

A WINNER-TAKE-ALL SPIKING NETWORK WITH SPIKING INPUTS

Matthias Oster and Shih-Chii Liu

Institute of Neuroinformatics, Uni/ETH Zurich
Winterthurer Str. 190, 8057 Zurich, Switzerland
{mao, shih}@ini.phys.ethz.ch

ABSTRACT

Recurrent networks that perform a winner-take-all computation have been studied extensively. Although some of these studies include spiking networks, they consider only analog inputs.

We present results from an analog VLSI implementation of a winner-take-all network that receives spike trains as input. We show how we can configure the connectivity in the network so that the winner will be selected after a pre-determined number of input spikes. To reduce the effect of transistor mismatch on the network operation, we use bursts of input spikes to compensate for this mismatch. The chip with a network of 64 integrate-and-fire neurons can reliably detect the winning neuron, that is, the neuron that receives spikes with the shortest inter-spike interval.

1. INTRODUCTION

Recurrent networks that perform a winner-take-all computation are of great interest because of the computational power they offer. They can be used to model attention and recognition processes in cortex and are thought to be a basic building block of the cortical microcircuit [2]. Descriptions of theoretical models [3] and analog VLSI (aVLSI) implementations of these models [4] can be found in the literature. Although the competition mechanism in these models uses spike signals, they usually consider the external input to the network to be either an analog input current or an analog value that represents the spike rate.

We describe the operation and connectivity of a winner-take-all network that receives input spikes. We consider the case of the hard winner-take-all mode, where only the winning neuron is active and all other neurons are suppressed. In Section 3, we will discuss a scheme for setting the excitatory and inhibitory weights of the network so that the winner which receives input with the shortest inter-spike interval is selected after a pre-determined number of input spikes. The winner can be selected with as few as two input spikes, making the selection process fast [3].

We implement this network on an aVLSI chip with 64 integrate-and-fire neurons and various synapses. The current implementation of this network has large mismatch in the transistors of the input synapses. To reduce the effect of this mismatch on the winner-take-all operation, we use a spike coding mismatch compensation procedure to store the individual synaptic weights. This procedure is described in Section 4. Once the network is calibrated, it reliably detects the winning neuron. We present results for the discrimination capability of the network in Section 5.

2. CONNECTIVITY

A possible connectivity for a winner-take-all network is shown in Figure 1. The network contains N identical spiking neurons. Each neuron receives external input spikes. It inhibits all other neurons and has self-excitation. The weights represent changes in the membrane potential of the neuron during a spike.

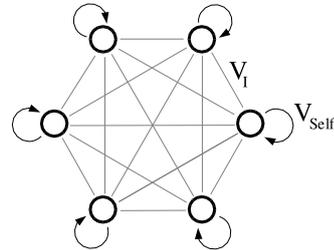


Figure 1: Connectivity of the winner-take-all network. Black circles: neurons (six shown); dark grey: bi-directional inhibitory connections of synaptic weight V_I ; black arrows: self excitation of synaptic weight V_{Self} .

We implement the network using an aVLSI chip which has a network of 64 integrate-and-fire neurons and circuits for communicating spikes on- and off-chip using an asynchronous transmission protocol (AER). When a neuron spikes, the chip outputs the address of this neuron (or spike) onto a common digital bus (see Figure 2). An external spike interface module (consisting of a custom computer board that can be programmed through the PCI bus) receives the incoming spikes from the chip, and retransmits spikes back to the chip using information stored in a routing table. This module can also monitor spike trains from the chip and send spikes from a stored list. Through this module and the AER protocol, we implement the connectivity needed for the winner-take-all network in Figure 1. We do not make use of any on-chip connections in this work. All components have been used and described in previous work (e.g. [5, 1]).

3. NETWORK CONNECTIVITY CONSTRAINTS FOR A WINNER-TAKE-ALL MODE

We first discuss the conditions under which a network operating in a hard winner-take-all mode will select the winning neuron after receiving a pre-determined number of input spikes. The winning neuron is the one receiving the input with the smallest inter-spike interval. For this analysis, we consider only the case where the neurons receive regular spiking inputs. Assume that all neurons

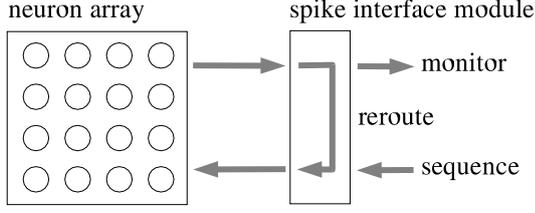


Figure 2: The connections are implemented by transmitting spikes over a common bus (grey arrows). Spikes from aVLSI neurons in the network are recorded by the digital interface and can be monitored and rerouted to any neuron in the array. Additionally, externally generated spike trains can be transmitted to the array.

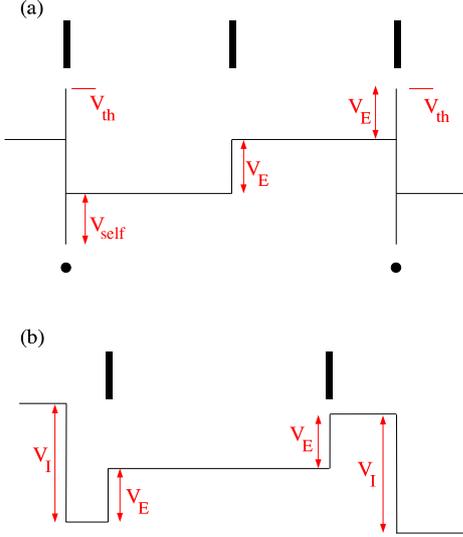


Figure 3: Membrane potential of the winning neuron k (a) and another neuron in the array (b). Black bars show the times of input spikes. Trace shows the changes in the membrane potential caused by the various synaptic weights. Black dots show the times of output spikes of neuron k .

$i \in 1 \dots N$ receive excitatory input spike trains of constant frequency r_i . Neuron k receives the highest input frequency ($r_k > r_i \forall i \neq k$). Every neuron will inhibit all other neurons when it produces a spike. Each excitatory or inhibitory input spike causes a fixed discontinuous jump V_E or V_I in the membrane potential V_i of the neurons. The time course of the synaptic currents and the transmission delay of the spikes are neglected. A neuron spikes when $V_i \geq V_{th}$, is reset to $V_i = 0$, and receives a self-excitation signal right after its own spike, resulting in $V_i = V_{self}$. All potentials satisfy $0 \leq V_i \leq V_{th}$, in particular, an inhibitory spike cannot drive the membrane potential below 0. Schematic traces for neuron k and another neuron are shown in Figure 3. The following constraints have to be fulfilled if the network is to select a winner after a pre-determined number n of input spikes:

(a) Neuron k (the winning neuron) spikes only after receiving $n_k = n$ input spikes that cause its membrane potential to exceed threshold. After every spike, the neuron is reset to V_{self} :

$$V_{self} + n_k V_E \geq V_{th} \quad (1)$$

(b) As soon as neuron k spikes once, no other neuron $i \neq k$ can spike because it receives an inhibitory spike from neuron k . Another neuron can receive up to n spikes even if its input spike frequency is lower than that of neuron k because the neuron is reset to V_{self} after a spike, as illustrated in Figure 3. The resulting membrane voltage has to be smaller than before:

$$n_i \cdot V_E \leq n_k \cdot V_E \leq V_I \quad (2)$$

(c) If a neuron j other than neuron k spikes in the beginning, there will be some time in the future when neuron k spikes and becomes the winning neuron. From then on, the conditions (a) and (b) hold, so a neuron $j \neq k$ can at most have a few transient spikes. Let us assume that neurons j and k spike with almost the same frequency (but $r_k > r_j$). For the inter-spike intervals $\Delta_i = 1/r_i$ this means $\Delta_j > \Delta_k$. Since the spike trains are not synchronized, an input spike to neuron k has a changing phase offset ϕ from an input spike of neuron j . At every output spike of neuron j , this phase decreases by $\Delta\phi = n_k(\Delta_j - \Delta_k)$ until $\phi < n_k(\Delta_j - \Delta_k)$. When this happens, neuron k receives $(n_k + 1)$ input spikes before neuron j spikes again and crosses threshold:

$$(n_k + 1) \cdot V_E \geq V_{th} \quad (3)$$

We choose $V_{self} = V_E$ and $V_I = V_{th}$ to fulfill the inequalities (1)-(3) and we adjust V_E to achieve the desired n_k .

Case (c) happens only under certain initial conditions, for example when $V_k \ll V_j$ or when neuron j initially received a spike train of higher frequency than neuron k . If we incorporate a decay of the membrane potential in the model, we can assume that all membrane potentials are discharged ($V_i = 0$) at the onset of a stimulus. In that case, neuron k will have the first output spike.

4. MISMATCH COMPENSATION

The neurons in our present aVLSI implementation of the winner-take-all network show significant mismatch in the transistors of the input synapses. This causes a large variation of the excitatory synaptic weight, even though the weight is set by a global bias. We will describe a procedure to compensate for the mismatch in the synaptic weights and in the reset voltage of the neurons.

We characterize the mismatch on a functional level by stimulating the neurons with a spike train of a constant frequency of 100Hz. We configured the number of input spikes needed for a neuron to reach threshold to be 9. The raw spikes are postprocessed to check for abnormal spike statistics. The vector of output spike rates \mathbf{r} has a mean of $\mu_r = 11.28\text{Hz}$. On average a neuron needs $\langle n_i \rangle = 8.86$ input spikes to reach threshold. With $V_{th} = 1.5\text{V}$, $\langle V_E \rangle = V_{th}/\langle n_i \rangle = 169\text{mV}$. The standard variation of the output rates is $\sigma_r = 11.69\text{Hz}$ and the coefficient of variation $\text{CV} = \sigma_r/\mu_r = 103.7\%$. With this variation, the network always select the neuron with the highest effective excitatory weight due to mismatch as the winner ($\max(\mathbf{r}) = 42.5\text{Hz}$). To select a different neuron, we have to increase its input frequency to $f \cdot 100\text{Hz}$ with $\langle f \rangle = \max(\mathbf{r})/\mu_r \approx 3.77$. Obviously, the computational capability of the uncalibrated network is quite low. For other settings of the neurons, the mismatch is even higher.

To compensate for the mismatch in the excitatory synapses, we transmit a burst of spikes for each input spike to a neuron. This compensation is done by repeating the same target address in the routing table of the external spike interface module. We checked

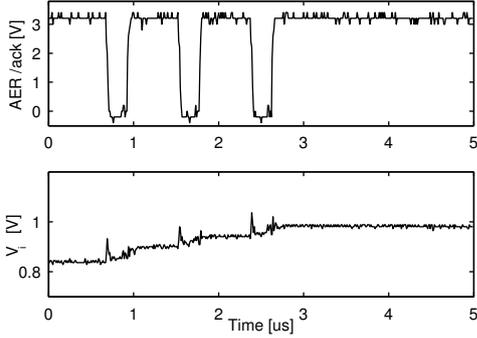


Figure 4: A burst of spikes sums on the membrane potential, shown for 3 spikes. Top: one of the AER communication signals indicating that the chip has received a event from the interface module (active low acknowledge signal of the chip); bottom: membrane potential of the neuron V_i .

that multiple spikes sum linearly on the membrane potential (Figure 4).

We vary the number of spikes in the burst m_i for each neuron until the neuron spikes with the desired output frequency. The output frequency can only be an integer division of the input frequency, because the neuron needs an integral number of spikes to reach threshold. The global bias that sets the excitatory weight has to be set to a lower value than in the uncalibrated case. We used a simple algorithm to automatically adjust m_i : If the output frequency of the neuron is below the target frequency, m_i is incremented by one, otherwise it is decremented.

In the same way we compensate for the mismatch in the reset mechanism of the neurons. Because of this mismatch, some neurons are not completely discharged while others show a finite refractory period (Figure 5). We use the internal reset mechanism of

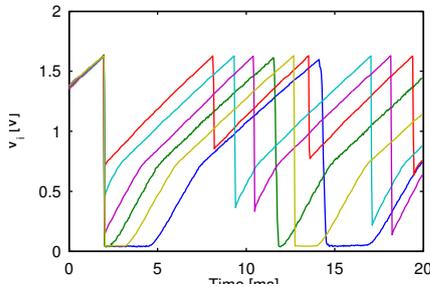


Figure 5: Mismatch in the reset voltage of the neurons, shown for some example traces. Some neurons are not completely discharged while others show a refractory period.

the neurons to discharge the membrane potential by a small voltage V_c (determined by the hysteresis capacitance, see [5] for a discussion of the neuron circuit). We then send a burst of p_i inhibitory spikes to a neuron once we receive the output spike of the neuron. The number of spikes p_i is adjusted with the calibration algorithm described earlier so that all neurons are reset to the same voltage (Figure 6).

Figure 7 shows the resulting output firing rates. The mean frequency $\mu_r = 11.60\text{Hz}$ is approximately equal to the mean in the uncalibrated case, but the standard deviation is now only $\sigma_r = 1.006\text{Hz}$, resulting in a coefficient of variation of 8.6%.

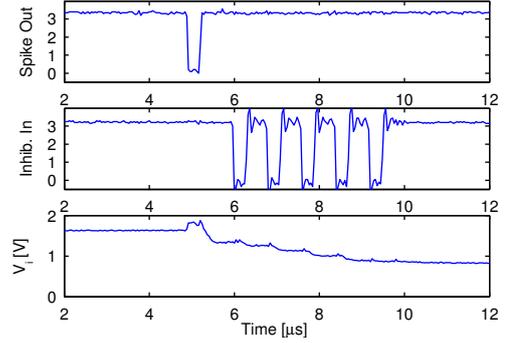


Figure 6: Compensation procedure of the reset voltage mismatch. An output spike of the neuron (top) gets routed back as a burst of 5 spikes to the inhibitory synapse (middle), discharging the membrane (bottom) to a defined reset voltage.

The weights of the inhibitory connections are calculated from the reset voltage compensation, which decreases the membrane potential by $V_{th} - V_c$. We choose the inhibitory weight such that the neuron is discharged by $V_i = V_{th}$.

When the full connectivity of the network is enabled, an output spike of a neuron i is translated into a large block of spikes: Inhibitory spikes to reset the neuron, inhibitory spikes to all other neurons and self-excitatory spikes. The total number of spikes is approximately

$$s_i = p_i + \sum_{\substack{k=1 \\ k \neq i}}^N q_k + m_i \approx 750.$$

Because only the winner emits spikes and the communication is fast, the spike interface module can transmit this block of spikes without saturating the communication bus.

5. RESULTS

We characterized the discrimination capability of the winner-take-all network by stimulating all neurons except one with a spike train of a constant frequency of 100Hz. This single neuron received an increased frequency of $f \cdot 100\text{Hz}$ (Figure 8). For each neuron we measured the minimum value of f at which the network selects

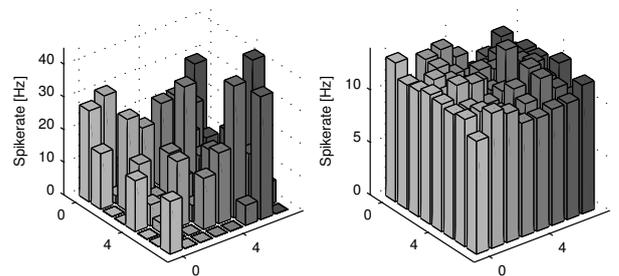


Figure 7: Output firing rates of the 8x8 neuron array, without (left) and with (right) mismatch compensation. Each neuron is stimulated with a constant input frequency of 100Hz. X/Y axis: neuron address; bar height: spike rate.

this neuron as the winner and all other neurons are completely suppressed. The network can discriminate an input of higher spike rate if this frequency is at least by $(f - 1) \cdot 100\text{Hz}$ higher than the other inputs.

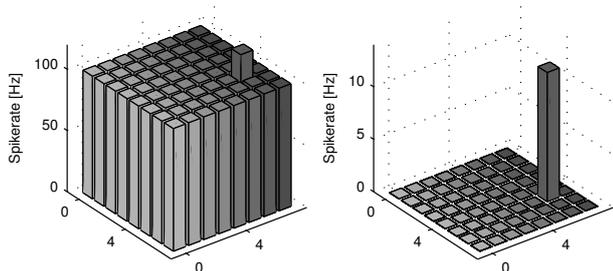


Figure 8: Input stimulus to the array (left) and winner-take-all output (right). Each neuron is stimulated with a constant input frequency of 100Hz, except one that receives an increased frequency of $f \cdot 100\text{Hz}$. This neuron suppresses all other neurons. X/Y axis: neuron number; bar height: spike rate.

The histogram of the minimum factors f for all neurons is shown in Figure 9. On average, the network is sensitive to a difference in the input frequency of 10%. In the worst case, for the neuron with the smallest excitatory weight, the frequency difference is 20%. Since we only use the timing information of the spike trains, the results can be extended to other input frequencies.

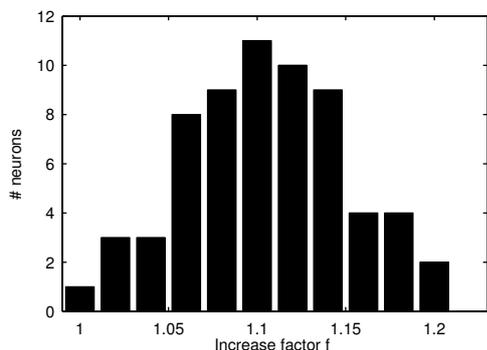


Figure 9: Discrimination capability of the winner-take-all network. X-axis: factor f to which the input frequency of a neuron has to be increased, compared to the input rate of the other neurons, to select that neuron as the winner. Y-axis: histogram of all 64 neurons.

In this experiment, the parameters of the neurons were configured to reach threshold after about $n_k = 9$ input spikes to demonstrate the capability of the mismatch compensation, however, any value of n_k is possible. For large n_k , the input represents a rate-code. For $n_k = 1$, the winner codes the latency of the input spike trains. In this case, the delay of a spike after a global reset determines the strength of the signal. The winner is selected with the first input spike to the network. If all neurons are discharged at the onset of the stimulus, the network does not require the global reset. In general, the computation is finished at a time $n_k \cdot \Delta_k$ after stimulus onset.

6. DISCUSSION

We analysed the performance and behavior of a winner-take-all network that receives input spike trains. The neuron that receives spikes with the shortest inter-spike-interval is selected as the winner after a pre-determined number of input spikes. Assuming a non-leaky integrate-and-fire model neuron with constant synaptic weights, we derived constraints for the strength of the inhibitory connections and the self-excitatory connection of the neuron. A large inhibitory synaptic weight is in agreement with previous analysis for analog inputs [3].

The assumption of a constant input frequency is too restricted. We are currently extending our analysis to spike trains with Poisson distribution. In that analysis we will also use a leaky integrate-and-fire neuron model and conductance-based synapses.

We described a spike-coding calibration procedure to compensate for the mismatch in our chip. A burst of spikes is sent to the neuron for each input spike from the spike interface module. The number of spikes in a burst is equivalent to the weight of the synapse. This procedure provides a simple solution to the long-discussed storage of synaptic weights for aVLSI neurons compared to analog techniques like floating-gate transistors.

We calibrated the excitatory synaptic weights, which could be done simultaneously for all neurons at once using the spike interface. For the calibration of the reset voltage, we had to measure the analog membrane potential for each individual neuron. The next chip revision will include an adjustable reset voltage that is not sensitive to mismatch. The complete connectivity of the network was implemented through the asynchronous spike interface module. The calibration procedure results in a large number of spikes (here: ~ 750) that have to be routed for each output spike. For a network of 64 neurons, this load does not saturate the communication bus, but introduces jitter to the other spike trains since the bus is blocked during this time. With an adjustable reset voltage in the next chip version, we can implement an inhibitory connection with only one spike per target neuron, reducing the number of spikes to 64. The results show that the aVLSI network can perform a reliable winner-take-all computation.

7. REFERENCES

- [1] K. A. Boahen. Point-to-point connectivity between neuromorphic chips using address-events. *IEEE Transactions on Circuits & Systems II*, 47(5):416–434, 2000.
- [2] R. Douglas and K. Martin. Cortical microcircuits. *Annual Review of Neuroscience*, 27(1f), 2004.
- [3] D. Z. Jin and H. S. Seung. Fast computation with spikes in a recurrent neural network. *Physical Review E*, 65:051922, May 2002.
- [4] J. Lazzaro, S. Ryckebusch, M. A. Mahowald, and C. A. Mead. Winner-take-all networks of $O(n)$ complexity. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 1, pages 703–711. Morgan Kaufmann, San Mateo, CA, 1989.
- [5] S.-C. Liu, J. Kramer, G. Indiveri, T. Delbrück, T. Burg, and R. Douglas. Orientation-selective aVLSI spiking neurons. *Neural Networks: Special Issue on Spiking Neurons in Neuroscience and Technology*, 14(6/7):629–643, 2001.