knowledge indicated that this problem was in fact a case of the shift-invariant classification of signals containing differing line orientations, in the presence of noise. A variety of MBNNs was tried, and eventually a design was found, specified by only 22 parameters, which achieved 95.70 ± 6.25 percent correct classification of the test set after training with only 10 example patterns, using simulated annealing. Using a

1344 parameter TNN, Plaut and Hinton (1987) achieved 97.8 percent correct classification, after training with 250,000 example patterns using backpropagation. Again, the MBNN has achieved equivalent performance, but both the number of parameters and the training set size have been reduced by orders of magnitude.

8.2 The Invariance Signature Neural Network Classifier

In Chapter 5, a new invariant feature of two-dimensional contours was developed: the Invariance Signature. We believe that the Invariance Signature is a powerful descriptor of contour shape, which is closely-related to measures employed in human perception. Its application is by no means limited to Model-Based Neural Networks. It is a useful contribution in its own right.

Nevertheless, the development of the Invariance Signature was inspired by the desire to find an invariant contour descriptor which was suitable for calculation in a MBNN, and which corresponded well to theories of human contour perception. Since Lie group theory provides the link between the local changes in the positions of points under the action of a transformation and the global specification of the transformation, it provides the natural starting point. The Invariance Signature is a global measure of the degree of invariance of a given contour with respect to a set of Lie transformations, which, however, is constructed from *local* calculations. It is this that makes the Invariance Signature so attractive for use in a neural network.

The core of the Invariance Signature approach is this: rather than seeking individual invariant features of a contour, the Invariance Signature measures the *degree to which the contour is invariant* under a transformation. The statistics of these departures from invariance are themselves an invariant descriptor of the contour.

In Chapter 6, it was shown that a MBNN could be constructed which calculated a discrete version of the Invariance Signature, and used this as the basis for the classification of contours presented at the input layer. Since the Invariance Signature is shift-, rotation-, scale- and reflection-invariant, the output of this network is *guaranteed* to be invariant under the application of these transformations to the input layer. This network is called the Invariance Signature Neural Network Classifier. The ISNNC is a MBNN which consists of a wide variety of modules. These modules perform tasks as diverse as tangent vector estimation and the binning of data. The weights of all modules in the ISNNC except the final MLP classifier are specified independently of the training set; some are specified directly, others are determined by training the module on a sub-task. As with the weighting-function networks, the final result is just a collection of simple nodes joined by weighted connections, exactly as for a TNN.

In order to be useful, the Invariance Signature must not only be invariant, but it must retain sufficient information for contour classes to be distinguished. That this is so is demonstrated in Chapter 7. The application chosen for the demonstration of the efficacy of ISNNCs was optical character recognition, the task being the classification of arbitrarily shifted and rotated lower-case alphabetic characters from a sans serif font. Two groups of experiments were performed.

The first group of experiments used noise-free, unambiguous computer-generated characters: perfect data. Networks were trained with only one upright example of each of the 22 unambiguous characters. The test set consisted of 4 differently shifted and rotated examples of each character. The best test test classification performance obtained by TNNs on these data was 15.00 ± 0.45 percent correct. Using NOEM (see §7.2.5), the ISNNC achieved 96.60 ± 0.00 percent correct classification, and with eigenvector-based local orientation extraction, 100.00 ± 0.00 percent correct: with perfect data, the ISNNC produces perfect results.

This preliminary experiment with synthetic "perfect" data demonstrated that the Invariance Signature was indeed sufficient to distinguish all the unambiguous letters of the alphabet. The second group of experiments used data obtained by scanning characters printed at 18 different orientations. Shifts were introduced by the segmentation process. These data were then thinned. The resulting data were far from perfect, the dominant problems being quantization noise and thinning artifacts. The data set was divided into a training set and a test set, each containing 9 examples of each character. Trained with this data, the TNNs achieved a best performance of 13.93 ± 0.30 percent correct. The ISNNCs obtained 72.74 ± 0.97 percent correct on ambiguous data, and 86.20 ± 1.18 percent correct on unambiguous data. Further inspection of the training and test data indicated other ambiguities. When corrected for these, ISNNC performance improved to 93.16 percent correct.

In all these cases, both for perfect data and for the scanned character task, the dimensionality of the classification module was very much lower for the ISNNCs than it was for the TNNs. For the scanned data task, the (minimal) TNN had 83018 parameters, whereas the ISNNC had only 881 parameters to be determined during training. Moreover, the dimensionality of the ISNNC classifier can be varied by the designer, by changing the number of bins for the discrete Invariance Signatures. The training time for these ISNNCs was correspondingly reduced.

These experiments indicate that the Invariance Signature can be successfully employed for the recognition of scanned characters independent of rotations and shifts, and that this technique can be implemented in a MBNN. The test set performance obtained is comparable to that obtained by others using thousands of training examples, and is remarkable considering the amount of noise present. Since the ISNNCs are guaranteed to be invariant under shift, scaling, rotation and reflection, and training set performance on the noisy data was 99.06 ± 0.63 percent correct, it can be concluded that test set performance below this level is due to failure to generalize in the presence of noise, not failure to generalize in the sense of invariance. If the size of the training set were increased, test set performance could be expected to approach 100 percent correct.

8.3 Conclusion

Throughout this thesis, a number of different ways of designing Model-Based Neural Networks has been introduced. In every case, the MBNN was shown to out-perform a TNN on the same task. It might be said that the comparisons were unfair, since the training sets used were inadequate for good TNN performance to be expected. That only emphasizes one of the key advantages of the MBNN approach: large training sets are not required. Also, in every case the number of parameters required to specify the MBNN was smaller than that for the TNN, often dramatically so. The training time required was correspondingly less.

A new and powerful invariant descriptor for two-dimensional contours, the Invariance Signature, has been developed It has been demonstrated that the local nature of the initial calculations required to obtain the Invariance Signature makes it particularly suitable for implementation in a neural network. Networks based on the Invariance Signature have been designed, and successfully applied to the shift- and rotation-invariant recognition of scanned characters.

The Model-Based Neural Network approach to invariant pattern recognition has been successfully shown to provide a framework for a network designer to compile expert *a priori* knowledge of the problem domain into a neural network. These Model-Based Neural Networks have guaranteed invariances, require less training data, have fewer parameters, are faster to train, and out-perform their traditional counterparts.