A System for Transmitting a Coherent Burst of Activity Through a Network of Spiking Neurons

J. Bose, S.B. Furber, and J.L. Shapiro

School of Computer Science, University of Manchester,
Manchester M13 9PL, UK
bosej@cs.manchester.ac.uk,
{steve.furber, jonathan.shapiro}@manchester.ac.uk

Abstract. In this paper we examine issues involving the transmission of information by spike trains through networks made of real time asynchronous spiking neurons. For our convenience we use a spiking model that is has an intrinsic delay between an input and output spike. We look at issues involving transmission of a desired average level of stable spiking activity over many layers, and show how feed-back reset inhibition can achieve this aim. We then deal with the coherence of spike trains and show that it is possible for a burst of spikes emitted by a layer to not diverge when passing through different layers of neurons. We present the results of simulations done on a multi layered feed-forward system to illustrate our method.

1 Introduction

Spiking neural models have been a source of interest [1] due to their biological plausibility and computational power. We are interested in engineering high level systems such as associative memories out of low level components such as spiking neurons, and in this paper we deal with some of the modelling issues in any such undertaking. Spiking neurons transmit information in the form of electrical pulses called spikes, whose firing times carry information. High level systems transmit information as symbols, which can be translated to a series or burst of spikes in the equivalent low level model. In a spiking neuron model of a high level system, in order to preserve the integrity of the transmitted information it is important for these spike bursts forming symbols to be stable (not die out or explode as it is transmitted through layers of neurons) and coherent (different bursts of spikes should not interfere with each other: there should be an appreciable time lag between them so that the symbols can be distinguished). In this paper we seek to tackle these two issues. There has been previous work done in these issues: studies have been made of the dynamics of activity in synfire chains [2, 3], consisting of neurons linked in a feed-forward chains propagating spiking activity.

In this paper, we have used simulations of spiking systems to illustrate our solutions, mainly because we encountered the mentioned problems during our efforts to model a real memory through spiking neurons. For our simulation

B. Apolloni et al. (Eds.): WIRN/NAIS 2005, LNCS 3931, pp. 44–48, 2006.

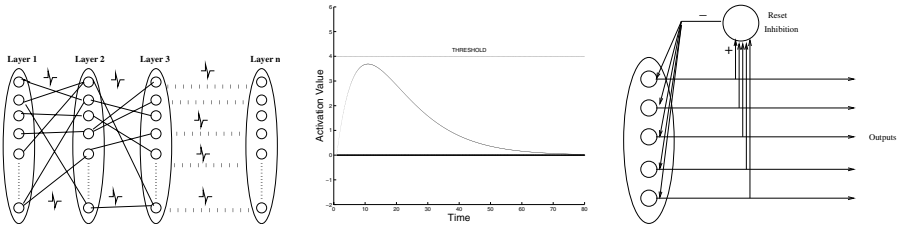

Fig. 1. (a) Architecture of the simulated network. The neurons in each layer are connected to those in the next layer with partial connectivity. The first layer fires a burst of spikes, which is fed to the second layer, whose outputs are fed to the third layer and so on. The temporal widths of the output spike bursts are measured. (b) Plot of a typical RDLIF neural activation with time, when the neuron receives a single input spike at time t=0. (c) Use of feed-back reset inhibition.

purposes we have taken a feed-forward system of layers of neurons emitting bursts of spikes (a synfire chain), as illustrated in figure 1(a). Such a system is simple and sufficient for manifesting the problems we described.

1.1 Model of Spiking Neuron

We use a rate driven leaky integrate and fire (RDLIF) model of spiking activity. As the name indicates, it is similar to the standard leaky integrate and fire (LIF) model [1], the only difference being that incoming spikes increase the driving force or the first derivative of the activation, rather than the activation itself. In this model, the behaviour of a neuron can be described by two variables: the activation a, the quantity which induces the neuron to emit a spike if it exceeds a threshold, and the activation driving force r, which controls the rate of increase of the activation. Both activation and driving force decay with time, the rate of decay being governed by their respective time constants τ_a and τ_r.

The driving force, or rate r_i of the i^{th} neuron increases with incoming spikes and decays with time t, till it reaches a resting value r_0

$$\dot{r_i} = \sum_j w_{ij} x_j - (r_i - r_0)/\tau_r \tag{1}$$

Here $x_j = \sum_n \delta(t - t_n)$ is the sum total of impulse functions of the input spikes emitted from the j^{th} input neuron and w_{ij} is the connection strength.

This driving force r_i drives the activation a_i, which itself decays with time to a resting value a_0.

$$\dot{a_i} = r_i - (a_i - a_0)/\tau_a \tag{2}$$

If the activation of the neuron exceeds its local threshold, it fires a spike and immediately its activation is reset to a refractory level, and driving force to 0 to prevent the activation from increasing.

Figure 1(b) shows the shape of the activation curve of an RDLIF neuron, following a single input spike at time 0. We see that the activation at first

increases due to the increased driving force caused by the incoming spike, but after a time the decay becomes dominant. For an RDLIF neuron to fire, it should get sufficient number of input spikes within a specified time, to enable it to reach the threshold before it 'dies out' because of the decay of the activation and activation rate. There is an inherent time lag between the input spike and the maximum activation reached by the neuron. If the system contains a feed-back loop, such a time lag is necessary, or else at least one input neuron would have to fire a spike at the same time as an output neuron fires a spike, and there would be no temporal separation between the input and output bursts. The standard LIF model cannot achieve this property. This motivated our choice of neural model. Although we have performed simulations on feed-forward networks only, in principle they should work equally well if feed-back loops are present.

2 Simulation of a Multi-layered System

We simulated a feed-forward network of partially connected layers of RDLIF neurons. We fed the first layer a uniformly distributed random set of spikes, and fed them into the second layer, the second layer spikes into the third layer and so on. Delays in the system are solely due to the second order dynamics inherent in the equations in the previous section. We then measured and plotted the temporal separation of the spike burst when passing through different layers.

2.1 Implementation Method

In this section we describe our simulation method for modelling an infinite number of different feed-forward layers of neurons. Our simulation program has a loop, each of whose iterations represents a propagation from layer i to layer (i+1). There is a different weight matrix in each iteration, representing different layers with the same average connectivity. The first layer is given a random set of spike firing times. We have an inner loop to count the time in time-steps in each such iteration, and in each time-step we check if any input or output neuron has fired. Each input spike increases the gain of the connected neuron proportional to the connection, as per our RDLIF model. The firings in the i^{th} layer cause spikes to fire in the $(i+1)^{th}$ layer. We found in our simulations that there was a time above which the gain and activation of all the neurons in a layer would decay and there would not be sufficient stimulus for any neuron to fire. We wait for this amount of time, found through simulation, which is sufficient for all the neurons in a layer to fire output spikes. We argue that this method (using time-steps and waiting for a specified time in each layer before moving on to the next layer) can be considered similar to an event-driven system.

The process of propagating the spike firings of one layer on to the next is repeated for the next iteration after copying the output vector of spike timings to the input layer. In each iteration the input is simply a vector of firing times and we get an output vector of firing times. We measure the temporal dispersion by taking the difference of first and last neuron firing times in that burst.

2.2 Sustaining Stable Activity in a Population of Neurons

Like we mentioned earlier, stability of the spiking activity over many layers is an important issue in modelling systems of spiking neurons. We found that for a given network, there was no threshold such that the system could sustain a stable average level of spiking activity over infinitely many layers. If the threshold is too low, the spiking activity dies out within a few layers. If it is too high, the activity increases with each layer, saturating to an unacceptably high level. The behaviour of the network abruptly switched from dying out of the spiking activity to saturation, as the threshold was increased. One of the ways of getting over this problem is through the use of feed-back reset inhibition. This can enable us to have a system in which a stable activity of firings could be sustained over a number of layers.

2.3 Effect of Feed-Back Reset Inhibition

Feed-back reset inhibition can be implemented by a neuron that is fed the output spikes from a layer, and fires an inhibitory spike once it gets a desired number of input spikes, say N (see figure 1(c)). This strong inhibitory spike resets all of the neurons in the layer, suppressing output activity of the layer. An RDLIF neuron with a threshold equal to the number of allowed spikes is equivalent to such a counter, provided the activation rate time constant τ_r is small and activation time constant τ_a large with respect to the input dispersion. Such systems can be used to implement N-of-M codes (when the inhibitory spike fires after exactly N neurons have fired out of a total of M), such as those used by Furber [4]. In our simulation, we implemented an 11-of-256 code, which could be sustained over infinitely many layers in this way.

2.4 Simulation Results

The parameters in our system are the individual neuron parameters (time constants, threshold) and system parameters (input time spreads and connectivity). Our model had 200 layers with 10% connectivity from layer to layer. The weights

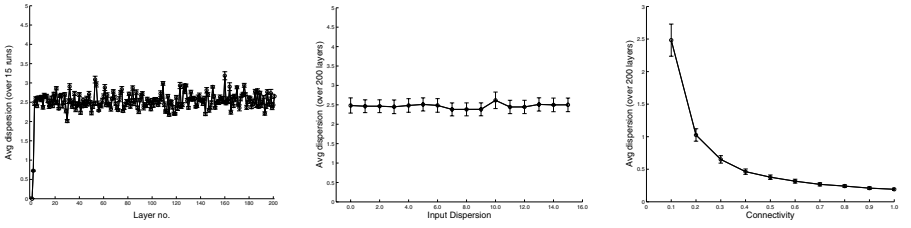

Fig. 2. (a) Plot of the average output dispersion, with initial input dispersion 0, of a burst of spikes passing through the 200-layer network. (b) Plot of the average output dispersion with varying input dispersions. (c) Plot of the variation of average output dispersion with network connectivity. As the connectivity increases, the dispersion decreases.

are real values between 0 and 0.1, chosen from an uniform distribution. The neurons have thresholds of 50 and reset values of -1, time-steps are 10 msec wide, and both time constants (τ_a and τ_r) are 1 sec each.

Figure 2(a) shows the variation of the average output dispersion (over 15 runs) with input dispersion when the initial temporal dispersion was 0. We find that the dispersion quickly tends to settle down into a range, from its input value of 0, and does not disperse much. We repeated this experiment with different values of initial input dispersion and found that we got the same behaviour with different input parameters. Thus, it is quite a stable and robust system.

We then varied other input parameters. Figure 2(b) shows the variation of the average output dispersion with the initial dispersion we gave to the first layer. We see that there is no appreciable change in the average dispersion, regardless of the input dispersion value. Figure 2(c) shows the variation with network connectivity. When the connectivity is low, the neurons have difficulty reaching the threshold and so the average spread is higher, and vice versa.

There are two important system-level time constants in our model, one for the temporal separation within a burst (our waiting time in each iteration from layer i to i+1), the other for separation between bursts (which can model axonal delays). Since we have shown that it is possible to design a system which can propagate a spike burst of desired spiking activity which can maintain coherence when passing through different layers, we can ensure that the inter-burst separation is sufficient (by inserting delays) so that successive waves of spikes do not impinge on each other. On the basis of this, we argue that it is possible to model a reliable system transmitting useful information using spiking neurons.

3 Conclusions and Future Work

In this paper we have studied issues involved in modelling systems of spiking neurons, and have shown that it is possible to propagate a coherent burst of spiking activity of a desired level over many layers. Work needs to be done in implementing real-time systems built with spiking neurons.

References

1. Wolfgang Maass, Christopher M. Bishop. Pulsed Neural Networks. MIT Press, 1999.
2. M. Abeles. Corticonics: Neural circuits of the cerebral cortex. Cambridge University Press, 1991.
3. David C. Sterratt. Spikes, synchrony, sequences and Schistocerca's sense of smell. PhD Thesis, University of Edinburgh, 2002.
4. S. B. Furber, J. M. Cumpstey, W. J. Bainbridge and S. Temple. Sparse distributed memory using N-of-M codes. Neural Networks, 2005, 10.