Recognition and anticipation of hand motions using a recurrent neural network

Peter Vamplew and Anthony Adams
Artificial Neural Networks Research Group
Department of Computer Science
University of Tasmania
email: vamplew@cs.utas.edu.au

Abstract: *Previous work in recognition of hand gestures has concentrated on classification of hand shapes, with relatively little work done on hand motions. This paper describes a recurrent neural network which has been trained to classify sixteen different hand trajectories, including relatively complex paths such as circles and back-and-forth motions. The network's ability to anticipate the classification of an incomplete gesture is also examined, and its implications for segmentation of gestures is discussed.*

Introduction

Computer recognition of human hand gestures has potential for application in many fields such as virtual reality interfaces, robotic control and automated sign language translation. Hand data can be captured either via a camera and image processing techniques, or directly through an instrumented glove worn by the user. Pattern recognition techniques can then be applied to this data to classify the gesture made by the user.

Components of hand gestures

Most of the analysis of hand gestures has originated from research into Deaf sign languages. The most common method of describing gestures is in terms of their four primary components - handshape, orientation, place of articulation (or location) and motion [1]. Handshape refers to the flexion of the fingers and wrist, orientation to the angle of the hand, and place of articulation to the location of the gesture relative to the body. Motion is the most complex feature as it can consist of changes over time in any combination of the other three features. For example, opening and closing of the fingers changes the handshape, twisting the wrist changes the orientation and moving the hand through space changes the location.

Existing work on gesture recognition

Previous research has focused primarily on recognising handshapes [2], [3]. Standard feedforward neural networks have been successfully applied to this problem. For example, Fels and Hinton (4) developed the GloveTalk system, which discriminates between 66 different handshapes with an accuracy rate of about 98%. The orientation and location components of signs are similar in nature to handshapes, and therefore the extension of existing techniques to these aspects of gestures should be relatively straight-forward.

In contrast the recognition of hand motions is a more complex problem which has been relatively lightly researched. GloveTalk can distinguish between six different motions (back-and-forth motions along the primary axes), whilst the GIVEN system [5] recognises seven different motions (movement in either direction along these axes, plus stationary). Murakami and Taguchi (6) applied recurrent neural networks to recognising ten different signs from Japanese sign language, for which both handshape and motion were important, achieving an accuracy of 96%.

Data gathering

As part of research on the SLARTI sign language recognition system [7],[8], a prototype hand motion classifier has been developed based on a recurrent neural network. Motion data was gathered from a user wearing a CyberGlove equipped with a Polhemus sensor for measuring the location and orientation of the hand. Examples of gestures were gathered from three different users, who made several examples each of sixteen different motions. These consisted of the same motions used by GIVEN, as well as back-and-forth motions along the main axes, and circling movements in these axes (both clockwise and counterclockwise). The start and end of a motion were indicated by the user via a button on the CyberGlove's wrist (this was the only data gathered from the CyberGlove which otherwise acted only as a mount for the Polhemus). 560 of these examples were used as a training set, and the remaining 320 as a test set.

Rather than working directly from the raw data the three position values from the Polhemus were pre-processed by calculating the difference between the current location and the previous one. Using these differences as the input to the network was intended to improve the system's spatial invariance.

Network architecture

A recurrent network was used in preference to a feedforward network presented with the entire sequence at once for a number of reasons. Primary amongst these was the length of the sequences in the data set. The longest of these was 10 time frames. A feedforward network would need 30 inputs to manage this example, whereas a recurrent network would need only 3 inputs. The resultant reduction in the number of free parameters in the network should improve the generalisability of the system. In addition a recurrent

network should be more immune to variations in the speed of gestures.

The network used consisted of 3 input nodes, completely interconnected to a single layer of processing nodes. This layer consisted of 16 output nodes and 14 additional hidden (or state) nodes, and all nodes in this layer were recurrently connected to every other node in the layer (including self-recurrent connections). The processing nodes all used the symmetric sigmoid as a squashing function. This architecture varies from the more commonly used Elman network in not having a hidden layer, and in having recurrent links to and from the output nodes. Whilst the absence of a hidden layer makes some problems such as a temporal XOR impossible, the state nodes can act a replacement for the hidden layer, but only in the next time-frame. The addition of recurrent links between the output nodes allows the development of positive self-recurrent links and negative recurrent links to the other output nodes, which appears to help to stabilise the network's behaviour.

The network was trained using the backpropagation through time (BPTT) algorithm. Effectively this unrolls the network to form a non-recurrent network the length of the training sequence and then backpropagates the error through this network (for a more technical description see [9]). One issue which arose during this process was the nature of the training signal to be presented to the network. The only point at which the desired output of the network is known is at the end of the sequence, when one output should be 0.4 and the rest -0.4. To use BPTT effectively it is necessary to generate training values for the time frames earlier in the sequence. A ramped signal (where the training signals were linearly interpolated from all 0 at the start of the sequence to their desired values at the end) was tested, but training was slower and less effective than using a flat signal where the end training values were also used for all the earlier frames in the sequence. The results reported in this paper are for the networks trained with this flat signal.

Recognition results

Ten networks were trained from different starting weights, with results as summarised in Table 1. A step size of 0.05 was used and the networks were trained for a maximum of 50,000 pattern presentations, testing their performance every 1000 presentations. During training the weights of the network for its best test set performance were retained and used to generate the results in Table 1, which shows that the networks achieved an extremely high level of accuracy on the test data.

	Training set	Test set
Mean	95.9	98.9
Minimum	95.0	98.4
Maximum	96.6	99.4

Table 1: Summary of the classification rate of ten recurrent neural networks trained to distinguish between 16 different hand motions

Anticipatory classification

An interesting feature of recurrent networks is that they produce an output for each time-frame in a sequence rather than producing only a single output at the end of the input sequence. By examining these intermediate outputs of the network it may be possible to classify a sequence correctly before the end of the sequence is actually reached. This has the benefit of reducing the response rate of the network which can be important for real-time applications. It may also have specific benefits for gesture recognition as a foundation for identifying individual gestures within a continuous series of hand motions, such as is used in signing.

The anticipatory abilities of the motion recognition were tested by applying a threshold to the value of the highest output node at each time step in the sequence. If the threshold was exceeded the sequence was classified as that gesture and the rest of the sequence was ignored. If the end of the sequence was reached without the threshold being exceeded then the sequence was classified as normal on the basis of the final output values. A range of different threshold values were examined on the test data for their hit rate (percentage of signs exceeding the threshold), the accuracy of their classification and the speedup they obtained (in terms of the percentage of the sequence actually processed prior to classification). The results of these experiments are summarised in Table 2.

From these results it can be seen that it was possible to classify many of the sequences well before their actual end with relatively little impact on the classification accuracy. For example by using a threshold of 0.25 we can reduce the network's response time by almost 40% whilst still maintaining a classification accuracy of 99%. This ability of the network to classify early in gestures leads to the possibility of automatically detecting the end of a gesture by performing this anticipatory classification and signalling the end of the gesture when that output node falls below a second threshold value. This would remove the need for the user to manually flag the end point of a gesture and would greatly improve the flexibility of a gesture recognition system.

Threshold	Hit rate (a)	% correct on thresholded gestures (b)	Speedup on thresholded gestures	% correct on all gestures	Speedup on all gestures	Segmentation accuracy (a x b)
-0.4	100.0	21.5	12.9	21.5	12.9	21.5
-0.35	100.0	21.5	12.9	21.5	12.9	21.5
-0.3	100.0	39.4	19.8	39.4	19.8	39.4
-0.25	100.0	55.6	26.2	55.6	26.2	55.6
-0.2	100.0	68.7	31.0	68.7	31.0	68.7
-0.15	100.0	79.7	35.3	79.7	35.3	79.7
-0.1	100.0	88.0	39.6	88.0	39.7	88.0
-0.05	99.9	92.5	43.0	92.5	43.1	92.4
0	99.7	94.7	46.2	94.7	46.4	94.4
0.05	99.4	96.6	48.7	96.6	49.0	96.0
0.1	99.1	97.6	51.4	97.6	51.8	96.7
0.15	98.3	98.5	54.1	98.4	54.9	96.8
0.2	97.6	98.9	57.8	98.9	58.8	96.5
0.25	96.6	99.0	60.5	99.0	61.9	95.6
0.3	94.3	99.1	63.7	99.1	65.8	93.5
0.35	88.5	99.1	66.9	99.2	70.7	87.7
0.4	71.3	99.2	69.2	99.2	78.0	70.7

Table 2: Mean results over ten networks of anticipatory classification of test data using different threshold values

To implement a segmentation algorithm it is necessary to choose a threshold which produces close to 100% both in gestures exceeding the threshold and in classifying those gestures. The final column in Table 2 summarises performance in this area by multiplying the hit rate and thresholded classification accuracy. For this problem thresholds in the range from 0.05 to 0.2 provide suitable performance in both of these categories (around 96-97% segmentation accuracy), which means they could be used for detecting the start of a gesture. Due to the pre-segmented nature of the data used it has not been possible to test the ability of the network to detect the end of a gesture, but it appears likely that similar thresholding techniques should prove equally effective for that task. A possible extension to the thresholding algorithm which may be useful for noisier data would be the inclusion of a temporal aspect to the threshold, such that the network's output must remain above the threshold for a certain period of time before the threshold is activated.

Conclusion

The recurrent network developed improves on both the number and complexity of hand motions recognised by previous systems, whilst maintaining a high level of generalisation to unseen examples. The thresholding technique described appears to have a great deal of potential for segmentation of gestures, which will allow the network to be applied to continuous sequences of gestures.

References

[1] Johnston, T., Auslan Dictionary - A Dictionary of the Sign Language of the Australian Deaf Community, Deafness Resources, Petersham, Australia, 1989

[2] Kramer, J. and Leifer, L., The Talking Glove: A Speaking Aid for Nonvocal Deaf and Deaf-Blind Individuals, RESNA 12th Annual Conference, New Orleans, Louisiana, 1989

[3] Takahashi, T. and Kishino, F., Hand Gesture Coding Based on Experiments Using a Hand Gesture Interface Device, in SIGCHI Bulletin, April 1991, pp 67-73

[4] Fels, S. and Hinton, G., Glove-Talk: A Neural Network Interface Between a Data-Glove and a Speech Synthesiser, in IEEE Transactions on Neural Networks, vol 4, no 1, January 1993, pp2-8

[5] Väänänen, K. and Böhm, K., Gesture Driven Interaction as a Human Factor in Virtual Environments - An Approach with Neural Networks, in Earnshaw, R.A., Gigante, M.A. and Jones, H. (eds.), Virtual Reality Systems, Academic Press, London, 1993, pp 93-106

[6] Murakami, K. and Taguchi, H., Gesture recognition using recurrent neural networks, in CHI91 Conference Proceedings, 1991, pp 237-242

[7] Vamplew, P. and Adams, A, The SLARTI System: Applying Artificial Neural Networks to Sign Language Recognition, in Proceedings of the Conference on Technology and Persons with Disabilities, California State University, Northridge, 18-21 March, 1992

[8] Vamplew, P., Sign Language Recognition Using Virtual Reality Gloves, in Loeffler, CE and Anderson, T, Virtual Reality Casebook, Van Nostrand Reinhold, 1994, pp 123-126

[9] Werbos, P.J., Backpropagation Through Time: What it does and how to do it, in Proceedings of the IEEE, vol 78, no 10, pp 1550-1560, 1990