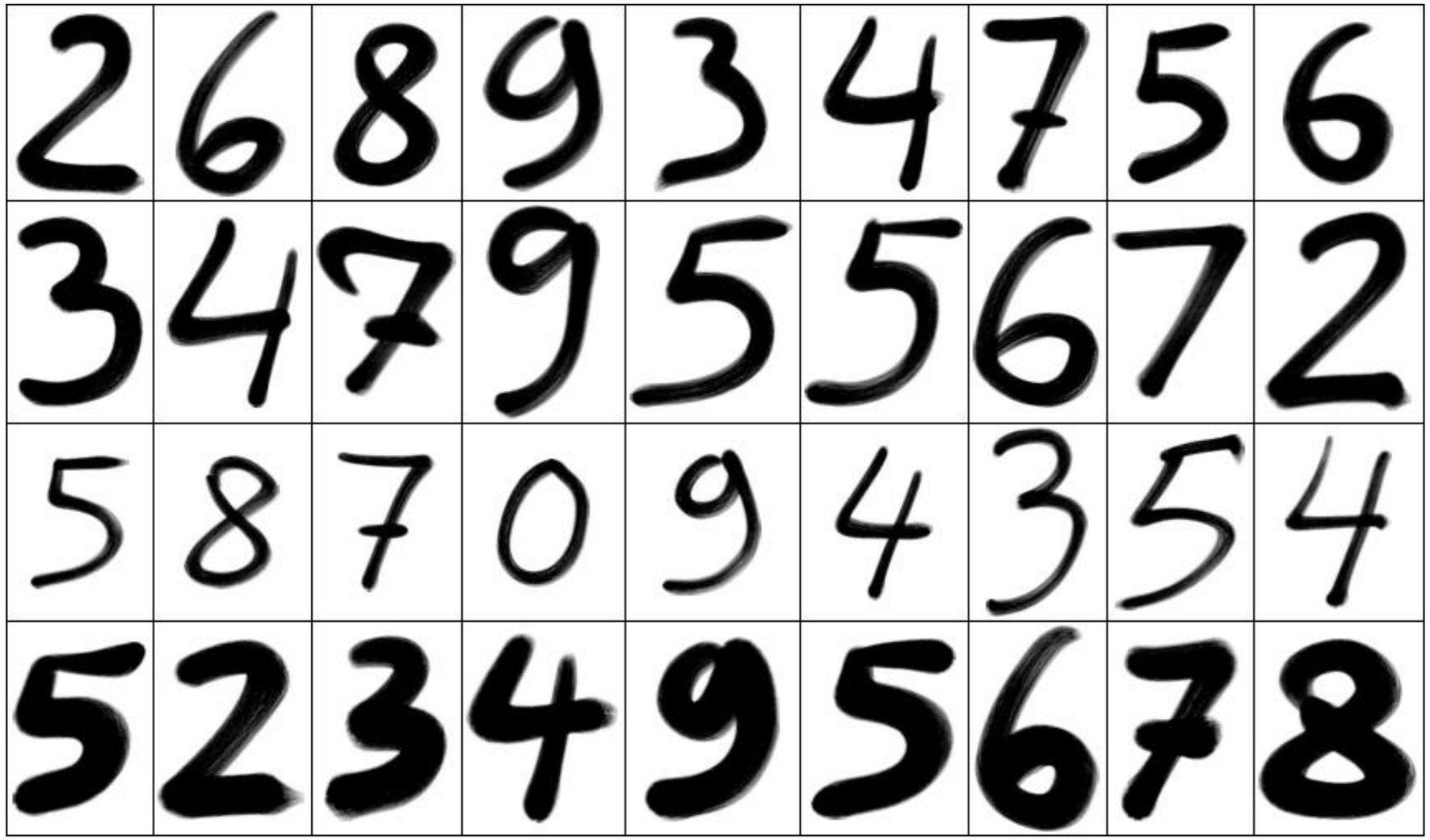
Adaptive Intelligence

Prof. Eleni Vasilaki

Types of Learning

- Supervised
- Unsupervised
- Reinforcement

Supervised Learning



MNIST Database

Supervised Learning

$$\{ (x^1, t^1)(x^2, t^2) \cdots \quad (x^p, t^p) \}$$

- An expert (supervisor) has provided labels for the data (e.g. class 1, class 2 etc).

- Parameters are updated so that the total error, evaluated using the training (labeled) data, is minimised.

- The system, if correctly trained, is expected to perform also with unseen samples from the same classes.

Unsupervised Learning

Unsupervised Learning

$$\{\ x^1, x^2, x^3, \quad \cdots \quad x^p\ \}$$

- Just data - NO labels!
- Detect intrinsic structure in the data.
- Leads to clusters (not classes).

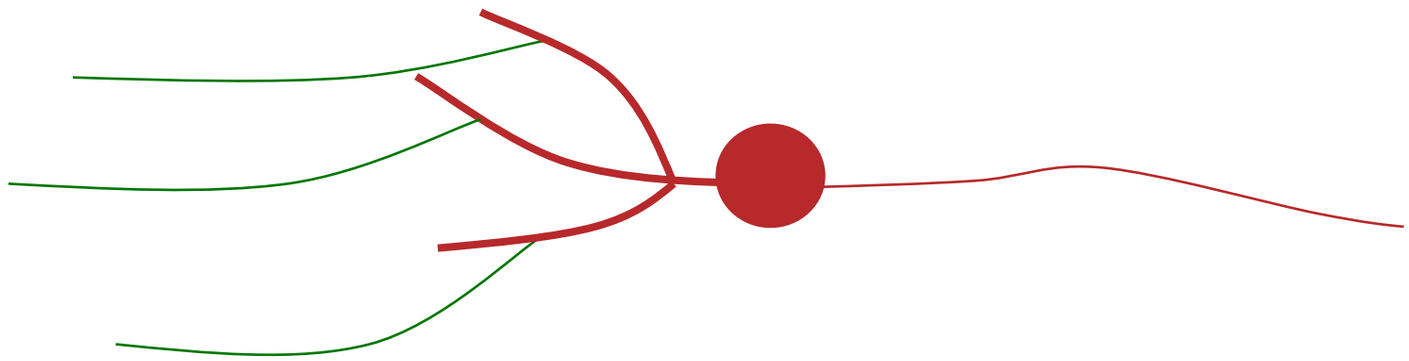
Reinforcement Learning

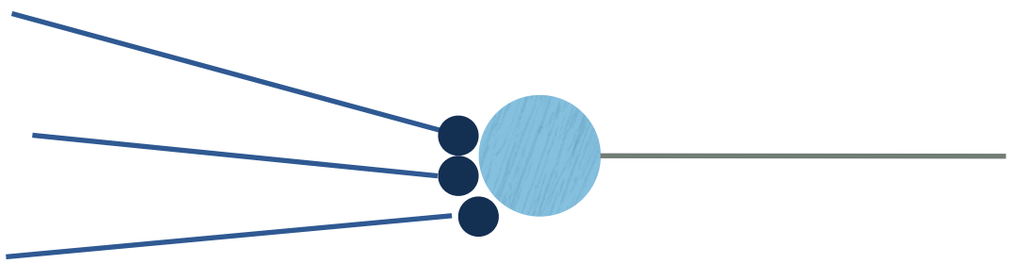
Reinforcement Learning

$$\{ \ (x^1, a^1, r^{11}) \ (x^1, a^2, r^{12}) (x^2, a^1, r^{21}) \ \ \ldots \ \ \}$$

- An agent receives an input/stimulus.
- Choice of actions.
- Reward upon correct action (or actions - it may not be immediate!).
- Parameters are updated so that the total Reward is maximised.

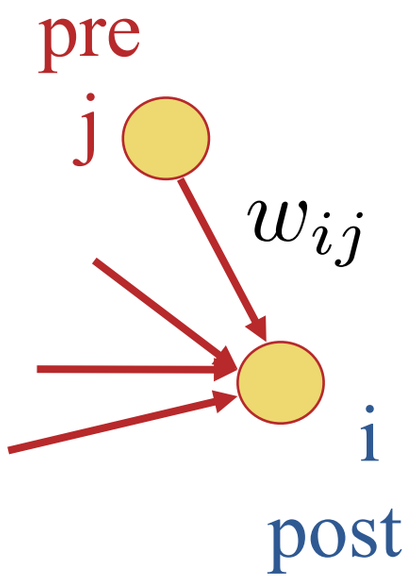
Artificial Neurons





$$\nu_i = g\left(\sum_j w_{ij}\nu_j\right)$$

The Hebbian rule



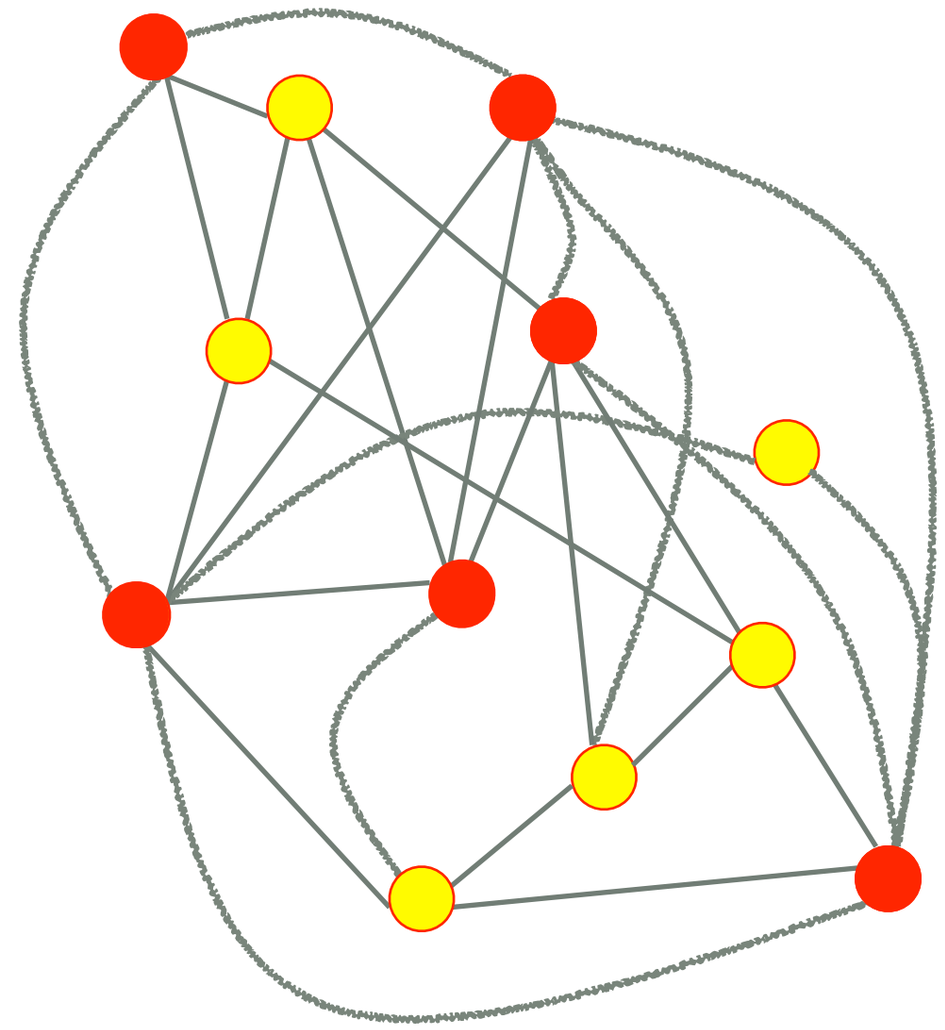
Correlations

Donald Hebb

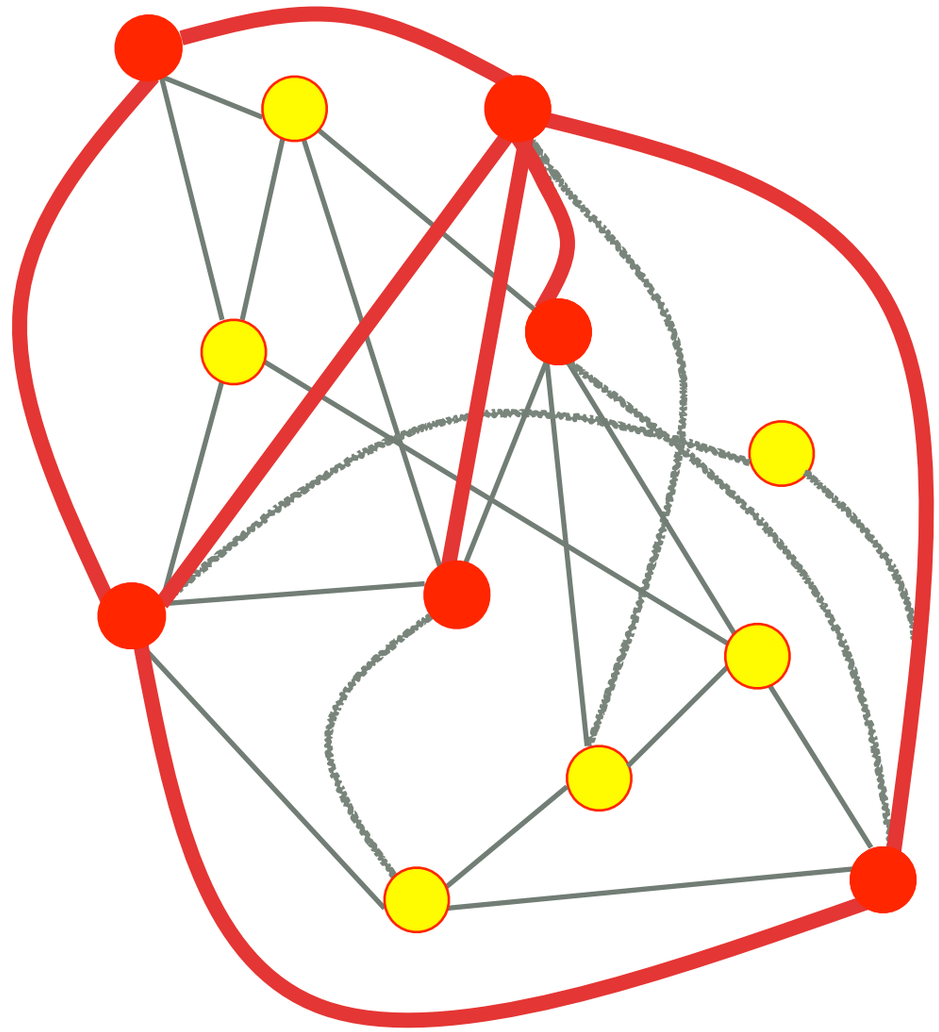
“Neurons that fire together wire together.”

–A compact form of Donald Hebb’s postulate

Hebbian Learning



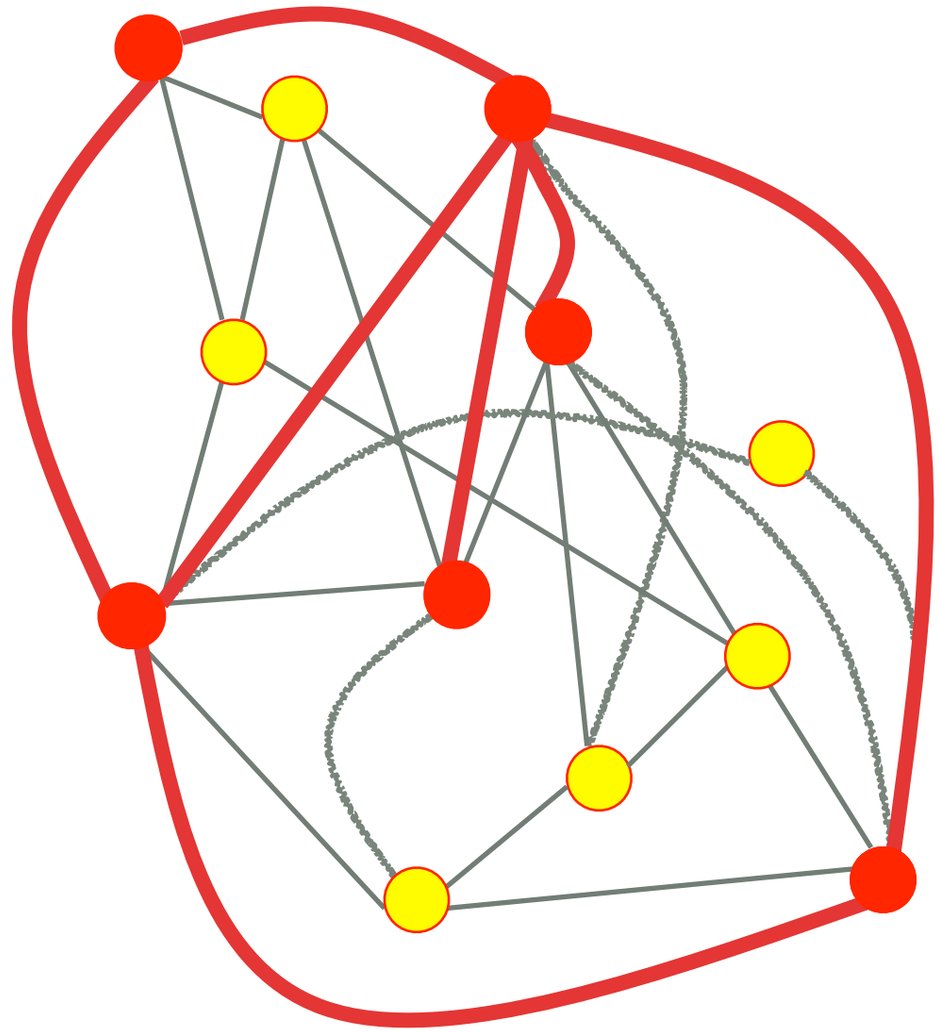
Hebbian Learning



item memorized

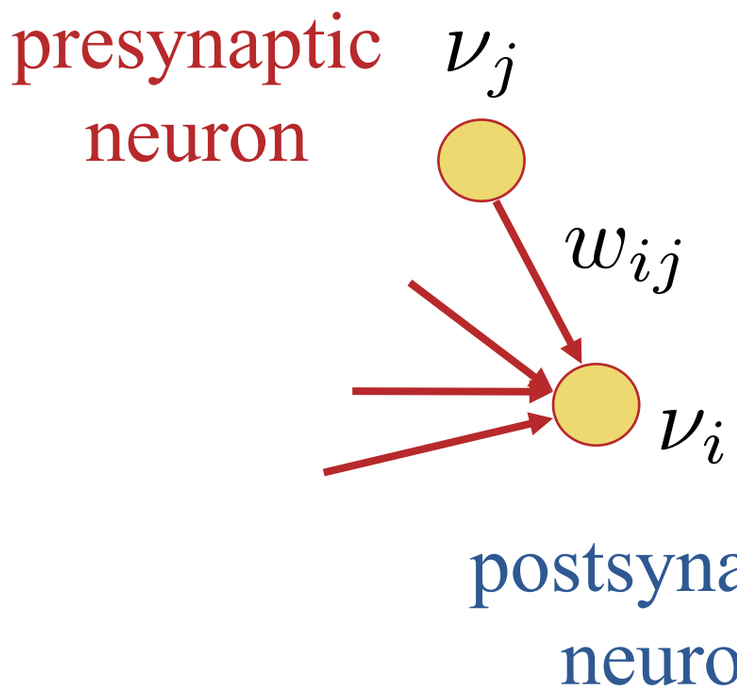
Hebbian Learning

Recall:
Partial info



item recalled

The “minimal” Hebbian Rule



In equations

$$\Delta w_{ij} = w_{ij}^{new} - w_{ij}^{old} = \alpha \nu_i \nu_j$$

w_{ij} : weight from neuron j to neuron i

Δw_{ij} : weight change

α : learning rate, >0

Learning rule notation

Discrete time

$$\Delta w_{ij} = f\left(\nu_i, \nu_j, w_{ij}\right)$$

$$w_{ij} \to w_{ij} + f(\nu_i, \nu_j, w_{ij})$$

Continuous time

$$\frac{dw_{ij}}{dt} = g(\nu_i, \nu_j, w_{ij})$$

Hebbian rules

$$\Delta w_{ij} = \alpha \nu_i \nu_j$$

$$\Delta w_{ij} = \alpha \nu_i \nu_j - c$$

$$\Delta w_{ij} = \alpha (\nu_i - \theta) \nu_j$$

$$\Delta w_{ij} = \alpha (\nu_i - \theta)(\nu_j - \theta)$$

Common “ingredient”

$$\Delta w_{ij} = \alpha \nu_i \nu_j$$

Which rules are Hebbian?

Variable definitions. x: stimulus (neuronal input), y: neuronal activity, w: synaptic weight, ϵ: small positive number (learning rate), r : reward. Unless otherwise stated, x,y are mean firing rates.

1. Rescorla-Wanger (reward predicting) rule. $w \to w + \epsilon\delta x$, where $\delta = r - y$, $x \in \{0, 1\}$.

2. Homeostatic rule. $w \to w + \epsilon\,(y - \theta)$, where θ is the homeostatic threshold.

3. Node perturbation. $w \to w + \epsilon r y \xi$, where ξ represents noise (random variable drawn from a Gaussian distribution).

Which rules are Hebbian?

5. Sejnowski rule. $w \to w + \epsilon\,(y - \langle y \rangle)\,(x - \langle x \rangle)$, with $\langle . \rangle$ being the mean activity of the signals y and x across time.

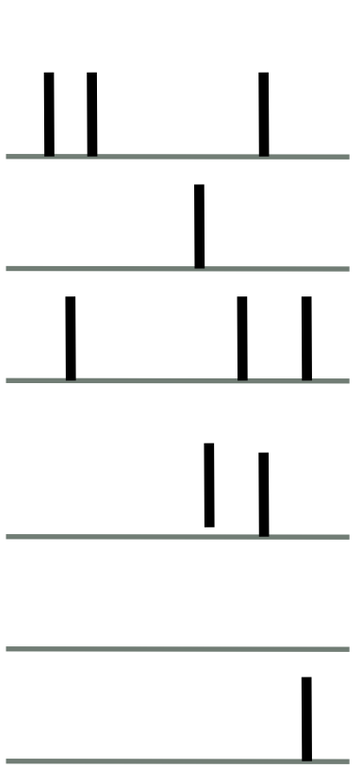
6. Oja rule. $w \to w + \epsilon y\,(x - yw)$.

7. Learning in Hopfield networks. $w \to w + \epsilon yx$, with $x, y \in \{-1, 1\}$.

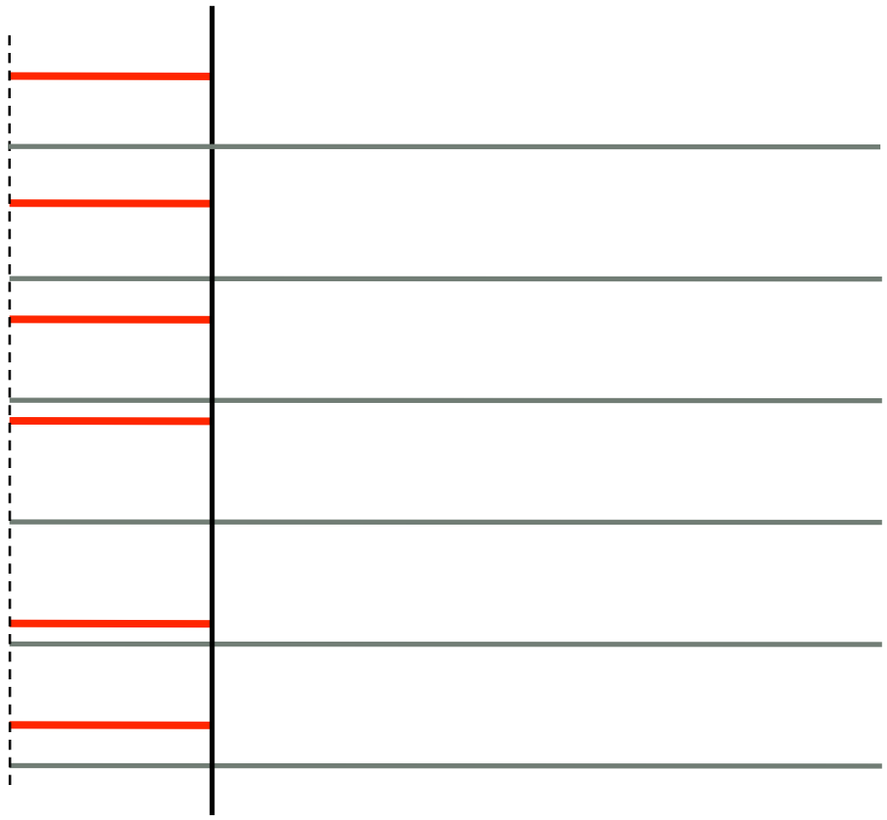
8. Delta rule. $w \to w + \epsilon \delta x$, where $\delta = t - y$, with t being the target for input x.

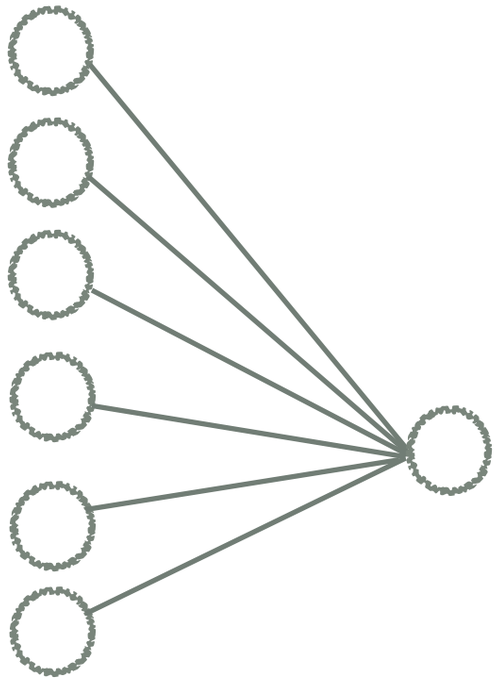
9. Associative Reward Inaction $w \to w + \epsilon r\,(y - P(y))\,x$, where $P(y)$ the probability of y to be active, $y \in \{0, 1\}$.

Functional Consequences of Hebbian Learning

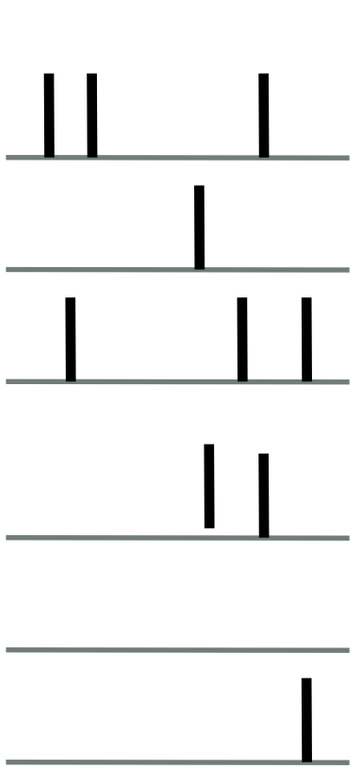


$$\vec{x}^1 = \begin{bmatrix} \nu_1^1 \\ \nu_2^1 \\ \nu_3^1 \\ \nu_j^1 \\ \vdots \\ \nu_N^1 \end{bmatrix}$$

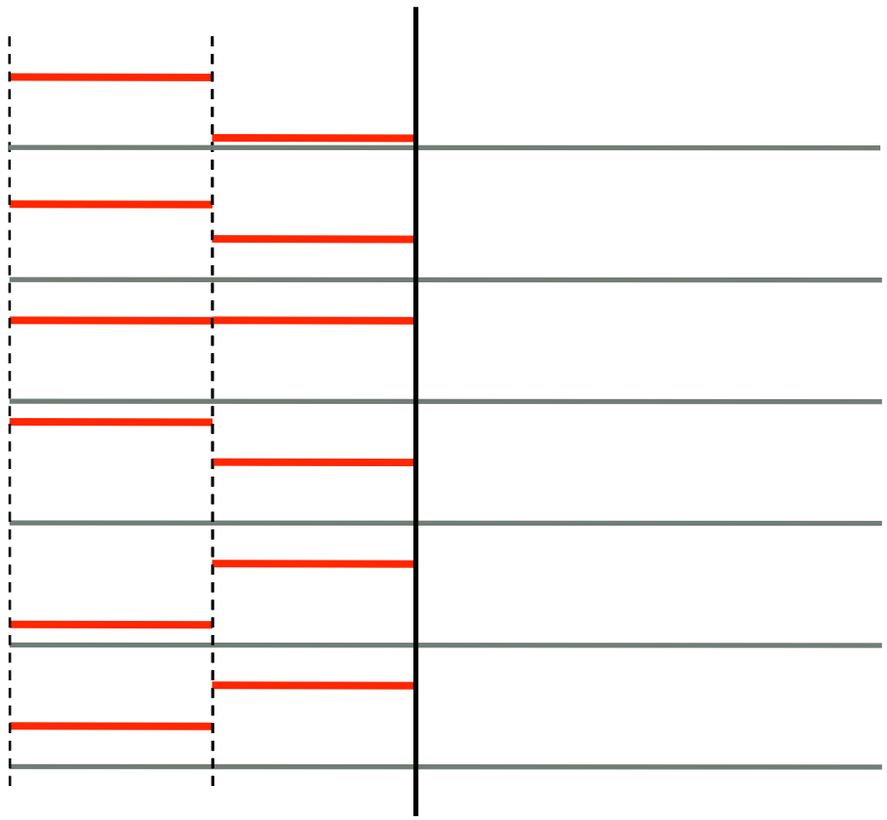


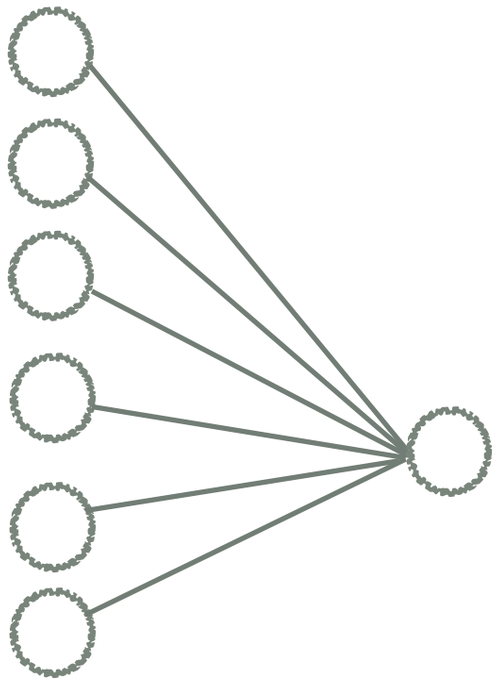


Functional Consequences of Hebbian Learning

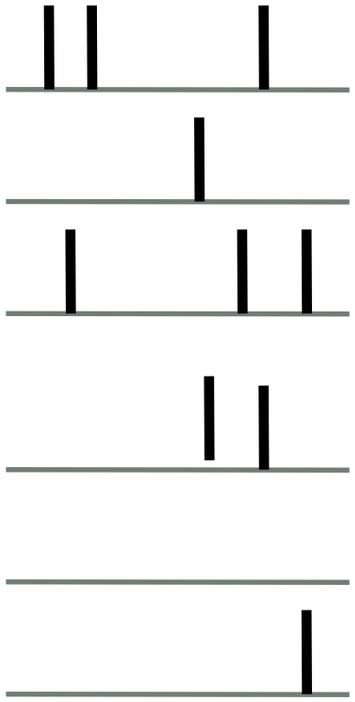


$$\vec{x}^2 = \begin{bmatrix} \nu_1^2 \\ \nu_2^2 \\ \nu_3^2 \\ \nu_j^2 \\ \vdots \\ \nu_N^2 \end{bmatrix}$$

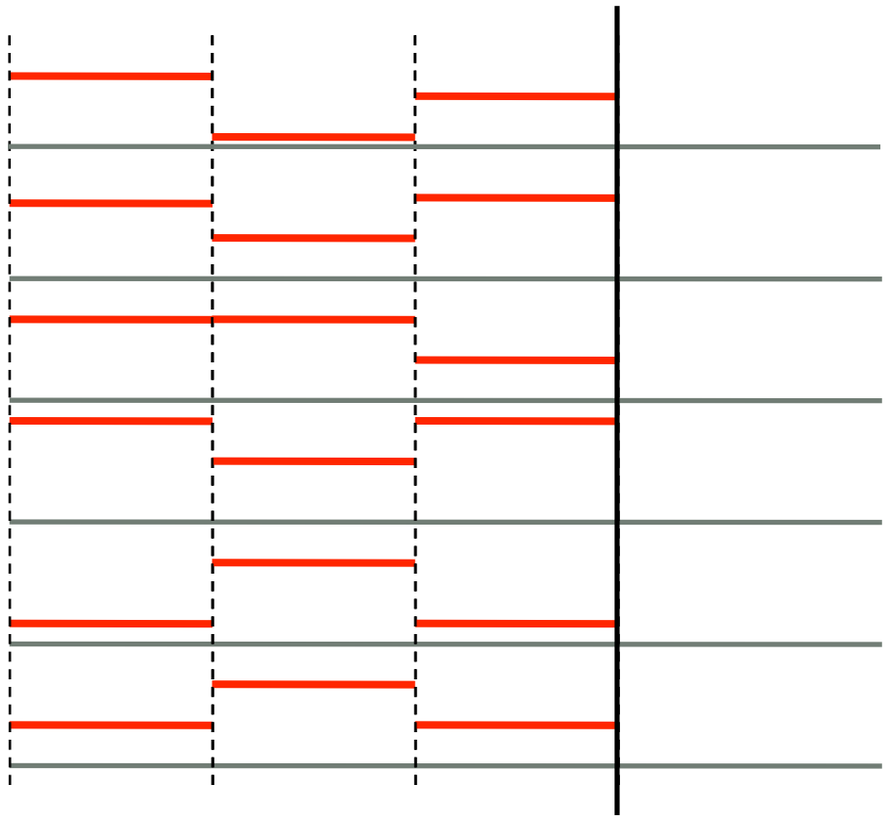


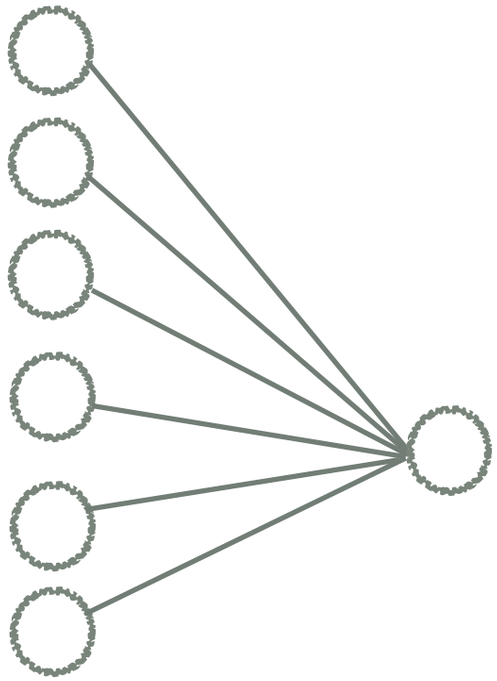


Functional Consequences of Hebbian Learning

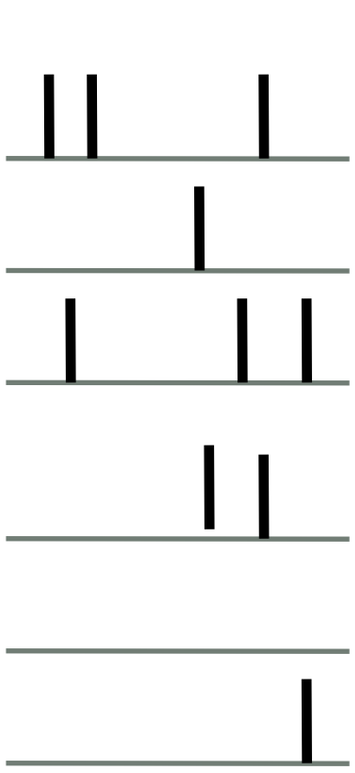


$$\vec{x}^3 = \begin{bmatrix} \nu_1^3 \\ \nu_2^3 \\ \nu_3^3 \\ \nu_j^3 \\ \vdots \\ \nu_N^3 \end{bmatrix}$$

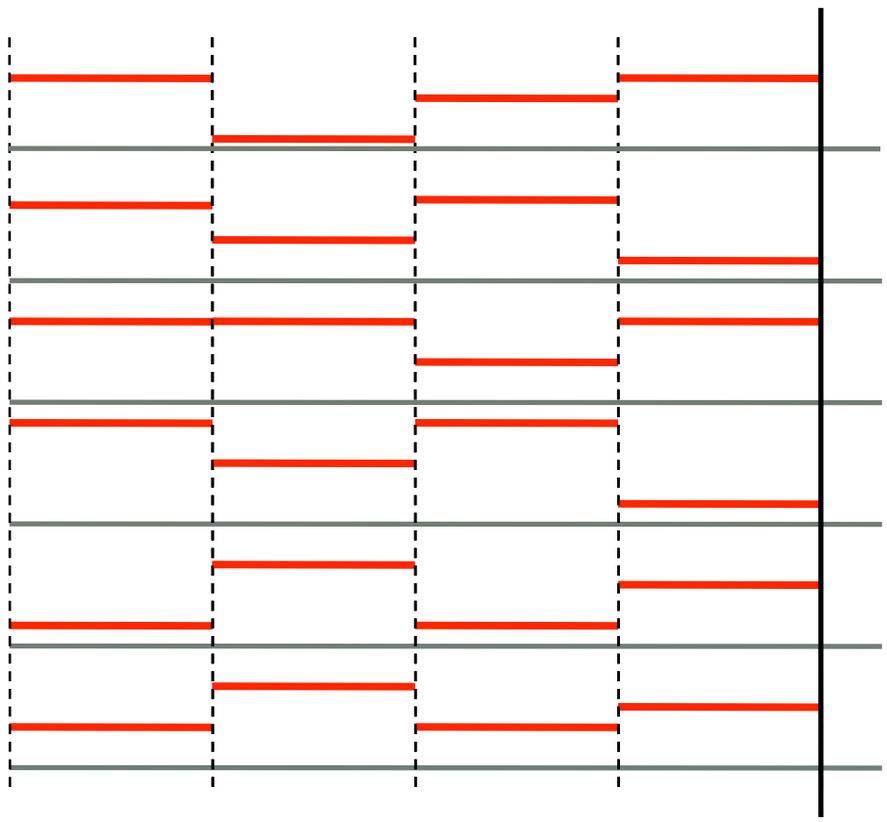


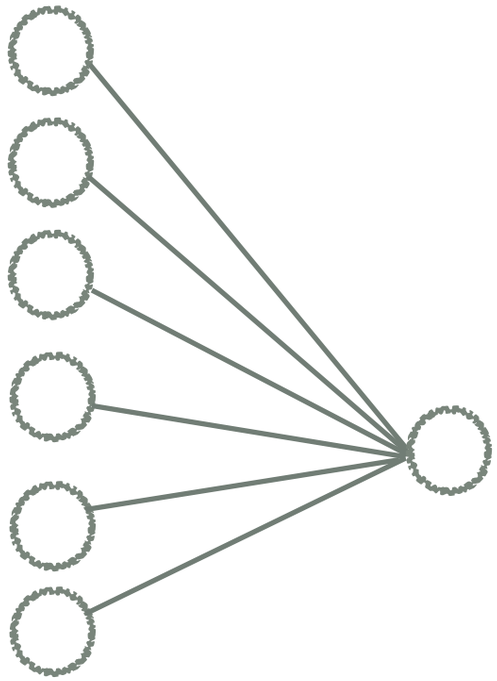


Functional Consequences of Hebbian Learning

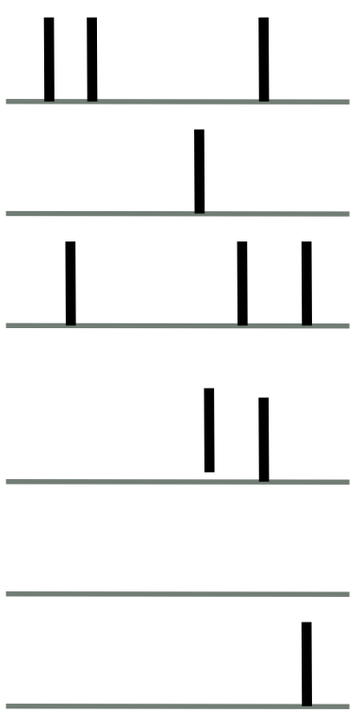


$$\vec{x}^4 = \begin{bmatrix} \nu_1^4 \\ \nu_2^4 \\ \nu_3^4 \\ \nu_j^4 \\ \vdots \\ \nu_N^4 \end{bmatrix}$$

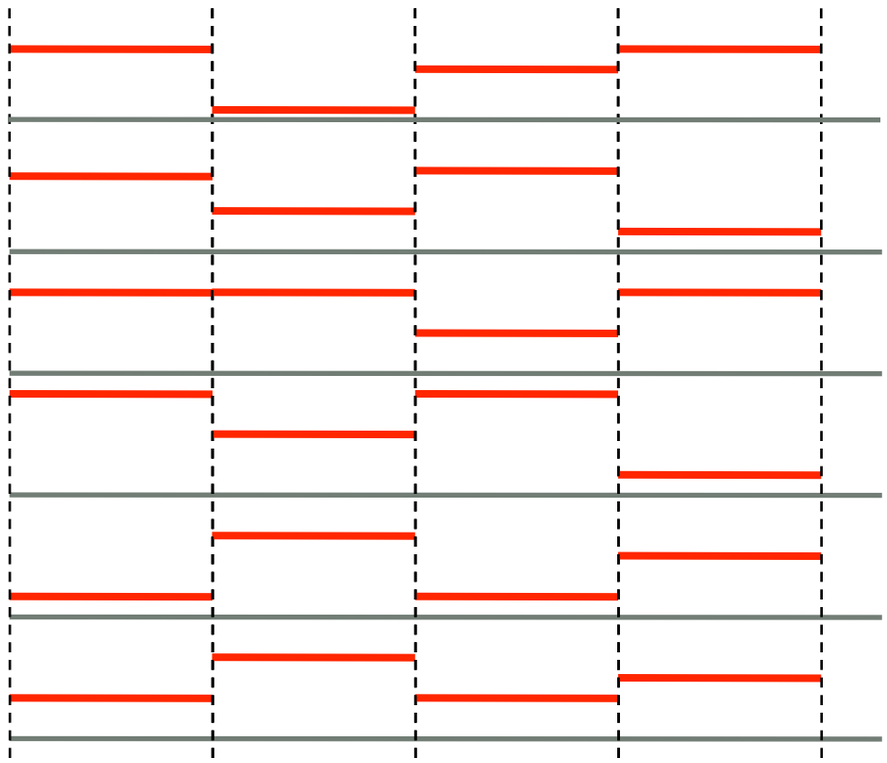


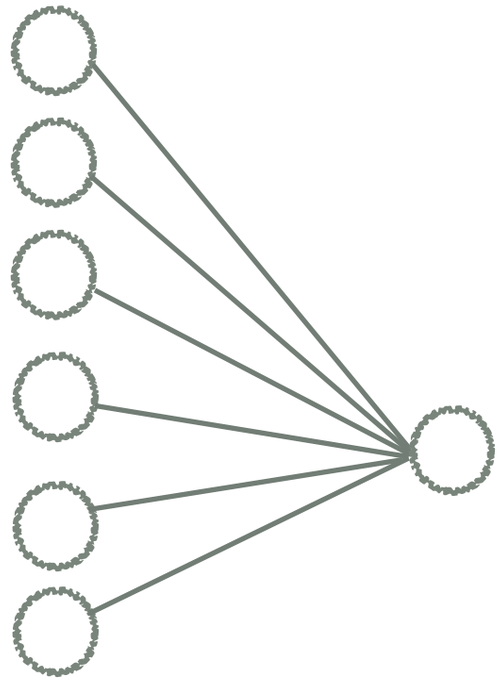


Functional Consequences of Hebbian Learning



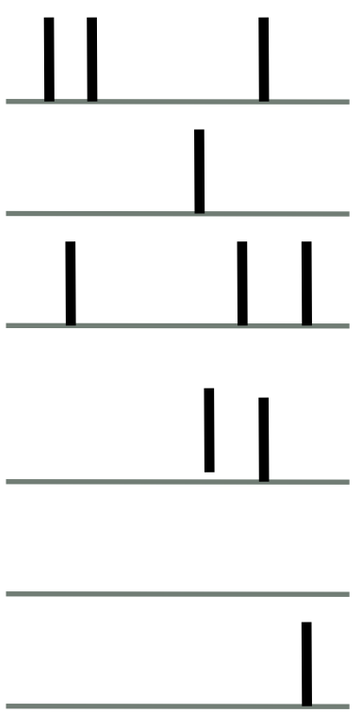
$$\vec{x}^{\mu} = \begin{bmatrix} \nu_1^{\mu} \\ \nu_2^{\mu} \\ \nu_3^{\mu} \\ \nu_j^{\mu} \\ \vdots \\ \nu_N^{\mu} \end{bmatrix}$$



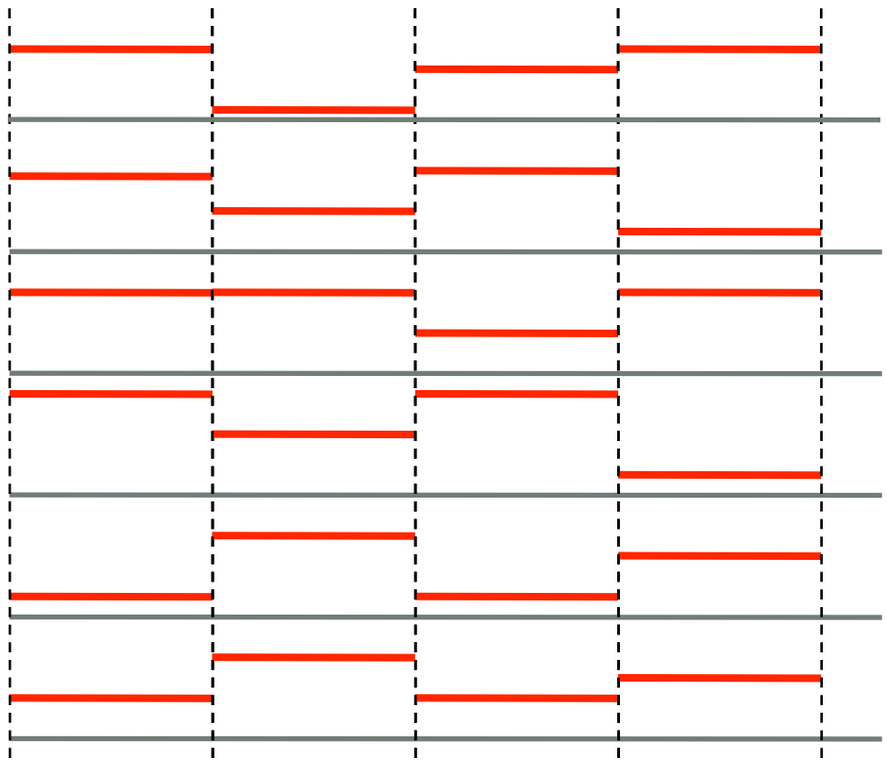


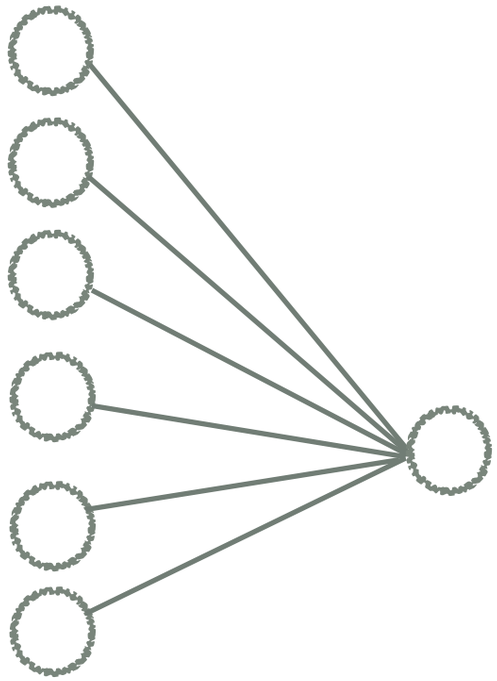
$$\{\vec{x}^{\mu}; 1 \leq \mu \leq P\}$$

Functional Consequences of Hebbian Learning



$$\vec{x}^{\mu} = \begin{bmatrix} \nu_1^{\mu} \\ \nu_2^{\mu} \\ \nu_3^{\mu} \\ \nu_j^{\mu} \\ \vdots \\ \nu_N^{\mu} \end{bmatrix}$$





Correlation: $$C_{kj} = \langle \nu_k \nu_j \rangle = \frac{1}{P} \sum_{\mu=1}^{P} \nu_k^{\mu} \nu_j^{\mu}$$

Calculate the Correlation Matrix

$$C_{kj} = \langle \nu_k \nu_j \rangle = \frac{1}{P} \sum_{\mu=1}^{P} \nu_k^\mu \nu_j^\mu$$

Neurons 1 and 2 exhibit the following pattern:

ν_1	1	-1	-1	1
ν_2	1	-1	-1	1

Calculate the Correlation Matrix

$$C_{kj} = \langle \nu_k \nu_j \rangle = \frac{1}{P} \sum_{\mu=1}^{P} \nu_k^\mu \nu_j^\mu$$

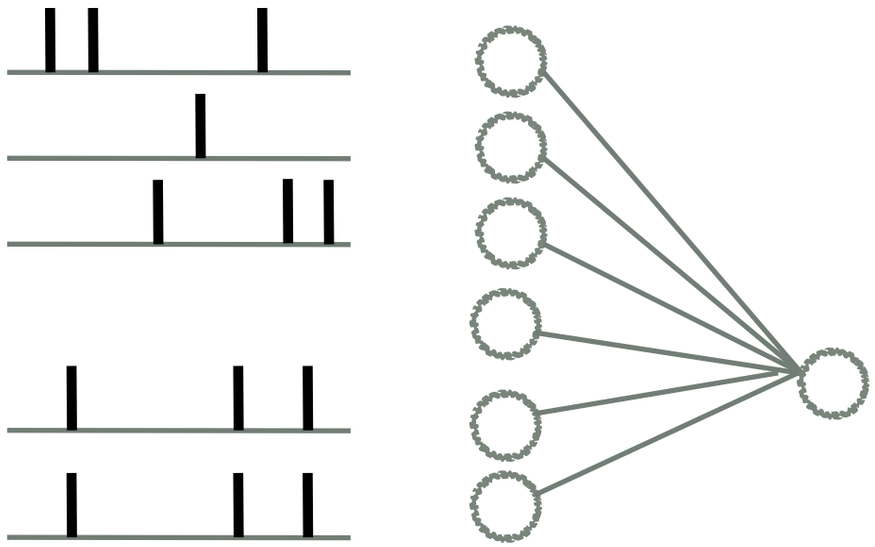
Neurons 1, 2 and 3 exhibit the following pattern:

ν_1	1	-1	-1	1
ν_2	1	-1	-1	1
ν_3	-1	1	1	-1

Functional Consequences of Hebbian Learning

Uncorrelated input

Jointly variant (correlated) input



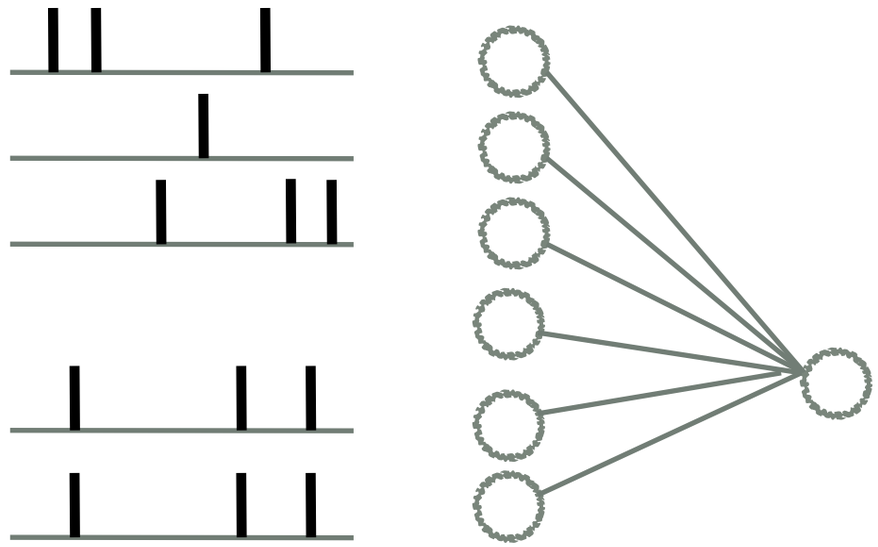
$$\Delta w_{ij} = \alpha \nu_i \nu_j$$

Claim: Detects correlations in the inputs

Functional Consequences of Hebbian Learning

Uncorrelated input

Jointly variant (correlated) input



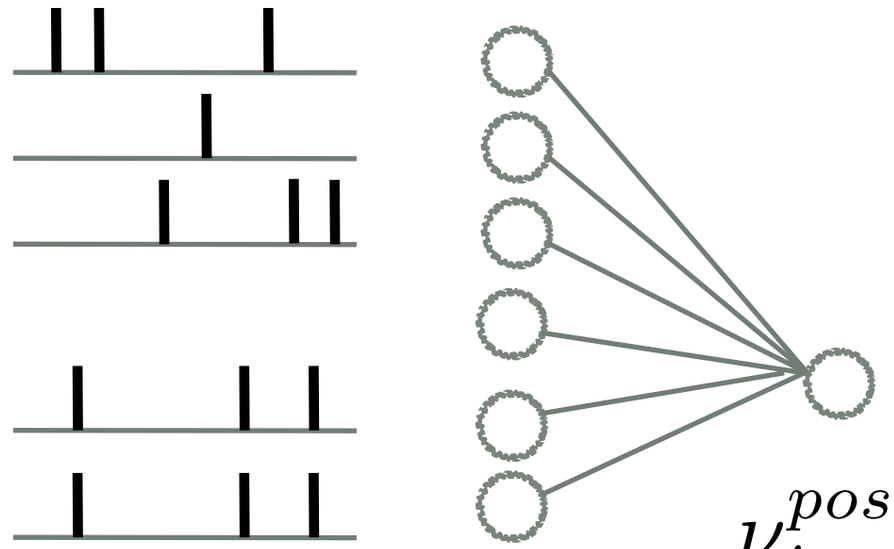
$$\Delta w_{ij} = \alpha \nu_i^{post} \nu_j^{pre}$$

Claim: Detects correlations in the inputs

Functional Consequences of Hebbian Learning

Uncorrelated input

Jointly variant (correlated) input



$$\nu_i^{post} = \sum_k w_{ik} \nu_k^{pre}$$

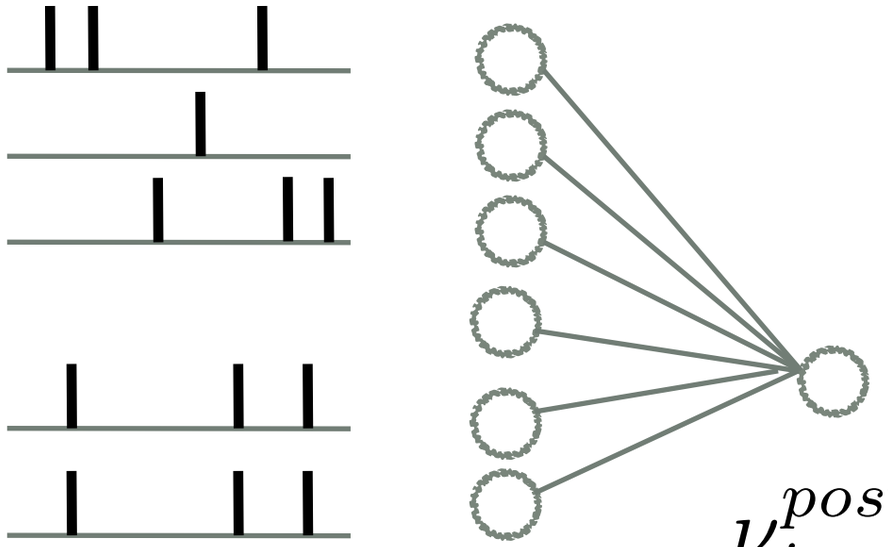
$$\Delta w_{ij} = \alpha \nu_i^{post} \nu_j^{pre}$$

Claim: Detects correlations in the inputs

Functional Consequences of Hebbian Learning

Uncorrelated input

Jointly variant (correlated) input



$$\nu_i^{post} = \sum_k w_{ik} \nu_k^{pre}$$

$$\Delta w_{ij} = \alpha \nu_i^{post} \nu_j^{pre}$$

$$C_{kj} = \langle \nu_k \nu_j \rangle = \frac{1}{P} \sum_{\mu=1}^{P} \nu_k^{\mu} \nu_j^{\mu}$$

Functional Consequences of Hebbian Learning

Learning rule

$$\Delta w_{ij} = \alpha \nu_i^{post} \nu_j^{pre}$$

$$\Delta w_{ij} = \alpha \left(\sum_k w_{ik} \nu_k^{pre} \right) \nu_j^{pre}$$

$$\frac{1}{P} \sum_{\mu=1}^{P} \Delta w_{ij}^{\mu} = \frac{1}{P} \sum_{\mu=1}^{P} \alpha \left(\sum_k w_{ik} \nu_k^{\mu} \right) \nu_j^{\mu}$$

$$\langle \Delta w_{ij} \rangle = \alpha \sum_k w_{ik} \langle \nu_k \nu_j \rangle$$

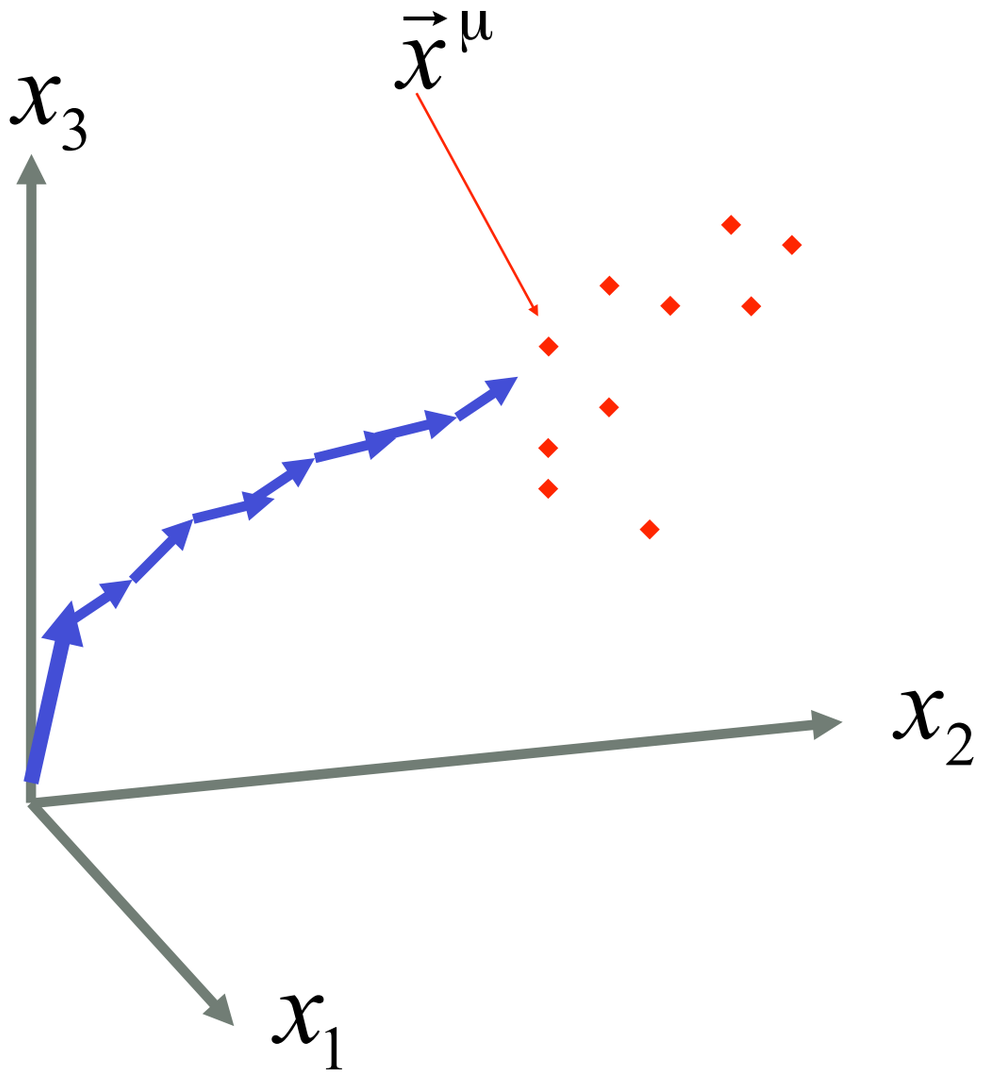
Neuron model

$$\nu_i^{post} = \sum_k w_{ik} \nu_k^{pre}$$

$$C_{kj} = \langle \nu_k \nu_j \rangle = \frac{1}{P} \sum_{\mu=1}^{P} \nu_k^{\mu} \nu_j^{\mu}$$

Detects correlations in the inputs

The minimal Hebbian rule



$$\Delta w_{ij} = \alpha \nu_i^{post} \nu_j^{pre}$$

Unstable!

The BCM rule

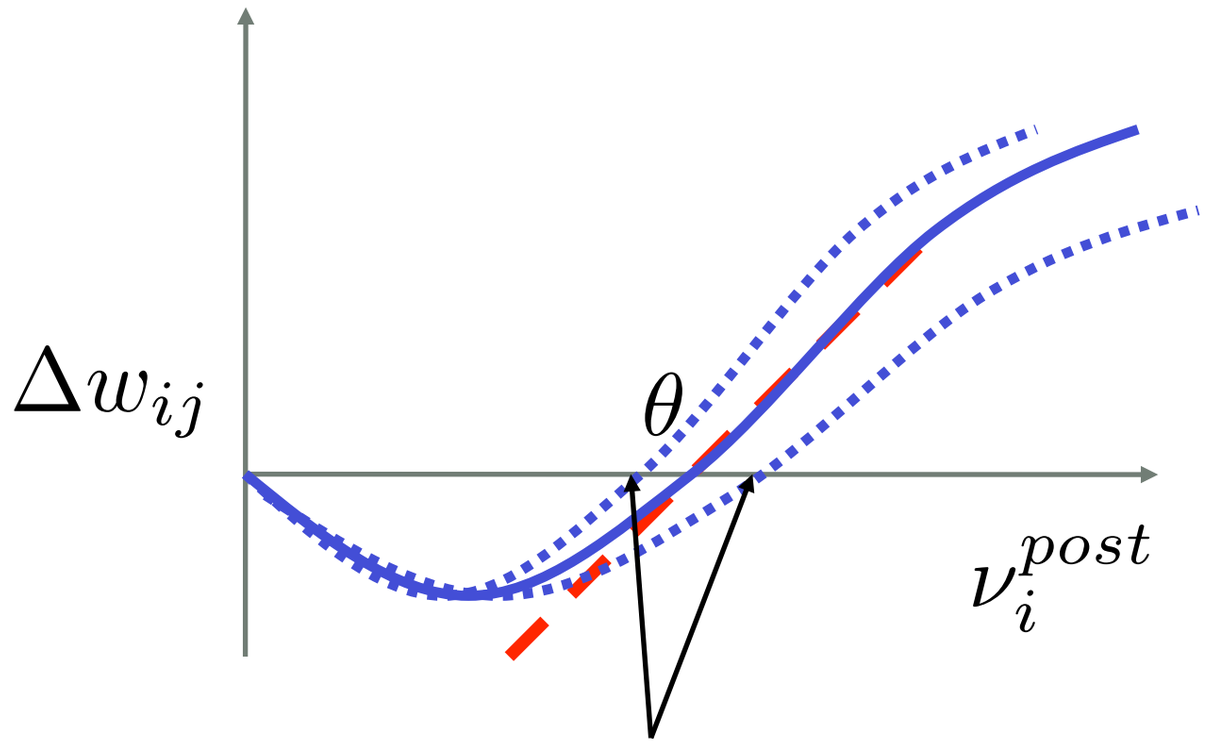
$$\Delta w_{ij} = \alpha \Phi \left(\nu_i^{post} - \theta \right) \nu_j^{pre}$$

Bienenstock, Cooper, Munro 1982

$$\Delta w_{ij} = \alpha \left(\nu_i^{post} - \theta \right) \nu_j^{pre}$$

presynaptically gated

$$\theta = f(< \nu_i^{post} >)$$

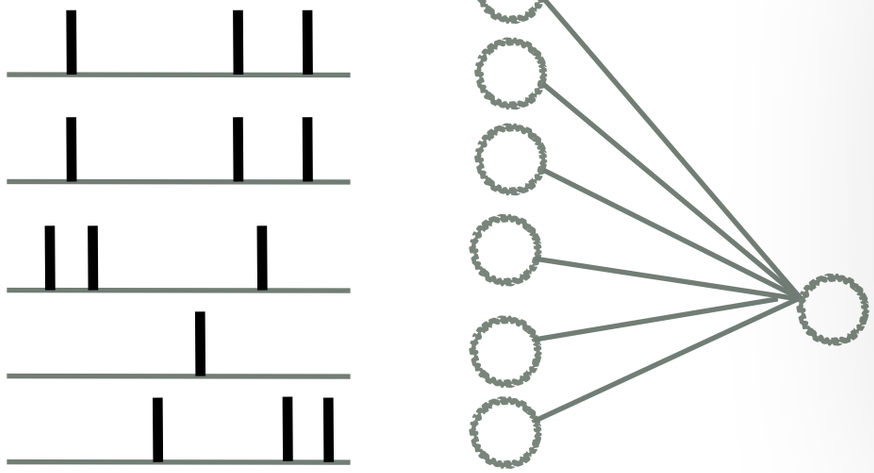


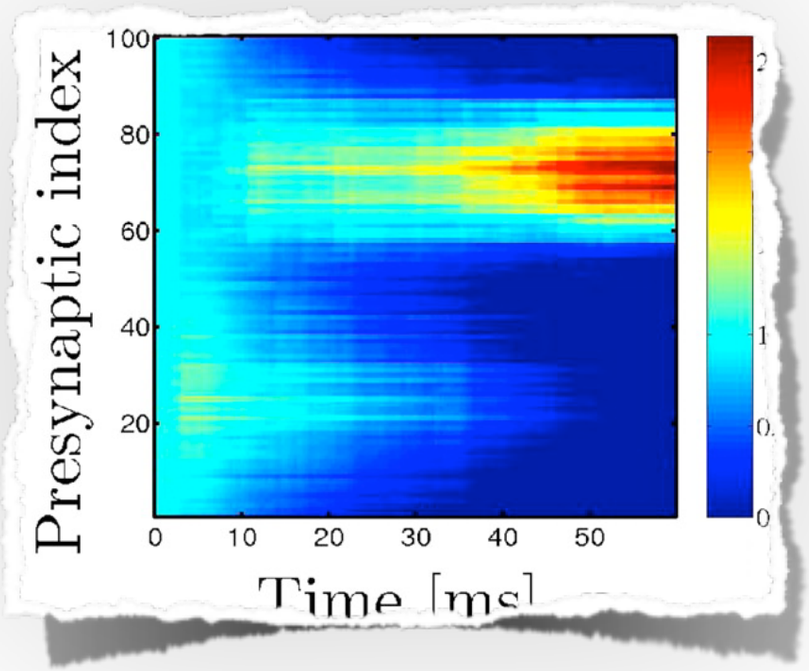
homeostasis

BCM

Jointly variant (correlated) input

Uncorrelated input





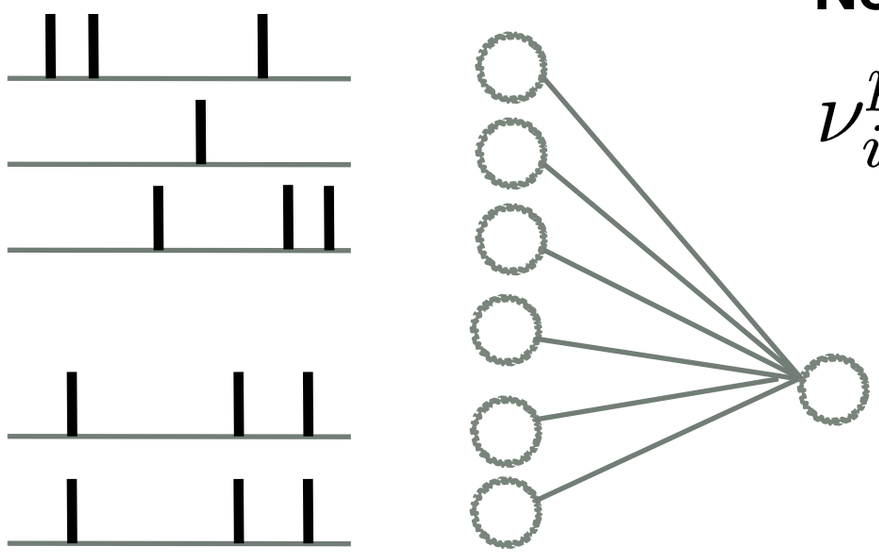
Weight evolution

$$\Delta w_{ij} = \alpha \Phi \left(\nu_i^{post} - \theta \right) \nu_j^{pre}$$

Is this a hebbian rule?

Detects where the ‘interesting’ input occurs (correlations): some synapses strengthened at the expense of others

BCM Rule



Neuron model

$$\nu_i^{post} = \sum_k w_{ik} \nu_k^{pre}$$

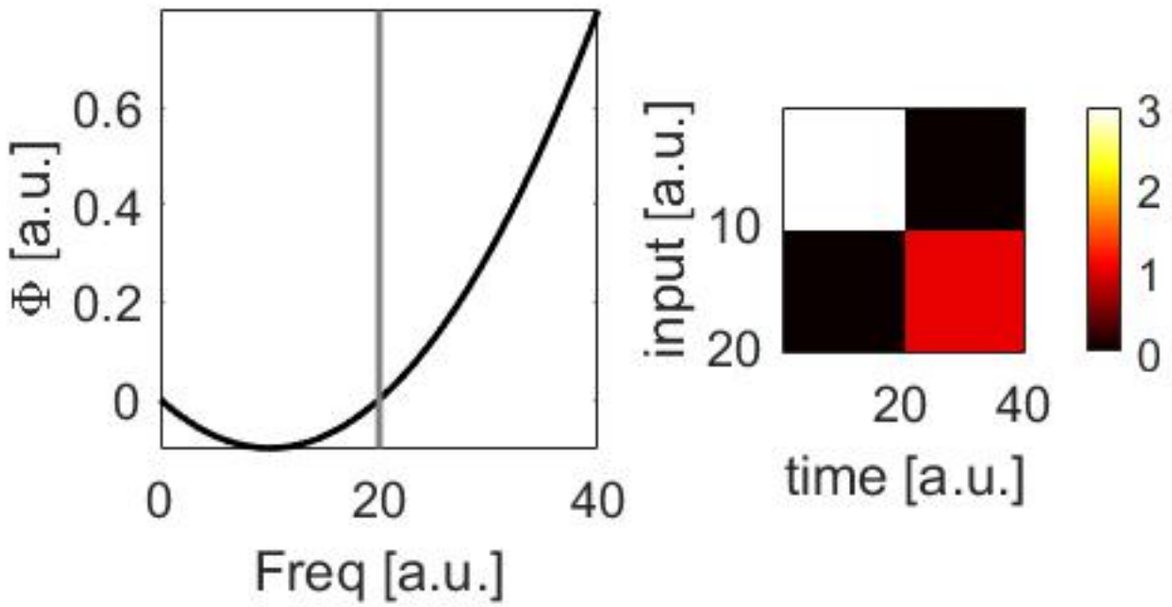
$$\Delta w_{ij} = \alpha \Phi \left(\nu_i^{post} - \theta \right) \nu_j^{pre}$$

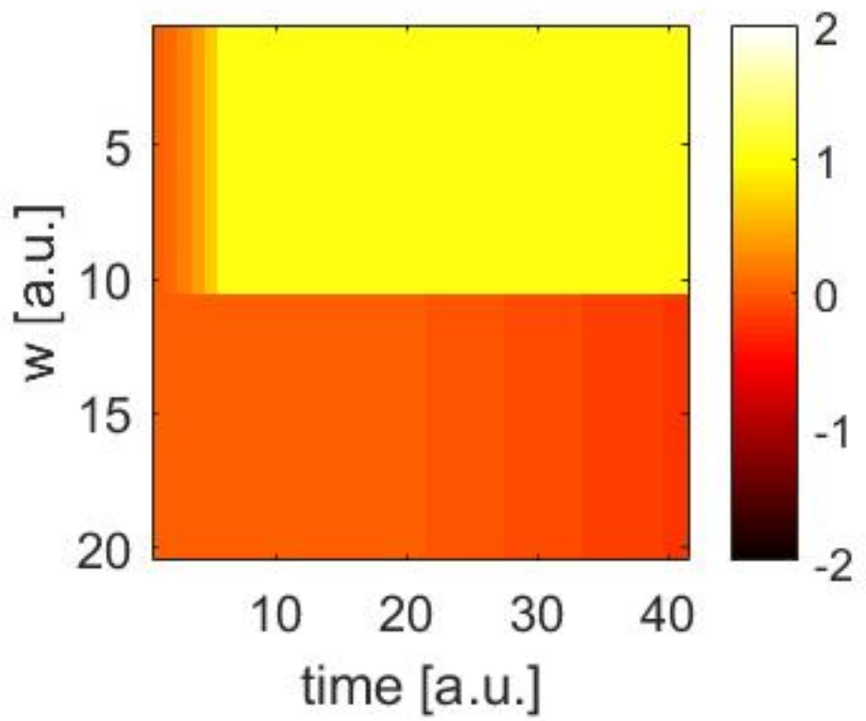
Assume two (presynaptic) groups of 10 neurons each. All weights = 1 and threshold *θ= 20Hz .*

1. Group 1 fires at 3 Hz, then group 2 at 1 Hz. What happens?
2. Group 1 fires at 3 Hz, then group 2 at 2.5 Hz. What happens?
3. As in 2, but make theta a function of the averaged rate. What happens?

BCM Rule

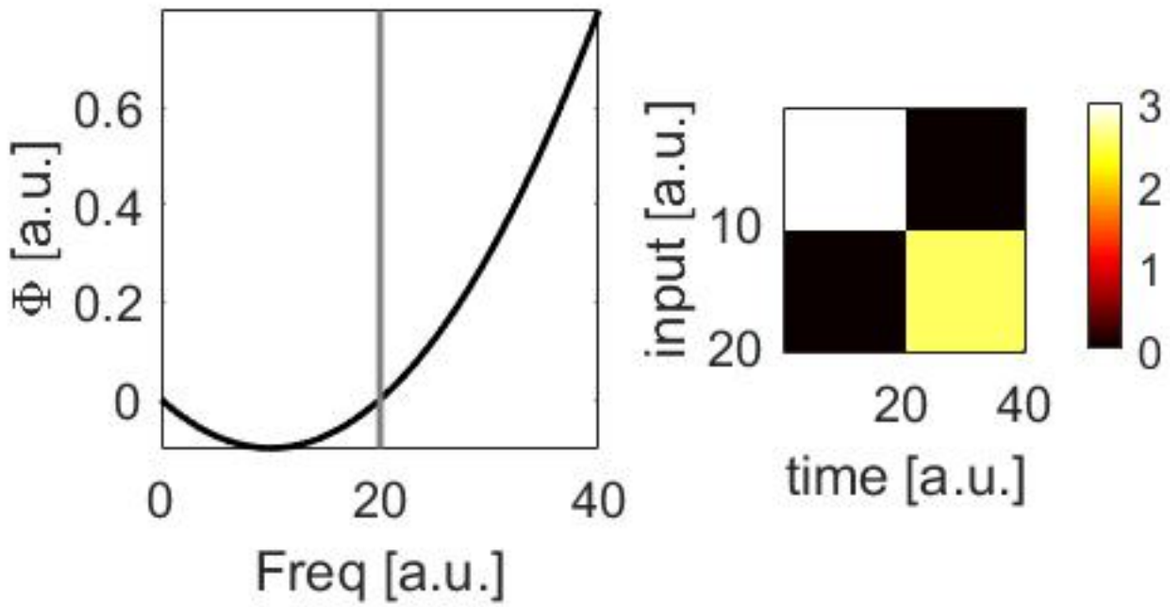
1.

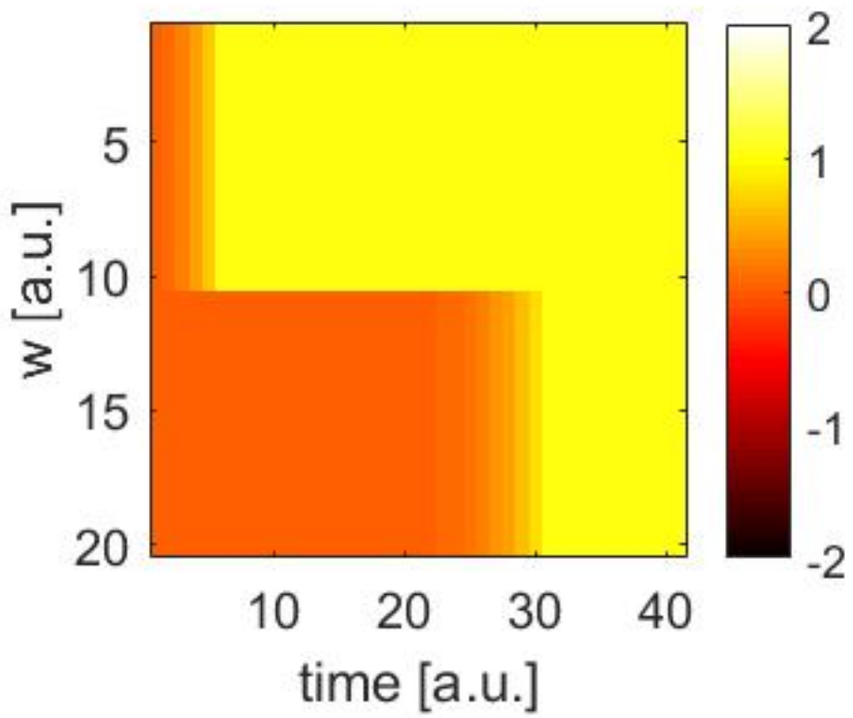

Φ [a.u.]
0.6
0.4
0.2
0
0
20
40
Freq [a.u.]
input [a.u.]
10
20
20
40
time [a.u.]
3
2
1
0


w [a.u.]
5
10
15
20
10
20
30
40
time [a.u.]
2
1
0
-1
-2

BCM Rule

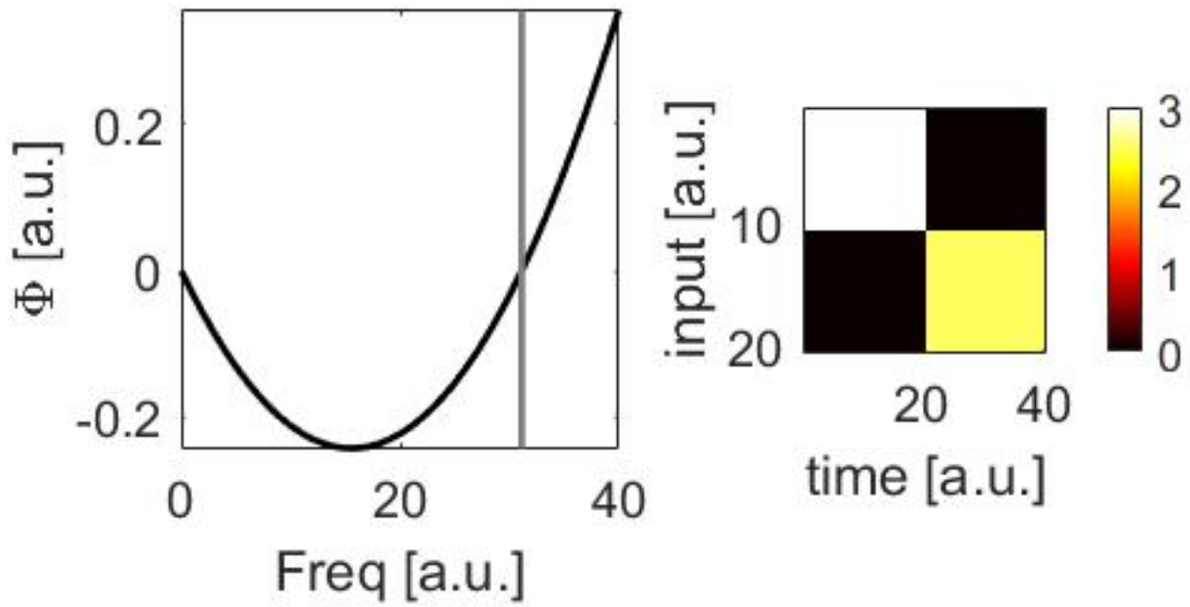
2.

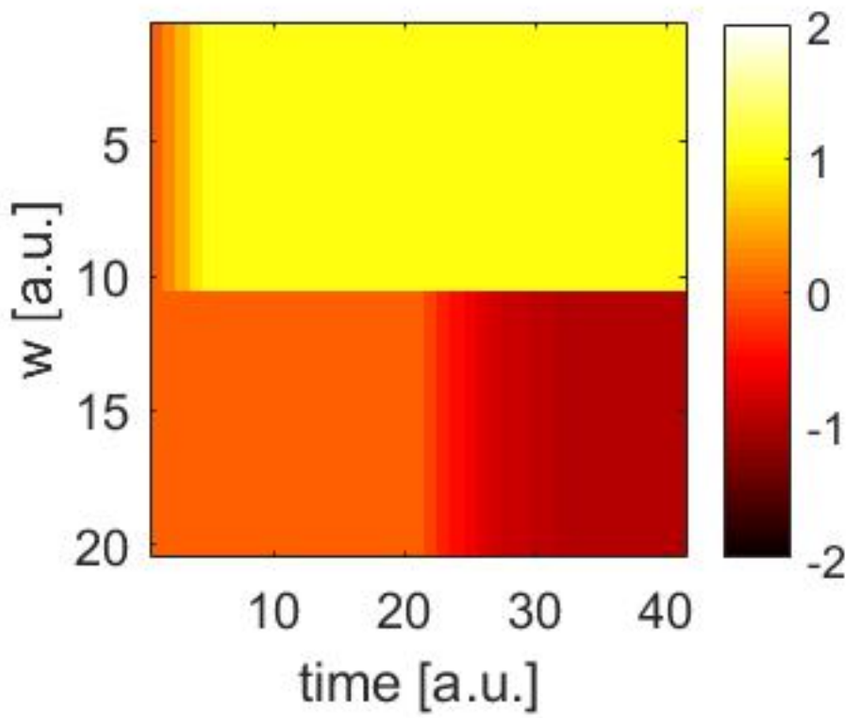

Φ [a.u.]
0.6
0.4
0.2
0
0
20
40
Freq [a.u.]
input [a.u.]
10
20
20
40
time [a.u.]
3
2
1
0


w [a.u.]
5
10
15
20
10
20
30
40
time [a.u.]
2
1
0
-1
-2

BCM Rule

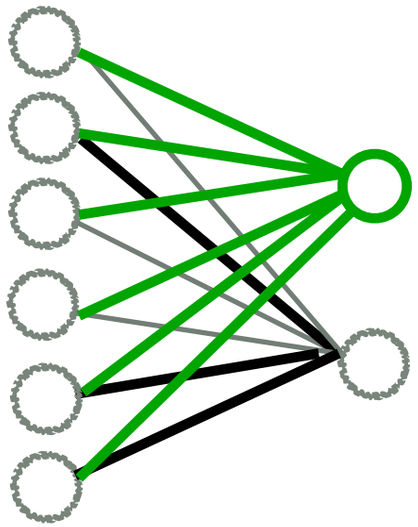
3.


Φ [a.u.]
0.2
0
-0.2
0
20
40
Freq [a.u.]
input [a.u.]
10
20
20
40
time [a.u.]
3
2
1
0


w [a.u.]
5
10
15
20
10
20
30
40
time [a.u.]
2
1
0
-1
-2

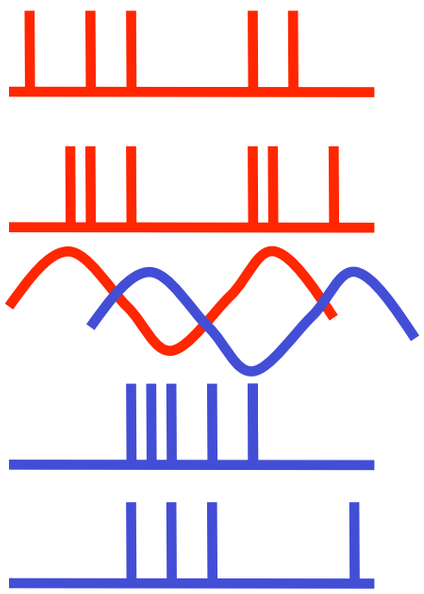
BCM leads to specialised neurons

Initial: random connections



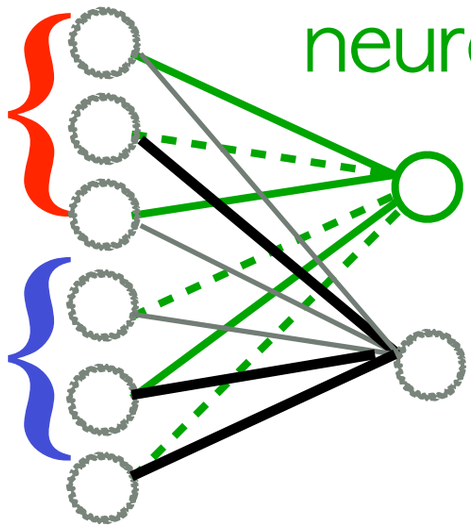
unselective neurons

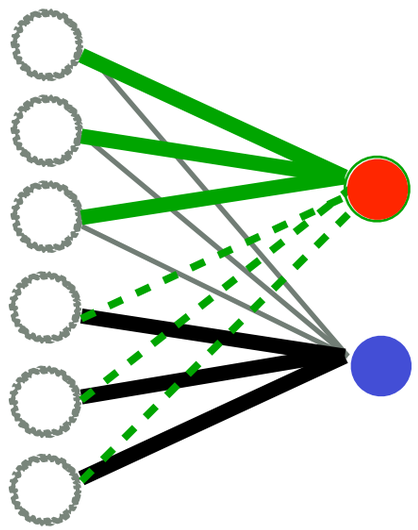
Receptive field development



Correlated input

output neurons





output neurons specialize

Summary

- Hebbian rules detect correlation in the input.
  - Mathematical proof.
  - True for all hebbian rules.
- The minimal hebbian rule is unstable.
- The BCM rule is a hebb rule with a homeostatic term.
  - Leads to specialised neurons.
  - Useful for receptive field development.

Thank you!

Acknowledgements

- This module is an adaptation of the MSc module “Unsupervised and Reinforcement Learning in Neural Networks” by Prof. Wulfram Gerstner at EPFL, Switzerland.

Bibliography

- Neuronal Dynamics, by Gerstner et al.
- Introduction to the Theory of Neural Computation" by Hertz, Krogh & Palmer
- Reinforcement Learning: An Introduction" by Sutton & Barto
- Scholarpedia