脑启发人工智能导论
Introduction to Brain-Inspired Artificial Intelligence

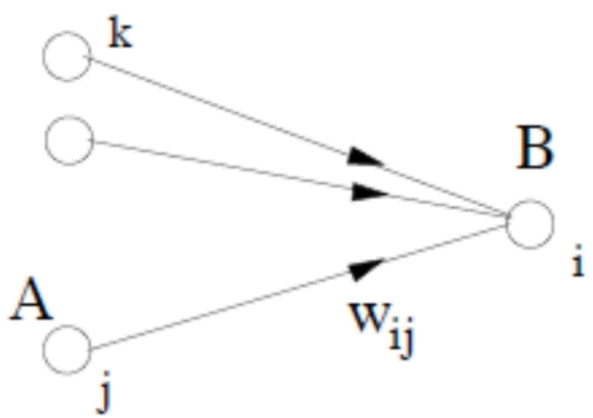
唐华锦 教授
浙江大学计算机学院
htang@zju.edu.cn
https://person.zju.edu.cn/htang

STDP与无监督学习

Hebb rule

When an axon of cell A is near enough to excite cell B or repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A’s efficiency, as one of the cells firing B, is increased.

Donald Hebb(1949)



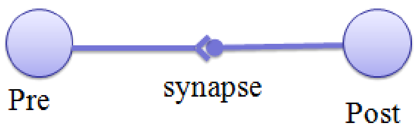
The change at synapse w_ij depends on the state of the presynaptic neuron j and the postsynaptic neuron i and the present efficacy w_ij , but not on the state of other neurons k.

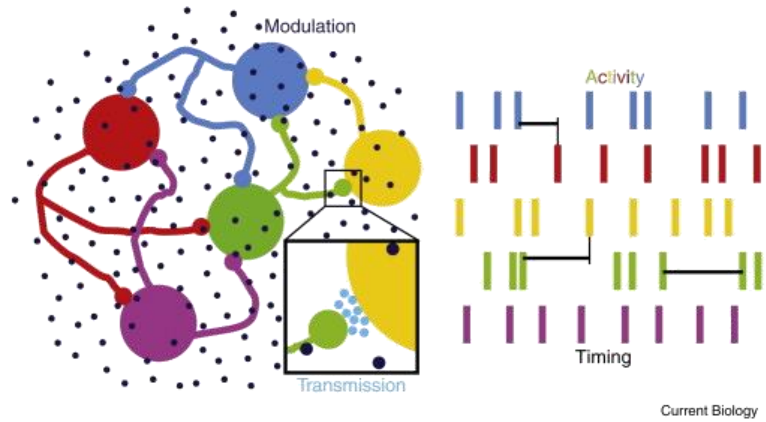
Hebb rule

- In the formal theory of neural networks the weight w_ij of a connection from neuron j to i is considered as a parameter that can be adjusted so as to optimize the performance of a network for a given task. The process of parameter adaptation is called learning and the procedure for adjusting the weights is referred to as a learning rule.

- A learning rule, viz., synaptic changes that are driven by correlated activity of pre- and postsynaptic neurons. This class of learning rule can be motivated by Hebb's principle and is therefore often called 'Hebbian learning'.

Wiring to Form A Network





- Knowledge resides in **synaptic weights**.
- Synapses play an important role in development, memory and learning of neural structures.
- Synaptic plasticity (learning) describes how changes in synaptic efficacy occur.

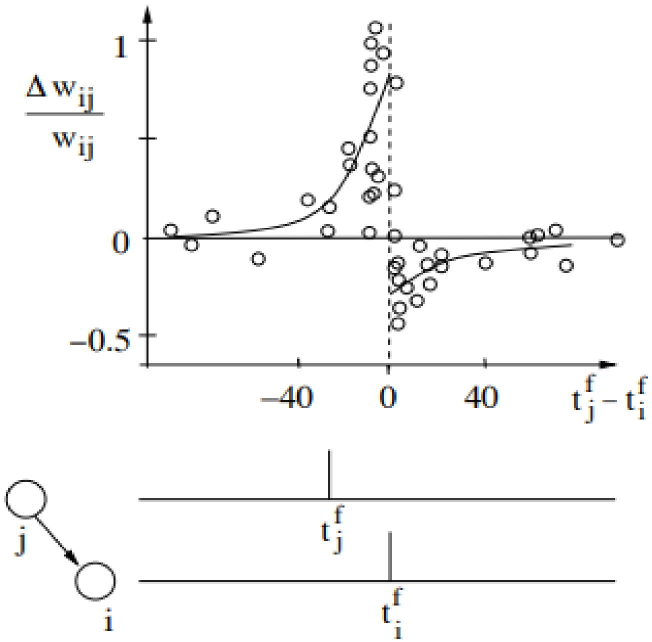
Temporal Aspects

- In 1998, Guo-qiang Bi and Mu-ming Poo found through experiments that the relative time of pre- and postsynaptic neurons firing determines the direction and degree of synaptic changes.

- Presynaptic neurons are repeatedly activated within the 20 millisecond time window before the activation of the postsynaptic neuron, resulting in long-term potentiation (LTP), while presynaptic neurons are repeatedly activated within the 20 millisecond time window after the activation of the postsynaptic neuron, resulting in long-term inhibition (LTD).

- The experimental results reflect the importance of precise firing time.

Temporal Aspects



Timing requirements between pre- and postsynaptic spikes. The circle is the experimental data, and the solid line is the curve fitting of the experimental data.

- When the spike of the presynaptic neuron is generated several milliseconds before the spike of the postsynaptic neuron, that is $t_j^{pre} < t_i^{post}$, the weight of the synapse connecting the two neurons becomes larger.

- On the contrary, the weight of the synapse connecting the two neurons becomes smaller when $t_j^{pre} > t_i^{post}$.

STDP

- Spike-Timing-Dependent Plasticity (STDP) learning rule was proposed by Henry Markram, and it is the most commonly used synaptic plasticity rule that has been verified by experiments.
- STDP can be considered as a temporally precise form of Hebbian synaptic plasticity, induced by isolated spikes in pre- and postsynaptic neurons.
- Based on the correlation between the firing time of pre-synaptic and post-synaptic neurons, the strength of the connections between neurons is adjusted according to the sequence of nerve firing spikes.

STDP

When the pre-synaptic neuron fired before the post-synaptic neuron, it will cause long-term potentiation (LTP); when the pre-synaptic neuron pulse is fired after the post-synaptic neuron, it will cause long-term inhibition (LTD).

For a neuron i:

- If neuron j transmits information before it reacts, the connection G(j→i) between neuron i and neuron j will strengthen
- If neuron j transmits information after it reacts, the connection G(j→i) between neuron i and neuron j will be weakened

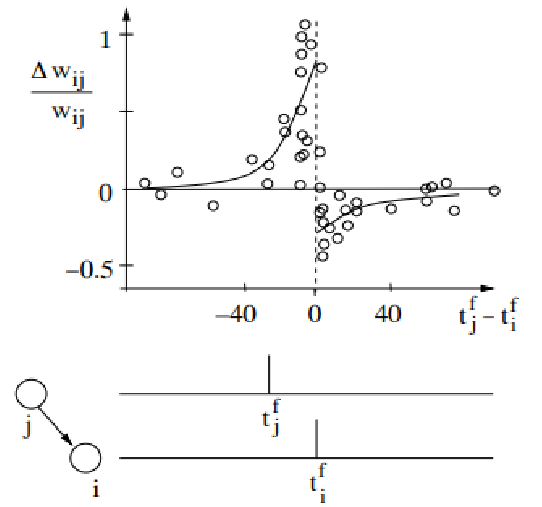
STDP

STDP weight update rules:

$$\Delta w(s) = \begin{cases} A_{+} exp(s/\tau_1) & for \;\; s < 0, \\ -A_{-} exp(-s/\tau_2) & for \;\; s > 0, \end{cases}$$

$s = t_j^{pre} - t_i^{post}$ is the difference between the arrival time of the pre-synaptic spike j and the firing time of the post-synaptic spike i

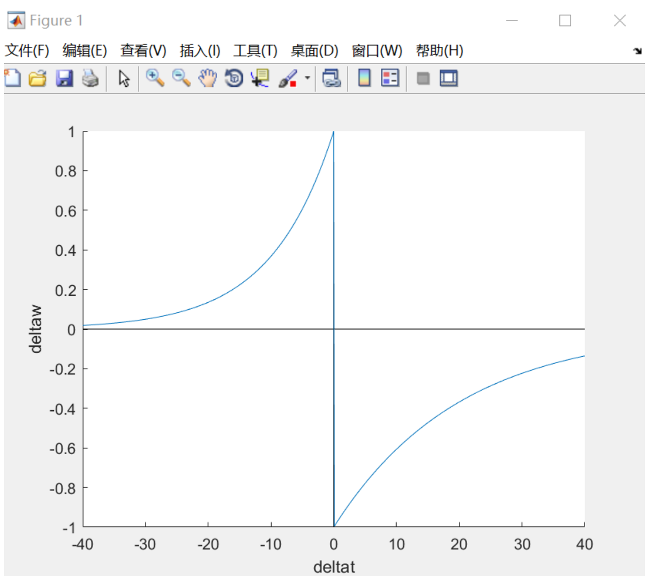
A_{+} , A_{-} , τ_1 , τ_2 are constants



STDP

$$\Delta w(s) = \begin{cases} A_+ exp(s/\tau_1) & for\ \ s < 0, \\ -A_- exp(-s/\tau_2) & for\ \ s > 0, \end{cases}$$

$$A_+ = A_- = 1, \qquad \tau_1 = 10ms, \tau_2 = 20ms, \qquad dt = 0.1ms$$



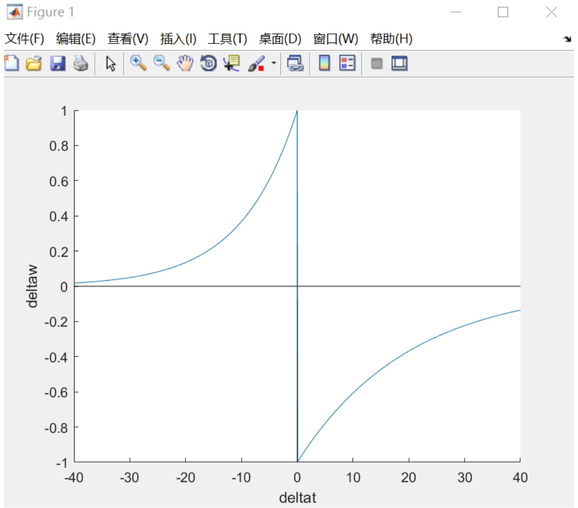
The relationship between s and $\Delta w(s)$

The weight change value depends on the time interval between two events. The closer the pre-synaptic and post-synaptic spike events, the smaller the time interval, the greater the changed value; on the contrary, the smaller the changed value.

STDP

```
clear;clc;
T=[-40:0.1:40]; % ms
A1 = 1;
A2 = 1;
t1 = 10; % ms
t2 = 20; % ms
delta_w = zeros(size(T));
[p,n] = size(T);
for i=1:n
    t = T(i);
    if t<0
        delta_w(i)=A1*exp(1.0*t/t1);
    else
        delta_w(i)=-A2*exp(-1.0*t/t2);
    end
end
```

```
figure(1);
line([-40 40], [0 0], 'color', 'k');
hold on;
line([0 0], [-1, 1], 'color', 'k');
hold on;
plot(T, delta_w);
xlabel('deltat');
ylabel('deltaw');
```



基于STDP无监督学习的应用（1）：
目标识别

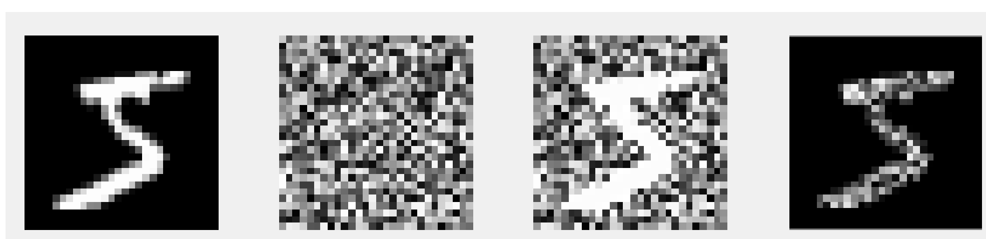
Task1: STDP

Input:

A MNIST image.

Output:

Visualization of input image, initial weight, trained weight, weight difference between the initial weight and the trained weight.



Parameter initialization:

```
clear;clc;
load('img.mat');

dt = 1;
T = 100;
iter = 12;
in_num = 784;
out_num = 1;

w = rand(in_num, out_num);
spikeTrains = zeros(in_num, T);
```

The structure is a fully connected network of 784*1

Main process:

```
for ite=1:iter
    spike_output = [];
    v = 0; % v of postsynaptic neuron
    ref = 0;
    for t = dt:dt:T
        spikeTrains(:, t) = possionCode(img);
        [v, spike_output, ref] = forward(spikeTrains(:, t),
spike_output, v, w, t, ref);
    end
    w = STDP(spikeTrains, spike_output, w, dt, T);
end
```

```
function [v, spike_output, ref] = forward(spikeTrain,
spike_output, v, w, t, ref)
    v_r = 0;
    v_th = 15;
    ref_time = 5;
    if ref<=0
        v = v + sum(w(find(spikeTrain == 1)));
    end
    ref = max(0, ref - 1);
    if v>v_th
        v = v_r;
        ref = ref_time;
        spike_output = [spike_output, t];
    end
end
```

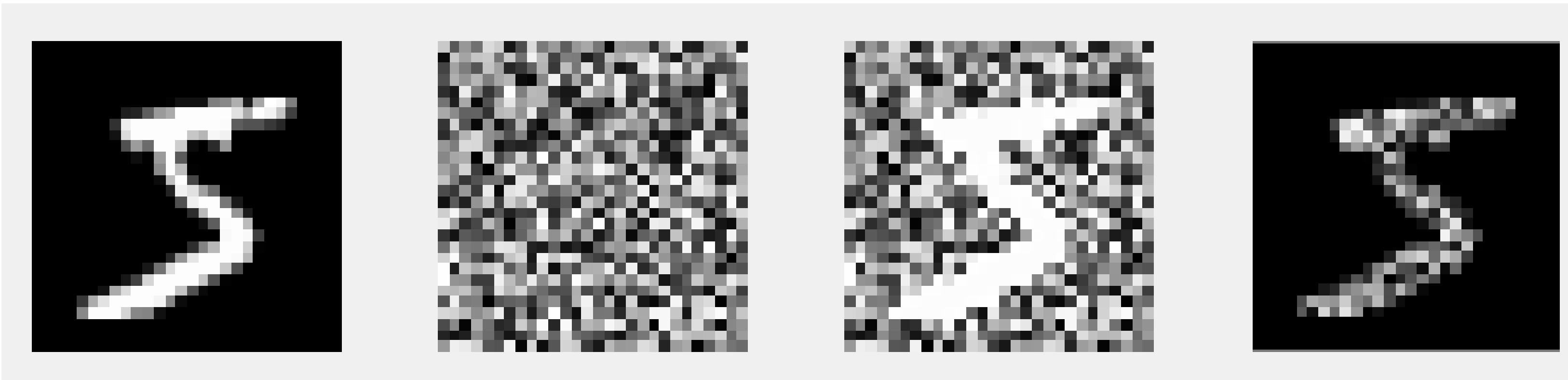
```matlab
function w = STDP(spikeTrains, spike_output, w, dt, T)
    A1 = 1; A2 = 1; tau1 = 20; tau2 = tau1/2; eta = 0.001;
    for t_pre = dt:dt:T
        spikeTrain = spikeTrains(:, t_pre);
        ind = find(spikeTrain == 1);
        for i = 1:length(spike_output)
            t_post = spike_output(i);
            s = t_pre - t_post;
            delta_w = 0;
            if s<0
                delta_w = A1 * exp(s / tau1);
            end
            if s>0
                delta_w = -A2 * exp(-s / tau2);
            end
            w(ind) = w(ind) + eta * delta_w;
            w = checkBound(w, 0, 1);
        end
    end
end
```

STDP in MNIST

```
function spikeTrain = possionCode(img)
    spikeTrain = img > (rand(1, 784)*255);
End
```

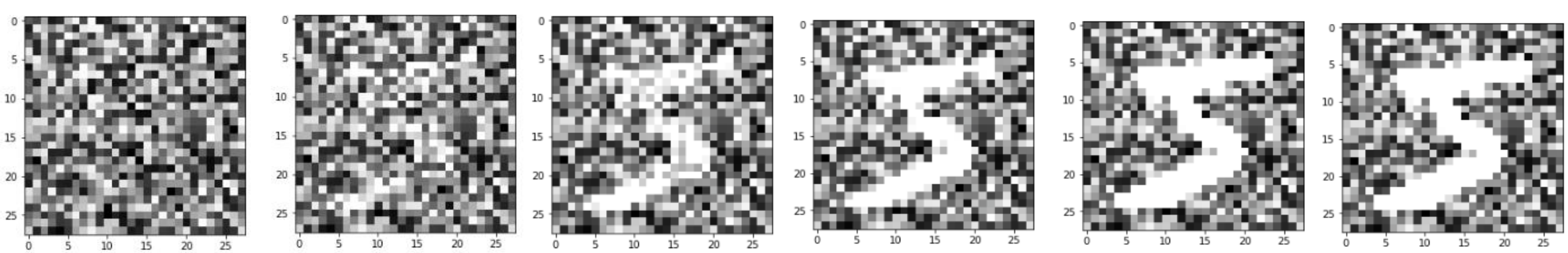
```
function w = checkBound(w, wmin, wmax)
    w(find(w<wmin)) = wmin;
    w(find(w>wmax)) = wmax;
end
```

STDP For MNIST Recognition



input initial weight trained weight weight difference

The following figure shows the change process of the training weight matrix



基于STDP无监督学习的应用（2）：目标检测

STDP finds the start of the repeating pattern

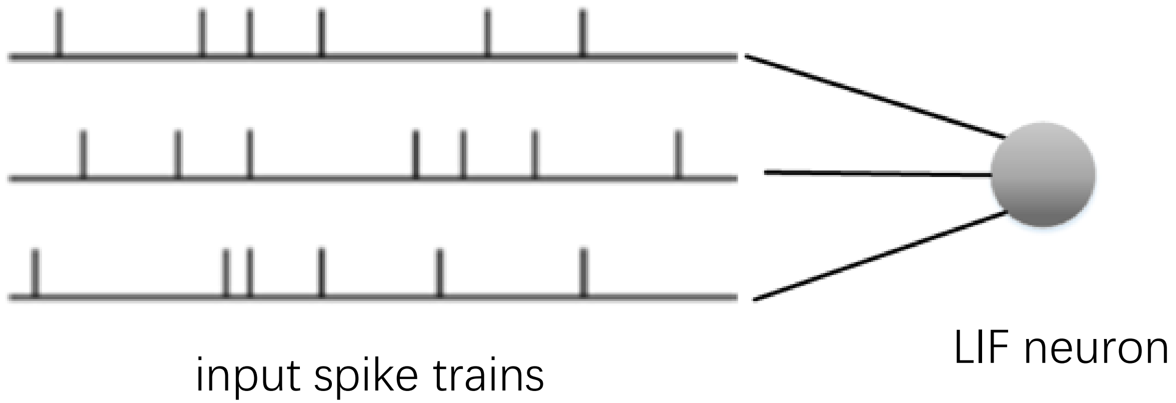
A repeating pattern hidden in an equally dense interfering spike trains can be detected and learned by a single neuron equipped with spike time-dependent plasticity (STDP).

When a neuron is presented successively with discrete volleys of input spikes STDP has been shown to learn 'early spike patterns', that is to concentrate synaptic weights on afferents that consistently fire early, with the result that the postsynaptic spike latency decreases, until it reaches a minimal and stable value.

Timothée, M., Rudy, G., & Thorpe, S. J. (2008). Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains. Plos One, 3(1), e1377.

STDP finds the start of the repeating pattern

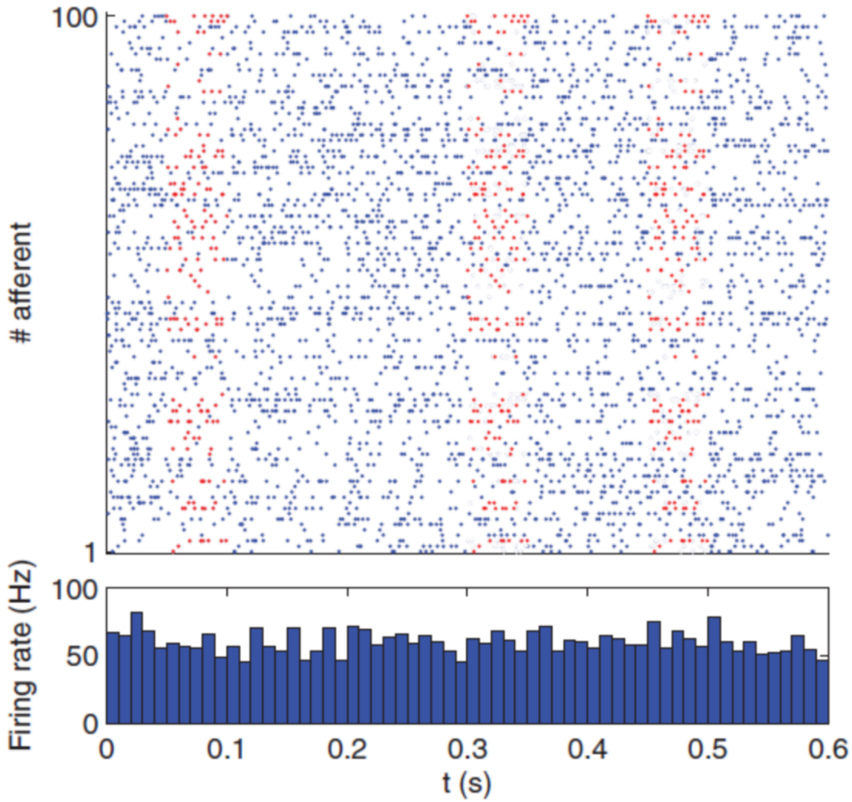
Structure diagram of a spiking neural network that recognizes a repetitive pattern



A LIF neuron is connected to 2000 afferent neurons (only three are shown in the figure). The initial value of all synaptic weights is equal to 0.475

Timothée, M., Rudy, G., & Thorpe, S. J. (2008). Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains. Plos One, 3(1), e1377.

STDP finds the start of the repeating pattern



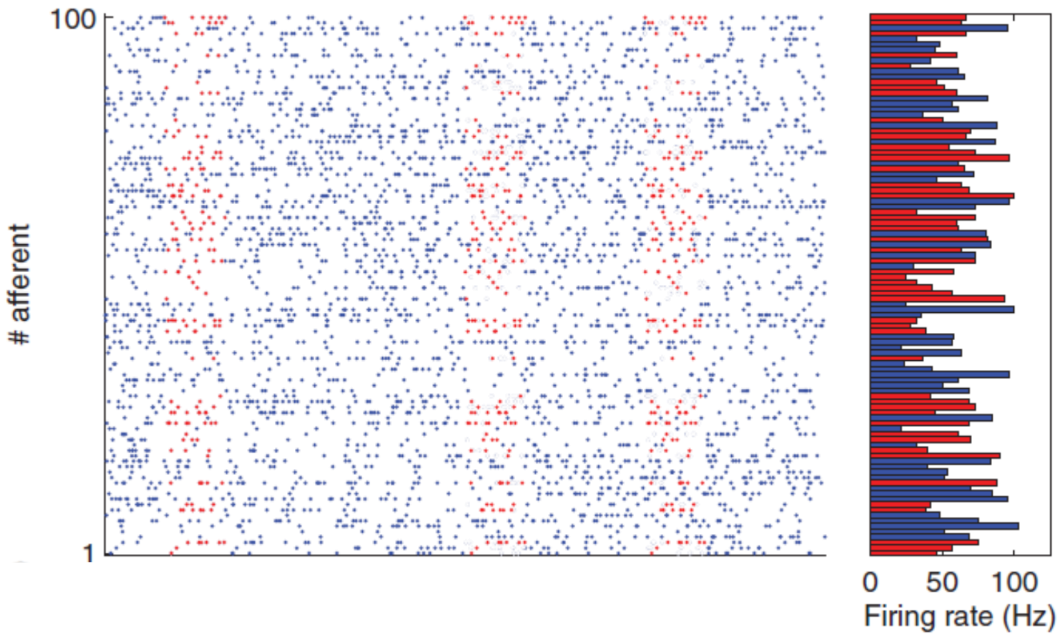
- The left panel shows in red a repeating 50 ms long pattern that concerns 50 afferents among 100. Neurons involved in the pattern are shown in red. Blue represents background noise, which is generated by a Poisson process with a variable instantaneous firing rate (varying between 0 and 90 Hz).

- The bottom panel plots the population-averaged firing rates over 10 ms time bins (equal to the time constant of the spiking neuron).

Timothée, M., Rudy, G., & Thorpe, S. J. (2008). Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains. Plos One, 3(1), e1377.

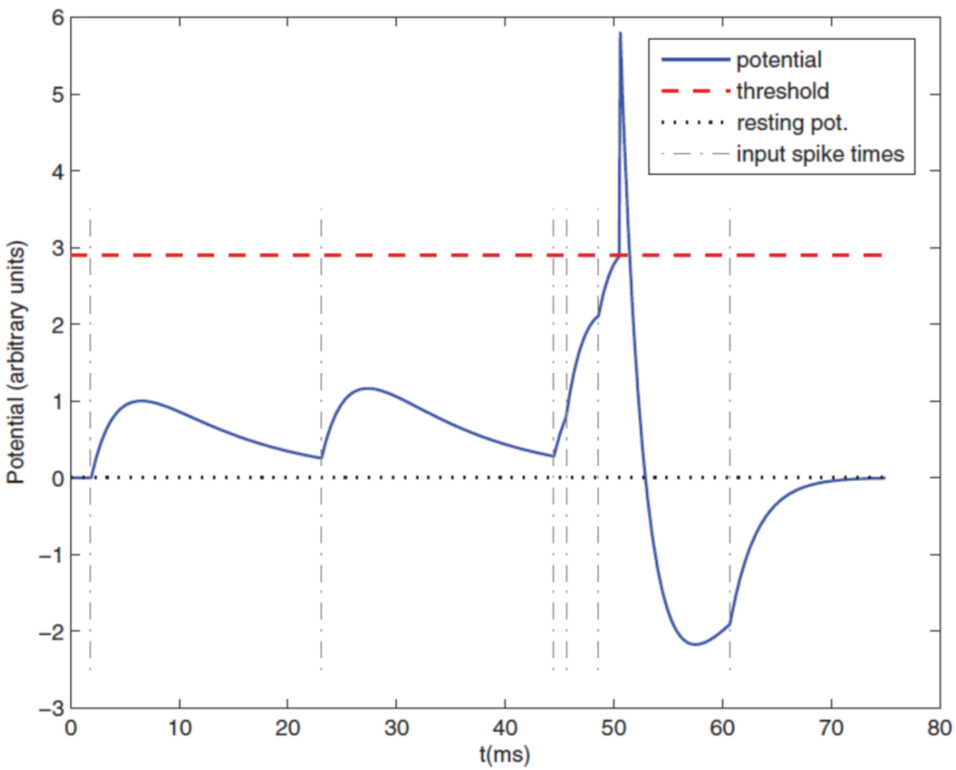
STDP finds the start of the repeating pattern

Spike patterns with background noise



- The right panel plots the individual firing rates averaged over the whole period. Neurons involved in the pattern are shown in red.
- Again, nothing characterizes them in terms of firing rates. Detecting the pattern thus requires taking the spike times into account.

STDP finds the start of the repeating pattern

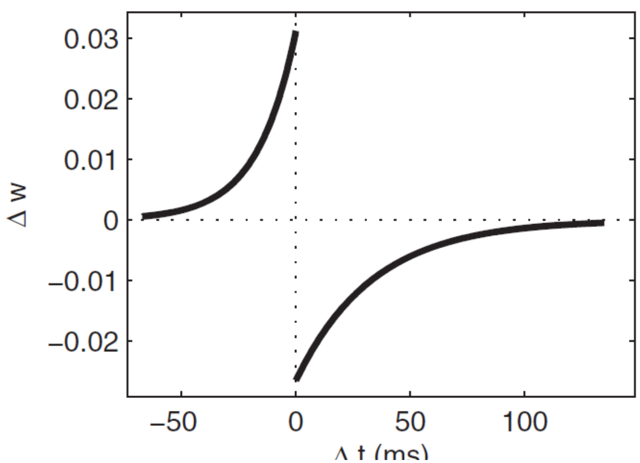


- Here is an illustrative example with only 6 input spikes. The graph plots the membrane potential as a function of time, and clearly demonstrates the effects of the 6 corresponding Excitatory PostSynaptic Potentials (EPSP).

- Because of the leak, for the threshold to be reached the input spikes need to be nearly synchronous. The LIF neuron is thus acting as a coincidence detector. When the threshold is reached, a postsynaptic spike is fired. This is followed by a refractory period of 1 ms and a negative spike-afterpotential.

Timothée, M., Rudy, G., & Thorpe, S. J. (2008). Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains. Plos One, 3(1), e1377.

STDP finds the start of the repeating pattern



The STDP modification function. The additive weight updates are modled as a function of the difference between the presynaptic spike time and the postsynaptic one. We used an exponential law. The left part corresponds to Long Term Potentiation (LTP) and the right part to Long Term Depression (LTD).

$$\Delta w_j = \begin{cases} a^+ \cdot \exp\left(\frac{t_j - t_i}{\tau^+}\right) & \text{if} \quad t_j \leq t_i \quad \text{(LTP)} \\ -a^- \cdot \exp\left(-\frac{t_j - t_i}{\tau^-}\right) & \text{if} \quad t_j > t_i \quad \text{(LTD)} \end{cases}$$

The time constants τ^+ = 16.8 ms and τ^- = 33.7 ms

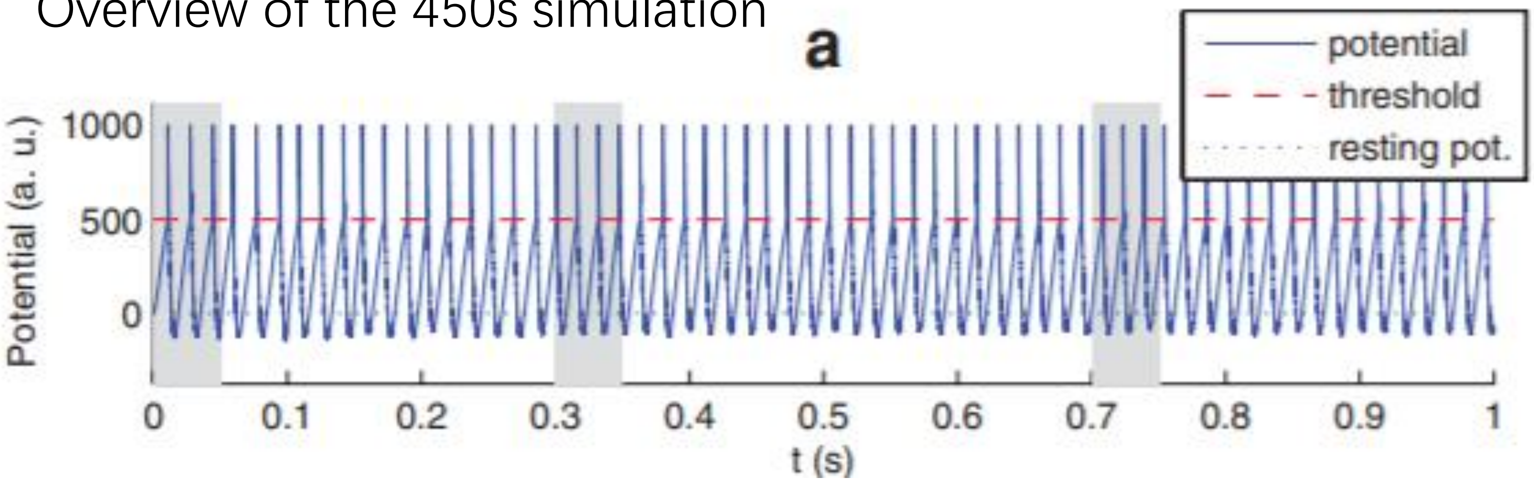
a^+ = 0.03125 and a^- = 0.85 a^+

We restricted the learning window to $[t_i - 7 \cdot \tau^+, t_i]$ for LTP and to $[t_i, t_i + 7 \cdot \tau^-]$ for LTD.

Timothée, M., Rudy, G., & Thorpe, S. J. (2008). Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains. Plos One, 3(1), e1377.

Simulation Results

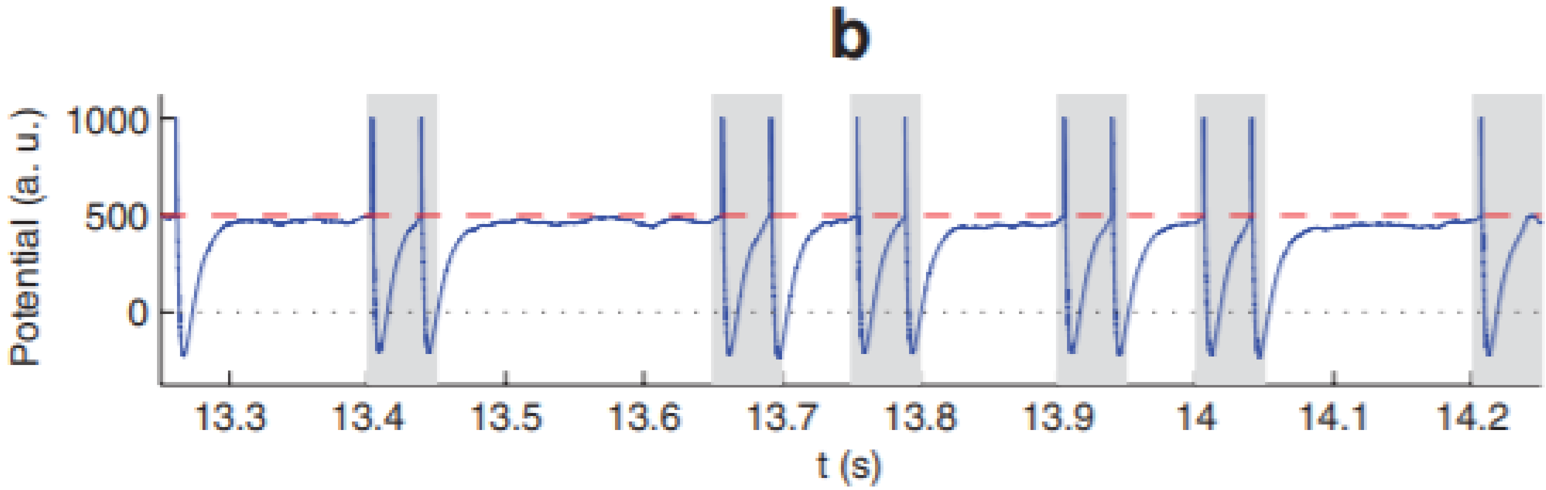
Overview of the 450s simulation



- At the beginning of the simulation the neuron is non-selective because the synaptic weights are all equal.

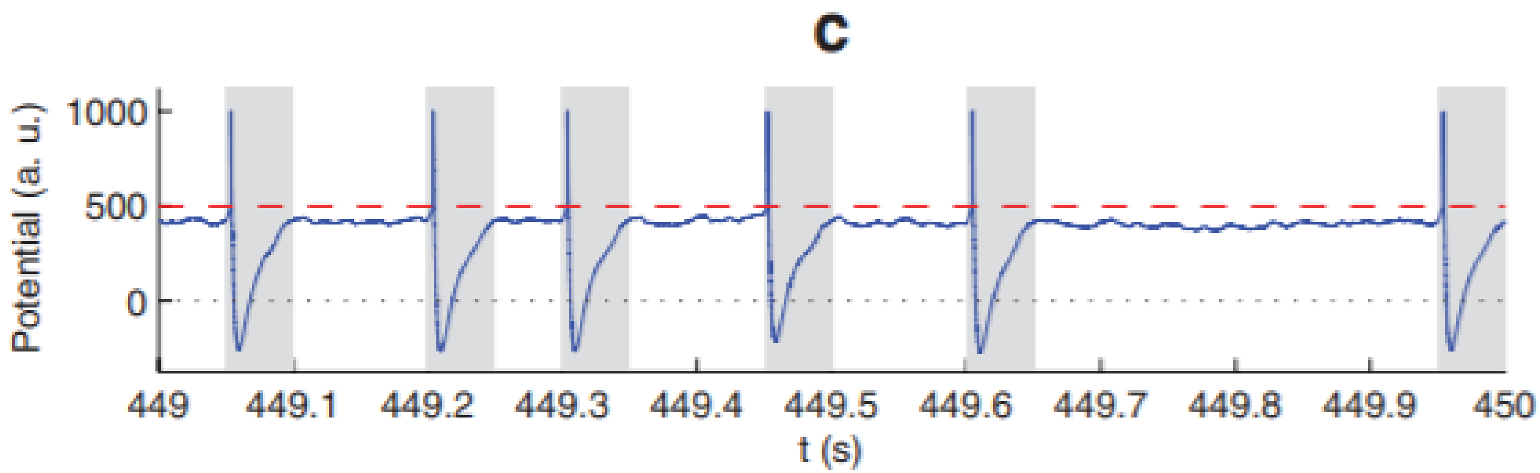
- It thus fires periodically, both inside and outside the pattern. (grey rectangles indicate repeated input patterns)

STDP finds the start of the repeating pattern



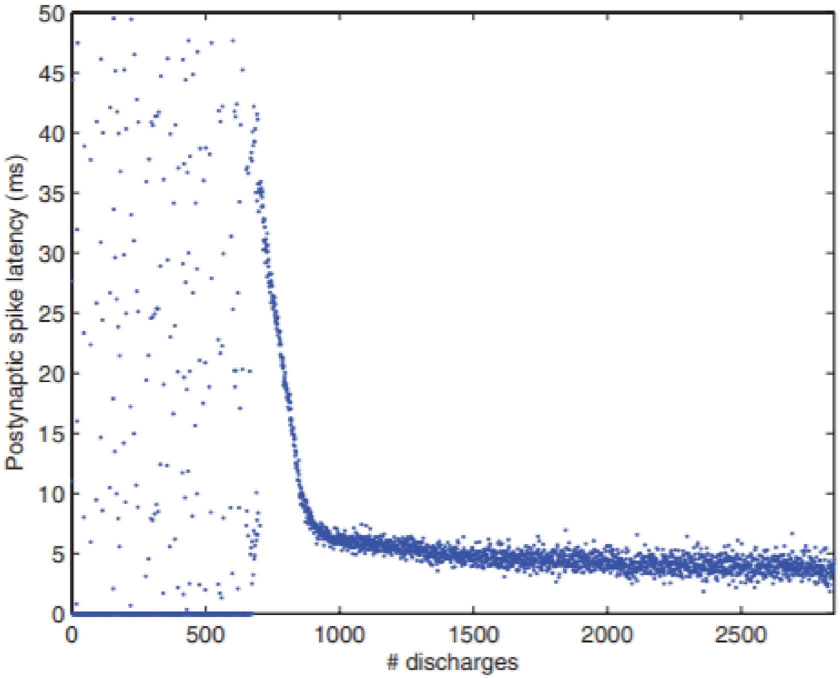
- By reinforcing the synaptic connections with the afferents that took part in firing the neuron, STDP increases the probability that the neuron fires again next time the pattern is presented (reinforcement of causality link) .
- At t<13.5s, after about 70 pattern presentations and 700 discharges, selectivity to the pattern is emerging: gradually the neuron almost stops discharging outside the pattern (no false alarms), while it does discharge most of the time the pattern is present (high hit rate), here even twice.

STDP finds the start of the repeating pattern

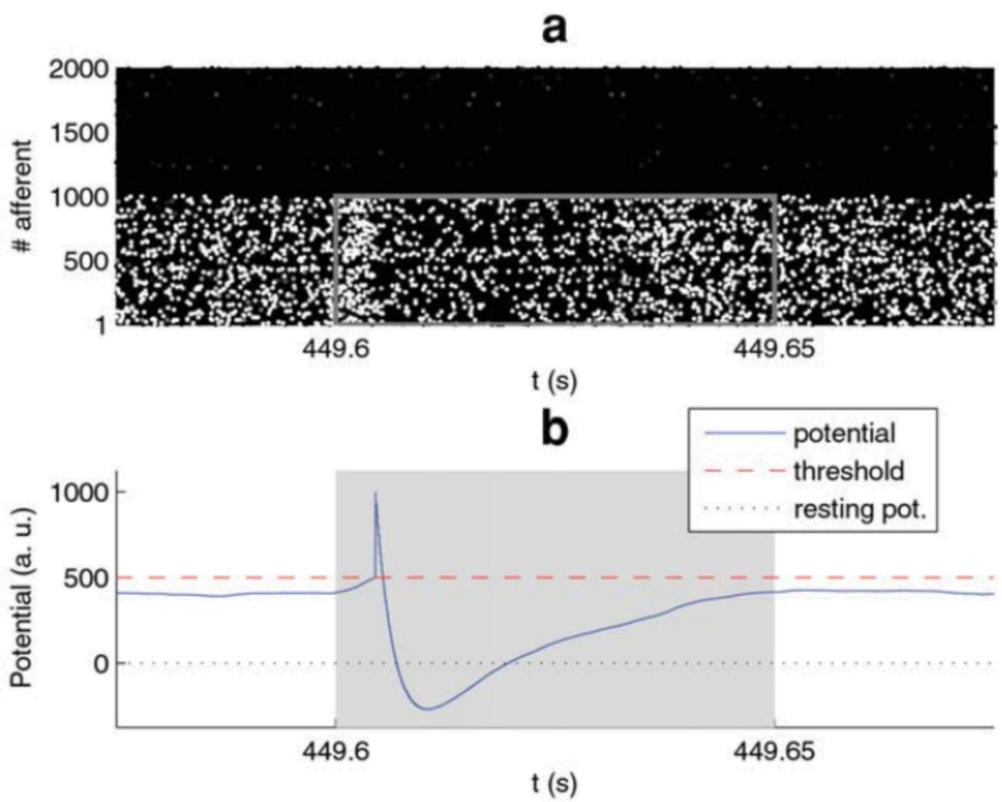


- Once selectivity to the pattern has emerged, **STDP has another major effect:** Each time the neuron discharges in the pattern, it reinforces the connections with the presynaptic neurons that fired slightly before in the pattern. As a result, next time the pattern is presented the neuron is not only more likely to discharge to it, but **it will also tend to discharge earlier**.
- In other words, the postsynaptic spike latency locks itself to the pattern and decreases steadily (with respect to the beginning of the pattern).
- However, it cannot decrease endlessly. There is a convergence by saturation when all the spikes in the pattern that precede the postsynaptic spike already correspond to maximally potentiated synapses, and all are necessary to reach the threshold.

Postsynaptic spike latency varies with the number of postsynaptic spikes



- At the beginning, during the appearance of the first 700 postsynaptic spikes, the neuron is not selective (the post-synaptic spike latency is 0 which means firing outside the pattern).
- In the middle, after the appearance of 700 postsynaptic spikes, selectivity to the pattern is emerging. STDP increases the weight of afferent neurons that contribute to the output spikes, leading to the advancement of the firing time of postsynaptic neurons, that is, the reduction of latency.
- Finally, the postsynaptic latency tends to a stable value, generating a fast and reliable pattern detector.



Convergence in the final stage. Reorder the afferents so that afferents 1–1000 are involved in the pattern, and afferents 1001–2000 are not.

- STDP enhanced most of the synapses corresponding to the earliest spikes in the pattern, and the synaptic connections of afferent neurons not involved in the pattern were completely inhibited.

- Use a color code ranging from black for spikes that correspond to completely depressed synapses (weight = 0) to white for spikes that correspond to maximally potentiated synapses (weight = 1).

基于STDP无监督学习的应用（3）：
多目标检测

Competitive Learning through STDP

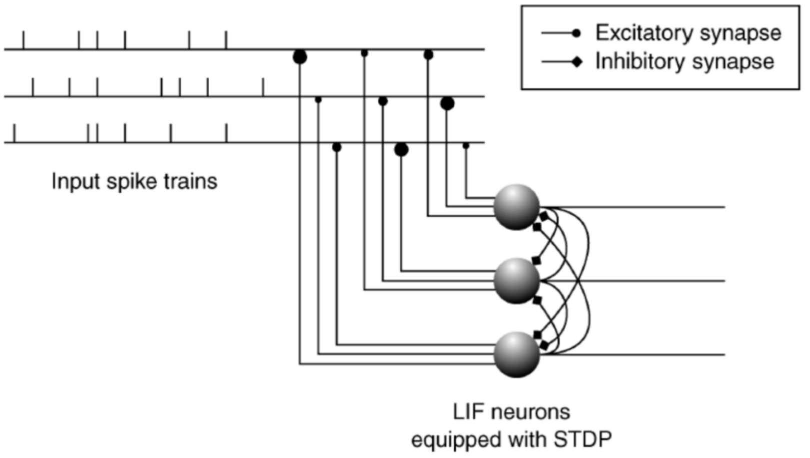
- A single neuron cannot recognize multiple repetitive patterns. As learning progresses, postsynaptic neurons become more and more selective, making it unable to respond to several patterns. To learn other patterns, other neurons are needed.

- Multiple neurons equipped with STDP can "monitor" input spike trains with multiple repeating patterns. There is a competitive mechanism among different output neurons, which can learn more than one repeating pattern present in the inputs or the different parts of a long pattern. This mechanism is implemented through inhibitory horizontal connections between neurons, such that as soon as one neuron fires, it could prevent other cells from learning the same pattern.

- This can be called **one-winner-take-all** mechanism.

Masquelier, T., Guyonneau, R., & Thorpe, S. J. (2009). Competitive stdp-based spike pattern learning. Neural Computation, 21(5), 1259.

Competitive Learning through STDP



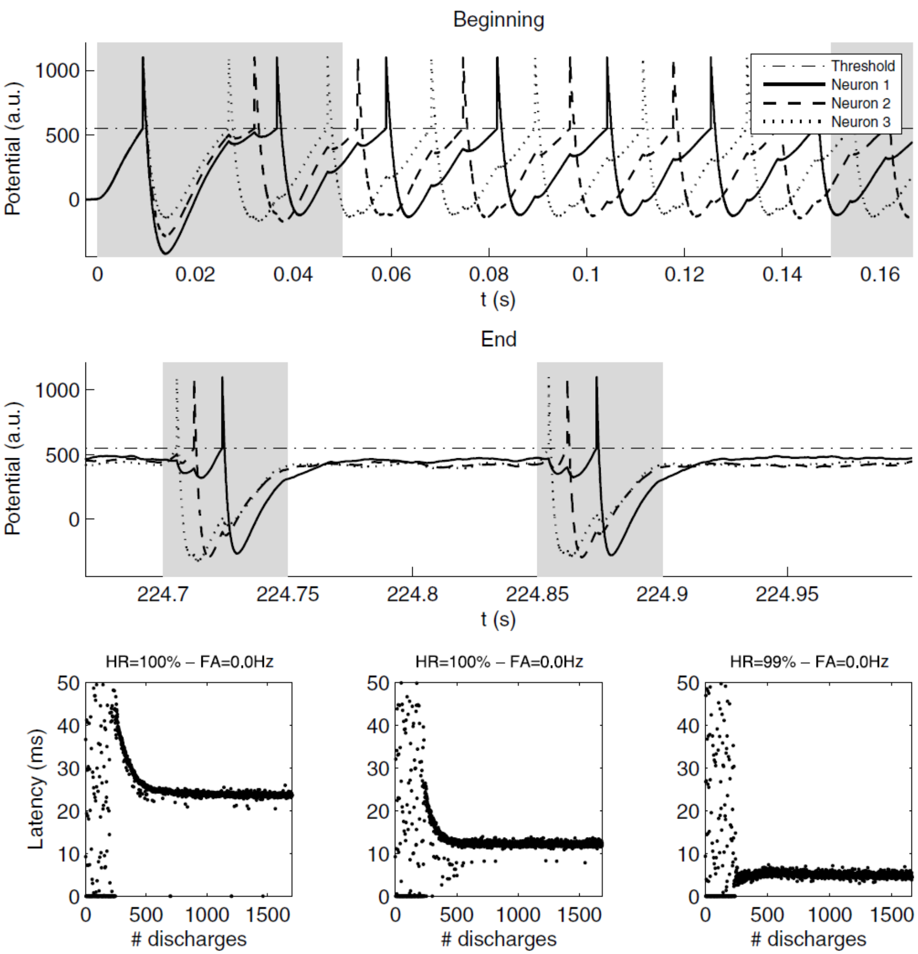
The architecture diagram of the network that recognizes multiple repetitive patterns. Several（here three） LIF neurons are connected to 2000 afferent neurons (only three are shown in the figure). Initialize synapse weights randomly (subject to equal probability distribution in the range of [0,1]).

- Lateral inhibitory connections are set up between them, so that as soon as a neuron fires, it sends a strong IPSP to its neighbors. This is a biologically plausible way to implement a one-winner-take-all mechanism.
- When detecting repetitive patterns, inhibition prevents neurons from learning the same parts.
- Unlike the excitatory connections, the inhibitory connections are hard-wired and not plastic.

Masquelier, T., Guyonneau, R., & Thorpe, S. J. (2009). Competitive stdp-based spike pattern learning. Neural Computation, 21(5), 1259.

Competitive Learning through STDP

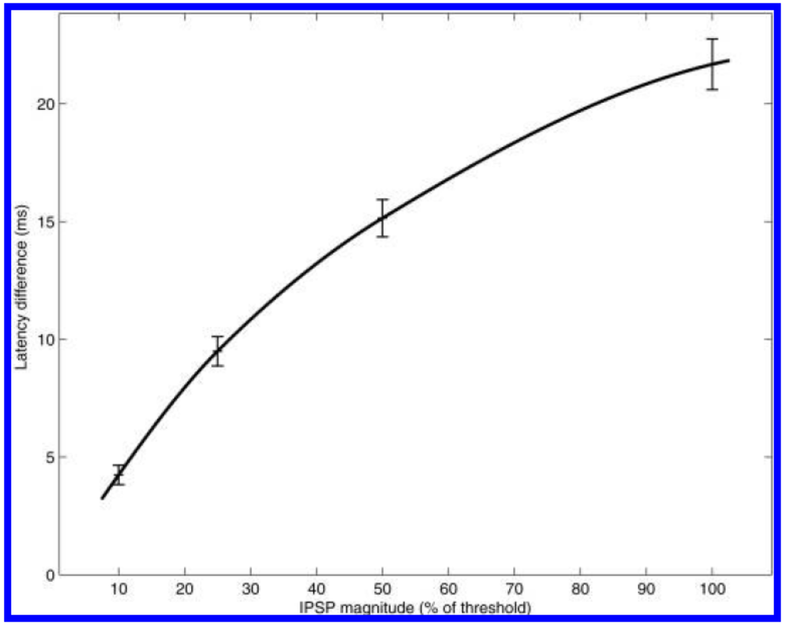
1. One Pattern, *n* Neurons



The top and middle panels plot the membrane potential of the three neurons (arbitrary units), as a function of time. Peaks correspond to postsynaptic spikes and are followed by a negative spike afterpotential. Notice that each time a neuron fires, it sends an IPSP to the two others. This tends to prevent neurons from firing too close to each other in time.

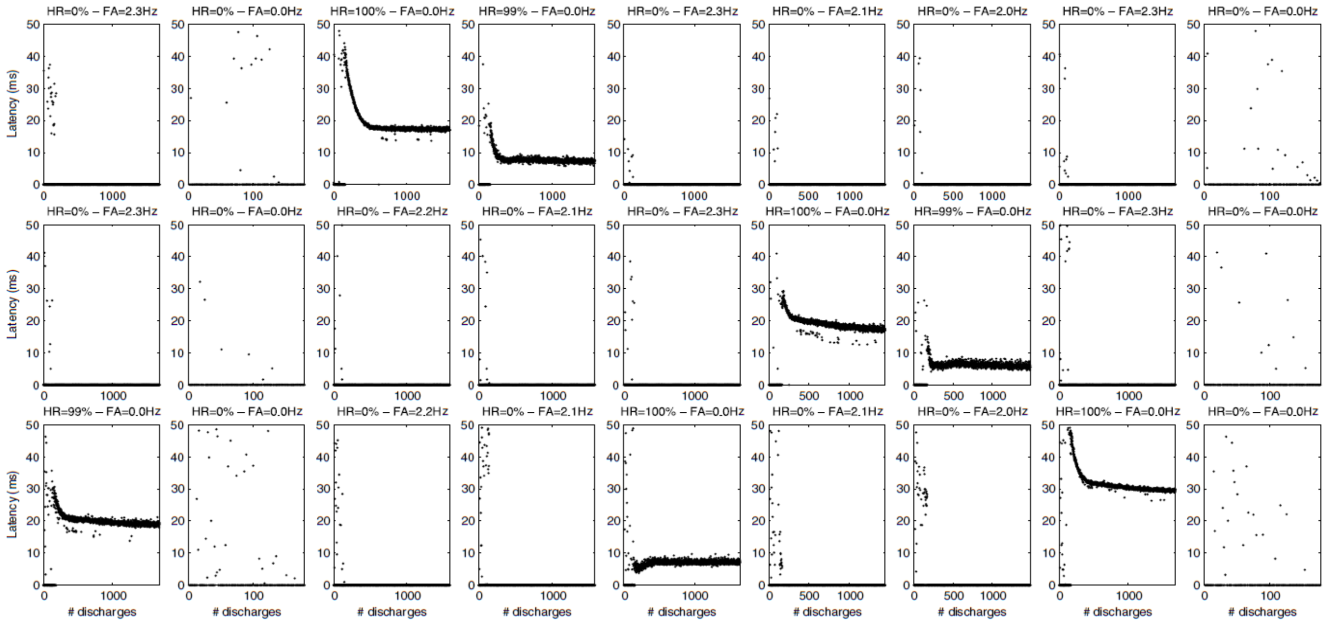
- (Top) Beginning of the simulation. The neurons are not selective. They fire pseudoperiodically, both inside and outside the pattern (shown in gray) indifferently.
- (Middle) End of the simulation. Each neuron has become selective to a different part of the pattern (i.e., has a different latency with respect to the beginning of the pattern).
- (Bottom) Postsynaptic spike latencies of the three neurons during the learning process.

Competitive Learning through STDP



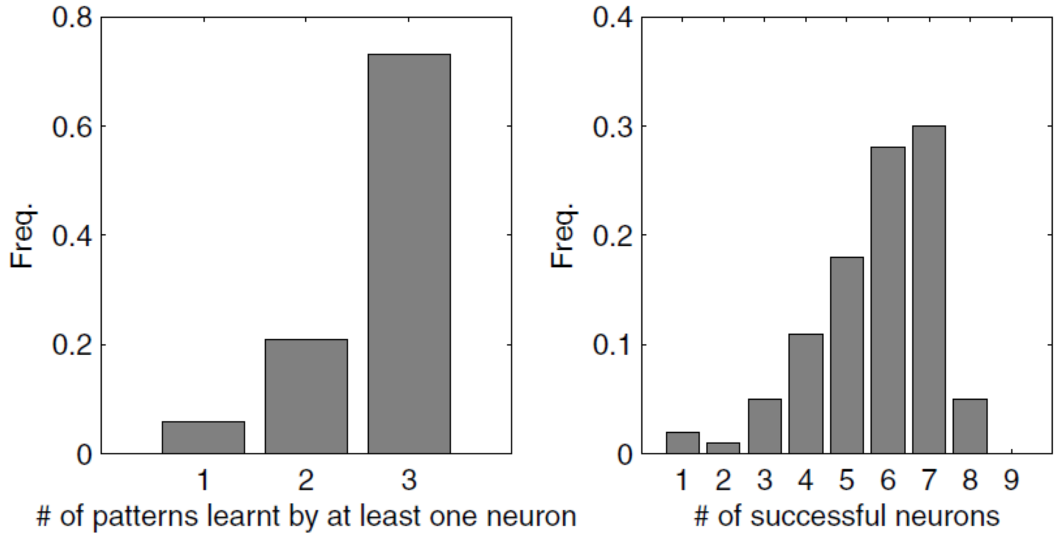
Effect of inhibition on latency differences. Here we plotted the mean latency difference between two successive successful neurons (criteria: hit rate above 90%, false alarm rate below 1 Hz) for various inhibition strengths (expressed as the magnitude of the IPSP with respect to the neuron's threshold). As expected, **the stronger the inhibition, the longer the intervals between two successive firings**.

2. *n* Patterns, *m* Neurons



Latencies with multiple patterns. Here we show a typical result with three patterns (shown on each column) and nine neurons (in rows). We plotted the latencies for each neuron with respect to the beginning of each pattern. Again the latency is said to be zero if the neuron fired outside a pattern (i.e., generated a false alarm). For instance, it can be seen that neuron 1 became selective to pattern 3 and ends up with a latency of 19 ms, a hit rate of rate of 99%, and no false alarms (from the point of view of patterns 1 and 2, it generates only false alarms). Pattern 1 has also been learned by neurons 5 and 8 with latencies of, respectively, 7 and 30 ms, with no false alarms and a hit rate of 100%. Neurons 3 and 4 learned pattern 1. Neurons 6 and 7 learned pattern 2. Neurons 2 and 9 "died"—stopped firing after too many (about 180) discharges outside the patterns, causing too much depression.

(2) *n* Patterns, *m* Neurons

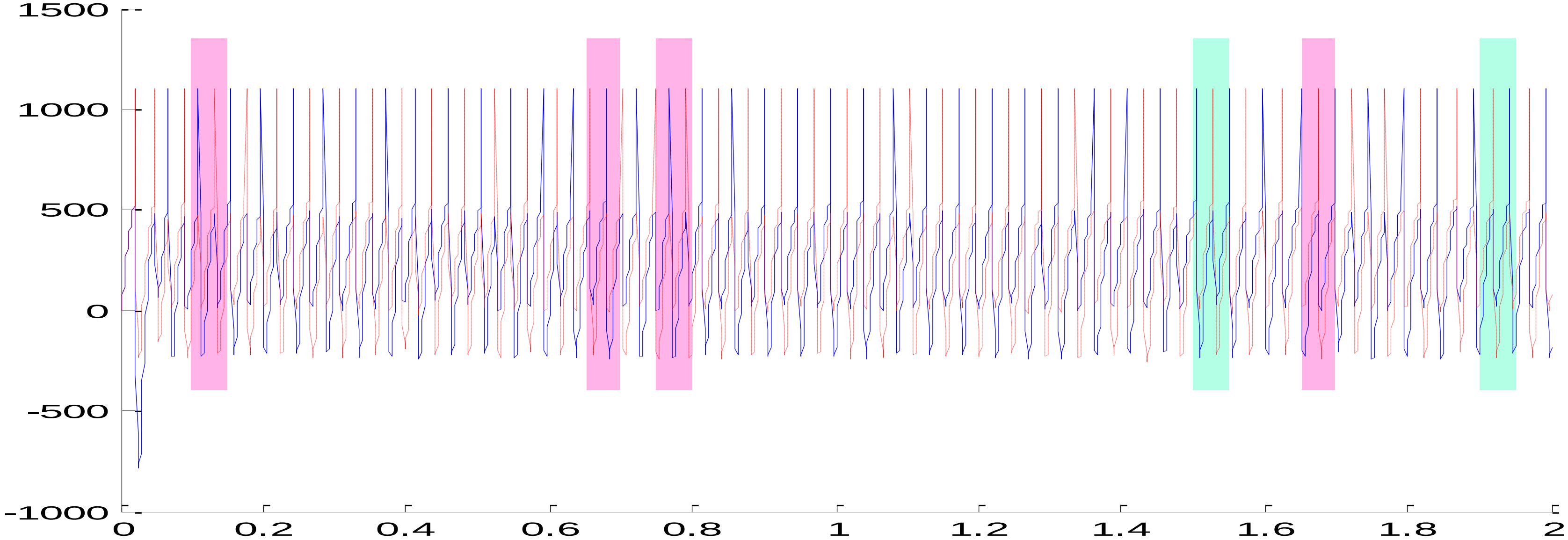


Statistics over the 100 runs. In more than two-thirds of the cases, each of the three patterns was learned by at least one neuron.

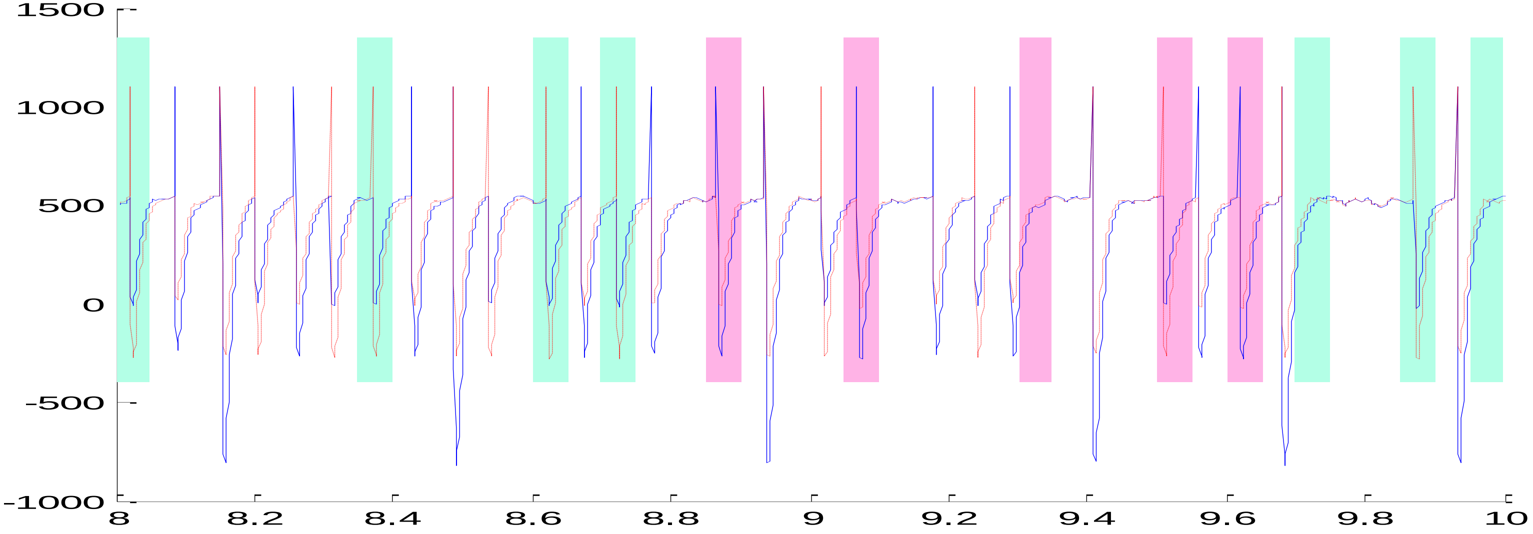
Performance with multiple patterns. The left panel plots the distribution of the number of patterns learned by at least one neuron across the 100 simulations. In more than two-thirds of the cases, all three patterns were learned by at least one neuron. The right panel plots the distribution of the number of successful neurons (criteria: hit rate above 90%, false alarm rate below 1 Hz). The mean is 5.71/9 = 63%.

Simulation Results

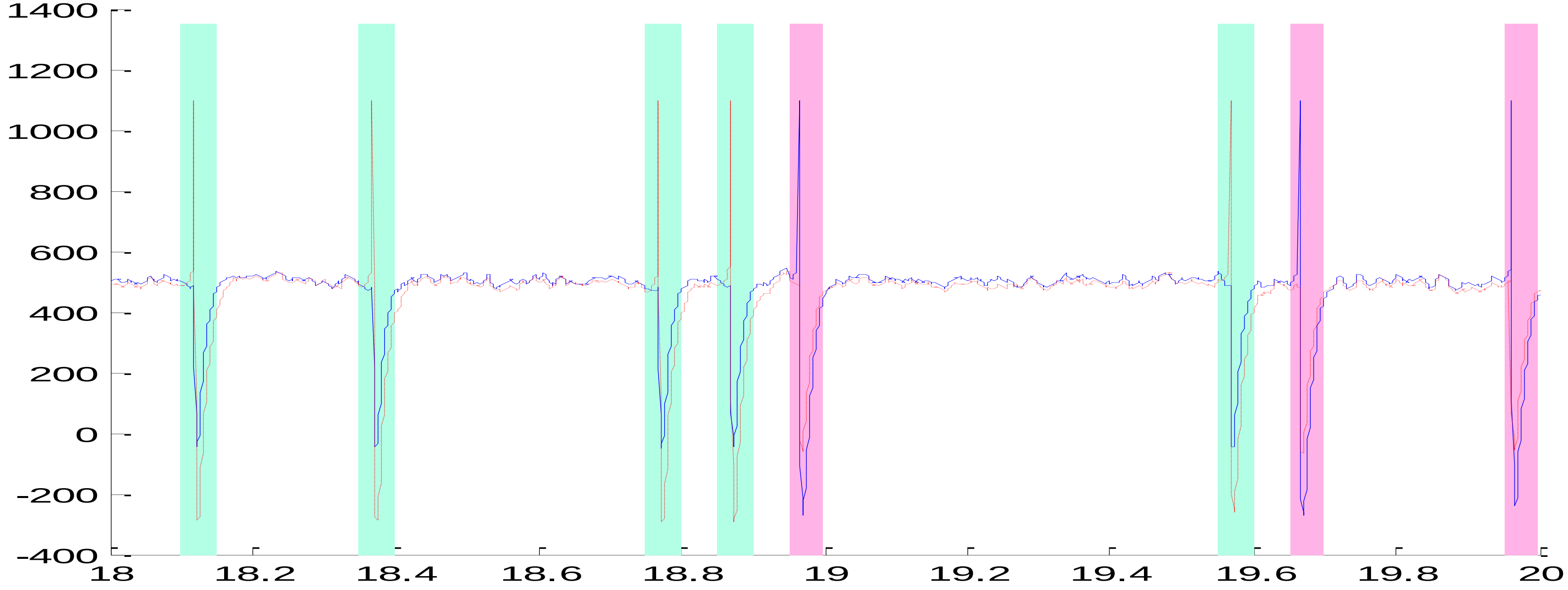
two neurons learn two patterns



Beginning: No neuron is selective.



- Selectivity gradually appears, and different neurons respond to different patterns.
- At this time, there are still firings outside of the pattern, and need to continue learning.

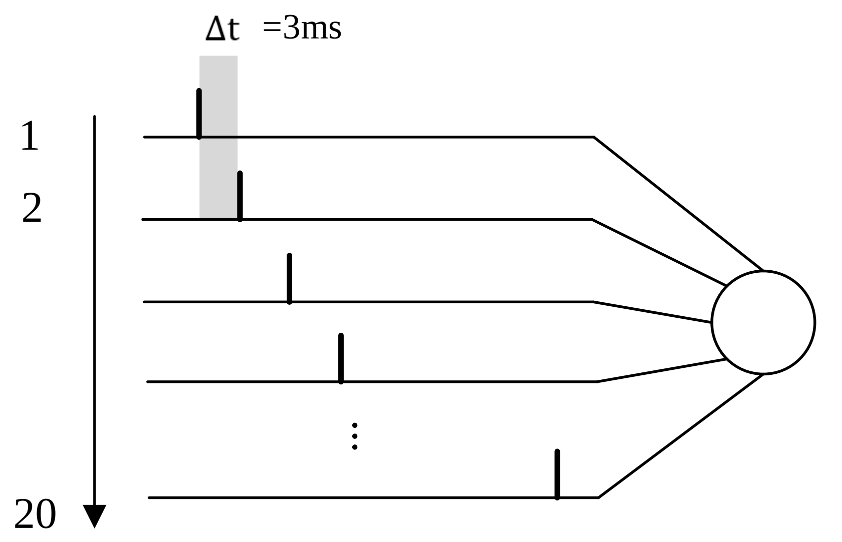


After convergence, reach a steady state and each neuron learns a pattern

STDP的其他特性

STDP: Learning to be Fast

- 用STDP规则对权重进行训练可以**使得突触后神经元更快发放脉冲**，从而加快信号处理并减小反应时间。按以下参数设置：



模型参数：

initial weight: 0.21

presynaptic neurons: 20

postsynaptic neurons: 1

input spike intervals: 3ms

T: 100ms

dt: 1ms

STDP参数：

$A_+ = A_- = 1$

$\tau_1 = 10ms$

$\tau_2 = 20ms$

学习率：0.001

LIF神经元参数：

$\tau_m = 10ms$

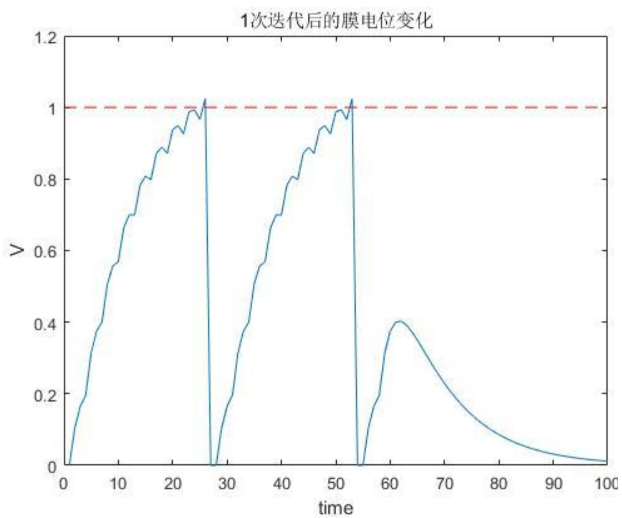
$\tau_s = \frac{\tau_m}{4}$

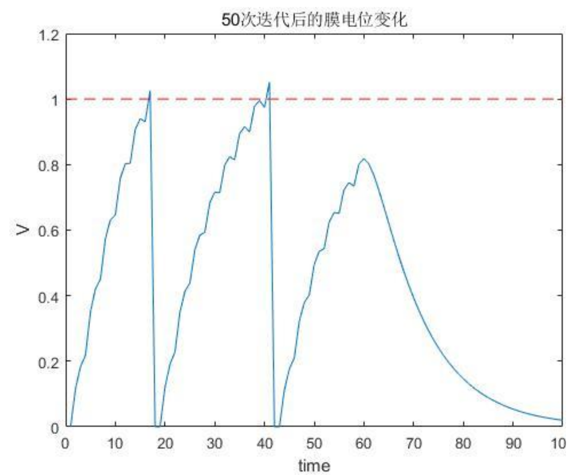
静息电位：0mV

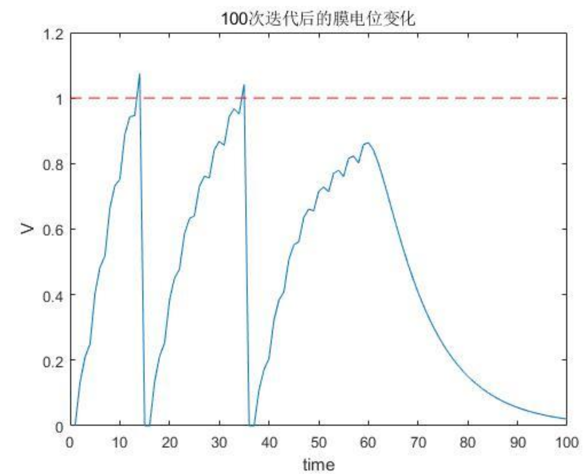
阈值：1mV

STDP: Learning to Be Fast

第1次迭代、第50次迭代、第100次迭代结果分别如图所示，由实验结果可知，随着训练次数的增加，**突触后神经元脉冲发放时间逐渐提前。**







[26,53]

[17,41]

[14,35]

STDP函数

```
function w = STDP(spikeTrains, spike_output, w, dt, T)
%Train synaptic weights with STDP rules
    A1 = 1; A2 = 1; tau1 = 20; tau2 = tau1/2; eta = 0.01;
    for t_pre = dt:dt:T
        spikeTrain = spikeTrains(:, t_pre);
        ind = find(spikeTrain == 1);
        for i = 1:length(spike_output)
            t_post = spike_output(i);
            s = t_pre - t_post;
            delta_w = 0;
            if s<0
                delta_w = A1 * exp(s / tau1);
            end
            if s>0
                delta_w = -A2 * exp(-s / tau2);
            end
            w(ind) = w(ind) + eta * delta_w;
            w = checkBound(w, 0, 1);
        end
    end
end
```

$$W(s) = \begin{cases} A_+ \exp(s/\tau), & \text{if } s < 0 \\ A_- \exp(-s/\tau), & \text{if } s > 0 \end{cases}$$

input:

spikeTrains: input spikes of presynaptic neurons ,with the shape nNeurons*T

spike_output: output spikes of postsynaptic neurons

w:synaptic weights before training

dt:time step

T:time window

return:

w: synaptic weights after training

checkBound函数

```
function w = checkBound(w, wmin, wmax)
%check synaptic weight boundaries
    w(find(w<wmin)) = wmin;
    w(find(w>wmax)) = wmax;
end
```

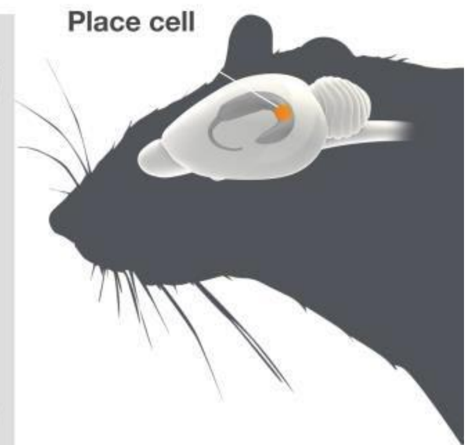
input：
wmin: minimum value of w
wmax: maximum value of w
w:synaptic weights before checking
return：
w: synaptic weights after checking

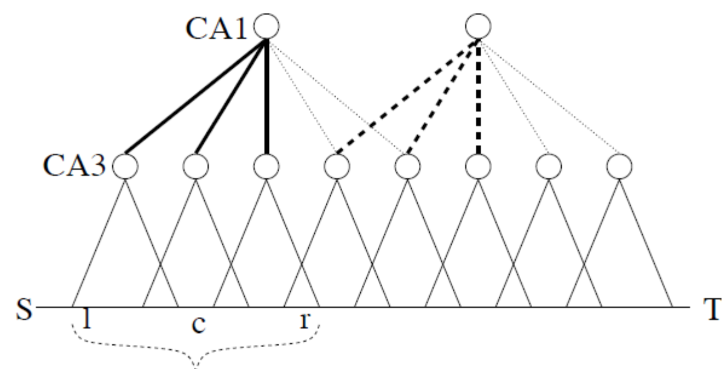
STDP: Hippocampal place fields

位置细胞是啮齿动物海马体中的神经元，对动物在环境中的空间位置很敏感。敏感区域称为细胞的位置域。

例如，如果一只老鼠在从**起点**S到**目标点**T的线性轨道上奔跑，该运动将首先激活位置域靠近S的细胞，然后是位置域在轨道中间的细胞，最后是那些位置域接近T的。

在一个简单的海马前馈模型中，CA1位置细胞接收来自CA3几个位置细胞的输入。在实验过程中，老鼠从左到右反复移动。在每次运动期间，相同序列的CA3细胞被激活。这会对从CA3细胞到CA1细胞的连接产生影响。

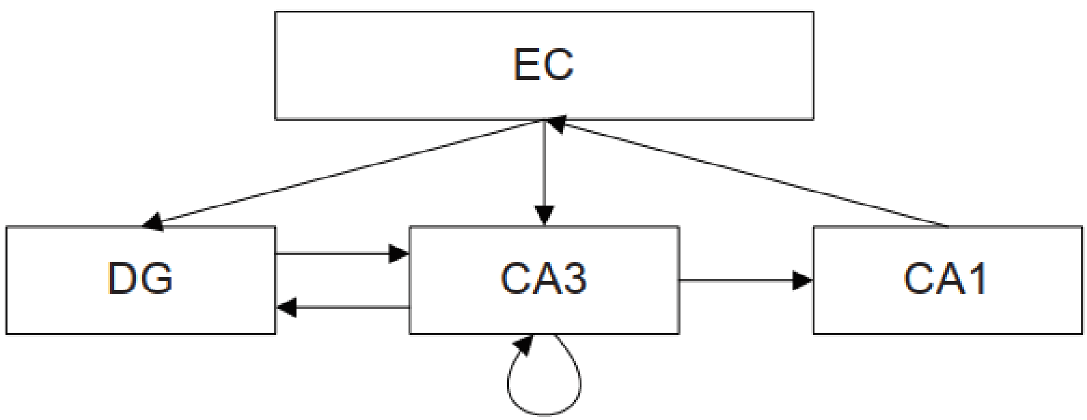




Structure of Hippocampus

- One of the functional roles of the **hippocampus** is the **storage** and **recall** of **associative memories**.

- Highly processed neocortical information from all sensory inputs converges onto the medial temporal lobe and enters the hippocampus via the entorhinal cortex (EC). There are connections from the EC to all parts of the hippocampus, including the dentate gyrus (DG), CA3 and CA1 through perforant pathway, from the DG to CA3 through mossyfibres, from CA3 to CA1 through schaffer collaterals, and then from CA1 back to EC. There are also strong recurrent connections within the DG and CA3 regions.

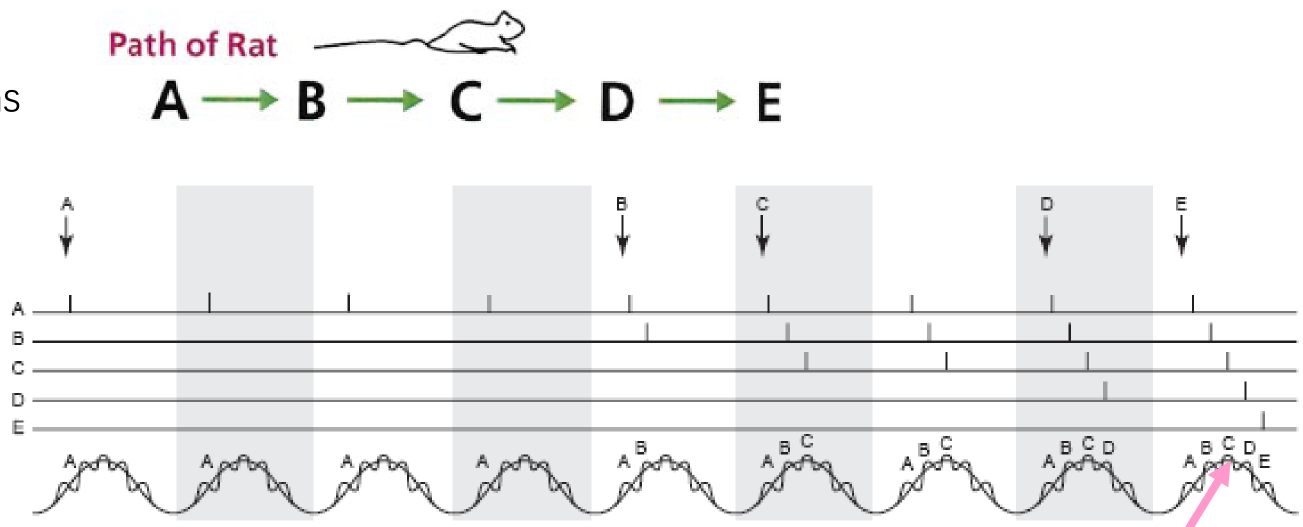


Block diagram of hippocampus

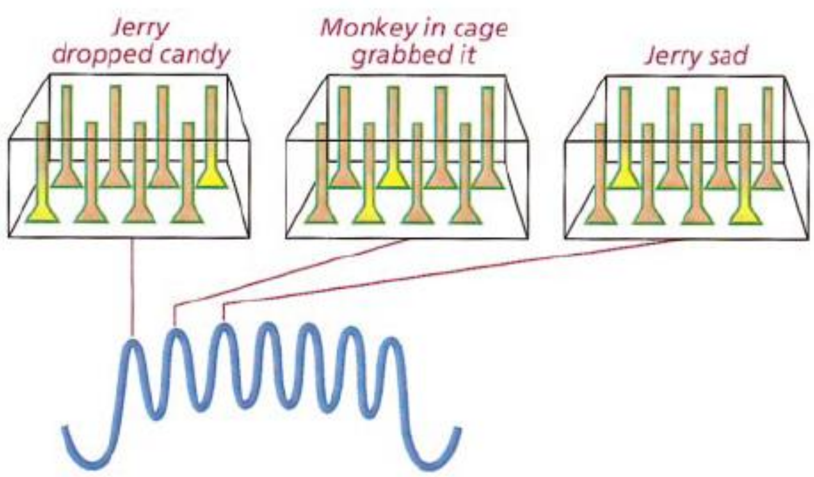
Chin Hiong Tan ,Eng Yeow Cheu,Jun Hu,Qiang Yu, and Huajin Tang. "Associative Memory Model of Hippocampus CA3 Using Spike Response Neurons." International Conference on Neural Information Processing Springer, Berlin, Heidelberg, 2011

Different Memory Patterns

Sequence of positions



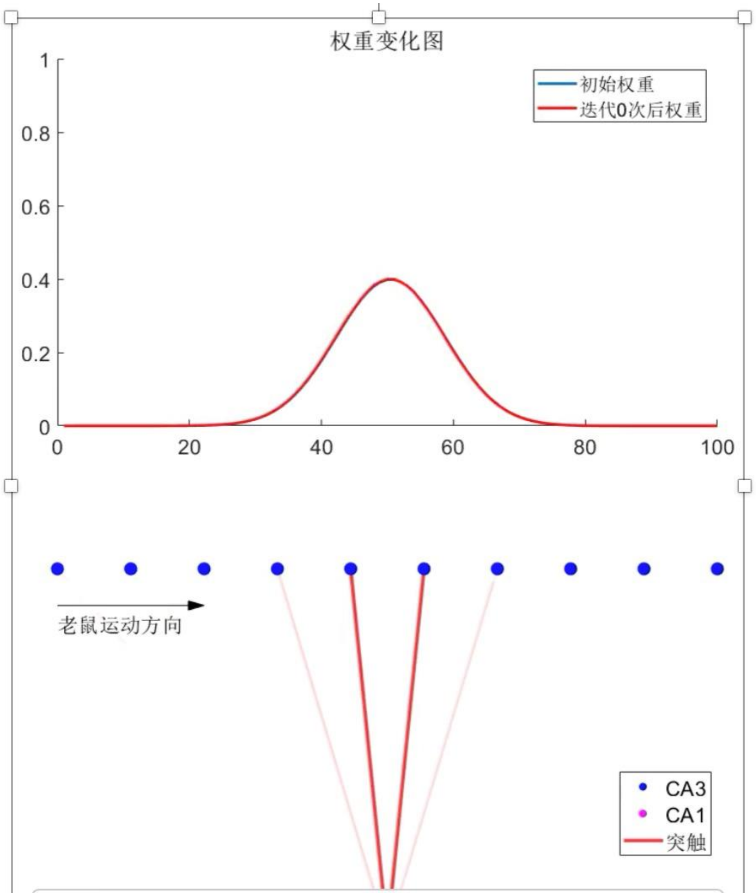
Sequence of events

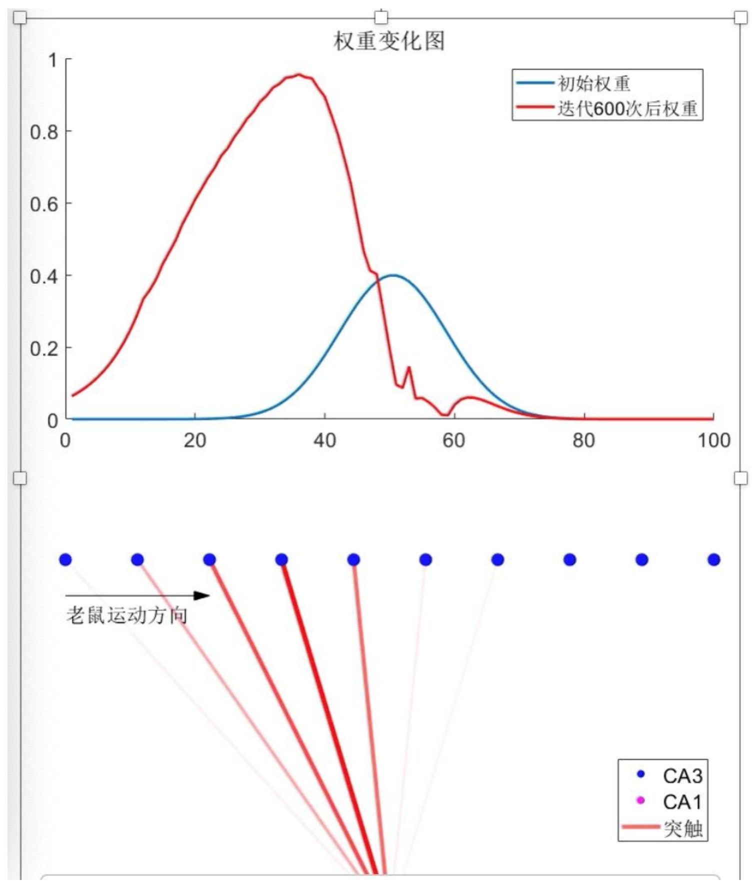


Neurons representing different position or event fires at different phase in time in each rhythm to encode the sequence.

STDP: Hippocampal place fields

位置细胞是啮齿动物海马体中的神经元，对动物在环境中的空间位置敏感。敏感区域称为单元格的**位置场**。虽然位置场在开始时是对称的，但随着经验的增加，它们变得**高度不对称**。





STDP: Hippocampal place fields

- 在右侧视频所示实验中，海马体CA1区与CA3区位置细胞连接的突触权重初始化为正态分布，老鼠从左到右反复移动600次，即每个运动序列中，CA3区细胞被依次激活。
- 具有不对称学习窗口的STDP**增强**了突触前神经元在**序列早期**激发的那些连接，而在**序列后期**激发的神经元的连接被**削弱**。因此，CA1中单元格的位置场的中心向左移动。

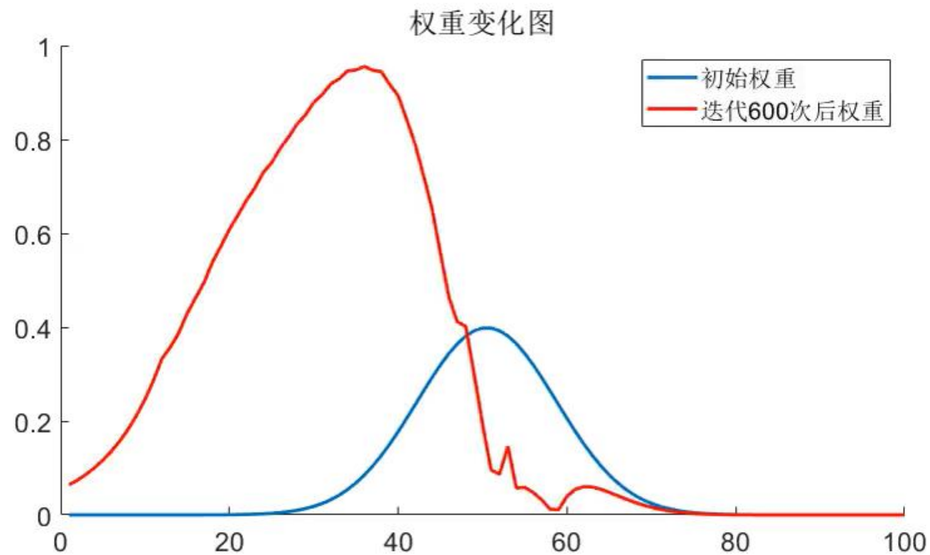
模型参数：

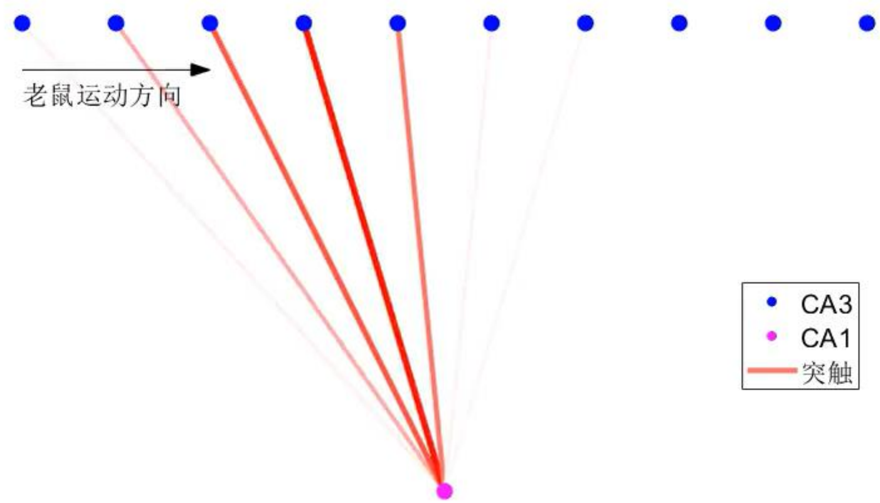
initial weight:normpdf(linspace(-6,6,100),0,1)

presynaptic neurons:100

postsynaptic neurons:1

其余参数同上





Hippocampal place fields 绘图函数

```
% 设置绘图图窗属性
skip_frame = 1; % 调控绘图速度
gcf = figure(1);
set(gcf,'color','w');
set(gcf,'position',[200,300,800,1000]);
set(gca,'fontsize',16);
% 绘制权重变化图（初始）
init_w = w;
ax1=subplot(2,1,1);
cla;
hold on;
plot(init_w,'LineWidth',2);
w_line = plot(w, 'r','LineWidth',2);
set(gca,'FontSize',16);
ylim([0 1]);
hold off;
title('权重变化图','FontSize',16);
legend('初始权重','迭代0次后权重
','FontSize',16);
```

```
% 绘制权重连接图（初始）
subplot(2,1,2);
cla;
hold on;
lines = [];
for i=1:10
    lines = [lines,
line([i,5.5],[1,0],'lineWidth',2*(1+w(i*11-10)))];
    lines(i).Color=hsv2rgb([0,w(i*11-10)/max(w),1]);
end
s1 = scatter(1:10,ones(1,10),100,'b','filled');
s2 = scatter(5.5,0,100,'m','filled');
arrow([1,0.9],[3,0.9]);
text(1,0.85,'老鼠运动方向','FontSize',16);
legend([s1 s2 lines(5)],'CA3', 'CA1', '突触
','Location','southeast','FontSize',16);
hold off;
ylim([0 1]);
axis off;
```

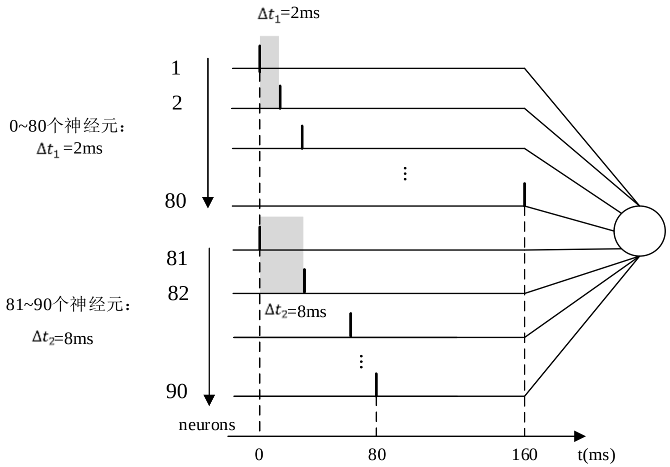
Hippocampal place fields 绘图函数

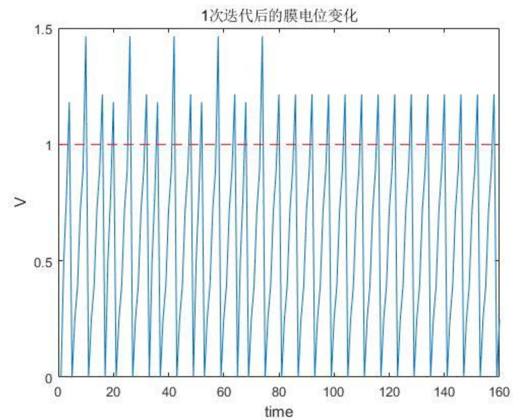
```
for ite=1:iter
    ……
     （脉冲发放计算过程）
    ……
    w = STDP(spikeTrains, spike_output, w, dt, T); % 更新权重
    if mod(ite,skip_frame)==0
        % 更新权重变化图的数据
        w_line.YData = w;
        % 更新权重连接图的数据
        for i=1:10
            lines(i).LineWidth = 2*(1+w(i*11-10));
            lines(i).Color=hsv2rgb([0,w(i*11-10)/max(w),1]);
        end
        drawnow;
        legend(ax1,'初始权重',sprintf('迭代%d次后权重',ite));
    end
end
```

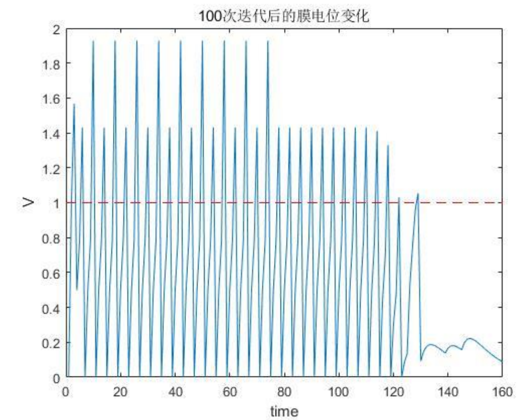
STDP: Learning to be Precise

模型参数
initial weight:0.5
presynaptic neurons、T见下图
其余参数同上

STDP规则可以**选择性地强化传递较高时间精度脉冲的突触，而减弱传递较低时间精度脉冲的突触**。因此STDP规则可以降低膜电位噪声水平并提高突触后神经元发放脉冲的时间精度。



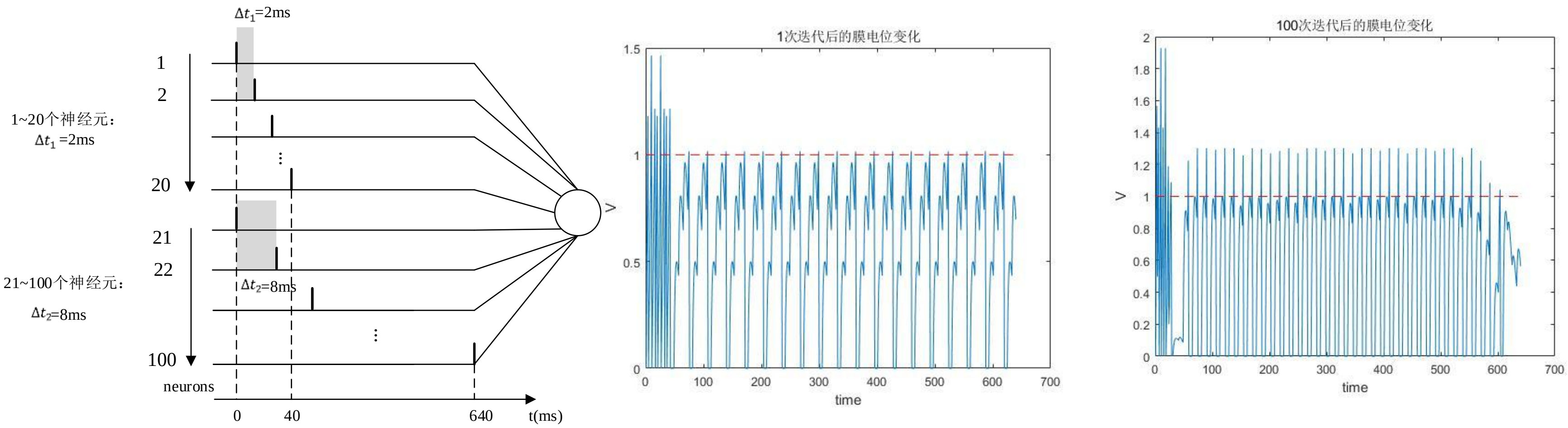




由实验结果可知，训练后STDP强化了传递较高时间精度（即$\Delta t_1 = 2ms$）脉冲的突触。

- 第一个实验结果中，高低精度神经元组同时刺激(0~80ms)和只有高精度神经元组刺激(80ms~160ms)，STDP训练前后，突触后神经元脉冲发放模式基本一致

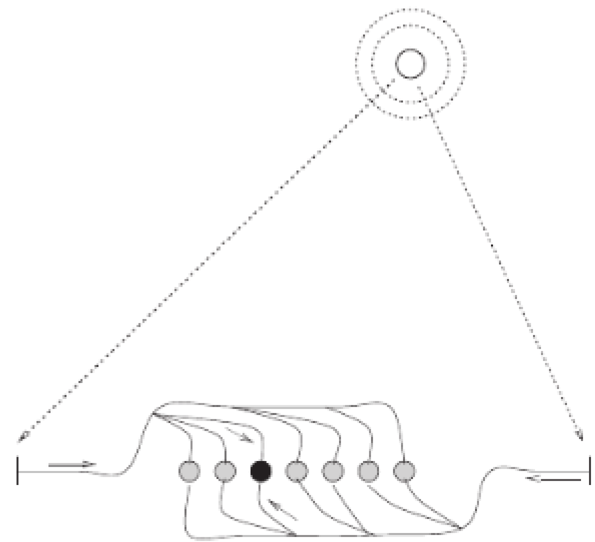
STDP: Learning to be Precise



- 第二个实验结果中，高低精度神经元组同时刺激（0~40ms）和只有低精度神经元组刺激（80ms~160ms），STDP训练前后，只有低精度神经元组刺激时，突触后神经元脉冲发放模式相差较大，**脉冲发放趋向于高时间精度**。这是因为STDP强化了传递较高时间精度脉冲的突触。

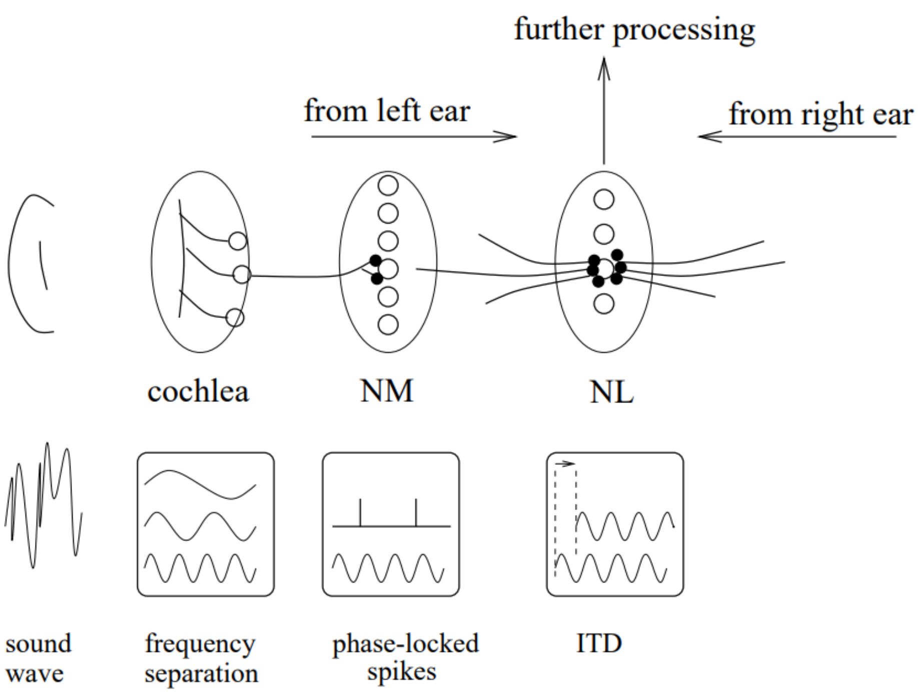
Barn owls auditory system

从行为学实验中可知，猫头鹰即使在完全黑暗的环境下，也能在水平面上以**大约 1-2 度**角的精度定位声源，这对应于左右耳声波之间**几微秒**的时间差异。这些微小的时间差异必须由猫头鹰的听觉系统检测和评估。



Jeffress模型：从猫头鹰头部右侧的声源发出的声波(虚线圈)到达两只耳朵并在那里激发神经元的活动。信号沿着传输线传输到**同步探测器阵列**(灰色圆圈)。如果双方的信号同时到达，探测同步的神经元会做出反应。由于**传输延迟**，信号激活的同步探测器(黑色填充圆)的**位置**取决于外部声源的位置。

Auditory Localization Pathway



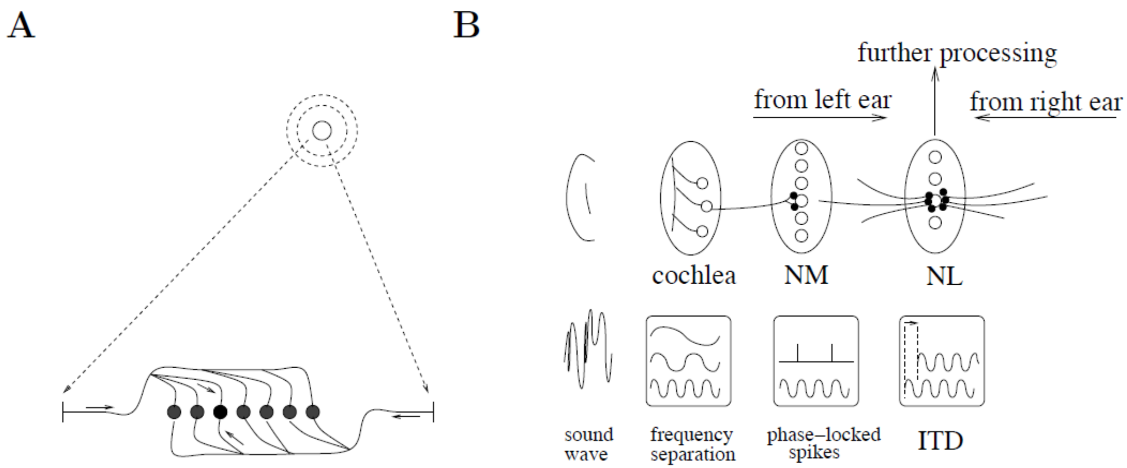
听觉通路的三个重要处理步骤：

- frequency separation：在耳蜗，声波被分成其**频率分量**。
- phase locking：**锁相**的脉冲沿听觉神经传输到前庭神经外侧核（NM），这是一个**中间处理步骤**。NM输出的动作电位也是锁相的。
- phase-correct averaging：来自双耳的信号在椎板核（NL）相遇，NL中的神经元对**耳间时间差（ITD）**敏感。

在进一步的处理步骤中，将**不同频率**的神经元的输出**组合起来**以解决其他的模糊性，以检索时间差异，从而检索声源在水平平面中的位置。

Gerstner, Wulfram, et al. "14 Hebbian Learning of Pulse Timing in the Barn Owl Auditory System." (1998).

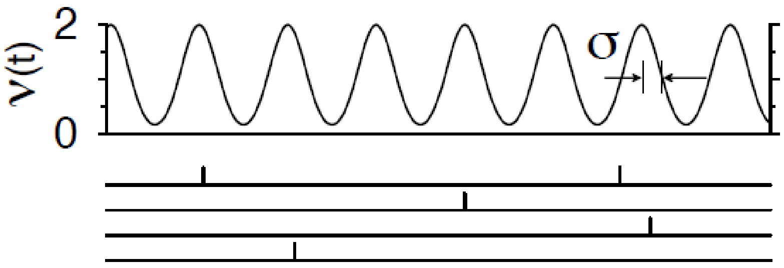
Barn owls auditory system

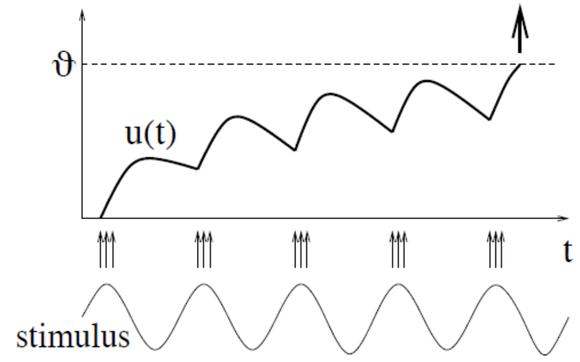


A. Jeffress model. Sound waves (dashed circles) from a source located to the right of the owl's head arrive at the two ears where they excite neuronal activity. Neuronal signals travel along transmission lines to an array of coincidence detectors (grey filled circles). Due to transmission delays, the position of the coincidence detector activated by the signals (black filled circle) depends on the location of the external sound source.

B. Auditory pathway. At the cochlea a sound wave is separated into its frequency components. Phase locked spikes are transmitted along the auditory nerve to the nucleus magnocellularis (NM, 前庭神经外侧核), an intermediate processing step. Action potentials at the output of the NM are phase locked as well. The signals from both ears meet in the nucleus laminaris (NL, 层状核). Neurons in the NL are sensitive to the interaural time difference (ITD) and can be considered as the coincidence detectors of the Jeffress model. In further processing steps, the output of neurons with different frequencies is combined to resolve remaining ambiguities;

Barn owls auditory system





- Phase locking can be observed in the auditory nerve connecting the cochlea and the nucleus magnocellularis, in the nucleus magnocellularis, and also in the nucleus laminaris. The phase jitter σ even decreases from one processing step to the next so that the temporal precision of phase locking increases from around 40μs in the nucleus magnocellularis to about 25μs in the nucleus laminaris. The precision of phase locking is the topic of the following subsection.

- Action potentials arrive periodically and are phase-locked to the stimulus in bundles of spikes (bottom). The postsynaptic potentials evoked by presynaptic spike arrival are summed and yield the total postsynaptic potential u(t) which shows a pronounced oscillatory structure. Firing occurs when u(t) crosses the threshold. The output spike is phase locked to the external signal.


Thank You!