IMPROVEMENTS IN SPINNAKER SIMULATOR

SpiNNaker – a spiking neural network simulator developed by APT group – The University of Manchester

SERGIO DAVIES
11/03/2010
Research Areas

• Currently working on:

  • **Synaptic plasticity:** specifically a new version of the STDP algorithm which aims to simplify the original version;

  • **New simulator framework:** Porting Scott's simulator code onto the test chip;

  • **Graphical User Interface (spinnGUI):** condensing all the tools in a nice front-end.
Synaptic plasticity

- Modification of synapse parameters (i.e.: weight);
- Deletion of synapses with a very low weight;
- Creation of new synapses for each removed synapse;

The last two events are generally referred to as “synaptic rewiring”.

![Neuron Image]
Synaptic weight modification (1/3)

- Weight modification dependent on the sequence of spikes:
  - Causality in the input – output causes the weight of the input synapse to increase (+);
  
  ![Diagram](image1)

  ![Diagram](image2)

  - Anti-causality in the input – output causes the weight of the input synapse to decrease (-);
The modification of the synaptic weight is described by an exponential law:
“Spike-Timing-Dependent Plasticity” (STDP) to evaluate the weight of the synapse:

- The original algorithm computes the time difference between spikes for all the possible combination of input and output spikes (in a time window).

This algorithm is very complex and needs future information to compute the weight in a specific moment.
Deferred event-driven (DED) model

- To avoid the need of future parameters the deferred event-driven model has been implemented. The execution is triggered on the arrival of a pre-synaptic spike.

- When some spikes are pushed out of the STDP time window, the STDP algorithm is triggered.
Disadvantages of the standard STDP: computational power and memory.

The first simplification takes into account only the nearest sequence of input – output spikes:

Similar to the DED STDP model. Less computations, but similar amount of memory.
To avoid the need of memory, a new model for the STDP algorithm has been proposed.
Forecast of the next outgoing spike:

- Running average historical spiking rate
- The future spike time-stamp of a neuron is computed according to an estimated firing rate.
The estimated firing rate is updated at every outgoing spike according to the previous estimated firing rate and the time between the last two spikes.

\[ \text{FR}(n) = \frac{1}{2} \text{FR}(n-1) + \frac{1}{2} \text{IST}(n) \]

Where:

- \( \text{FR}(n) \): firing rate estimated after \( n \) outgoing spike
- \( \text{IST}(n) \): Inter Spike Time between spike \( n-1 \) and spike \( n \)

Sudden variation of spiking rate doesn't have an immediate effect.
Before the simulator expired, there was the possibility to run only one simulation:

- The algorithm has been simulated and compared with the original STDP with a random network of 50 neurons with an input to 10 of these neurons.

- The input is characterized by fast and strong spikes.
Running average STDP
Results (1/2)

Raster plots of the simulations:

Standard STDP

Running average STDP

Behaviour seems similar.
Running average STDP Results (2/2)

Numerical results of the simulation.

Values are ratio between the variation of the synaptic weights in both models.

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>0.71</td>
<td>0</td>
<td>0</td>
<td>1.07</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.64</td>
<td>0.57</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>26</td>
<td>0</td>
<td>0.62</td>
<td>0</td>
<td>0</td>
<td>0.43</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.98</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0.67</td>
</tr>
<tr>
<td>27</td>
<td>0</td>
<td>0.93</td>
<td>0</td>
<td>0</td>
<td>0.83</td>
<td>0</td>
<td>0</td>
<td>0.8</td>
<td>0</td>
<td>0</td>
<td>0.97</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>28</td>
<td>0</td>
<td>0.72</td>
<td>0</td>
<td>1.08</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.43</td>
<td>0</td>
<td>0</td>
<td>0.65</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>29</td>
<td>0</td>
<td>0.54</td>
<td>0</td>
<td>0.67</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.9</td>
<td>0.44</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0.76</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.13</td>
<td>0.95</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>31</td>
<td>0.68</td>
<td>0</td>
<td>0</td>
<td>0.6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.03</td>
<td>0</td>
<td>0</td>
<td>1.1</td>
<td>0</td>
</tr>
<tr>
<td>32</td>
<td>0.75</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.76</td>
<td>0</td>
<td>0</td>
<td>0.76</td>
<td>0.77</td>
</tr>
<tr>
<td>33</td>
<td>0.51</td>
<td>0</td>
<td>0</td>
<td>0.68</td>
<td>0</td>
<td>0.54</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.47</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>34</td>
<td>0</td>
<td>0</td>
<td>0.48</td>
<td>0</td>
<td>0.75</td>
<td>0</td>
<td>0.68</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1.12</td>
<td>0</td>
</tr>
<tr>
<td>35</td>
<td>0</td>
<td>0.46</td>
<td>0</td>
<td>0.79</td>
<td>0</td>
<td>0.72</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.58</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>36</td>
<td>0.76</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.97</td>
<td>0</td>
<td>1.24</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.8</td>
</tr>
<tr>
<td>37</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.52</td>
<td>0.46</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.64</td>
<td>0</td>
</tr>
<tr>
<td>38</td>
<td>0.57</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.57</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1.07</td>
<td>0</td>
<td>0.71</td>
<td>0</td>
</tr>
<tr>
<td>39</td>
<td>0</td>
<td>0</td>
<td>1.11</td>
<td>0</td>
<td>0</td>
<td>0.67</td>
<td>1.16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.12</td>
<td>0</td>
</tr>
</tbody>
</table>
Running average STDP
Future work

- Raster plots indicate a comparable behaviour.
- Synapse weights table shows the opposite.

WHAT TO BELIEVE???

- We are more interested in the behaviour of the network
- More simulations and study of the parameter is necessary
New simulator

- SoC Designer has expired. Simulations running now on the test chip.

- Some adaptations are needed to load data into and to retrieve data from the simulator.
SpinnGUI

A new front-end for SpiNNaker simulator.

- Expandability for new future tools and modules
- Portability to various operating systems
- Controls the whole tool-chain: from the compilation to the retrieval of simulation results.
SpinnGUI explained

Expandability:

- Front-end is modular
- Modules of the simulator are described in XML.

Portability:

- The Qt framework supports multiple OS.
- The Xerces-C library as well.
SpinnGUI – State of development

XSD Schema
XML Description
spinnGUI
SpinnGUI – Future work

- A graphical interface to describe the neural network
- Tools to configure the simulation on the chip
- Output analyser tools
- ...
Thank you!!!