IEEE International Symposium on Circuits and Systems, 2000, 4, 405-408

REVERSE ENGINEERING OF THE VISUAL SYSTEM USING NETWORKS OF SPIKING NEURONS

S.J. Thorpe, A. Delorme, R. Van Rullen, W. Paquier

Centre de Recherche Cerveau & Cognition
133, route de Narbonne
31062, Toulouse, France

ABSTRACT

Recent research has shown that the speed of image processing achieved by the human visual system is incompatible with conventional neural network approaches that use standard coding schemes based on firing rate. An alternative is to use networks of asynchronously firing spiking neurones and use the order of firing across a population of neurones as a code. In this paper we summarize results that demonstrate a number of advantages of such coding schemes: (1) they allow very efficient transmission of information, (2) they are intrinsically invariant to variations in stimulus intensity and contrast, (3) they can be used in very large scale processing architectures to solve difficult problems including categorisation of objects in natural scenes, and (4) they are particularly suited for implementation in low-cost multi-processor hardware.

1. INTRODUCTION

Over the last two decades, artificial neural networks have been used in a wide range of applications in areas as diverse as engineering and control systems as well as financial prediction. However, in many areas, even the most sophisticated artificial systems look feeble when compared with their biological counterparts. Take the case of vision. Both humans and monkeys can decide very rapidly whether a briefly flashed colour photograph contains an animal [2, 5], even when the image has never been seen before. Such levels of performance are way beyond the capabilities of current machine vision systems. What is it that makes biological vision so effective? Is it possible to use knowledge of visual processing in biological systems to devise novel engineering solutions that could one day rival human vision?

In this paper, we will argue that there is one particular feature of biological visual systems which is absent from virtually all systems that use artificial neural networks. Conventional neural networks use large arrays of processing elements, roughly equivalent to neurones, each of which is characterised by an activity level which is often a continuous variable in the range 0-1. However, real neurones do not in general transmit information in the form of a continuous analog signal. Instead, they send a series of all-or-none pulses or spikes. It is only relatively recently that researchers have come to realize that the use of spikes dramatically changes the types of computation that can be performed by a neural network [4]. In this paper we will argue that the use of networks of spiking neurons may be a key feature underlying the efficiency of biological vision systems, and that spikes may provide a particularly efficient way of implementing neural networks in parallel digital hardware.

2. CODING WITH SPIKES

2.1 Problems with Conventional Rate Codes

The analog activation value attributed to units in artificial neural networks is often taken to correspond to the firing rate of biological neurones. But recent work on the speed of processing in the visual system has raised questions about the viability of such a scheme[3]. For example, in a scene classification task, monkeys can have behavioural reaction times that can be as short as 180 ms. If one subtracts roughly 80 ms for initiating and executing the motor response, this leaves only about 100 ms for visual processing. Interestingly, this is roughly the onset latency of neurones in the inferotemporal cortex, the highest order visual processing stage in the primate visual system, implying that much if not all of the underlying processing can be achieved with a single feed-forward pass through the various levels of the visual pathways. Indeed, to get to inferotemporal cortex, information about the image has to traverse a number of processing stages that include ganglion cells in the retina, relay cells in the thalamus, cortical areas V1, V2, V4, and the posterior inferotemporal cortex. Calculations suggest that each of these stages only has about 10 ms to do the necessary computation. Ten milliseconds may seem plenty of time given the speed of today's electronics, but the neurones in the visual system rarely generate pulses at more than 100 or so spikes per second. This means that in many cases, each neurone will only fire one spike during the criticial 10 ms available for processing. How can the visual system function with so little time for processing by each individual element?

The usual response is to use large numbers of neurons in parallel. Suppose that we wish to code a particular parameter, such as the grey-scale value of a pixel, with 10 possible values and that we want to do this within a 10 ms time period, too short to allow any individual neuron to fire more than one spike. Obviously we could choose to use 10 different neurones, and simply count the number of cells that fire within the 10 ms period. This will certainly work, but has the drawback that it requires very large numbers of units. For example, we know that the optic nerve contains roughly 1 million fibres. This number of fibres is clearly sufficient to transmit the contents of the entire visual field to the brain, so it seems unlikely that there can be much redundancy available for coding individual pixels in the image. Rather, it would appear that the retina is using a much more efficient coding scheme to transmit information to the brain, but what is it?

2.2 Rank Order Coding

One possibility takes advantage of the fact that a neuron can be thought of as an analog-delay convertor. It acts somewhat like a capacitance which is progressively charged by an input until it reaches a threshold, at which point it generates an output pulse – the action potential or spike. Such neurons will naturally fire earliest when the input is strong, and will take progressively longer to fire when the input is weaker. In this way, the time at which a neuron fires (its response latency) can be used to code the intensity of the stimulus.

However, this sort of code requires knowledge of when the stimulation started, information which is not generally available in the case of the biological visual system. There is, however, a way round this. Consider what happens when several neurons are used in parallel. In this case, even without knowing the precise moment at which the stimulus came on, information can be obtained by looking at the order in which the neurones fire [6].

It is not difficult to see that the order of firing of a group of neurons is potentially a very rich source of information about the input pattern. Take for instance the 10 neurons that we previously used to code ten grey scale levels for a single pixel. Suppose that those 10 neurons are connected to 10 different pixels, and that we can look at the order in which they fire. With 10 neurons, there are factorial 10 possible orders than can occur – a total of over 3.6 million. Each of these corresponds to a different intensity profile and thus provides a great deal of information about the stimulus within a very short period of time.

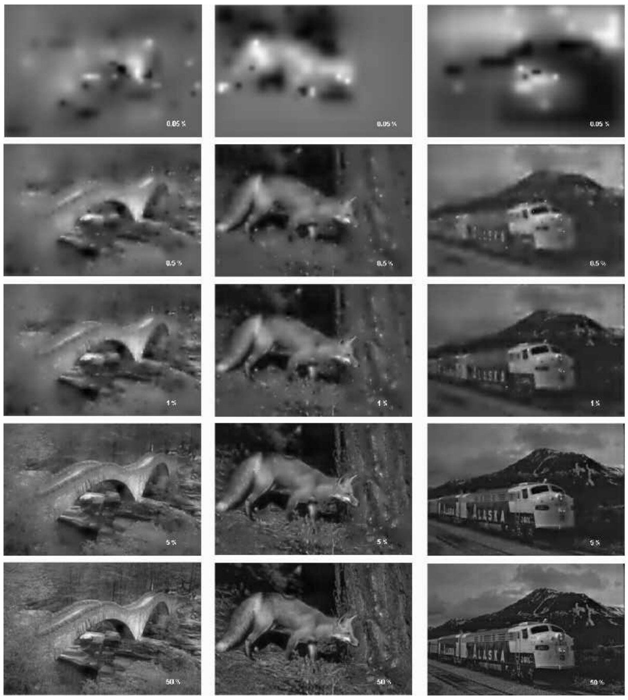
3. BIOLOGICAL IMAGE COMPRESSION

To illustrate how such a scheme can work, we will look at the work of Rufin van Rullen, who has recently examined how this sort of rank-order coding scheme could be used by the retina to transmit information to the brain[8]. Van Rullen used a very simple model of the retina, in which two different sets of neurones were used – On-centre and Off-centre cells. ON-centre cells respond best to a bright spot of light on a dark background, whereas the Off-centre cells prefer dark spots on a bright backgroung. The use of the analog-delay conversion scheme that we just described means that the time of firing will vary depending on the local contrast, with high contrasts leading to short response latencies. Van Rullen also used ON- and OFF-centre cells at different scales, starting with one cells per pixel at the highest resolution, one cell for every four pixels when the scale was doubled and so on.

Imagine now that we are in the brain, listening to the activity in the hundreds of thousands of nerve fibres as a result of the presentation of a natural image. Our job is to try and reconstruct the image as well as we can, simply on the basis of the spikes in the optic nerve. How might we go about this? Let us suppose that the first spike occurs in a cell that we know corresponds to a large OFF-centre retinal cell centred at a particular location in the retina. Obviously we can plug the receptive field shape of this cell in the reconstructed image. Subsequently, as more and more spikes arrive, we can progressively fill in the image using the receptive field profile of the neurons that fire. Note however, that we can be a little more sophisticated. We know that in principle, the first neurons to fire will correspond to the places in the image where the contrast is highest, whereas those neurons that fire later on should be given a lower impact. In principle, there is no strict law determining how one should decrease sensitivity, but Van Rullen calculated the average contrast in natural images associated with particular orders using a representative set of 3000 natural scenes.

By using this empirical Look-Up Table for contrast as a function of order, it is possible to see how well one can reconstruct an image as a function of the number of cells that have fired. This is illustrated in Figure 1, which shows that even when only 1-2% of cells have fired one spike, it is often possible to make quite clear statements concerning the contents of the image.

Figure 1. Progressive reconstruction of images



based on the order of firing of retinal ganglion cells [8].

Comparisons with coding schemes based on neurons that uses rate coding showed that coding based on the order of firing of retinal ganglion cells is considerably more efficient. Furthermore, it is interesting to note that this sort of coding scheme has another important advantage in that it performs an automatic normalisation of the image with respect to contrast and luminance. Since the only thing that is important for the reconstruction is the order of firing, exactly the same output image would be produced if the contrast of the image was reduced or if the luminance was lower. Of course, we lose the ability to recover the absolute grey level values of the pixels in the image, but that is true for human vision too – humans are also notoriously inaccurate at estimating true image luminance levels.

Van Rullen's study of retinal coding demonstrates that the order of firing of spiking neurons can be used to code information efficiently. In the next section we will show that a simple biologically plausible mechanism can be used to make neurons sensitive to the order of firing of their inputs.

4. RANK ORDER DECODING

Suppose that we want to make a neuron that will respond selectively to the order in which its inputs fire. One obvious possibility would be to use a series of delay lines so that the inputs will arrive synchronously only if the correct delays are used. This strategy is used in a range of sensory systems that use temporal differences between arrival times of sensory stimuli, as for example in the case of auditory sound localisation. However, it is an expensive strategy that needs a relatively large amount of specialist hardware.

An alternative "trick", much simpler to implement, is to use fairly standard neurons, but include a mechanisms that progressively desensitizes the post-synaptic neuron following each incoming spike [6]. Consider a neuron with 3 inputs, A, B and C, that have relative synaptic weights of 3, 2 and 1 units. Suppose that each time an input arrives, the sensitivity of the neuron is cut by a factor of 50%. Now, how much exicitation will the cell receive if the inputs fire in the order A>B>C? It is easy to see that the final activation will be equal to 3 + (2*0.5) + (1*0.25) = 4.25 and that this is the highest possible activation that can be produced when each input is only allowed to fire once. For example, the opposite order, C>B>A, would only produce 2.75 units of excitation. Thus by setting the threshold at an appropriate level, one can make the neuron as selective as one wants. With three inputs, the total number of possible orders is very low – just 6, but with more realistic numbers of inputs, neurons can be made extremely selective.

5. SPIKENET

To test the impact of such ideas, we have developed a simulation system called SpikeNET that is specifically designed for simulating very large networks of asynchronously firing integrate-and-fire neurons[1]. The results have been extremely encouraging and suggest that the computational potential of networks of spiking neurons may be very high. In the final sections of this chapter, we will illustrate how SpikeNET can be used to develop image processing systems of a radically new type that are (a) capable of performing challenging visual categorisation tasks using natural images, and (b) suitable for implementation on low cost parallel computer hardware.

5.1 Face Detection with SpikeNet

Van Rullen et al showed that a four layer feed-forward network of asynchronously spiking neurons could be used to detect the presence of a face, even when none of the neurons fires more than one spike [7]. The network was an extremely simple one composed of three layers (see figure 2). Layer one corresponds very roughly to the retina and contains On- and Off-centre cells that generate at most one spike, with a latency that depends on local contrast, just as in the model presented in the previous section. These cells project to 8 arrays of neurons in layer two, each tuned to respond optimally to contours at a particular orientation. Thus cells in the map corresponding to 0° will respond very rapidly when a high contrast vertically oriented edge is present at a particular location in the image. These second level units project to a set of maps in the third layer, that were "trained" to respond selectively to the presence of a right eye, a left eye or a mouth in the image. The training scheme uses a a supervised learning algorithm that fixes synaptic weights as a function of the order in which the inputs fire – inputs that tend to fire early are given high weights, whereas those that fire late are given low weights.

This arrangement, together with the progressive desensitisation mechanism described in the previous section, makes the neurons in the third layer selective to the particular face features. Finally, these maps feed on to a final map that responds to the presence of appropriately located activity in the three input maps, namely a left eye, right eye and mouth.

Figure 2. A simple three layer feed-forward architecture that is capable of accurate face identification [7].

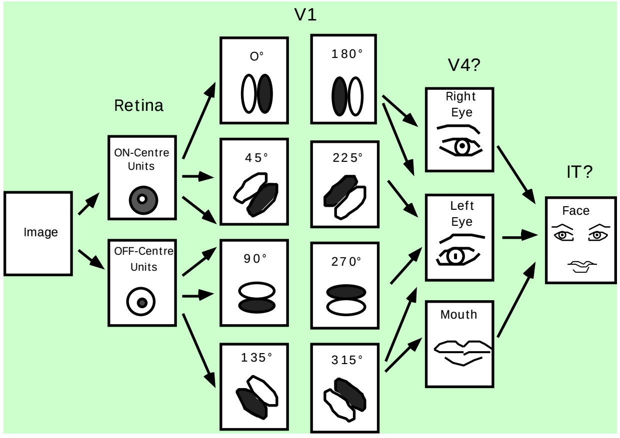
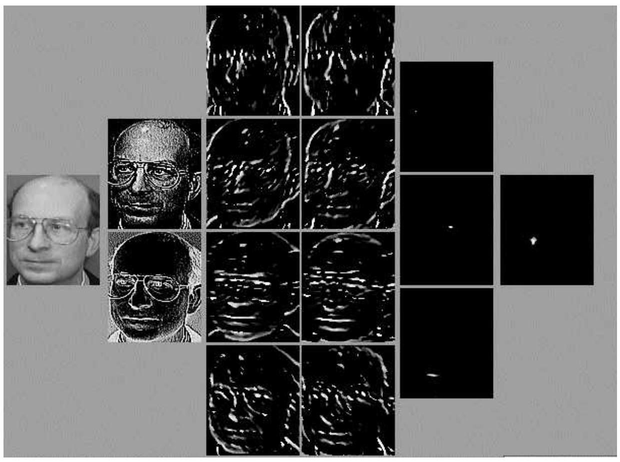


Figure 3 shows how activity propagates through such a network in response to an image containing a face. Within each map, the grey scale values are used to depict the order of firing – bright points correspond to places where the neurons fired early on, whereas neurons that fired later on are increasingly dimmer. It can be seen that activity in the the three feature maps and within the "face" map is localised and can be used to determine the position of the face in the image.

Figure 2. Propagation of activity within the face detection network described by Van Rullen et al [7].



5.2 Parallel implementation

Even on a single desktop computer, SpikeNet is capable of simulating large networks of neurons efficiently. For example, recent work has involved networks containing as many as 30 million neurons and over 3 billion synaptic connections. But despite their large size, such networks can still be run on a single workstation, as long as there is plenty of RAM available. Obviously the penalty one has to pay is speed, because with such a large system, propagating a single image can take 30 seconds or more. However, the very nature of SpikeNet means that it is relatively straightforward to take advantage of clusters of workstations and even low cost multiprocessor hardware. The reason is that each array of neurons can be handled by a separate processor using local memory. In this case, communications between processors are essentially limited to sending lists of spikes, i.e. the neurons that fired during the previous time step.

Let us take a concrete example. Suppose that we want to simulate a network with 5 million neurons using 5 workstations linked together in a cluster, and that real-time performance requires an average firing rate of 1 spike/second per neuron. Each second, each workstation would need to transmit the coordinates of 1 million units, which, using a code that used 20 bits per neuron, would require a debit of 20 Mbits per second. By using multicast techniques it would theoretically be possible to simulate such a 5 million neuron network in real time without requiring more than standard 100 Mbit Fast Ethernet connection technology. And with techniques such as Gigabit Ethernet and the VIA protocol (Virtual Interface Architecture), it will soon be possible to increase the network traffic by a factor of 10.

We are also working on multiprocessor hardware that can be used to further improve parallelism. In collaboration with Simtec Electronics (http://www.simtec.co.uk), a UK-based company we have developed a computational module based on the low-power StrongARM processor. Each module is composed of the CPU, 64 Mbytes of SDRAM, a Flash Memory chip, and a PCI bridge chip. Eight such modules can be mounted on a single PCI board, and four such boards can be fitted in a single desktop PC. In conventional workstation clusters, communications between processors are handled using standard Ethernet interconnects (typically limited to 100 Mbits per second). But in the case of the multiprocessor boards, communications between processors make use of local PCI interconnections that allow transfers at up to 132 MBytes per second – more than 10 times the debit of fast Ethernet. Furthermore, since each PCI board would effectively have its own local PCI bus, the total aggregate bandwidth within a PC equipped with 4 of the 8 CPU boards could be as high as 2 Gigabytes per second. In addition, second generation PCI bridge chips will mean that the speed of data transfers could be increased by a factor of four.

Interprocessor communication limits the number of spikes that can be handled per second. But it is the local processor and memory architecture that defines how many synapses can be implemented. Suppose that a particular processor receives a spike from a processor located somewhere else. At this point, the SpikeNet kernel has to update all the local neurons that receive inputs from that neuron. The number involved could be large, maybe hundreds or even thousands. However, all these local computations only involve local memory accesses, which are independent of the rest of the system. Furthermore, there is enormous potential for optimising the code using multimedia instruction such as MMX, SSE and Altivec. For this reason, it seems likely that by using fast interprocessor communications for transmitting spikes, and highly optimised local processing, it should soon be possible to develop architectures capable of allowing real-time simulation of spiking neural networks with millions of neurons and billions of connections.

6. PERSPECTIVES

We have really only just begun what will certainly be a long term project aimed at reverse engineering the primate visual system. For the moment, we have only explored some very simple architectures, involving essentially just feed-forward architectures involving a relatively small numbers of layers. However, the results obtained so far indicate that we may well be on the right track. Performance in tasks such as face identification is remarkably good, given the simplicity of the architectures used, and we are convinced that adding features such as horizontal connections between neurons at the same level, and feedback connections between structures will improve performance even more. In the years to come, we will strive to incorporate as many of the computational tricks used by the primate and human visual system as possible. More to the point, it seems that by adopting the spiking neuron approach, it will soon be possible to develop sophisticated systems capable of simulating very large neuronal networks in real time. The enormous potential of such an approach has led to the creation of a start-up company, SpikeNet Technology (http://www.spikenet-technology.com), that is aimed at making the potential of spiking neural networks available to end-users in a wide range of application areas.

7. REFERENCES

[1] Delorme, A., Van Rullen, R., Gautrais, J., and Thorpe, S., SpikeNET: a simulator for modeling large networks of integrate and fire neurones. *Neurocomputing,* 26-27:989-996, 1999.

[2] Fabre-Thorpe, M., Richard, G., and Thorpe, S.J., Rapid categorization of natural images by rhesus monkeys. *NeuroReport,* 9:303-308, 1998.

[3] Gautrais, J. and Thorpe, S., Rate coding versus temporal order coding: a theoretical approach. *Biosystems,* 48:57-65, 1998.

[4] Maass, W. and Bishop, C.M., *Pulsed neural networks*. Cambridge, Mass., MIT Press, 1999.

[5] Thorpe, S., Fize, D., and Marlot, C., Speed of processing in the human visual system. *Nature,* 381:520-2, 1996.

[6] Thorpe, S.J. and Gautrais, J., *Rank Order Coding*, in *Computational Neuroscience: Trends in Research 1998*, J. Bower, Editor. 1998, Plenum Press: New York. p. 113-118.

[7] Van Rullen, R., Gautrais, J., Delorme, A., and Thorpe, S., Face processing using one spike per neurone. *Biosystems,* 48:229-39, 1998.

[8] Van Rullen, R. and Thorpe, S., Ultra-Rapid Visual Categorisation of natural scenes: Order of spiking in ganglion cells as a code. *Perception,* (in press), 1998.