WHAT CAN A NEURON LEARN WITH SPIKE-TIMING-DEPENDENT PLASTICITY?
Legenstein, Naeger & Maass

Vered Volansky, Irina Lipov, Ala Shiban
SOME DEFINITIONS...

- Spiking Neurons – computational models from input spike trains to output spike trains
- Plasticity – The ability of a synapse to change its strength
- STDP – Spike Timing Dependent Plasticity
- Teacher Forcing – A current that enforces the desired behaviour
- EPSP – Excitatory PostSynaptic Potentials
- IPSP – Inhibitory PostSynaptic Potentials
INTRODUCTION

- Examine to what extent biologically realistic models can be taught via STDP to implement a given transformation $F$
- Show that no theoretical guarantee can be given for the convergence of STDP for arbitrary input
- Prove that in some cases the Perceptron convergence theorem holds for STDP
- Show computer simulations where STDP holds
INTRODUCTION

- There exist many maps from input spike trains to output spike trains that cannot be realized by a neuron for any parameter settings.
  - High rate output spike train in the absence of any input spikes
  - No parameter setting for a neuron to represent a transformation without an equilibrium point
INTRODUCTION

- Adjustable parameters: \( w \) & \( U \)

- Focus on transformations \( F \) that can be implemented by a neuron in a stable manner for some values \( p \) of its adjustable parameters and ask which of these can be learnt by such a neuron

- Not clear which parameters influence the behaviour of a biologically realistic synapse

- Only biological realistic dynamic synapses are considered in the computer simulations in this article

- The neuron is taught to fire at particular points in time via extra input currents
INTRODUCTION

- Section 2 – models for neurons and synapses and STDP rule
- SNCC – The Spiking Neuron Convergence Conjecture
- SNCC is closely related to the Perceptron convergence theorem
- Section 3 – STDP & Perceptron + SNCC fails for worst-case scenarios
- Section 4 – STDP analysis for the average case
- Section 5 – Computer Simulations
MODELS FOR NEURONS, SYNAPSES AND STDP

- Neuron

Standard leaky integrate-and-fire neuron model

\[ \tau_m \frac{dV_m}{dt} = -(V_m - V_{\text{resting}}) + R_m \cdot (I_{\text{syn}}(t) + I_{\text{background}} + I_{\text{inject}}(t)) \]

\[ \tau_m = C_m \cdot R_m \]
SYNAPSE MODEL – CONT.

- **Synapse**


  The model predicts the amplitude $A_k$ of the EPSC for the $k$th spike in a spike train
SYNAPSE MODEL – CONT.

\[ A_k = w \cdot u_k \cdot R_k \]

\[(2.1)\]

\[ u_k = U + u_{k-1}(1 - U) \exp(-\Delta_{k-1}/F) \]

\[ R_k = 1 + (R_{k-1} - u_{k-1}R_{k-1} - 1) \exp(-\Delta_{k-1}/D) \]

- U, D, F – randomly chosen from Gaussian distributions
- Mean values were chosen to be:
  - 0.5, 1.1, 0.05 (E)
  - 0.25, 0.7, 0.02 (I)
STDP EFFECT

- STDP effect is commonly tested by measuring the amplitude of the EPSP in the postsynaptic neuron for a single spike

\[ A_1 = w \cdot U \cdot R_1 \]

- The change in the Amplitude \( \Delta A \) can be caused by:
  - Either \( w \) or \( U \) changes (or other synaptic parameters).
  - \( W \) is most commonly assumed to change
  - \( U \) is favored by Markram & Tsodyks (1996)
  - The latter is not considered
EPSPS’ CHANGE IN AMPLITUDE

\[
A(\Delta t) = \begin{cases} 
W_+ \cdot e^{-\Delta t/\tau_+}, & \text{if } \Delta t > 0 \\
-W_- \cdot e^{\Delta t/\tau_-}, & \text{if } \Delta t \leq 0
\end{cases}
\]

\[W_+ , W_- , \tau_+ , \tau_- > 0 , \quad 0 < A_1 < A_{\text{max}}\]
\[t^{\text{post}} = t^{\text{pre}} + \Delta t\]

- \(W_+ \) and \( W_- \) are constants based on the synaptic weight
- \( \tau_+ \) and \( \tau_- \) are time constants
- As \( \Delta t \) is shorter the \( \exp \) is closer to 0 and \( \Delta A \) is larger.
LINEAR NEURON MODEL

- Spike trains are represented as the sum:
  \[ S(t) = \sum_k \delta(t - t_k) \]

- The leaky integrate-and-fire neuron is replaced by a linear neuron model.

- This neuron model outputs a spike train \( S^{post}(t) \), which is a realization of a Poisson process with the underlying instantaneous firing rate \( R^{post}(t) \).

  The effect of an input spike at input \( i \) at time \( t \) is modeled by an increase in the instantaneous firing rate of an amount \( w_i \( t' \) e(t - t') \), where \( e \) is a response kernel and \( w_i \( t \) is the synaptic efficacy of synapse \( i \) at time \( t' \)
LINEAR NEURON MODEL – CONT.

- The contributions of all inputs are summed up linearly:

\[ R^{post}(t) = \sum_{j=1}^{n} \int_{0}^{\infty} ds \ w_j(t - s) \ \epsilon(s) \ S_j(t - s) \]

where \( S_1, \ldots, S_n \) are the \( n \) presynaptic spike trains and \( \epsilon(s) = 0 \) for \( s < 0 \)

- The generation of an output spike is independent of previous output spikes
SNCC FOR STDP VS. PERCEPTRON CONVERGENCE

- $\Delta w$ of the weight of a synapse is proportional to:

$$
\begin{cases}
W_+ \cdot e^{-\Delta t / \tau_+}, & \text{if } \Delta t > 0 \\
-W_- \cdot e^{\Delta t / \tau_-}, & \text{if } \Delta t \leq 0
\end{cases}
$$

(3.1)

while: $0 < w < w_{\text{max}}$

- STDP changes the synaptic weight to according to the rule: $w_{\text{new}} = w_{\text{old}} + \Delta w$

$$
\begin{cases}
\min\{w_{\text{max}}, \ w_{\text{old}} + W_+ \cdot e^{-\Delta t / \tau_+}\}, & \text{if } \Delta t > 0 \\
\max\{0, \ w_{\text{old}} - W_- \cdot e^{\Delta t / \tau_-}\}, & \text{if } \Delta t \leq 0
\end{cases}
$$

(3.2)

With predefined parameters $W_+, W_- > 0$
STDP COMMON RULE

There exists some analogy to other learning rules with very simplified neuron models

\[ x = \langle x_0, \ldots, x_n \rangle \in \mathbb{R}^{n+1} \]

\[ y \in \mathbb{R} \]

\[ w = \langle w_0, \ldots, w_n \rangle \in \mathbb{R}^{n+1} \]

\[ y = w \cdot x \]

\[ \Delta w = \eta \cdot x \cdot y \]

\[ \eta \geq 0 \]

Learning rate

input

output

weights

Hebbian rule for changing the weights

Linear neuron output
SUPERVISED LEARNING

- There exists a target value $y_{\text{teacher}}$ for the output of a neuron.

Replace $y$ in the Hebb rule (3.3) with $y_{\text{teacher}} - y$:

$$\Delta w = \eta \cdot x \cdot (y_{\text{teacher}} - y) \quad (3.4)$$

Where $y_{\text{teacher}}$ is the tagging and $y$ the actual output.
PERCEPTRON LEARNING RULE AND CONVERGENCE

\( y \in \{0,1\} \)

\[ \Delta w = \begin{cases} 
\eta \cdot x, & \text{if } y_{\text{teacher}} = 1 \text{ and } y = 0 \\
\eta \cdot (-x), & \text{if } y_{\text{teacher}} = 0 \text{ and } y = 1 \\
0, & \text{otherwise} \end{cases} \quad (3.5) \]

- Learning with the perceptron learning rule converges for a given list \( L \) of examples iff the list \( L \) is linearly separable.

- If \( L \) is linearly separable, then the weight vector to which this learning rule converges is autonomically a solution of the corresponding classification problem.
SPIKING NEURON CONVERGENCE CONJECTURE - SNCC

- Any setting of $w$ that allows a Perceptron to solve a given classification task is automatically stable with regard to the Perceptron learning rule.
- Such automatic stability is not provided by STDP.
- We consider in this section only learning tasks for spiking neurons for which a solution exists that is stable with regard to STDP.
In other words - clarify whether in a supervised paradigm where the output is clamped to the teacher signal, STDP enables a spiking neuron, starting from any initial weights, to learn any transformation $F$ from input spike trains to output spike trains that it can possibly implement in a stable manner.
PERCEPTRON VS. STDP

Different structure of inputs and outputs

**Perceptron**
- static vectors of numbers

\[(x_1, x_2, \ldots, x_n)\]

**STDP**
- functions of time (spike trains):
  The transformation \( F \) from inputs to outputs computed by a spiking neuron that maps \( n \) functions \( S_i \) that represent \( n \) input spike trains \( S_1, \ldots, S_n \) onto some output spike train \( S \) of the same form
The sign of any weight $w_i$ of a perceptron can be changed by the perceptron learning rule.

STDP doesn’t turn an excitatory synapse into an inhibitory synapse, or vice versa.
If an example $x$ that should be classified negatively is incorrectly classified through the current weight vector $w$ (i.e., $y = \text{sign}(w \cdot x) = 1$ but $y_{\text{teacher}} = 0$), the perceptron rule changes $w$ in a way that makes a reoccurrence of this mistake less likely.

When an output neuron fires incorrectly it is likely to reoccur:

- **Supressing teacher currents**: no changes of synaptic parameters are triggered during training. Hence, this mistake is likely to show up again during testing.

- **No Supressing teacher currents**: rules 2.1 to 2.3, 3.1, and 3.2 for STDP change the synaptic parameters in a way that positively reinforces future reoccurrences of this mistake.
## PERCEPTRON VS. STDP – CONT.

<table>
<thead>
<tr>
<th><strong>Perceptron</strong></th>
<th><strong>STDP</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>the weights of the Perceptron are unchanged when there is no mistake (i.e., ( y_{teacher} = y; ) see the third line of equation 3.5)</td>
<td>continue to change synaptic parameters even if the neuron fires exactly at the desired times ( t ) (even if this firing occurs without the help of an extra “teaching current”) – This is due to the dependency on ( \Delta t )</td>
</tr>
</tbody>
</table>
DIFFERENCES INFLUENCE

- **W Sign changes** – not crucial for convergence since the perceptron convergence theorem also holds for a sign-constrained version of the Perceptron learning rule.

- **Wrong classification**
  Very serious and in worst-case scenarios entails falsification of the SNCC for STDP.
DISREGARDING NEGATIVE EXAMPLES

\[ \Delta w = \begin{cases} 
\eta \cdot x, & \text{if } y_{\text{teacher}} = 1 \text{ and } y = 0 \\
-\eta \cdot x, & \text{if } y_{\text{teacher}} = 0 \text{ and } y = 1 \\
0, & \text{otherwise}.
\end{cases} \quad (3.5) \]

- This will falsify the perceptron convergence theorem
When disregarding the negative example a decision boundary parallel to H will arise.
COUNTEREXAMPLE FOR SNCC

S1, S2, S3 denote three input spike trains to three synapses of a neuron.

A is a positive example where firing of the postsynaptic neuron at time t3 is desired.

B is a negative example where no firing of the postsynaptic neuron is desired.

- S1, S2, S3 denote three input spike trains to three synapses of a neuron.
- A is a positive example where firing of the postsynaptic neuron at time t3 is desired.
- B is a negative example where no firing of the postsynaptic neuron is desired.
COUNTEREXAMPLE FOR SNCC – CONT.

- A solution does exist:
  Initial values $w_1 = w_3 = w_{\text{max}}$ and $w_2 = 0$

- But if learning starts with other initial values, it won’t converge:
  initial values $w_1 = w_3 = w_{\text{max}}$ and $w_2 = w_{\text{max}}/4$
  Fire in both scenarios and the weights will never decrease
    initial values $w_1 = w_3 = 0$ and $w_2 = w_{\text{max}}/4$ while enabling only the desired fire (A). If (A) occurs a lot during training then the weights will converge to $w_1 = w_3 = w_{\text{max}}$ and $w_2 = w_{\text{max}}/4$ and again fire in both scenarios
COUNTEREXAMPLE FOR SNCC – CONCLUSION

- STDP does not converge from these initial weights to a solution of this learning problem, although a stable solution exists.
- Counterexamples can be constructed for any given positive values of the parameters $W_+, W_-.$
- This counterexample shows that no convergence theorem can exist for STDP that holds, like the Perceptron convergence theorem, for any given set of inputs.
Theoretical Results on STDP in the Context of Supervised Learning
Average Case Analysis

- STDP a convergence result are different from the perceptron convergence theorem: *No guarantee of convergence for any set of inputs* even *under the assumption that a suitable weight vector exists*

- Motivation: *Average case analysis of STDP for Poisson input spike trains*
  - *Poisson input*  
    - *Poisson distribution* - expresses the probability of a number of events occurring in a fixed period of time if these events occur with a known average rate and independently of the time since the last event.*  
      - Different spikes are independent
      - Refractory period
Average Case Analysis (Cont.)

- **Validity of the SNCC for STDP**
  - **Motivation from perceptron**
    - If the list \( L \) of training examples contained not just a single positive example but rather a larger set of positive examples covering the area above \( H^* \) then \( L \) would contain more positive examples \( x_1, x_2 \) with \( x_2 > x_1 \) than positive examples with \( x_2 < x_1 \).
    - This will cause a weight vector \( w \) with \( w_2 > w_1 \), since weight vector \( w \) is proportional to the sum of positive counterexamples that occur during learning. According to:

\[
\Delta w = \begin{cases} 
\eta \cdot x, & \text{if } y_{\text{teacher}} = 1 \text{ and } y = 0 \\
-\eta \cdot (-x), & \text{if } y_{\text{teacher}} = 0 \text{ and } y = 1 \\
0, & \text{otherwise}.
\end{cases}
\]  

\text{(3.5)}
Average Case Analysis (Cont.)

- **Validity of the SNCC for STDP (Cont.)**
  
  Poisson input provides:
  
  - more uniform distribution of input patterns for which the neuron is supposed to fire is generated
  
  - [Implicitly] information about the distribution of negative examples, that is, input patterns for which the neuron is not supposed to fire, and hence they can indirectly influence the learning process even without any explicit provision in the rule for STDP that discourages the firing of the neuron for such input patterns.

- **Observation:** In the average case a form of SNCC holds
Average Case Analysis (Cont.)

- **Weight changes resulting STDP**
  - Amplitude change according to **one** pre-post spike pair
    \[
    A(\Delta t) = \begin{cases} 
    W_+ \cdot e^{-\Delta t / \tau_+}, & \text{if } \Delta t > 0 \\
    -W_- \cdot e^{\Delta t / \tau_-}, & \text{if } \Delta t \leq 0
    \end{cases}
    \]
  - \(W_+/W_- > 0\) - learning rates
  - The \(\Delta w\) of **one** pre-post spike pair is proportional to Amplitude change

- **Total weight change in time interval** \(T\)
  - \(\Delta w_i(t) = w_i(t + T) - w_i(t)\)
  - All pre-post pairs in a time-interval \(T\) are considered [integrating up all the weight changes in the time-interval \(T\)]
Average Case Analysis (Cont.)

- Total weight change in time interval $T$

\[
\Delta w_i(t) = \int_t^{t+T} dt' \int_t^{t+T} dt'' A(t'' - t') S^*(t'') S_i(t').
\]

- Intuition to the formula:
  - The weight update is performed iff there were pre-synaptic and post-synaptic spike: $S^*(t'') S_i(t')$ serves to this indicator, getting value 1 in this case and 0 in other cases.

  when $S(t) = \sum_k \delta(t - t_k)$, where $t_k$ is the $k$th spike

  - As remembered weight change of one pre-post spike pair is proportional to Amplitude change
Average Case Analysis (Cont.)

- **Total weight change in time interval** $T$ (Cont.)
  
  - By substituting $s = t'' - t'$ we get
    \[
    \Delta w_i(t) = \int_t^{t+T} dt' \int_{t-t'}^{t+T-t'} ds \ A(s) S^*(t' + s) S_i(t').
    \]
  
  - Weight changes during the time interval $[t, t + T]$ can potentially also be caused by pre- or postsynaptic spikes that do not fall into this interval
    \[
    \Delta w_i(t) = \int_t^{t+T} dt' \int_{-\infty}^{\infty} ds \ A(s) S^*(t' + s) S_i(t').
    \]
A Necessary Condition on Input Spike Trains.

- If we assume that the statistics of input and output spike trains are constant over a learning trial, the total weight change over a sufficiently long time interval $T$ provides a good predictor for the end result of a learning process.

- Consider:
  - Neuron with $n$ synapses
  - Set $M \subseteq \{1, 2...n\}$
  - the neuron computes the target transformation $F^*$ if and only if $w_i = w_{\text{max}}$ for all $i \in M$ and $w_i = 0$ for all $i \notin M$
A Necessary Condition on Input Spike Trains (Cont.)

- All weights $w_i, i \in M$ have positive total weight change
- All weights $w_i, i \notin M$ have negative total weight change

\[
\begin{align*}
\int_t^{t+T} dt' \int_0^\infty ds \ W_+ S^*(t' + s)S_i(t') e^{-s/\tau_+} \\
- \int_t^{t+T} dt' \int_{-\infty}^0 ds \ W_- S^*(t' + s)S_i(t') e^{s/\tau_-} > 0.
\end{align*}
\]

\[
\frac{W_-}{W_+} < \frac{\int_t^{t+T} dt' \int_0^\infty ds \ S^*(t' + s)S_i(t') e^{-s/\tau_+}}{\int_t^{t+T} dt' \int_{-\infty}^0 ds \ S^*(t' + s)S_i(t') e^{s/\tau_-}}
\]

\[
\begin{align*}
\int_t^{t+T} dt' \int_0^\infty ds \ W_+ S^*(t' + s)S_i(t') e^{-s/\tau_+} \\
- \int_t^{t+T} dt' \int_{-\infty}^0 ds \ W_- S^*(t' + s)S_i(t') e^{s/\tau_-} < 0.
\end{align*}
\]

\[
\frac{W_-}{W_+} > \frac{\int_t^{t+T} dt' \int_0^\infty ds \ S^*(t' + s)S_i(t') e^{-s/\tau_+}}{\int_t^{t+T} dt' \int_{-\infty}^0 ds \ S^*(t' + s)S_i(t') e^{s/\tau_-}}
\]
A Necessary Condition on Input Spike Trains (Cont.)

○ Conclusion
  • A value in the middle between these maximum and minimum values for $W_{-}/W_{+}$ seems desirable to minimize the effects of noise in the learning process.
  • Note: noise in the learning process can be caused for example
    ○ source of noise that is specific to neurons arises from the finite number of ion channels in a patch of neuronal membrane
    Most ion channels have only two states: they are either open or closed. The electrical conductivity of a patch of membrane for ion type $i$ is proportional to the number of open ion channels.
    ○ source of noise, which is literally omnipresent, is thermal noise. Due to the discrete nature of electric charge carriers, the voltage $u$ across any electrical resistor $R$ fluctuates at finite temperature (Johnson noise). The variance of the fluctuations at rest is $\langle \Delta u^2 \rangle \propto R k T B$ where $k$ is the Boltzmann constant, $T$ the temperature and $B$ the bandwidth of the system
    Since neuronal dynamics is described by an equivalent electrical circuit containing resistors, the neuronal membrane potential fluctuates as well.
Correlated and Uncorrelated Poisson Input

In general, the spike trains $S_1, \ldots, S_n, S^*$ may not be known, only the process that generated them. Example:

- Only the statistics of the inputs is known [correlated Poisson spike trains]
- Postsynaptic spike generation process is stochastic and $S^*$ is therefore not known explicitly
- $w_i$ is a random variable with a mean drift and fluctuations around it
Correlated and Uncorrelated Poisson Input

- Bounded growth of weights:
  \[
  w_{new} = \begin{cases} 
  \min\{w_{\text{max}}, \ w_{old} + W_+ \cdot e^{-\Delta t/\tau_+}\}, \\
  \max\{0, \ w_{old} - W_- \cdot e^{\Delta t/\tau_-}\},
  \end{cases}
  \]

- Update dependent on the actual weight value
  - \( \Delta w = \begin{cases} 
  W_+ \cdot f_+(w) \cdot e^{-\Delta t/\tau_+}, & \text{if } \Delta t > 0 \\
  -W_- \cdot f_-(w) \cdot e^{\Delta t/\tau_-}, & \text{if } \Delta t \leq 0,
  \end{cases} \)

- \( \mu_+ / \mu_- \geq 0 \)
  - \( f_+(w) = \left((w_{\text{max}} - w)/w_{\text{max}}\right)^{\mu_+} \) and \( f_-(w) = \left(w/w_{\text{max}}\right)^{\mu_-} \)

- Basic additive update \( \mu_+ = \mu_- = 0 \)
- Linearly dependent on the current weight update \( \mu_+ = \mu_- = 1 \)
Correlated and Uncorrelated Poisson Input (Cont.)

- $\mu_+ = \mu_- = \mu$, $w_{\text{max}} = 1$

  \[ f_+(w) := (1 - w)^\mu \quad \text{and} \quad f_-(w) := w^\mu. \]

  \[
  \downarrow
  \]

  \[
  \Delta w = \begin{cases} 
  W_+ \cdot (1 - w^\mu) \cdot e^{-\Delta t / \tau_+}, & \text{if } \Delta t > 0 \\
  - W_- \cdot w^\mu \cdot e^{\Delta t / \tau_-}, & \text{if } \Delta t \leq 0.
  \end{cases}
  \]

  \[
  \downarrow
  \]

  The total weight change can be approximated:

  \[
  \Delta w_i(t) = \int_t^{t+T} dt' \left[ \int_0^\infty ds \ W_+ f_+(w_i(t)) e^{-s/\tau} S^*(t' + s) S_i(t') \\
  - \int_{-\infty}^0 ds \ W_- f_-(w_i(t)) e^{s/\tau} S^*(t' + s) S_i(t') \right].
  \]
Correlated and Uncorrelated Poisson Input (Cont.)

- The ensemble of all possible realizations of input and output spike trains given by some fixed spike generation processes for input and output spike trains should be consider.

- **Definition:** The average over this ensemble is in the following denoted by $\langle \cdot \rangle_E$ and called **ensemble average**.

\[
\frac{\langle \Delta w_i \rangle_E(t)}{T} = \frac{1}{T} \int_t^{t+T} dt' \left[ f_+^\mu(w_i(t)) \int_0^\infty ds \ W_+ e^{-s/\tau} \langle S^*(t'+s)S_i(t') \rangle_E \right. \\
- \left. f_-^\mu(w_i(t)) \int_{-\infty}^{0} ds \ W_- e^{s/\tau} \langle S^*(t'+s)S_i(t') \rangle_E \right]
\]
Correlated and Uncorrelated Poisson Input (Cont.)

- Ensemble average (Cont.)
  - The function $<S_i(t)S^*(t + s)>_E$ measures the correlation between $S_i$ and $S^*$, is defined as the joint probability density for observing an input spike at synapse $i$ at time $t$ and an output spike at time $t + s$.
  - A real neuron does not integrate over the whole ensemble; instead, learning is driven by a single realization of the stochastic process. But instead of averaging over several trials, we may also consider one single long trial during which input and output characteristics remain constant.
Correlated and Uncorrelated Poisson Input (Cont.)

- **Definition:** Temporally averaged correlation function

\[
C_i(s; t) := \frac{1}{T} \int_t^{t+T} dt' \langle S_i(t') S^*_i(t' + s) \rangle_E.
\]

- **Reminder:** A process \( \{N_t, t > 0\} \) is called a homogeneous Poisson process if

\[
\forall t, s \geq 0, Pr\{N_{t+s} - N_t \geq 2\} = o(s)
\]

- **Assumption:** spike trains are homogeneous

\[
C_i(s; t) = \langle S_i(t) S^*_i(t + s) \rangle_E
\]

- **Approximation:**

\[
\frac{dw_i(t)}{dt} \equiv \dot{w}_i(t)
\]

\[
\dot{w}_i(t) = W_+ f_+^{\mu}(w_i(t)) \int_0^\infty ds \ e^{-s/\tau} C_i(s; t)
\]

\[- W_- f_-^{\mu}(w_i(t)) \int_0^\infty ds \ e^{s/\tau} C_i(s; t).
\]
Remainder: Learning task-

Given an arbitrary set \( M \subseteq \{1, \ldots, n\} \) and that the target weight vector \( w^* \) which satisfies \( w^*_i = 1 \) if \( i \in M \) and \( w^*_i = 0 \) otherwise. The target output spike train \( S^* \) is produced by a neuron with synaptic efficacies \( w^* \) and input spike trains \( S_1, \ldots, S_n \). The question is whether a neuron with some rather arbitrary initial weight vector can learn the target transformation \( F^* \), which maps inputs \( S_1, \ldots, S_n \) to the target output \( S^* \), defined by \( S_1, \ldots, S_n, w^* \).

A precise mathematical characterization of those target transformations \( F^* \) (defined by some weight vector \( w^* \)), which can be learned by STDP, is complicated.

- Hard to analyze while clipping around the barriers 0 and \( w_{\text{max}} \)

\[
\begin{align*}
w_{\text{new}} = & \begin{cases} 
\min\{w_{\text{max}}, \ w_{\text{old}} + W_+ \cdot e^{-\Delta t/\tau_+}\}, & \text{if } \Delta t > 0 \\
\max\{0, \ w_{\text{old}} - W_- \cdot e^{\Delta t/\tau_-}\}, & \text{if } \Delta t \leq 0
\end{cases}
\end{align*}
\]

- Easier to analyze. But this rule no longer yields convergence to the target vector \( w^* \) (in the case of supervised training with teacher-enforced output spike train \( S^* \)), but yields instead convergence to some other weight vector that is now dependent on \( \mu \)

\[
\begin{align*}
\Delta w = & \begin{cases} 
W_+ \cdot (1 - \bar{w})^\mu \cdot e^{-\Delta t/\tau_+}, & \text{if } \Delta t > 0 \\
-W_- \cdot \bar{w}^\mu \cdot e^{\Delta t/\tau_-}, & \text{if } \Delta t \leq 0
\end{cases}
\end{align*}
\]
We express this weight vector through a function $W : \mathbb{R}^+ \rightarrow (0, 1)^n$, which maps each $\mu > 0$ onto a weight vector $W(\mu)$ (we set $\mathbb{R}^+ := \{x \in \mathbb{R} : x > 0\}$). For $\mu \rightarrow 0$, $W(\mu)$ converges to the target vector $w^*$. Thus, we have to replace a direct analysis of supervised learning by the analysis of the limit of supervised learning for $\mu \rightarrow 0$. 
Definition: *Equilibrium point*

A point $x^*$ in the state space of a dynamical system is called an *equilibrium point* if it has the property that whenever the state of the system starts at $x^*$, it remains at $x^*$ for all future times. A equilibrium point $x^*$ is said to be *stable* if the state of the system converges to $x^*$ for all sufficiently small disturbances away from it.
Correlated and Uncorrelated Poisson Input (Cont.)

Definition:

The target weight vector $w^* \in \{0, 1\}^n$ can approximately be learned in a supervised paradigm where the output is clamped to the teaching signal by STDP with soft weight bounds on homogeneous Poisson input spike trains (short: “$w^*$ can be learned”) if and only if there exists a function $W : \mathbb{R}^+ \rightarrow (0, 1)^n$ with $\lim_{\nu \rightarrow 0} W(\nu) = w^*$ and there exist $W_+, W_- > 0$ such that for all $\mu > 0$ the ensemble averaged weight vector $<w(t)>_E$ with learning dynamics given above converges to $W(\mu)$ for any initial weight vector $w(0)$.
Theorem 1

A target weight vector $\mathbf{w}^* \in \{0, 1\}^n$ can be learned if and only if there exists a function $\mathcal{W} : \mathbb{R}^+ \rightarrow (0, 1)^n$ with $\lim_{v \rightarrow 0} \mathcal{W}(v) = \mathbf{w}^*$ and there exist $W_+, W_- > 0$, such that for all $\mu > 0$, $\mathcal{W}(\mu)$ is a stable equilibrium point of the ensemble averaged weight vector $\langle \mathbf{w}(t) \rangle_E$ with learning dynamics given by equation
Theorem 1 - purpose

- There is a technicality included in terms of $\mu$. The $w^*$ we are interested in is one which consists of 0s and 1s. But for the proof we need the rule with soft-bound, i.e., with the $\mu$. However, in the limit $\mu \to 0$, the soft-bound rule becomes equivalent with the hard-bound rule (where we just clip weights at 0 and 1). We thus have to take the detour over the soft-bound rule to arrive at the result for the hard-bound rule and $w^*$ with 0s and 1s.
Theorem 1 + Theorem 2
When/What can be learned with STDP

- STDP cannot change the “sign” of a synapse
- STDP keeps changing synaptic parameters for inputs that are already processed in the desired way by the neuron

**Theorem 2.** A weight vector $w^*$ can be learned for homogeneous Poisson input spike trains with window correlation matrices $C^+$ and $C^-$ to a linear Poisson neuron with nonnegative response kernel if and only if $w^* \neq 0$ and

$$\frac{\sum_{k=1}^n w^*_k c^+_{ik}}{\sum_{k=1}^n w^*_k c^-_{ik}} > \frac{\sum_{k=1}^n w^*_k c^+_{jk}}{\sum_{k=1}^n w^*_k c^-_{jk}}$$

for all pairs $(i, j) \in \{1, \ldots, n\}^2$ with $w^*_i = 1$ and $w^*_j = 0$.

- Theorem 2 can be interpreted in the following way. The amount of correlation between input $i$ and the output also depends on other inputs $k$, which are correlated with this input. Furthermore, the impact of such a correlated input depends on its weight. In the linear model, these effects are summed up. Theorem 2 asserts a criterion on the fraction of such summed correlations in the positive and negative learning window. This fraction needs to be larger for synapses that should be potentiated than for synapses that should be depressed.
Theorem 1 + Theorem 2
When/What can be learned with STDP

- Respond selectively to the start of the repeating pattern
  - Acts as coincidence detector

- If the input contains more than one repeating pattern - a postsynaptic neuron picks one randomly, and becomes selective to it and only to it

- In order to learn the other patterns other neurons are needed. A competitive mechanism ensures that all the different patterns are cover and avoid learning the same ones (inhibitory horizontal connections between neurons)

- A long input pattern can be coded by the successive firings of several STDP neurons, each selective to a different part of the pattern

- STDP tracks back through the pattern, from one coincidence to the previous one, until the initial coincidence is reached and the chain of causality is stopped. After long time neuron is selective only to the simultaneous arrival of the pattern's earliest spikes, and can serve as ‘earliest predictor’ of the subsequent spike events at the risk of triggering a false alarm if these subsequent events don't occur, but with the benefit of being very reactive.
Theorem 1 + Theorem 2
When/What can be learned with STDP (Cont.)

- **Observation:** Neuron never fires during training except when it is supposed to fire.

- **Example:** In the subsequent computer simulations, the neuron received a strong depolarizing input when it was supposed to fire and a hyperpolarizing input, which prevented most (but not all) undesired firing, when it was not supposed to fire. It turns out that the use of such hyperpolarizing teacher input is not necessary if one instead starts the learning with small (randomly assigned) initial weights.

With large initial weights and without hyperpolarizing teacher input, learning capabilities are weak (results not shown).
Abbreviation

- \( C_i^{\text{pos}} \) for \( \frac{1}{\tau} \int_0^\infty ds \ e^{-s/\tau} C_i(s) \)

- \( C_i^{\text{neg}} \) for \( \int_{-\infty}^0 ds \ e^{s/\tau} C_i(s) \)

Theorem 1 proof
Theorem 1 Proof

To show the "if" part of theorem 1, we show that for any $\mu > 0$, the stable equilibrium point $W(\mu) = (w_{\mu 1}, \ldots, w_{\mu n})$ is the only equilibrium point of the system. Consider an arbitrary $\mu > 0$ and an arbitrary synapse $i$. Since $w_{\mu i}$ is a stable equilibrium point, the synaptic drift for small perturbations from $w_{\mu i}$ is such that $w_i$ converges to $w_{\mu i}$. We show that the synaptic drift has this property for all initial values $w_i(0) \in [0, 1]$ (since the system is time invariant, it suffices to consider perturbations at $t = 0$). For all $w_i(0) < w_{\mu i}$ with $w_i(0)$ sufficiently close to $w_{\mu i}$, we know that the synaptic drift is positive, because the equilibrium point is stable. From equation 4.12, we get $0 < \dot{w}_i(0) = W_+ C_i^{\text{pos}} (1 - w_i(0))^\mu - W_- C_i^{\text{neg}} w_i(0)^\mu$. By definition, we have $C_i^{\text{pos}}, C_i^{\text{neg}} \geq 0$, and $C_i^{\text{pos}} = C_i^{\text{neg}} = 0$ is impossible since this would imply $\dot{w}_i(0) = 0$ for all values of $w_i(0)$. Therefore, it holds for all $w_i'(0)$ with $0 \leq w_i'(0) < w_i(0)$ that $W_+ C_i^{\text{pos}} (1 - w_i(0))^\mu - W_- C_i^{\text{neg}} w_i(0)^\mu < W_+ C_i^{\text{pos}} (1 - w_i'(0))^\mu - W_- C_i^{\text{neg}} w_i'(0)^\mu$. Hence, the synaptic drift is positive for all weight values smaller than $w_{\mu i}$. A similar argument shows that the synaptic drift is negative for all weight values $w_i(0)$ with $w_{\mu i} < w_i(0) \leq 1$.

Together, this implies that $w_{\mu}$ is the only globally stable equilibrium point of the learning dynamics. Hence, the ensemble averaged weight vector $(\mathbf{w}(t))_E$ converges to $W(\mu)$ for any initial weight vector $\mathbf{w}(0) \in [0, 1]^n$. 
Theorem 1 Proof

We now show the “only if” part of theorem 1. If the target vector can be learned, then for some \( W_+, W_- > 0 \), we know that for any \( \mu > 0 \), the ensemble averaged weight vector \( \langle \mathbf{w}(t) \rangle_E \) converges to \( \mathcal{W}(\mu) \) for any initial weight vector \( \mathbf{w}(0) \in [0, 1]^n \). Since \( \mathcal{W}(\mu) \in (0, 1)^n \), we can draw \( \mathbf{w}(0) \) from a small surrounding of \( \mathcal{W}(\mu) \) which is still in \( [0, 1]^n \). This implies that \( \mathcal{W}(\mu) \) is a stable equilibrium point of \( \langle \mathbf{w}(t) \rangle_E \) under the learning dynamics. Hence, for these values of \( W_+ \) and \( W_- \), it holds for all \( \mu > 0 \) that \( \mathcal{W}(\mu) \) is a stable equilibrium point of \( \langle \mathbf{w}(t) \rangle_E \) under the learning dynamics. This implies the “only if” part of theorem 1.
WHAT CAN A NEURON LEARN WITH SPIKE-TIMING-DEPENDENT PLASTICITY?

Computer Simulations of Supervised Learning with STDP: Weight Modulations
SIMULATION STATION - GOALS

- Teach our STDP system to detect patterns within spike trains
SIMULATION STATION - GOALS
SIMULATION STATION - GOALS

- More specifically

- We’re trying to learn the target weight vector $w^*$ of a neuron
SIMULATION STATION - GOALS

- So how is this any different than what we’ve been doing so far??
SIMULATION STATION – THE SETUP

- We utilize STDP.

- We use the leaky integrate-and-fire neuron (We’ve already learned that)

- It receives its inputs from \( n = 100 \) dynamic synapses (and this is new)
SIMULATION STATION – THE SETUP

- We apply STDP to most Synapses (but not all)

- 90% of the Synapses function as excitatory synapses and STDP is applied to them

- 10% function as inhibitory Synapses – their parameters remain unchanged
Are the inhibitory synapses (we didn’t forget about those 10%)
What’s up with that?
Some parameters are set based on experimental biological data.

Maximal ability to change $w_{max}$ (chosen from the Gaussian distribution with mean 54 and SD 10.8, bounded by $54 \pm 3SD$).

90% and 10% - Excitatory / Inhibitory synapses.

There’s no consensus on how Inhibitory synapses change.
SIMULATION STATION – THE SETUP

- What's the target pattern we’re trying to detect?
SIMULATION STATION – THE SETUP

- We randomly selected one-half of the excitatory synapses and set their weights to the corresponding maximal efficacy $w_{\text{max}}$ (Gaussian).

- The weights of the other excitatory synapses were set to zero.
That’s nice… But that’s a setup, not a pattern

So let’s generate a pattern!
Using the target weight vector we’ve just setup two slides ago – we define a transformation $F$.

$F$ maps 100 input spike trains to one output spike train.

The spikes themselves are Poisson spike trains.
Are the inhibitory synapses (we didn’t forget about those 10%)
SIMULATION STATION – SHOCK THERAPY

Great. We have the model, the target pattern, the biological parameter picks – Let’s initialize the weights and START LEARNING!

we’re ready to go!
We replace the weights of all excitatory synapses by new, randomly chosen values according to a gamma distribution with mean 9 and standard deviation 6.3.
The theoretical analysis of section 4 had assumed that the neuron never fires during training except when it is supposed to fire.
In the subsequent computer simulations, the neuron received a strong depolarizing input when it was supposed to fire.

And a hyperpolarizing input, which prevented most (but not all) undesired firing, when it was not supposed to fire.
SIMULATION STATION – EXPERIMENT #1

Uncorrelated input
SIMULATION STATION – EXPERIMENT #1

- *Just a reminder on what we’re trying to do:*

- Run the data against the model with the parameters we learned

- Check the output spikes – Compare to the desired output spikes from the SETUP phase.
Are the inhibitory synapses (we didn’t forget about those 10%)
SIMULATION STATION – EXPERIMENT #1

And here are the results:
Angular:
Angle between the target and learned weight vectors

Weight deviation:
Difference between the target and learned weight vectors – normalized by the mean

Spike Correlation:
Turn previous slide into a State (0 or 1) through time, Smoothen (Spline-like) and look at the difference
SIMULATION STATION – EXPERIMENT #1

- 20 arbitrary target and trained synapses' weights
SIMULATION STATION – EXPERIMENT #2

The Noisy Teacher
(or – the Drunk Teacher)
In a realistic scenario of prediction learning, the predicted inputs are likely to have some timing jitter.
SIMULATION STATION – EXPERIMENT #2

- We therefore repeated experiment 1 with the timing of “teacher spikes” jittered by Gaussian noise with zero mean and SD 4 ms
SIMULATION STATION – EXPERIMENT #2

- Side Effects?

- Learning took considerably longer:
  
  - 65 ± 12 minutes convergence time until an angular error of ≤ 10 degrees was achieved for the case 100 input spike trains

  - (For 20 repetitions of the experiment)
SIMULATION STATION – EXPERIMENT #2

- **Spike Correlation and Angular error after 2 hours of training**

**Angular Correlation:**
Angle between the target and learned weight vectors

**Spike Correlation:**
Turn previous slide into a State (0 or 1) through time, Smoothen (Spline-like) and look at the difference
SIMULATION STATION – EXPERIMENT #2

- Results

- Training time needed until an angular error of less than 10 degrees is achieved.
SIMULATION STATION – EXPERIMENT #3

Correlated Input
SIMULATION STATION – EXPERIMENT #3

Are the inhibitory synapses (we didn’t forget about those 10%)
We have chosen the most difficult case:

- Target transformations $F$ that were generated by assigning within each of the 9 groups of the 10 excitatory synapses to 5 of them the weight 0 and to 5 of them their maximal weight value $w_{max}$

- (which was again chosen randomly for each synapse as in experiment 1).
Target transformations $F$ have to be learned that require different weights for highly correlated input spike trains.
SIMULATION STATION – EXPERIMENT #3

- (A) A typical target weight vector $w^*$ for experiment 3

- (B) Typical learned weight vector
(C) The result of experiment 4 with sharper correlation.

Dashed line: the spike correlation achieved by randomly drawn weight vectors (where half of the weights were set to $w_{max}$ and the other weights were set to 0).
Correlation Overdrive

Dependence of Learning Performance on Input Correlation
SIMULATION STATION – EXPERIMENT #4

- How does correlation affect things?
- How do we sharpen correlation?
SIMULATION STATION – EXPERIMENT #4

- How does correlation affect things?

- In order to make the effects of these correlated inputs more pronounced, the time constant $\tau_{cc}$ for the temporal decay of input correlations was reduced from 10 to 6 ms.

- Why?
Decay theory proposes that memory fades due to the mere passage of time.

(in reality, its controversial – and only applicable to short-term memory)
We repeat the above experiment for input correlations $cc = 0.1$, 0.2, 0.3, 0.4, and 0.5.
SIMULATION STATION – EXPERIMENT #4

- RESULTS!

![Graph showing correlation and weight error](image-url)
Predictions based on Theorem 2

The higher the weight error – the worse the STDP result

Some predicted to not be learnable situations, produced quite reasonable results – Why?

(Hint: Simplifications in the Proof)
CONCLUSIONS

- What can a spiking neuron learn with STDP?

- Any map $F$ from input to output spike trains that it could possibly implement in a stable manner
  
  *(This holds at least for uncorrelated and correlated Poisson input spike trains.)*

- STDP enables spiking neurons to learn to predict very complex temporal patterns
ありがとうございます
Thank You Very Much