The Influence of Prior Knowledge and Related Experience on Generalisation Performance in Connectionist Networks

F.M.Richardson [1,2], N.Davey[1], L.Peters[1], D.J.Done[2], S.H.Anthony[2]

F.1.Richardson, N.Davey, L.Peters, D.J.Done, S.H.1.Anthony@herts.ac.uk
Department of Computer Science[1], Department of Psychology[2], University of Hertfordshire, College Lane, Hatfield, Hertfordshire, AL10 9AB, UK

Abstract. The work outlined in this paper explores the influence of prior knowledge and related experience (held in the form of weights) on the generalisation performance of connectionist models. Networks were trained on simple classification and associated tasks. Results regarding the transfer of related experience between networks trained using back-propagation and recurrent networks performing sequence production, are reported. In terms of prior knowledge, results demonstrate that experienced networks produced their most pronounced generalisation performance advantage over naïve networks when a specific point of difficulty during learning was identified and an incremental training strategy applied at this point. Interestingly, the second set of results showed that knowledge learnt about in one task could be used to facilitate learning of a different but related task. However, in the third experiment, when the network architecture was changed, prior knowledge did not provide any advantage and indeed when learning was expanded, even found to deteriorated.

0. Introduction

Some complex tasks are difficult for neural networks to learn. In such circumstances an incremental learning approach, which places initial restrictions on the network in terms of memory or complexity of training data has been shown to improve learning [1], [2], [3]. However, the purpose of the majority of networks is not simply to learn the training data but to generalise to unseen data. Therefore, it can be expected, but not assumed, that using incremental learning may also improve generalisation performance.

The work reported in this paper extends upon the original work of Elman [3], in which networks trained incrementally showed a dramatic improvement in learning. In this paper three different ways of breaking down the complexity of the task are investigated, with specific reference to generalisation performance. In the first experiment an incremental training regime, in which the training set is gradually increased in size, is evaluated against the standard method of presenting the complete training set. In the second experiment the hypothesis that knowledge held in the weights of a network from one task might be useful to a network learning a related

task, is explored. The third experiment completes the investigation by determining whether knowledge transfer between networks such as those in the second experiment, may prove beneficial to learning across different types of networks, performing related tasks. The use of well-known learning rules (back-propagation and recurrent learning) allows this work to complement previously mentioned earlier work, allowing a more complete picture of learning and generalisation performance.

1. Experiment One: Investigating Prior Knowledge

The aim of this experiment was to compare the generalisation performance of networks trained using incremental learning in which the input to the network is staged *(experienced networks)*, with equivalent networks initialised with random weights *(naïve networks)*. Networks were trained with the task of classifying static depictions of four simple line-types as seen in Figure 1. Classification of the line-type was to be made irrespective of its size and location on the input array. In order to accomplish this task the network, must learn the spatial relationship of units in the input and then give the appropriate output in the form of the activation of a single classification unit on the output layer.

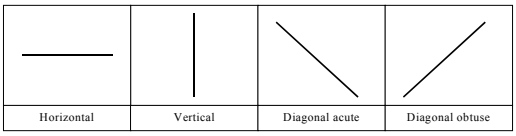

Fig. 1. Shows the four basic line-types which the network was required to classify.

1.1 Network Architecture

A simple feed-forward network consisting of an input layer of 49 units arranged as a 7x7 grid was fully connected to a hidden layer of 8 units, which was also connected to an output layer of 4 units, each unit representing a single line type was used (see Figure 2).

1.2 Training and Testing of Naïve Networks

A full set of patterns for all simple line-types of lengths ranging from 3 to 7 units, for all locations upon the input grid, were randomly allocated to one of two equal-sized training and testing sets (160 patterns per set). Three batches of networks (each consisting of 10 runs) were initialised with random weights (naïve networks). The first batch of networks was trained to classify two different line-types (horizontal and vertical lines), the second three and the third, all four different line-types. All networks were trained using back-propagation with the same learning rate (0.25) and momentum (0.9) to the point where minimal generalisation error was reached. At this stage the number of epochs that each network had taken to reach the stop-criterion of

minimal generalisation error was noted. These networks formed the basis of comparison with experienced networks.

1.3 Training and Testing of Experienced Networks

These networks were trained using the same parameters as those used for naïve networks. The training and testing of the four line-types was divided into three increments, the first increment consisting of two line-types (horizontal and vertical lines) with an additional line-type being added at each subsequent increment. The network progressed from one increment to the next upon attaining minimal generalisation error for the current increment. At this point the weights of the network were saved and then used as the starting point for learning in the following increment, patterns for the additional line-type were added along with an additional output unit (the weights for the additional unit were randomly initialised).

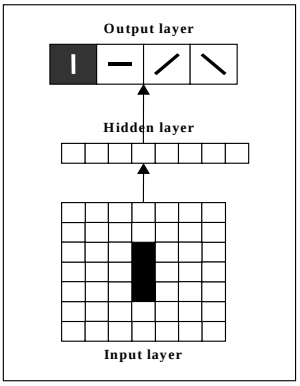

Fig. 2. Shows the network architecture of the 7x7-classification network. The network consisted of an input layer with units arranged as a grid, a hidden layer, and an output layer consisting of a number of classification units. Given a static visual depiction of a line of any length as input, the network was required to classify the input accordingly by activating the corresponding unit on the output layer. In the example shown the network is given the input of a vertical line with a length of 3 units, which is classified as a vertical line-type.

1.4 Results

Generalisation performance was assessed in terms of the number of output errors and poor activations produced by each type of network. Outputs were considered errors if activation of the target unit was less than 0.50, or if a non-target unit had an activation of 0.50 or higher. Poor activations were marginally correct classifications (activation of between 0.50-0.60 on the target unit, and/or activation of 0.40-0.49 on non-target units) and were used to give a more detailed indicator of the level of classification success.

The generalisation performance of naïve networks trained on two line-types was good, with networks classifying on average, 89.84% of previously unseen patterns correctly. However, performance decreased as the number of different line-types in the training set increased, with classification performance for all four line-types dropping to an average of 79.45%. In comparison, experienced networks proved marginally better, with an average of 81.76%.

Further comparisons between naïve and experienced networks revealed that a naïve network trained upon three line-types produced a better generalisation performance

than an experienced network at the same stage. It seemed that the level of task difficulty increased between the learning of three and four line-types. So the weights from naïve networks trained with three line-types were used as a starting point for training further networks upon four line-types, resulting in a two-stage incremental strategy. This training regime resulted in a further improvement in generalisation performance, with an average of 84.48%. A comparison of the results for the three different training strategies implemented can be seen in Figure 3.

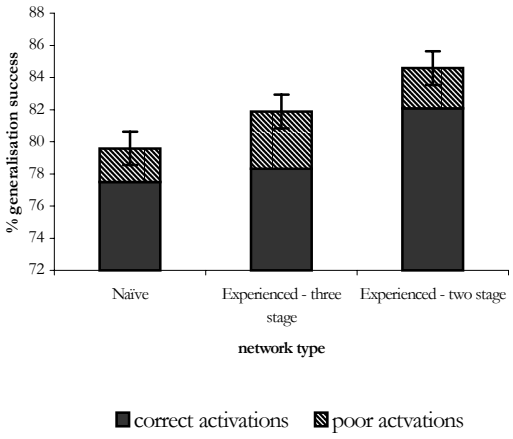

Fig. 3. Shows a comparison of generalisation performance between networks trained using the three different strategies. It can be seen that naïve networks produced the lowest generalisation success. Of the experienced networks, those trained using the two-stage strategy produced the best generalisation performance.

2. Experiment Two: Investigating Related Experience I

The aim of this experiment was to attempt to determine whether the knowledge acquired by networks trained to classify line-types in Experiment One would aid generalisation performance of networks trained upon different but related task. In the new task, networks were given the same type of input patterns as those used for the classification network, but were required to produce a displacement vector as output. This displacement vector contained information as to the length of the line and the direction in which the line would be drawn if produced (see Figure 4).

It was hypothesised that the knowledge held in the weights from the previous network would aid learning in the related task because the task of determining how to interpret the input layer had already been solved by the previous set of weights. The divisions between line-types created upon output in the classification task were also relevant to the related task, in that same line-types shared activation properties upon the output layer, for example all vertical lines are the result of activation upon the y-axis.

2.1 Network Architecture

All networks consisted of the same input layer as used in Experiment One, a hidden layer of 12 units and an output layer of 26 units.

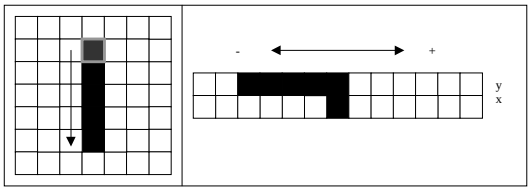

Fig. 4. Shows the static input to the network of a vertical line of a length of five units and the desired output of the network. Output is encoded in terms of movement from the starting point along the x and y-axis. This form of thermometer encoding preserves issometry and is position invariant [3].

2.2 Training and Testing

All networks were trained and tested in the same manner and using the same parameters as those used in Experiment One. Naïve networks were initialised with a random set of weights. For experienced networks, weights from networks producing the best generalisation performance in Experiment One were loaded. Additional connections required for these networks (due to an increase in the number of hidden units) were initialised randomly. Generalisation performance was assessed.

2.3 Results

Naïve networks reached an average generalisation performance of 59.59%. Experienced networks were substantially better, with an average generalisation performance of 70.42%. This result as seen in Figure 5, clearly demonstrates the advantage of related knowledge about lines and the spatial relationship of units on the input grid in the production of displacement vectors.

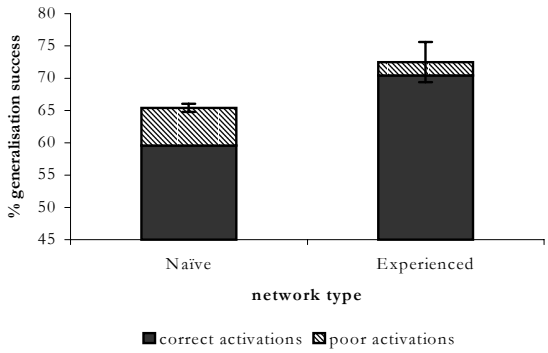

Fig. 5. Shows a comparison of generalisation performance between naïve and experienced networks. It is clear that the experienced networks produced the best generalisation performance.

3. Experiment Three: Investigating Related Experience II

The aim of this experiment was to examine whether weights from a classification task could be used to aid learning in a recurrent network required to carry out an extension of the static task. Initially, a standard feed-forward network (as shown in Figure 2) was trained to classify line-types of simple two-line shapes. Following this a recurrent network [5] was trained to carry out this task in addition to generating the sequence in which the line-segments for each shape would be produced if drawn (as shown in Figure 6).

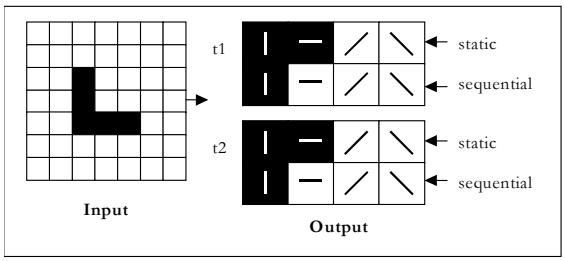

Fig. 6. Shows an example of the input and output for the sequential shape classification task. The input was a simple shape composed of two lines of different types. The output consisted of two components. The static, identifying the line-types, and the sequential generating the order of production. Networks used in Experiment One were trained to give the static output of the task. Weights from these networks were then used by recurrent networks to produce both the static and sequential outputs as shown above.

3.1 Training and Testing

Both naïve and experienced networks were trained on an initial sequence production task, using simple shapes composed of diagonal line-types only. Following this, the initial task was extended for both networks to include shapes composed of horizontal and vertical line-types.

3.2 Results

The generalisation performance of naïve and experienced networks for the initial and extended task was assessed. Initially, comparison showed no notable difference between naïve and experienced networks. However, performance deteriorated for experienced networks with the addition of the extended task. This drop in performance was attributed to a reduction in performance upon the initial task.

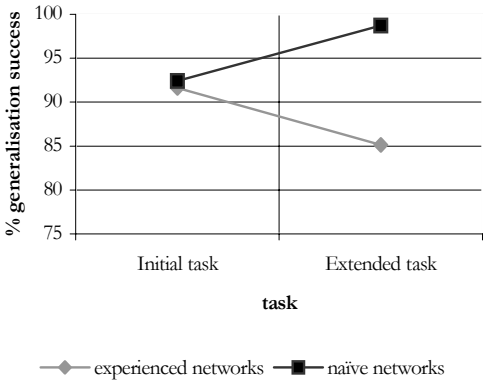

Fig. 7. Shows a comparison of generalisation performance between naïve and experienced networks, performing the initial sequence production task followed by the extended task. It can be seen that performance for the two types of networks for the initial task is relatively equal. However, for the extended task, performance of the experienced networks is poor in comparison to naïve networks trained with random weights

4. Discussion

The experiments conducted have provided useful insights into how prior knowledge and related experience may be used to improve the generalisation performance of connectionist networks. Firstly, it has been demonstrated that incremental learning was of a notable benefit. Secondly, by selecting the point at which learning becomes difficult as the time to increment the training set produces a further advantage. Thirdly, an interesting result is that knowledge learnt about in one task can be used to facilitate learning in a different but related task. Finally, the exploration into whether knowledge can transfer and aid learning between networks of different architectures has a less clear outcome; prior knowledge was successfully transferred, but was found to deteriorate as the network attempted to expand its learning. Further work involves exploring methods by which knowledge transfer between static and recurrent networks may prove beneficial in both learning and generalisation performance.

References

1. Altmann, G.T.M.: Learning and Development in Connectionist Learning. Cognition (2002) 85, B43-B50
2. Clarke, A.: Representational Trajectories in Connectionist Learning. Minds and Machines (1994) 4, 317-322
3. Elman, J.L.: Learning and Development in Neural Networks: the Importance of Starting Small. Cognition. (1993) 48, 71-99
4. Richardson, F.M., Davey, N., Peters, L., Done, D.J., Anthony, S.H.: Connectionist Models Investigating Representations Formed in the Sequential Generation of Characters. Proceedings of the 10th European Symposium on Artificial Neural Networks. D-side publications, Belgium (2002) 83-88
5. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning Internal Representations by Error Propagation. In: Parallel Distributed Processing. Vol. 1. Chapter 8. MIT Press, Cambridge (1986)