ESTIMATING MOVEMENT DIRECTION WITH A NEURAL NETWORK

William C. Treurniet

Communications Research Centre
Communications Canada

ABSTRACT

Various schemes have been developed to identify the direction of motion in a scene, often to assist in compressing the information required to broadcast a sequence of images. These schemes typically make sequential comparisons of blocks of pixels in order to arrive at a most likely direction of motion. This paper investigates how an artificial neural network may be used to perform the same task. The approach is interesting because these networks perform computations in parallel, thus allowing a form of top-down as well as bottom-up processing. Also, because computations are conceptually performed in parallel, it would be possible to consider performing the task in real time with an appropriate hardware implementation. Preliminary results show that a properly trained network has interesting properties similar to those of real neurons and can, indeed, report direction of movement based on binary pixel values.

KEYWORDS: movement estimation, neural network

1. Introduction

Improvement of television image quality can readily be achieved, but usually at the cost of a considerable increase in the bandwidth required to broadcast the images. Thus, current work on defining high-definition television systems includes development of methods for image data compression. Movement-adaptive encoding of television images is one method under consideration. This scheme encodes an area of the television frame more compactly if it is determined to be very similar to an adjacent area in the previous frame. In that case, the image content is presumed to have moved from one location to another. Transmitting only the direction and extent of movement of display areas can result in a considerable saving in the amount of information that needs to be transmitted. A number of techniques to measure the similarity of two blocks of pixels have been developed (e.g., Huang, 1981; Sabri, 1984; Storey, 1986; Tsuda and Hiraoka, 1986; Puri, Hang, and Schilling, 1987; Lee and Griffiths, 1987). Motivation for this work was the need for an efficient algorithm which could perform the necessary comparisons within the time required.

The central issue for movement-adaptive encoding of television signals is recognition of the direction and extent of movement in an image. Tracking movement in a visual field conceptualized as an array of points is not an easy task for image processing systems. In computational vision research, following individual points through space in successive time intervals has been termed the correspondence problem (e.g., Marr, 1982). That is, a given point at an instant in time must be identified with one of many possible points at a later instant. When only local information is available, even the motion of edges is ambiguous. Under certain conditions, however, the velocities of non-parallel edge segments belonging to the same pattern offer constraints which unambiguously identify the exact direction of motion (Glazer, 1981; Adelson and Movshon, 1982). The measurement of the velocity gradient may be further improved by computing a vector that best fits each set of neighbouring vectors. This procedure helps to compensate for discontinuities, intensity fluctuations, and noise in the sampled image (Glazer, 1981).

In human vision, Ramachandran and Anstis (1983) showed that perceived motion in different parts of a display are not always independent. Bistable patterns presented simultaneously in random locations all showed the same direction of motion. Apparently, the perceived direction of movement in the visual field may be affected by global as well as local relationships in an image. Therefore, methods that process an image in both a top-down and a bottom-up manner should be useful for ascertaining the direction and extent of movement of image content.

The need for both local and global

information suggests the possible utility of artificial neural networks as image representation devices for analysis of movement. Because information is represented in a distributed way in such a network, it's behaviour in response to a particular local activation is affected by the state of the rest of the network. Thus, the network's "bottom-up" response to an input may be modulated by "top-down" influences from global information distributed throughout the network. Fahlman, Hinton, and Sejnowski (1983) suggested that this type of processing may be ideal for real world recognition problems where "there is usually a single answer ... that is much better than any other, but this can not be found by pure bottom-up or pure top-down processing; instead, like the solution of a set of simultaneous equations, it must either emerge as a whole or be found by laborious iteration". Fahlman et al. presented the Boltzmann machine (a distributed network with a stochastic learning algorithm) as an ideal device for solving complex recognition problems.

Sereno (1987) used the Boltzmann machine to model the behaviour of cortical neurons involved in motion analysis. Input units to the network represented specific speeds and directions of motion of edges along single dimensions, while output units represented movements of patterns as a whole in two dimensions. After training, the network exhibited behaviours consistent with that of biological neural systems involved in motion recognition. The model demonstrates that two-dimensional motion measurements can be derived from the integration of one-dimensional measurements. More specifically, it shows that artificial neural networks can compute the unambiguous velocity vector from the constraints imposed by the ambiguous vectors obtained from local computations.

Sereno's model used as input data describing the one-dimensional velocity vectors, and the issue of how to create these vectors from low-level pixel data was not addressed. For real time operation on video image data, the network should be able to estimate motion directly from intensity values or other low-level information such as colour. The following work uses the neural network approach to estimate movement direction directly from intensity data. Rather than the Boltzmann machine, however, the back-propagation learning model of Rummelhart, Hinton and Williams (1986) was used. It has the required distributed processing property for integrating global and local information, and a deterministic learning procedure that is faster and thus, more useful for solving real world problems.

2. The Back-Propagation Model

The back-propagation model is a practical procedure for automatically adjusting weights on connections in a network in order to encode patterns from the environment. The model assumes a layered network topology where no restriction is placed on the number of layers or the number of nodes per layer. Further, connections are allowed between pairs of nodes in different layers but not between pairs within a layer. One layer is designated the input layer where patterns from the environment are encoded, and another is identified as the output layer where the results of processing the input patterns are obtained. The intervening layers contain "hidden" nodes in that their inputs and outputs are not directly set or read, respectively, by the environment.

Relationships between input and desired output patterns are "learned" by the network by iterative adjustments of the connection weights. In each iteration, the sizes of the weight adjustments are directly related to the errors at the output nodes. The method by which the weights are adjusted in order to reduce the error on subsequent iterations is described in Rummelhart et al. (1986).

3. Training to recognize movement

Although the network model is capable of accepting continuous values as inputs, the decision was made to represent each pixel value above a certain threshold as '1', and as '0' otherwise. Figure 1 shows how a block of pixels from an image might appear in successive instants of time. In this example, a network would be expected to report movement of the image from left to right. The following work is based on such simulated images.

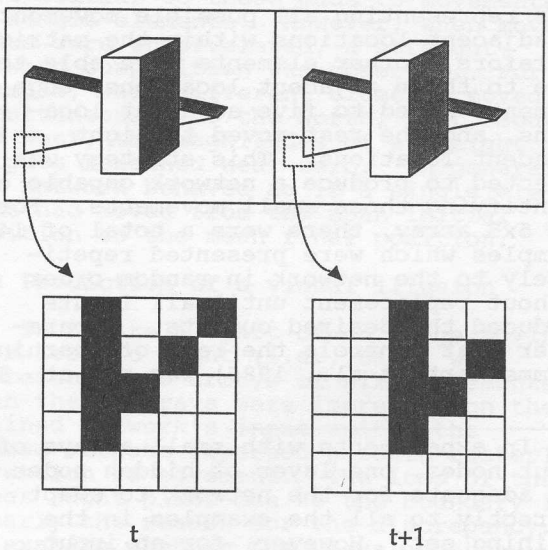

Figure 1: Representation of the movement estimation problem.

A 5x5 array of pixels was mapped onto an array of network input nodes, A(n), schematized in Figure 2. Two blocks of 25 nodes were used to encode

pixel values at two successive instants. Figure 2 represents a single pixel value initially at A(2) which has moved to the location indicated by A(33). The network's output array represented eight directions of movement. (Compass points, where "North" means "upward", are used as a convenience for indicating direction.) In the example shown, the output represents a movement in the South-East direction. There were two layers of hidden nodes with 20 nodes in the first layer and 10 in the second[1]. Adjacent layers of nodes were completely interconnected.

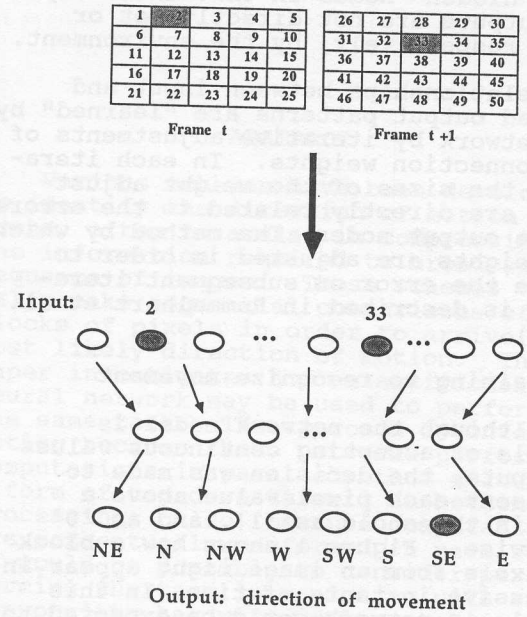

Figure 2: The meaning of network input and output units.

The training set consisted of examples representing all possible movements to adjacent locations within the matrix. Therefore, corner elements were able to move to three adjacent locations, edge elements moved to five adjacent locations, and the rest moved to eight adjacent locations. This strategy was expected to produce a network capable of identifying those small movements. For the 5x5 array, there were a total of 144 examples which were presented repetitively to the network in random order without replacement until all inputs produced the desired outputs. A parameter that controls the rate of learning (Rummelhart et al., 1986) was set at .8 and a "momentum" parameter was set at .9. The learning rate parameter was further adjusted to be proportional to the "fan-in" of each node (Plaut, Nowlan, and Hinton, 1986). Weight adjustments were made after every 8 examples, and learning continued until each value of each example's output was within .1 of the desired value of zero or one. Approximately 300 to 400 sweeps through all the examples were required to reach this criterion.

1. In experiments with small arrays of input nodes, one layer of hidden nodes was adequate for the network to adapt correctly to all the examples in the training set. However, for an input array representing areas larger than 4x4, there was an obvious symmetry in the pattern of examples not learned. Two layers of hidden nodes appeared to provide the complex decision space required to complete the classification task for the larger areas (Lippmann, 1987).

3.1 Response to untrained movements

The training procedure ensured that the network would give correct outputs to queries representing movements of pixel values to adjacent locations. It is interesting to observe the response of the network to queries representing movements not in the training set. Figure 3 shows the response to a movement from A(1) to A(48). Although this example had never been explicitly presented to the network during training, the magnitudes of the resulting output values seem to indicate correctly the direction of the movement. That is, the direction is mostly southward but also a bit eastward.

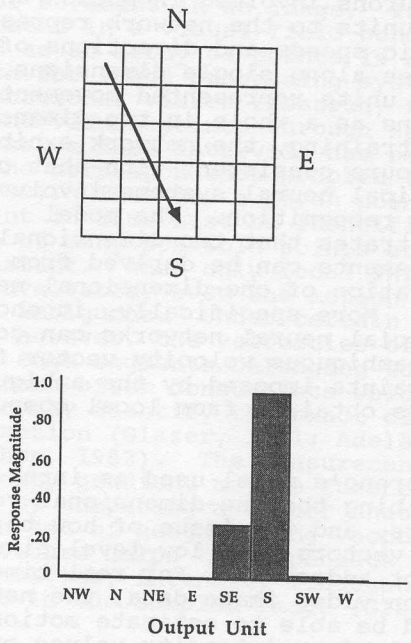

Figure 3: The response of network output units to an input representing an untrained movement.

The latter result suggests that summing the outputs as vector quantities may give a relatively precise indication of direction, even for movements not in the training set. This suggestion was confirmed by computing the directions of resultant vectors in response to movements to the edges from the center and the corner locations. Figures 4 and 5 show the computed directions for each location, respectively, plotted against the actual direction of the movements. Clearly, the correspondence is good. Figure 5 seems to indicate more

variability, but note that the scale is different in the two graphs.

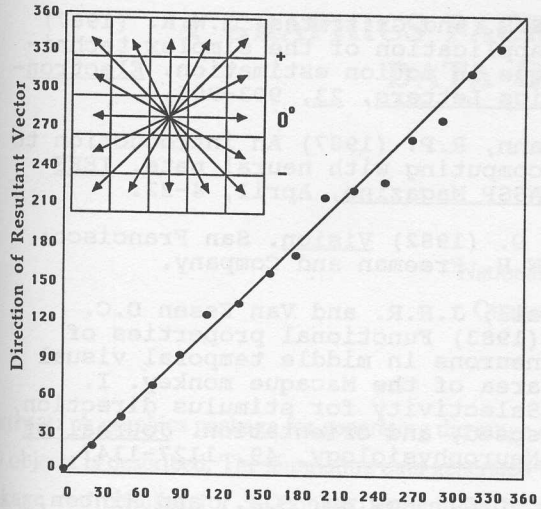

Figure 4: Vector sum of network outputs plotted against directions of movements originating at the center.

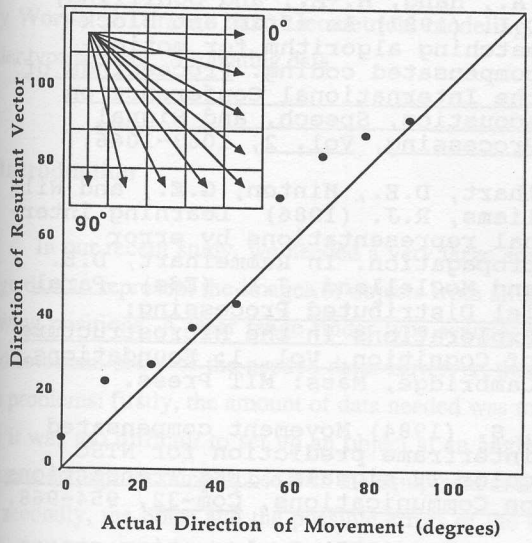

Figure 5: Vector sum of network outputs plotted against directions of movements originating at a corner.

The responses to the same movements were also examined to determine the "tuning curve" for each output node. Figure 6 was created by plotting, for each direction of movement, the response magnitude of a given output unit. (Curves are presented for only four of the eight units for clarity.) The figure shows that an output unit responds maximally to a range of directions. Figure 7 indicates that the tuning curve of a unit is relatively unaffected by the origin of the movement (i.e., the corner or the center).

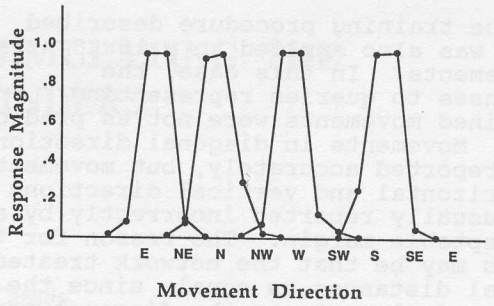

Figure 6: Tuning curves representing responses of four output units to the movements depicted in Figure 4.

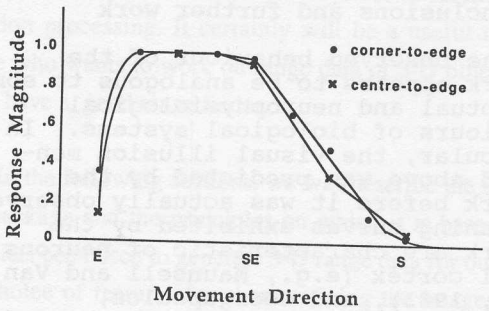

Figure 7: Tuning curves representing responses of the same unit to movements originating at the center or at the corner.

3.2 Response to moving several elements

The behaviour of the network in response to simultaneous movements by more than one element was also examined. For correlated movements, the responses were similar to those made to movements by single elements. However, when movements were uncorrelated, a straightforward competition was observed. For example, when different elements moved simultaneously to the North, East, South and West, the magnitudes of all eight outputs remained near zero. In general, the network behaved as if a single element had moved from the mean initial position to the mean final position.

3.3 Prediction of a visual illusion

In Figure 1, part of an image moving from left to right (i.e., East) was represented as arrays of pixel elements. When these arrays were impressed on the trained network's input units, the summed outputs reported a south-easterly direction of movement. The loss of the upper right pixel due to the boundary is clearly the reason for the discrepancy. To examine whether the visual system responds in the same way to a similar stimulus situation, an experiment was prepared to test the direction of apparent motion in an alternating display consisting of the two views. The perceived movement was, indeed, in the direction predicted by the network.

3.4 Problems with scaling up

The training procedure described above was also applied to a 15x15 array of elements. In this case, the responses to queries representing untrained movements were not as predictable. Movements in diagonal directions were reported accurately, but movements in horizontal and vertical directions were usually reported incorrectly by an unacceptable margin. The reason for the errors may be that the network treated unequal distances as equal, since the number of elements on the diagonal of the matrix was the same as the number along the edges. Thus, the network may have learned properties of a non-euclidean space, but this is apparent only with larger arrays.

4. Conclusions and further work

The observed behaviour of the network appears to be analogous to some perceptual and neurophysiological behaviours of biological systems. In particular, the visual illusion mentioned above was predicted by the network before it was actually observed, the tuning curves exhibited by the network are characteristic of neurons in visual cortex (e.g., Maunsell and Van Essen, 1983), and Georgopoulos, Schwartz, and Kettner (1986) observed that vector summation of the outputs of broadly tuned neurons in motor cortex precisely indicated directions of corresponding overt motor movements. Therefore, a network trained in the above manner appears to model some types of central nervous system behaviours.

From an applied research perspective, a network's ability to identify direction of movement of an ensemble of points was shown. Training procedures are being studied to create a network that can report on the extent of movement as well. The network's behaviour with real images is being examined, and adequate performance may eventually lead to development of special hardware to process video images in real time.

References

Adelson, E.H. and Movshon, J.A. (1982) Phenomenal coherence of moving visual patterns. Nature, 300, 523-525.

Fahlman, S.E., Hinton, G.E., and Sejnowski, T.J. (1983) Massively parallel architectures for AI: NETL, Thistle, and Boltzmann machines. Proceedings of the AAAI-83 Conference, Washington.

Georgopolous, A.P., Schwartz, A.B., and Kettner, R.E. (1986) Neuronal population coding of movement direction. Science, 233, 1416-1419.

Glazer, F. (1981) Computing optical flow. Proceedings of the IJCAI, Vol. 2, Vancouver.

Huang, T.S. (1981) Image Sequence Analysis. New York: Springer-Verlag.

Lee, B.S. and Griffiths, J.W.R. (1987) Application of the simplex technique in motion estimation. Electronics Letters, 23, 903-905.

Lippmann, R.P. (1987) An introduction to computing with neural nets. IEEE ASSP Magazine, April, 4-22.

Marr, D. (1982) Vision. San Francisco: W.H. Freeman and Company.

Maunsell, J.H.R. and Van Essen D.C. (1983) Functional properties of neurons in middle temporal visual area of the Macaque monkey. I. Selectivity for stimulus direction, speed, and orientation. Journal of Neurophysiology, 49, 1127-1147.

Plaut, D.C., Nowlan, S.J., and Hinton, G.E. (1986) Experiments on learning by back propagation (Tech. Rep. No. CMU-CS-86-126). Pittsburgh, PA: Carnegie-Mellon University, Department of Computer Science.

Puri, A., Hang, H.-M., and Schilling, D.L. (1987) An efficient block-matching algorithm for motion-compensated coding. Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, Vol. 2, 1063-1066.

Rummelhart, D.E., Hinton, G.E., and Williams, R.J. (1986) Learning internal representations by error propagation. In Rummelhart, D.E. and McClelland, J.L. (Eds), Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations, Cambridge, Mass: MIT Press.

Sabri, S. (1984) Movement compensated interframe prediction for NTSC color TV signals. IEEE Transactions on Communications, Com-32, 954-968.

Sereno, M.E. (1987) Implementing stages of motion analysis in neural networks. Ninth Annual Conference of the Cognitive Science Society, 405-416.

Storey, R. (1986) HDTV motion adaptive bandwidth reduction using DATV. BBC Research Department Report No. BBC RD 1986/5.

Ramachandran, V.S. and Anstis, S.M. (1983) Perceptual organization in moving patterns. Nature, 304, 529-531.

Tsuda, T. and Hiraoka, M. (1986) Efficient video bandwidth compression. Fujitsu Scientific and Technical Journal, 22, 355-366.