

A forecast-based biologically-plausible STDP learning rule

Sergio Davies, Alexander Rast, Francesco Galluppi and Steve Furber

Abstract— Spike Timing Dependent Plasticity (STDP) is a well known paradigm for learning in neural networks. In this paper we propose a new approach to this problem based on the standard STDP algorithm, with modifications and approximations, that relate the membrane potential with the LTP (Long Term Potentiation) part of the basic STDP rule. On the other side we use the standard STDP rule for the LTD (Long Term Depression) part of the algorithm. We show that on the basis of the membrane potential [5] it is possible to make a statistical prediction of the time needed by the neuron to reach the threshold, and therefore the LTP part of the STDP algorithm can be triggered when the neuron receives a spike. We present results that show the efficacy of this algorithm using one or more input patterns repeated over the whole time of the simulation. Through the approximations we suggest in this paper we introduce a learning rule that is easy to implement in simulators and reduces the execution time if compared with the standard STDP rule.

I. INTRODUCTION

LEARNING in neural networks was postulated for the first time by D. O. Hebb in 1949 [9]. On the basis of this general idea, various forms of learning have been tested, with the goal of determining a biologically realistic learning model. In 1996 [17] biological tests observed that synaptic weight modification occurs when pre- and post-synaptic spikes coincide at low frequencies. In 1998 [4] tests were run on cultures of hippocampal cells: the outcome of this research showed a relation between the relative timings of pre-synaptic spikes and post-synaptic spikes with potentiation or depression of the corresponding synaptic weight. This correlation is, however, limited by an efficacy time window: if the time difference between a pre- and a post-synaptic spike is greater than this effective time window (also known as the STDP time window), modifications in synaptic strength are zero. The function that summarizes this behaviour is shown in fig.1. The numerical values of this function (called the STDP curve) vary depending on the system and on the neuron type [1] [3] [21].

While this rule states how synapses are influenced by couples of pre- and post-synaptic spikes, it is not yet clear how multiple sets of pre- and post-synaptic spikes influence synaptic weights. The standard STDP rule deals with this condition by applying the algorithm for every pair of pre- and post-synaptic spikes in the relevant time window (see fig.2(a)) [21]. This algorithm, as described by the research done in 1998 [4], has been used in several simulation tests (e.g.: [8]) which have proven to replicate the biology. Burkitt et al. [6] found a mathematical and statistical description of the standard STDP rule.

All the authors are with the School of Computer Science, The University of Manchester, Manchester, U.K. (email: {daviess, rasta, galluppi, sfurber}@cs.man.ac.uk).

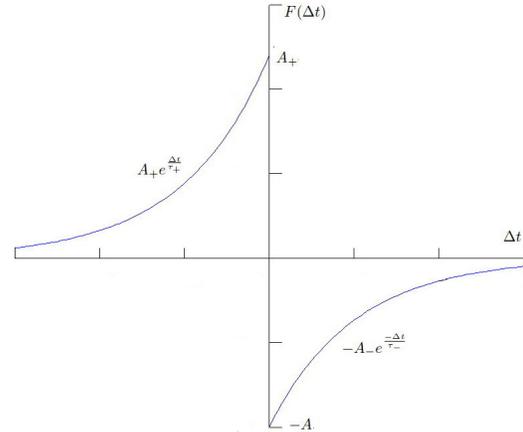


Fig. 1. STDP curve. The X axis represents the time between pre and post synaptic spikes ($\Delta t = t_{pre} - t_{post}$). The Y axis represents the synaptic weight modification.

An highlighted property of the STDP rule [8] is that the neuron which identifies an incoming repeated pattern moves the identification (fires) earlier and earlier in the pattern until it reaches the earliest group of spikes that introduces the pattern.

In [2], Banerjee states that while there are differences between an abstract model and a biological model of a neural network, the results demonstrate that the behaviour of an abstract system is generally consistent with what is observed in biological experiments. As an example of this, Indiveri et al. [10] use an abstract model which differs from the standard STDP rule.

There have been implementations of variants of the STDP rule and some of them take into account trains of spikes differently from the standard STDP model. Some implementations take into account series of three and four spikes [23] [20]. However these models are more complex than the standard STDP rule, because of the need to track different patterns across the sequence of pre- and post-synaptic spikes. The algorithm which implements the simplest learning rule is the “spike-pair” (also known as “nearest neighbour” or “nearest spike”) rule: only the nearest pairs of pre- and post-synaptic spikes interact to modify the synaptic weight (fig.2(b)). This particular rule is known to reproduce at least one biologically-relevant feature [14]: post-synaptic spikes propagate back into the dendritic spines, resetting them. For this reason the latest post-synaptic spike erases the effect of earlier pre-synaptic spikes. This rule has been used for various experiments [18] [19] and has proved to be reliable from the phenomenological point of view.

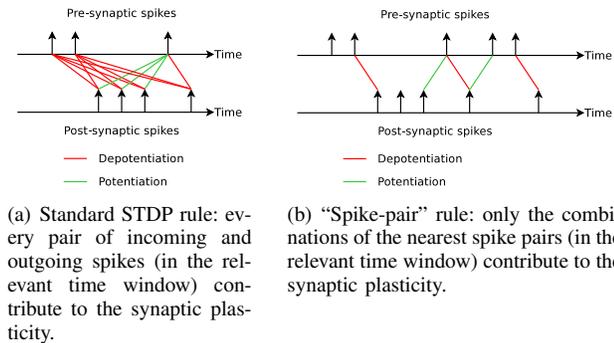


Fig. 2. STDP rules

Other variants of the STDP algorithm link the synaptic weight modifications to the (pre- or post-synaptic) neuron membrane potential. In this class of algorithms, generally, the synaptic weight modification is influenced by the membrane potential (or a low-pass filtered version of it) [7]. In analogue hardware implementations of neural network simulators, similar techniques are used in order to evaluate synaptic plasticity [5].

In our implementation, the standard STDP algorithm described has been modified using the "Deferred Event" model [16] [15] to allow triggering the STDP algorithm only when a pre-synaptic spike is received by the post-synaptic neuron. A statistical description of this algorithm is beyond the scope of this paper.

II. PROPOSED ALGORITHM

In this paper we present a learning algorithm for neural networks which is able to update the synaptic weight as soon as a pre-synaptic spike is received. To do this we trigger LTD using the standard STDP algorithm and we trigger LTP in relation to the membrane potential value. For this purpose we determine, with a statistical method, the relation between the membrane potential and the estimated time of the next neuron spike. The basic idea is that the higher the membrane potential, the sooner the neuron is supposed to fire. This estimation is called throughout this paper "Time-To-Spike" (TTS).

With this approximation we compute the LTP part of the STDP algorithm pairing the incoming pre-synaptic spike with an estimation of the first post-synaptic spike using a statistical rule. Therefore this part of the algorithm is similar to the "spike-pair" rule described before. For the LTD part of the algorithm, we use the standard STDP rule which pairs the post-synaptic spikes with all the pre-synaptic spikes received after the post-synaptic spike.

The model we present in this paper is able to compute both synaptic potentiation and depression (LTP and LTD) as soon as one pre-synaptic spike is received by a post-synaptic neuron, using only the event triggered by the receipt of a pre-synaptic spike.

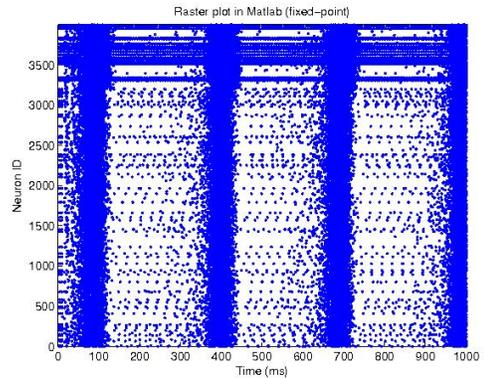


Fig. 3. Raster plot of a randomly connected network with constant input to random neurons.

III. MODEL DESCRIPTION

The approach proposed in this paper is based on a detection of the membrane potential of the neuron: the higher the membrane potential value, the sooner the neuron is supposed to fire. For the purpose of this paper we focus on the Izhikevich neuron model which is described by the two ordinary differential equations [11]:

$$\begin{cases} \dot{v} = 0.04v^2 + 5v + 140 - u - I \\ \dot{u} = a(b \cdot v - u) \end{cases} \quad (1)$$

$$\text{if } v \geq 30 \text{ mV then } v = c, u = u + d \quad (2)$$

This function is estimated statistically using a network composed of 4,000 Izhikevich neurons randomly interconnected. Two types of Izhikevich neurons are used in this network:

- 1) Tonic spiking neurons: ($a = 0.02$; $b = 0.2$; $c = -65$; $d = 8$);
- 2) Fast spiking neurons: ($a = 0.1$; $b = 0.2$; $c = -65$; $d = 2$);

There are 3,200 tonic spiking neurons each connected with excitatory synapses to 25 other neurons randomly chosen from all the other neurons in the population. The synaptic strength for excitatory connections is $10nA/\mu F$.

There are 800 fast spiking neurons each connected with inhibitory synapses to 25 other neurons randomly chosen from all the tonic spiking neurons in the population. The synaptic strength for inhibitory connections is $-5nA/\mu F$.

The network is excited by the injection of a constant current of $20nA/\mu F$ into 60 randomly chosen tonic spiking neurons and 20 randomly chosen inhibitory neurons. No synaptic plasticity is enabled for this test.

The simulator used to collect the results presented in this paper is a revised version of the Izhikevich neural network Matlab simulator [13]. The network is then simulated for 1,000 steps of $1msec$ each (for an overall simulation time of $1s$). Fig.3 shows the raster plot generated by the simulation.

At the end of the simulation all the values of the membrane potential stored for each millisecond of the simulation are

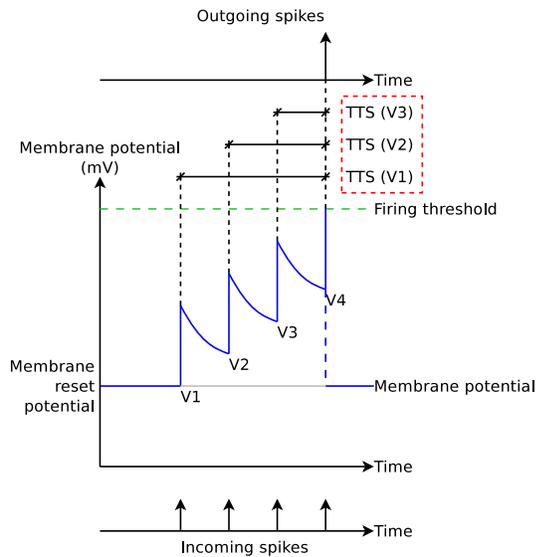


Fig. 4. Example of computation of the Time-To-Spike (TTS) of a neuron.

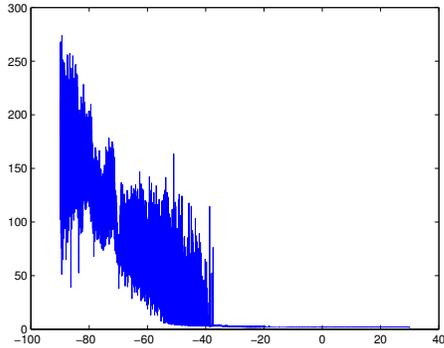


Fig. 5. Function that relates the membrane potential in mV (on the X axis) and the estimated time to spike in $msec$ (on the Y axis).

post-processed: for every spike received by a neuron, a couple (membrane potential, time-to-spike) is computed (see fig.4). Then those couples are sorted and grouped using the membrane potential value as the key. Finally we estimated a mean time-to-spike across all the elements of each of these groups. The result of this statistical approach is presented in fig.5.

In this graph the X axis represents the membrane potential value. The Y axis represents the mean time-to-spike estimated before. It is possible to note that the graph is divided in two sections: for values between (about) $-40mV$ and $30mV$ the estimated time to spike is linear in the interval between $3msec$ and $0msec$. For values smaller than $-40mV$ the graph is very noisy.

The difference between the two parts of the graph can be justified by looking at the phase plane of the Izhikevich neuron model (fig.6). In this graph the X axis represents the membrane potential (variable V of the Izhikevich neuron model), while the Y axis represents the variable U . If the

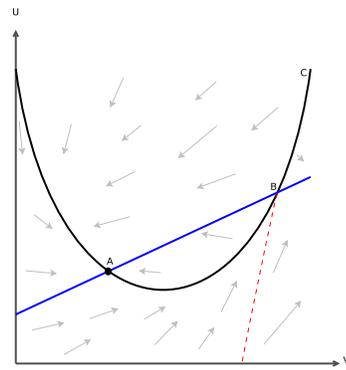


Fig. 6. Izhikevich neuron state phase plane.

neuron state is on the left of the threshold (the red dashed line and the BC part of the parabola) then in the absence of input it will move towards the equilibrium point A. If the neuron state is on the right of the threshold, it moves autonomously in the absence of input towards the spiking condition ($V_{membrane} \geq 30mV$). This threshold corresponds to the separation between the two sections of the graph in fig.5.

To remove some noise from the graph that relates the membrane potential with the estimated time to spike, we used a sliding window filter with size going from 64 samples to 1024 samples (left column of fig.7). Running this test multiple times, it is possible to note that the right part of the graph does not change significantly, while the left part varies more and more as the membrane potential moves left from the threshold point. Therefore we limit our analysis to a window of 32 milliseconds of forecast (right column of fig.7).

The function that results from this analysis is composed of two segments: the first passes through the points $(30mV; 0msec)$ and $(-40mV; 3msec)$. The second passes through the point $(-40mV; 3msec)$. We use the second edge of the segment as a parameter for the forecast (parameter “L”). The relation function between the membrane potential and the estimated time to spike can be then summarized in fig.8.

IV. NETWORK DESCRIPTION AND SIMULATION PARAMETERS

The simulations we ran have a neuron input layer composed of 200 neurons connected to an output layer which can be formed of one, two or four output neurons. All the neurons used in these tests are tonic spiking Izhikevich neurons. Multiple output neurons inhibit each other for 5 milliseconds with a strength that linearly increases from $-10nA/\mu F$ to $-6nA/\mu F$.

We used “voltage-jump” synapses with plasticity enabled. Here we will introduce two sets of experiments: the first uses the standard STDP algorithm, while the second uses the forecast-based learning algorithm. The STDP curve (fig.1) is defined by the function:

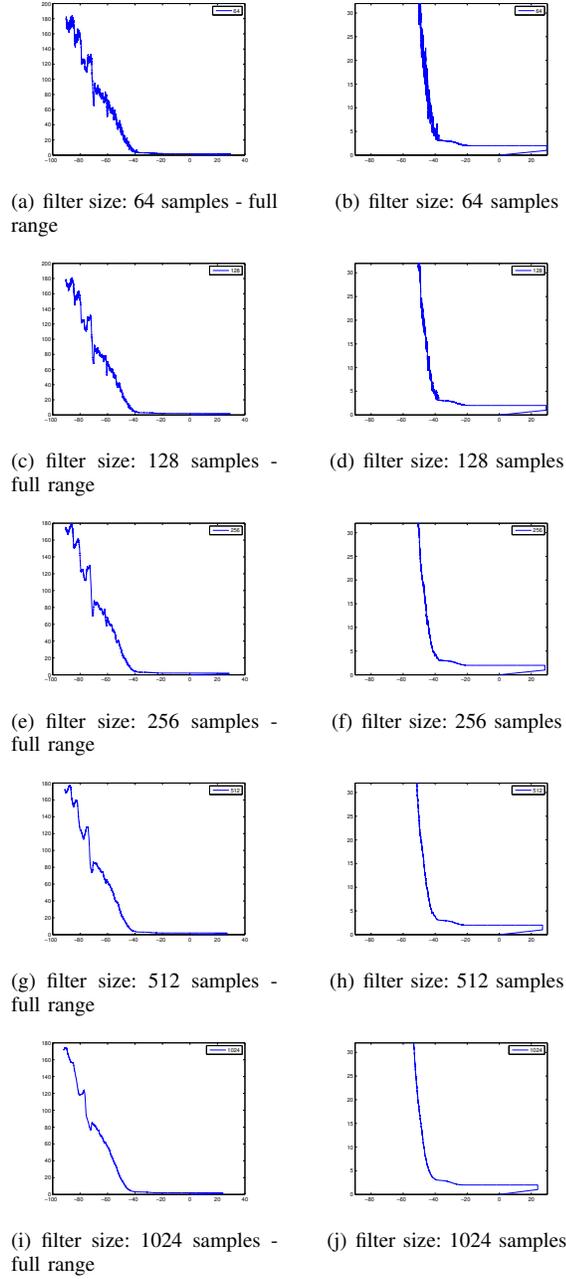


Fig. 7. Filtered version of the function in fig.5 that relates the membrane potential (in mV), on the X axis, and the estimated time to spike (in $msec$), on the Y axis. The filter applied is a sliding window with varying size. On the left column the full range filtered graph. On the right column a detailed part including values up to $32msec$ of the estimated time to spike.

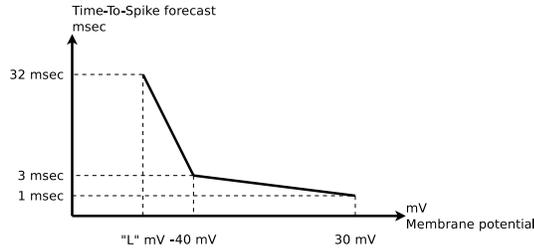


Fig. 8. Relation between membrane potential and time to spike of the neuron.

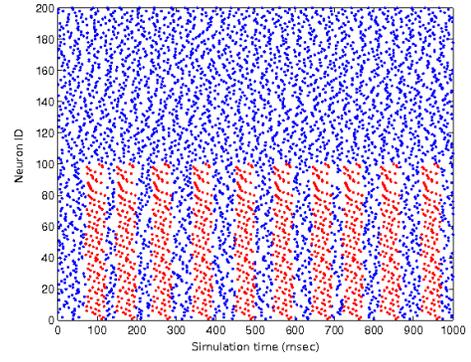


Fig. 9. Example of raster plot of the input pattern. In red the input pattern highlighted.

$$F(\Delta t) = \begin{cases} A_+ e^{\frac{\Delta t}{\tau_+}} & \Delta t < 0 \\ -A_- e^{-\frac{\Delta t}{\tau_-}} & \Delta t \geq 0 \end{cases} \quad (3)$$

The parameters we use are [16] [15] [22]: $\tau_+ = \tau_- = 20msec$, $A_+ = 0.1$ and $A_- = 0.12$.

The input layer is divided into two equally sized population: the first population stimulates the output layer with random spikes generated by a Poisson process with a mean firing rate of 50 Hz. The second input population stimulates the output layer using a 50 millisecond pattern interleaved in the random noise with an inter-pattern time which is itself a Poisson random process with a mean modulation frequency of 11 Hz. The mean firing rate of the pattern is the same as the noise (50 Hz), therefore the pattern is not clearly identifiable merely by looking at the input. In our implementation, patterns cannot be presented across the boundaries of a second of simulation. This reduces the number of mean occurrences of the pattern to 10 per second. An example of input is presented in fig.9. In the case where we input more than one pattern, these are interleaved between each other (e.g. 1-2-1-2-1-...).

V. SIMULATION RESULTS

To test the software we first use the standard STDP algorithm, then we compare this with the forecast plasticity model. The initial conditions of the networks used for these tests (unless otherwise specified) are: starting synaptic weights are random in the interval $[0; 2]$ and maximum synaptic weight is set to 2. The networks used are described in fig.10. The scatter plots we present in this section show when the output neuron fires with respect to the presentation of the input pattern. The X axis represents the simulation time (in $msec$). The Y axis represents the time from the beginning of the pattern injection to the output spike (in $msec$). Every dot represents an action potential emitted by the output neuron. The red line indicates when the pattern ends.

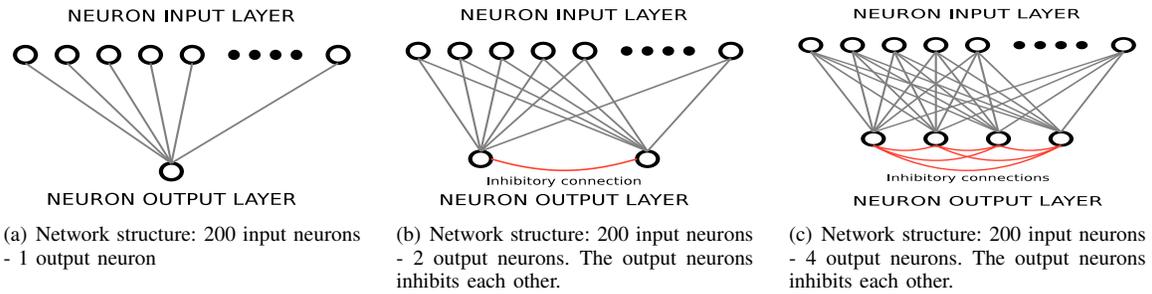


Fig. 10. Structure of the neural networks used in the tests: 200 input neurons and 1, 2 or 4 output neurons.

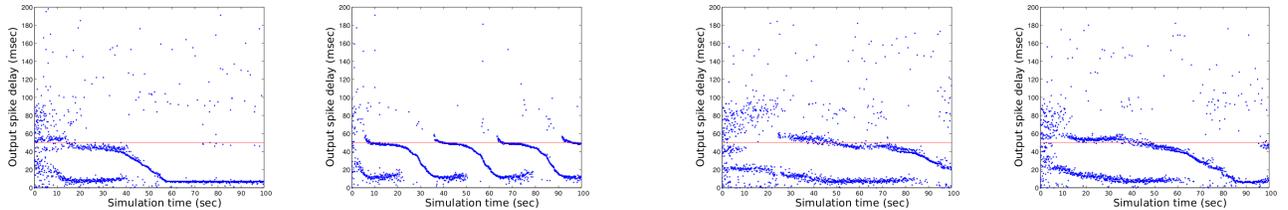


Fig. 11. Standard STDP: one input pattern, one output neurons.

A. Standard STDP results

1) *Single output neuron - Single input pattern:* The network used for this experiment is described in fig.10(a). We present here two scatter plots (fig.11) in which the network differs only by the initial and the maximum synaptic weights.

The output neuron in fig.11(a) identifies two different sections of the pattern, but around 30 seconds from the beginning of the simulation, the synaptic weights get stronger according to the last part of the pattern. Finally at 40 seconds of the simulation time, the output neuron identifies only the last part of the pattern and then, according to [8], the output neuron starts to advance in its identification until it reaches $10msec$ from the start of the pattern, then stabilizes around this value.

We reduced the maximum synaptic weights and we set random values (in the interval $[0; 2]$) as starting synaptic weights. This resulted in an unstable system (as it is possible to see in fig.11(b)). In this network, when the identification time reaches $20msec$ from the beginning of the pattern the neuron starts also to identify the final part of the pattern. Then the identification of the last part becomes stronger and stronger until the neuron stops identifying the first part of the pattern and then starts to tune to the earliest spikes. This process continues repeatedly. This behaviour is related to the low value for the maximum synaptic weight we are using, introducing the possibility for other patterns to distract the output neuron from the pattern that it is already identifying.

2) *Two output neurons - Single input pattern:* The network used for this experiment is described in fig.10(b). We present here two scatter plots (fig.12), one for each

(a) Scatter plot for the first output neuron

(b) Scatter plot for the second output neuron

Fig. 12. Standard STDP: one input pattern, two output neurons.

output neuron. It is possible to identify that the two output neurons lock their identification at two different times of the input pattern. This is a consequence of the lateral inhibition between the two output neurons.

3) *Two output neurons - Two input patterns:* The network used for this experiment is described in fig.10(b). We present four scatter plots, one for each neuron against each of the input patterns (fig.13). Figures 13(b) and 13(d) are the scatter plots with respect to the second pattern, and it is recognized at the very same moment by both the output neurons.

As it is possible to note, in figures 13(a) and 13(c), there is a “ghost” effect. This is connected with the presence of two different interleaved input patterns, whose inter-time is random. When a neuron identifies one of the two input patterns it fires with a delay from the beginning of the pattern identified and with a different delay (usually greater) from the beginning of the previous pattern occurrence. Given the randomness of time occurrence of the pattern, the result is the “ghost” effect in figures 13(a) and 13(c).

Because lateral inhibition acts with one millisecond delay from an output spike, two output neurons firing simultaneously will mutually inhibit one millisecond after the initial spike. Then as a result the first input pattern is not identified by any of the output neurons. The first input pattern is not identified by any of the output neurons: all the output neurons lock and consistently identify the second pattern.

4) *Four output neurons - Two input patterns:* The network used for this experiment is described in fig.10(c). We present eight scatter plots, one for each neuron in relation with each of the input pattern (fig.14). It is possible to see that the two first output neurons lock at the very same moment of the second pattern (figures 14(e) and 14(f)). The third neuron locks to the first pattern (fig.14(c)). The fourth neuron

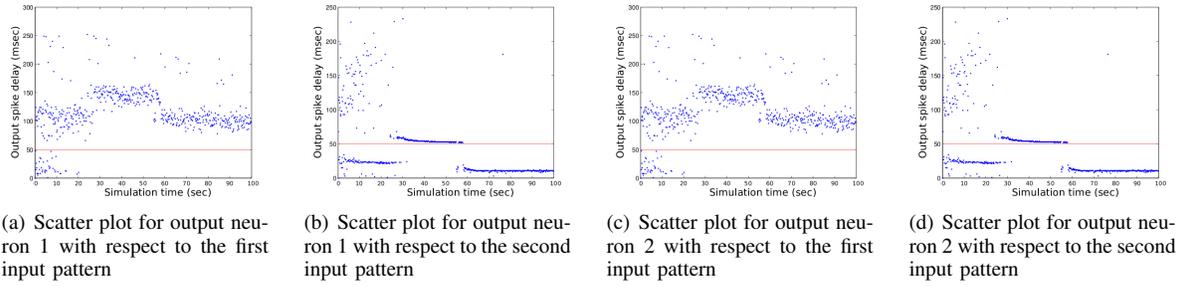


Fig. 13. Standard STDP: two input patterns, two output neurons

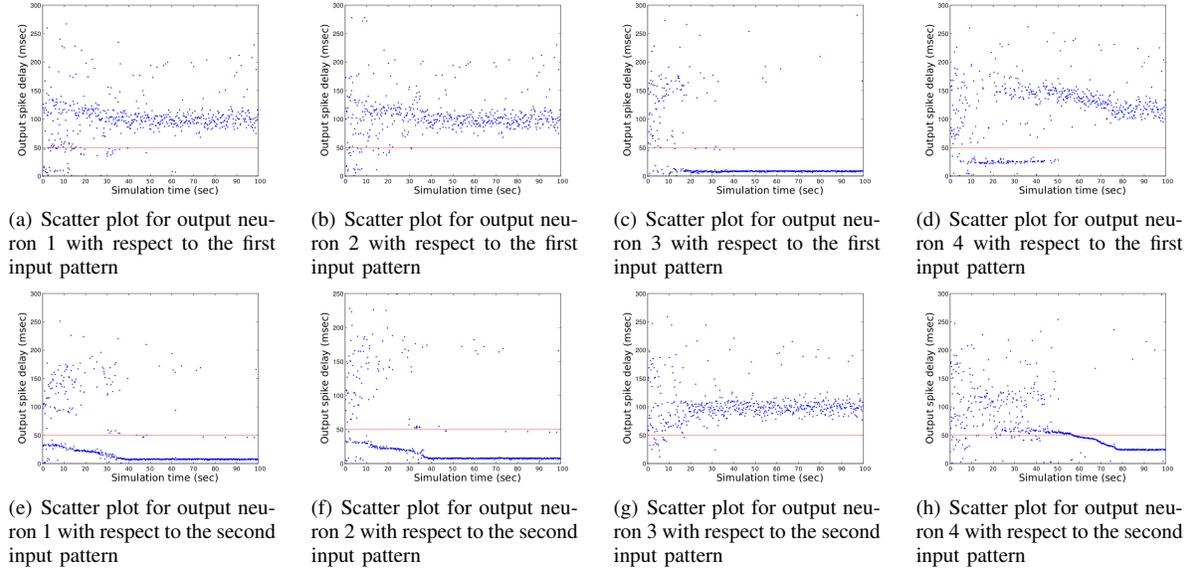


Fig. 14. Standard STDP: two input patterns, four output neurons.

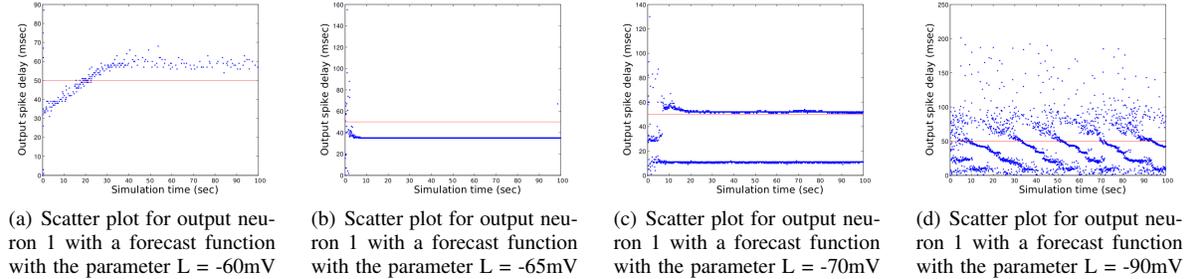


Fig. 15. STDP with TTS forecast: one input pattern, one output neuron. This is a comparison between the different values of the parameter “L” (see fig.8). The value used for the parameter “L” in all the other simulations will be $-65mV$.

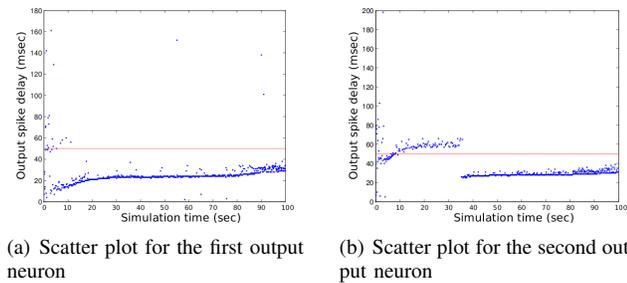


Fig. 16. STDP with TTS forecast: one input pattern, two output neurons.

initially tries to lock to the first pattern (fig.14(d)), but probably, due to the inhibition imposed by output neuron 3 it fails to lock and when the neurons 1 and 2 proceed towards the very beginning of the second pattern, neuron 4 locks to the end of the second pattern and starts advancing in the identification time, until it stabilizes at about $25msec$ (fig.14(h)).

B. Plasticity with Time-To-Spike forecast

We forecast the Time-To-Spike (TTS) of a neuron through a function which is composed of two line segments (see section III), as described in figure 8. In the first experiment

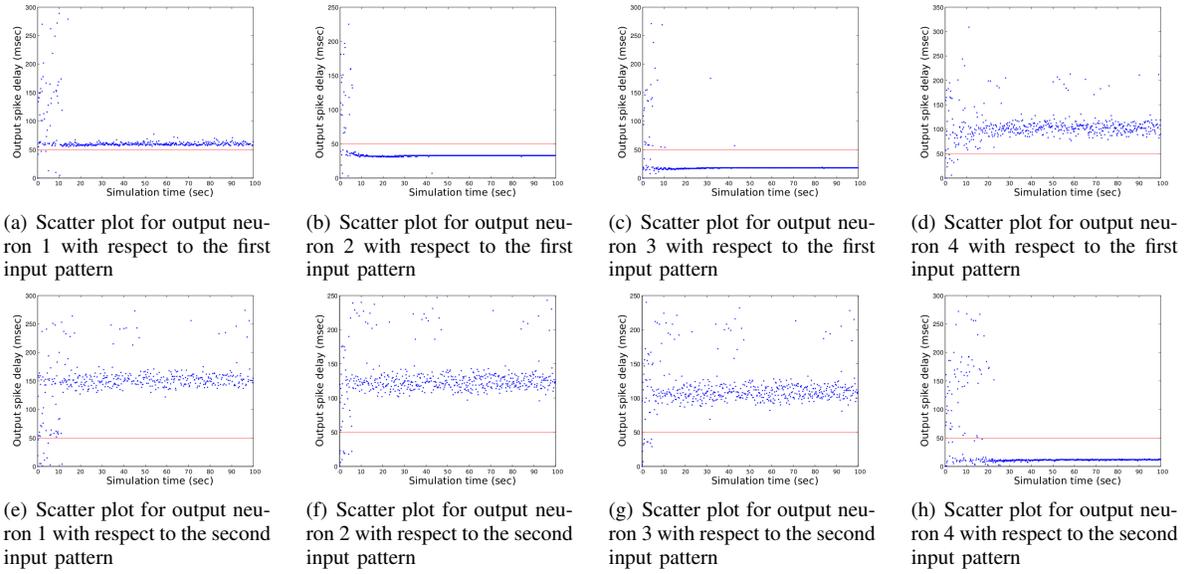


Fig. 17. STDP with TTS forecast: two input patterns, four output neurons.

we consider “ L ” a parameter and we analyse the learning behaviour of the output neuron compared with the standard STDP algorithm. After this we will use one value of this parameter to perform the next experiments.

1) *Single output neuron - Single input pattern:* The network used for this experiment is described in fig.10(a). We present here four scatter plots (fig.15), which are generated using various values of the parameter “ L ” described before (see section III). If this parameter is too small, the scatter plot shows that the neuron will identify the pattern later and later until the identification moves out of the pattern time and the neuron will eventually “die” (see fig.15(a)). If this parameter is too high, the result will be too noisy and the identification will be unstable (see fig.15(d)). If $L = -70mV$ (see fig.15(c)) the pattern identification will move very early in the pattern time, but this makes it possible for neuron to spike twice in the same pattern occurrence. Therefore, for all the next simulations we run, we consider $L = -65mV$ (see fig.15(b)).

2) *Two output neurons - Single input pattern:* The network used for this experiment is described in fig.10(b). Here we present two scatter plots, one for each output neuron (see fig.16). At the beginning of the simulation, it is possible to note that the output neurons lock to different parts of the input pattern. In particular, the first output neuron identifies the pattern at about $10msec$ and then moves later in time until it stabilizes around $25msec$. The second neuron gets inhibited by the first one and locks at about $40msec$ from the beginning of the pattern. Since the first neuron moves later in the identification of the pattern, the second neuron also has to move later and later until the identification moves out of the pattern time range. The neuron then locks to the same part of the pattern as the first neuron, and it stabilizes at the same time as the first neuron.

3) *Four output neurons - Two input patterns:* The network used for this experiment is described in fig.10(c). In this test the first three output neurons lock in different parts of the first pattern (figures 17(a), 17(b) and 17(c)). In particular the first neuron consistently identifies the presence of the pattern even if it fires after the end of the pattern. In these conditions the neuron uses the random background noise to reach the firing threshold, hence the firing delay changes from one occurrence of the pattern to the other. The fourth neuron locks at the beginning of the second pattern (see fig.17(h)).

VI. DISCUSSION

In this paper we introduced a new learning rule in neural networks with a statistical approach. This algorithm has a very important feature: to be able to compute both the potentiation and the depression of the synapses when a spike is received. This allows the operation to be performed in a single moment of the simulation: the spike arrived event. Some simulators (e.g. [16] [15]) may take advantage of this feature to achieve better performance.

This algorithm is based on a statistical approach, and we showed that small variations of the parameters involved may lead to very different behaviours. In particular, modification of the parameters of the forecast function leads to divergent behaviours, from tuning later and later in the spike input pattern, to unstable behaviours. In addition, we extracted the parameters for the forecast function starting from a network made of only two types of Izhikevich neurons. Changing the parameters of the neurons [12] may affect the forecast function. In the case of neurons with chaotic behaviour there is a possibility that this forecast function may not consistently predict the time-to-spike of the neuron. On the other hand, it is questionable whether purely chaotic neurons are able to analyse input and respond consistently with repeated patterns detection.

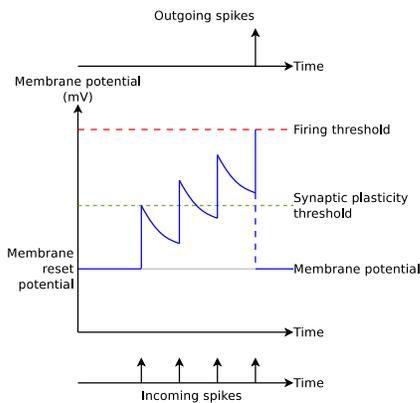


Fig. 18. Description of synaptic plasticity trigger. Only the last incoming spike triggers LTP because the membrane potential is above the plasticity threshold. Even if this synapse is strengthened the previous three synapses from which the three spikes are received are not influenced and the output spike cannot tune to the earlier spikes.

We have shown that this learning algorithm locks to the input pattern very quickly, and even *more* quickly than the standard STDP algorithm in some conditions.

In contrast to the standard STDP algorithm, this algorithm is not able to tune to the earliest spikes in the input pattern [8]. This can be justified by looking at the membrane potential values when the neuron receives spikes from an already identified pattern (see fig.18). Before the LTP part of the synaptic plasticity is triggered, the incoming spikes need to bring the membrane potential values above the threshold set with the parameter “L” (see fig.8) described before. In addition, if we move this parameter to a value which is too low, the network may adapt to random noise that may be present in the input pattern which increases the membrane potential value (in the mean). In the tests we ran, no noise was present on the input pattern. Adding some noise may affect these results and even the forecast function. At the same time, opening the forecast time window to values greater than what we chose in this paper may lead to unpredictable behaviours. This field needs to be explored to obtain the maximum performance from this algorithm.

Future work may include deeper analysis of the forecast function so that it can include more type of neurons and more network behaviour (e.g. oscillations, bursts) than what we used in the tests described in this paper.

This paper described an algorithm we implement that allows triggering the STDP only on one event, easing the implementation in simulators and reducing the time needed for computation, compared with the standard STDP algorithm. Such an abstraction could be part of a neural network simulator to develop a biologically-plausible learning rule.

VII. ACKNOWLEDGEMENTS

The SpiNNaker project is supported by the Engineering and Physical Sciences Research Council, grant EP/4015740/1, and also by ARM and Silistix. We appreciate the support of these sponsors and industrial partners. The au-

thors would like to thank Timothée Masquelier for providing useful information and Eugene Izhikevich for providing the source code of the simulator.

REFERENCES

- [1] L. F. Abbott and S. B. Nelson. Synaptic plasticity: taming the beast. *Nature neuroscience*, 3 Suppl:1178–1183, November 2000.
- [2] A. Banerjee. On the phase-space dynamics of systems of spiking neurons. I: model and experiments. *Neural computation*, 13(1):161–193, January 2001.
- [3] G. Bi and M. Poo. Synaptic modification by correlated activity: Hebb’s postulate revisited. *Annual review of neuroscience*, 24(1):139–166, 2001.
- [4] G.-Q. Bi and M.-M. Poo. Synaptic Modifications in Cultured Hippocampal Neurons: Dependence on Spike Timing, Synaptic Strength, and Postsynaptic Cell Type. *J. Neurosci.*, 18(24):10464–10472, December 1998.
- [5] J. M. Brader, W. Senn, and S. Fusi. Learning Real-World Stimuli in a Neural Network with Spike-Driven Synaptic Dynamics. *Neural Computation*, 19(11):2881–2912, September 2007.
- [6] A. N. Burkitt, M. Gilson, and J. L. van Hemmen. Spike-timing-dependent plasticity for neurons with recurrent connections. *Biological cybernetics*, 96(5):533–546, May 2007.
- [7] C. Clopath, L. Busing, E. Vasilaki, and W. Gerstner. Connectivity reflects coding: a model of voltage-based STDP with homeostasis. *Nature Neuroscience*, 13(3):344–352, March 2010.
- [8] R. Guyonneau, R. VanRullen, and S. J. Thorpe. Neurons Tune to the Earliest Spikes Through STDP. *Neural Computation*, 17(4):859–879, April 2005.
- [9] D. O. Hebb. *The Organization of Behavior: A Neuropsychological Theory*. 1949.
- [10] G. Indiveri, E. Chicca, and R. Douglas. A VLSI Array of Low-Power Spiking Neurons and Bistable Synapses With Spike-Timing Dependent Plasticity. *IEEE Transactions on Neural Networks*, 17(1):211–221, January 2006.
- [11] E. M. Izhikevich. Simple model of spiking neurons. *IEEE Trans. Neural Networks*, pages 1569–1572, 2003.
- [12] E. M. Izhikevich. Which model to use for cortical spiking neurons? *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 15(5):1063–1070, September 2004.
- [13] E. M. Izhikevich. Polychronization: Computation with Spikes. *Neural Computation*, 18(2):245–282, February 2006.
- [14] E. M. Izhikevich and N. S. Desai. Relating STDP to BCM. *Neural Computation*, 15(7):1511–1523, July 2003.
- [15] X. Jin, A. Rast, F. Galluppi, S. Davies, and S. Furber. Implementing spike-timing-dependent plasticity on SpiNNaker neuromorphic hardware. pages 1–8, July 2010.
- [16] X. Jin, A. Rast, F. Galluppi, M. Khan, and S. Furber. Implementing Learning on the SpiNNaker Universal Neural Chip Multiprocessor. In C. S. Leung, M. Lee, and J. H. Chan, editors, *Neural Information Processing*, volume 5863, chapter 48, pages 425–432. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [17] H. Markram and M. Tsodyks. Redistribution of synaptic efficacy between neocortical pyramidal neurons. *Nature*, 382(6594):807–810, August 1996.
- [18] T. Masquelier, R. Guyonneau, and S. J. Thorpe. Competitive STDP-Based Spike Pattern Learning. *Neural computation*, 21(5):1259–1276, December 2008.
- [19] T. Masquelier, R. Guyonneau, and S. J. Thorpe. Spike Timing Dependent Plasticity Finds the Start of Repeating Patterns in Continuous Spike Trains. *PLoS ONE*, 3(1):e1377+, January 2008.
- [20] J.-P. P. Pfister and W. Gerstner. Triplets of spikes in a model of spike timing-dependent plasticity. *J. Neurosci.*, 26(38):9673–9682, September 2006.
- [21] S. Song and L. F. Abbott. Cortical development and remapping through spike timing-dependent plasticity. *Neuron*, 32(2):339–350, October 2001.
- [22] S. Song, K. D. Miller, and L. F. Abbott. Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nature neuroscience*, 3(9):919–926, September 2000.
- [23] H.-X. X. Wang, R. C. Gerkin, D. W. Nauen, and G.-Q. Q. Bi. Coactivation and timing-dependent integration of synaptic potentiation and depression. *Nature neuroscience*, 8(2):187–193, February 2005.