Neurorobotics

Module 1: Background and Foundations

Lecture 4: Learning and memory. Types of learning; neural networks basics

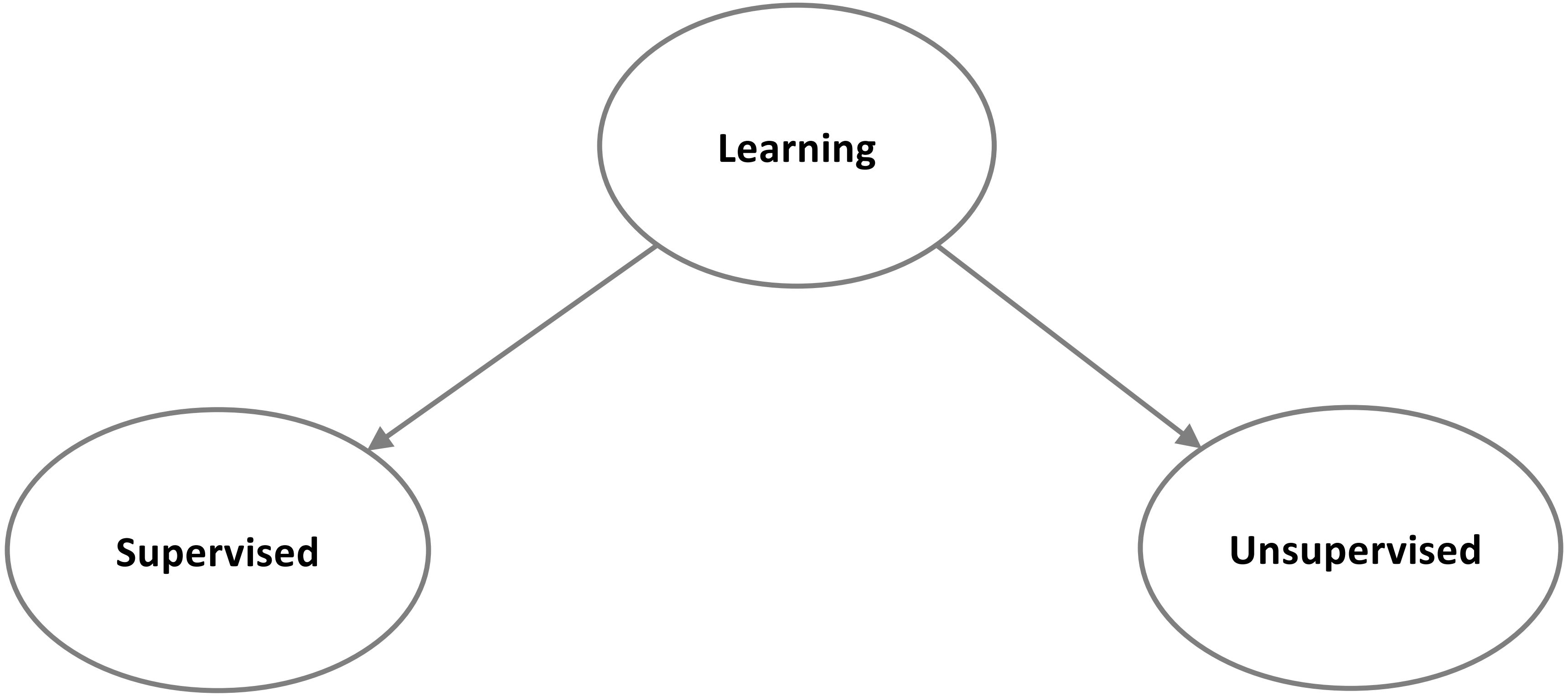
David Vernon
Carnegie Mellon University Africa

www.vernon.eu

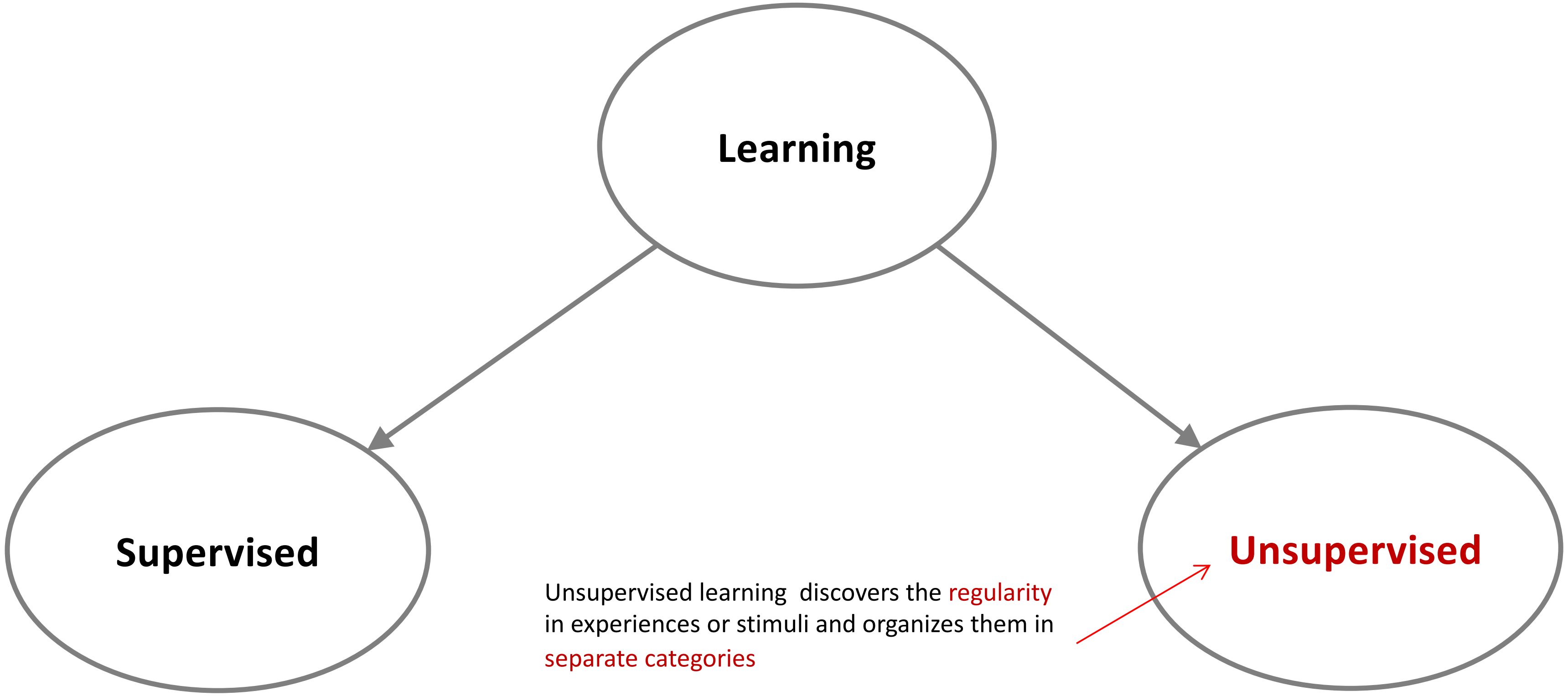
Learning and Memory

- Learning and memory enable neurorobots to adapt to a dynamic environment
- Without learning, adaptation can only take place by genetic mutation
  - Within species
  - Over many generations
- With learning, adapation happens with the lifetime of the organism
  - Learn from experience
  - Learn new associations between stimuli
  - Adapt associations when the environment changes

Types of Learning



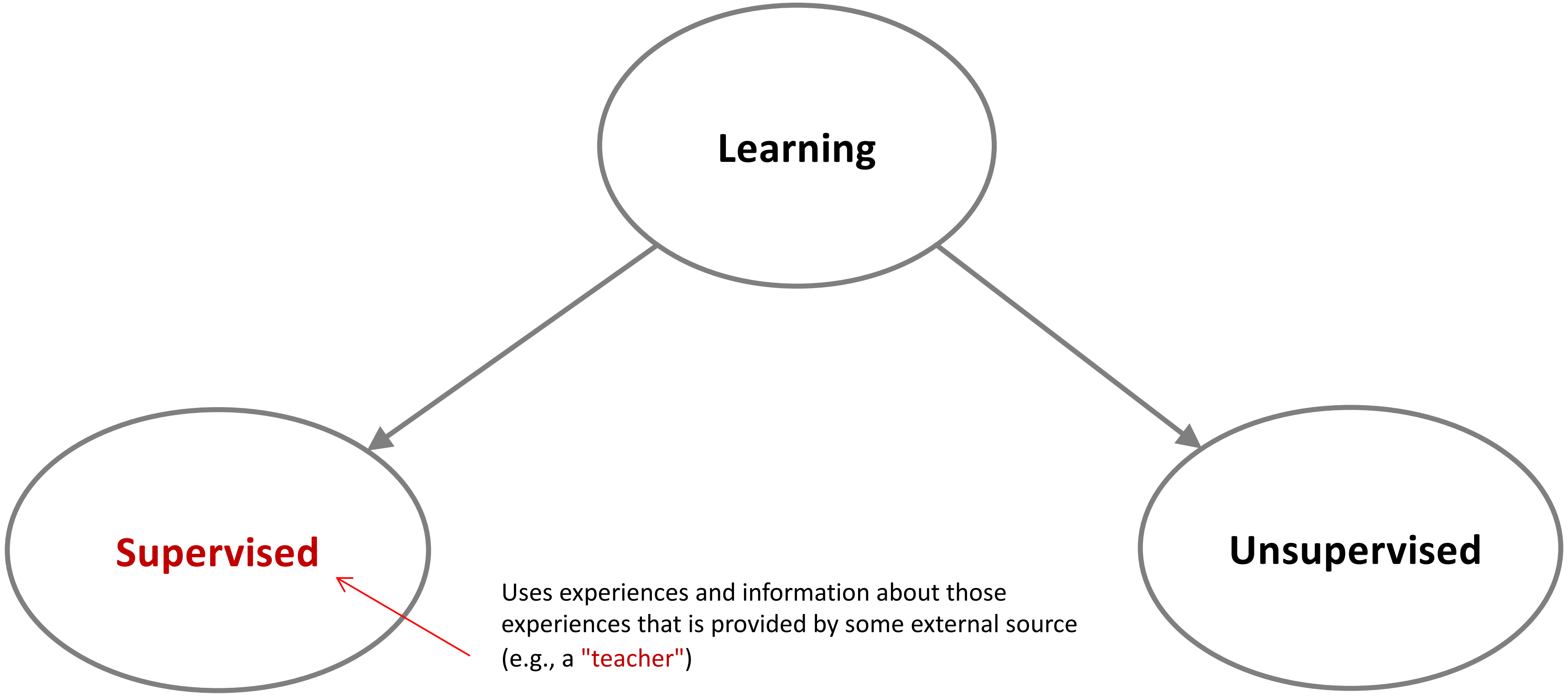
Types of Learning



Unsupervised learning discovers the regularity in experiences or stimuli and organizes them in separate categories

No "teacher"

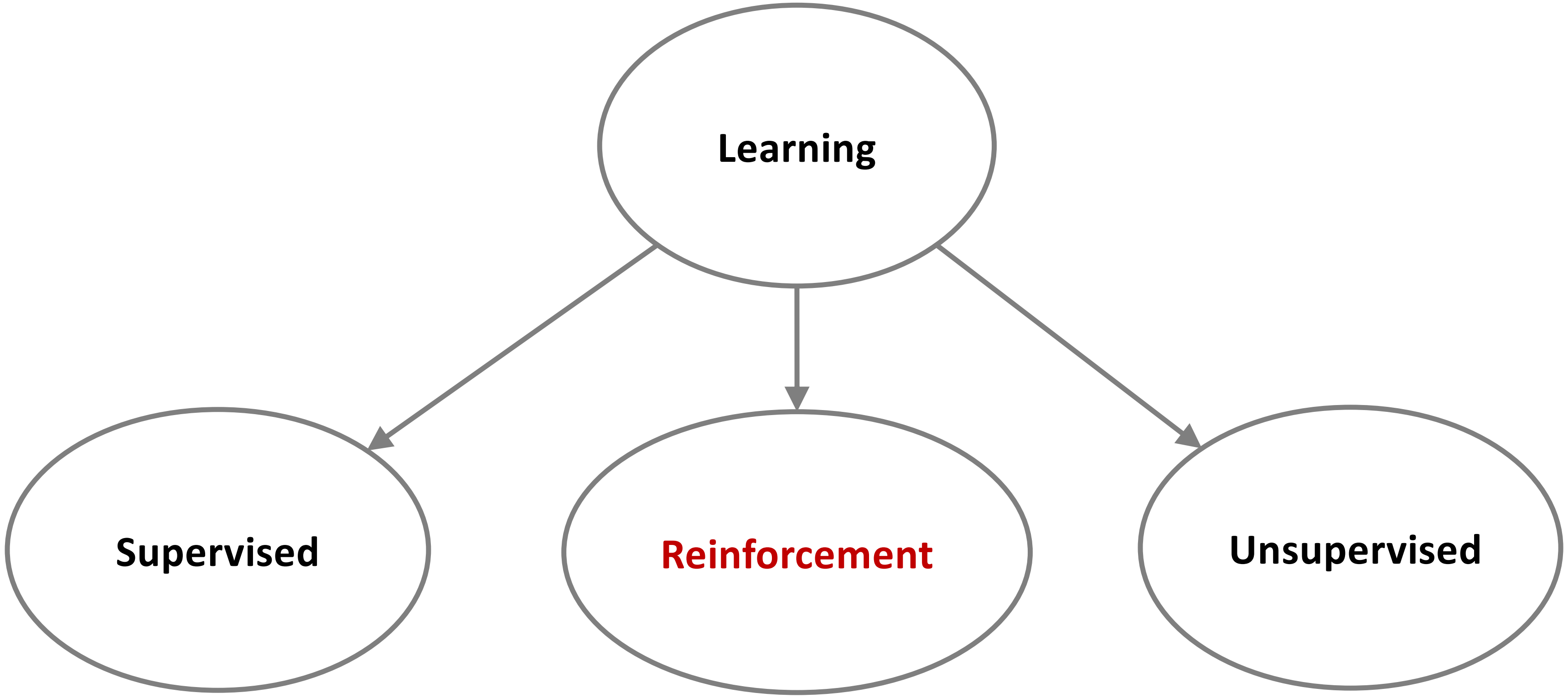
Types of Learning



Uses experiences and information about those experiences that is provided by some external source (e.g., a "teacher")

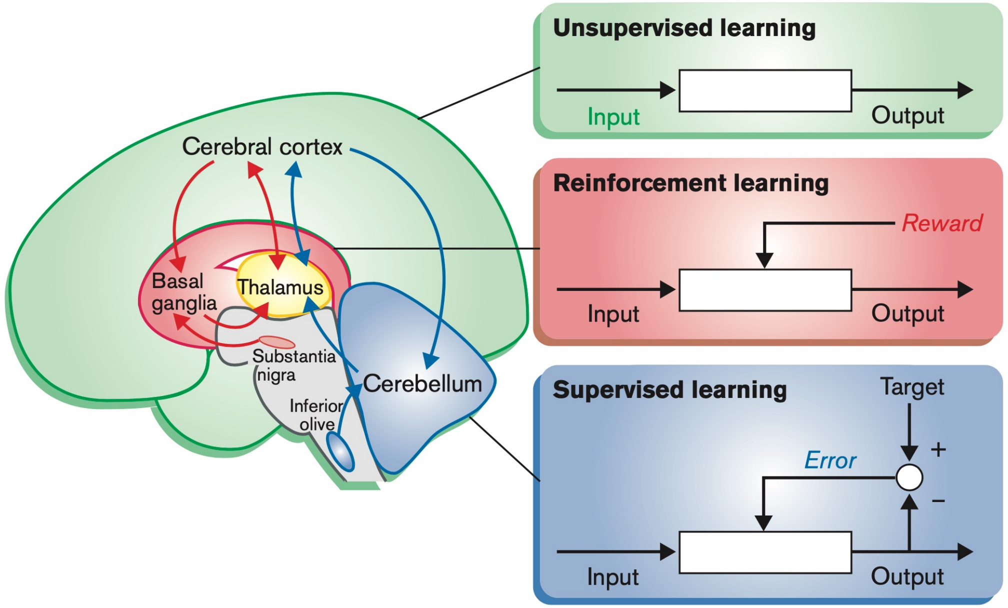
It is rare to have pure supervised learning in biological systems as the brain modulates (or interprets) the signal (or label) provided by the teacher based on some internal value system

Types of Learning



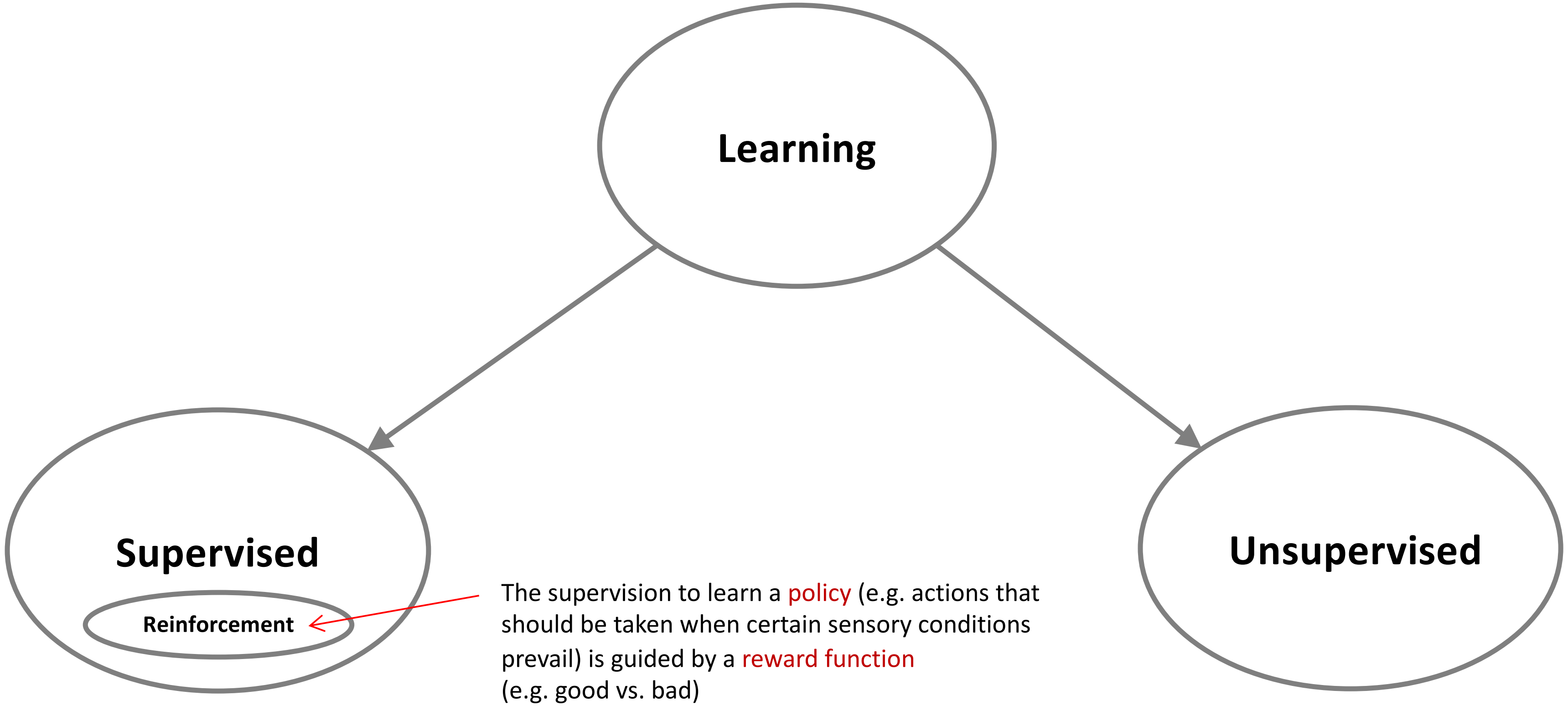
Uses positive or negative feedback from experience in the environment (e.g., a reward)

Types of Learning



K. Doya, 2000. Complementary roles of basal ganglia and cerebellum in learning and motor control, Current Opinion in Neurobiology, Vol. 10, pp. 732-739.

Types of Learning

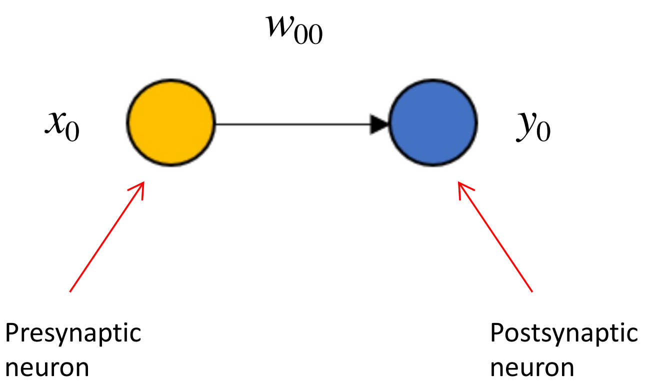


The supervision to learn a policy (e.g. actions that should be taken when certain sensory conditions prevail) is guided by a reward function (e.g. good vs. bad)

The cost function (or objective function) that governs the learning is based on maximizing the cumulative sum of rewards over time.

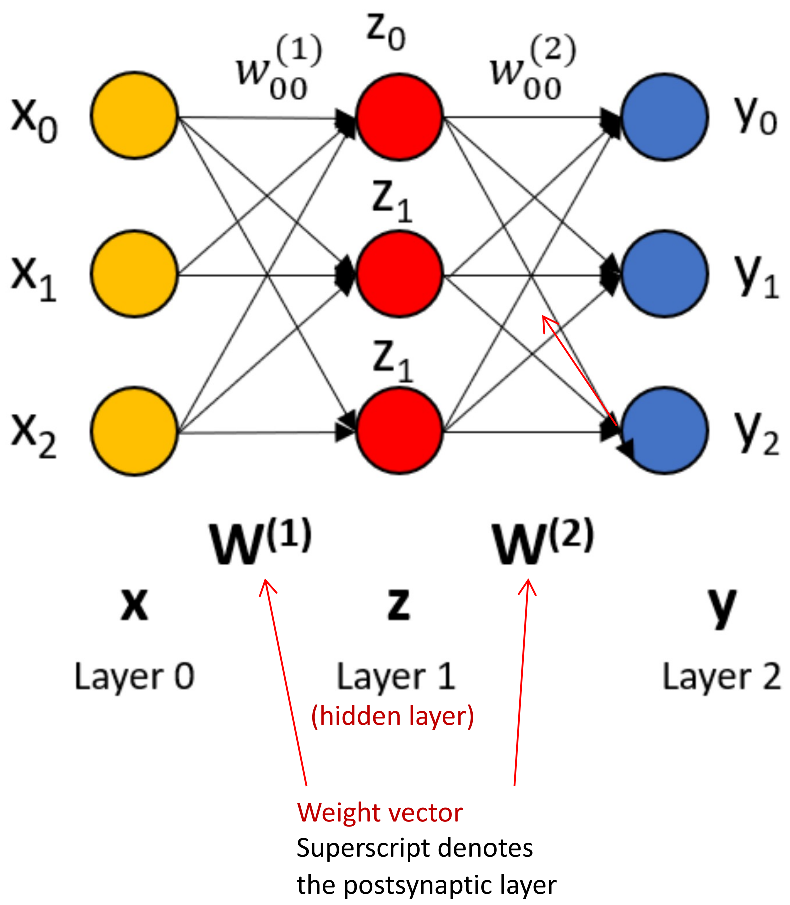
Neural Network Basics

- Neurons are also referred to as units
  - Input unit, e.g., x_0
  - Output unit, e.g., y_0
- Synapses are also referred to as connections or weights
  - Weight, e.g., w_{00}
  - Subscripts denote the presynaptic & postsynaptic neuron, respectively
  - In this example $y_0 = w_{00} x_0$



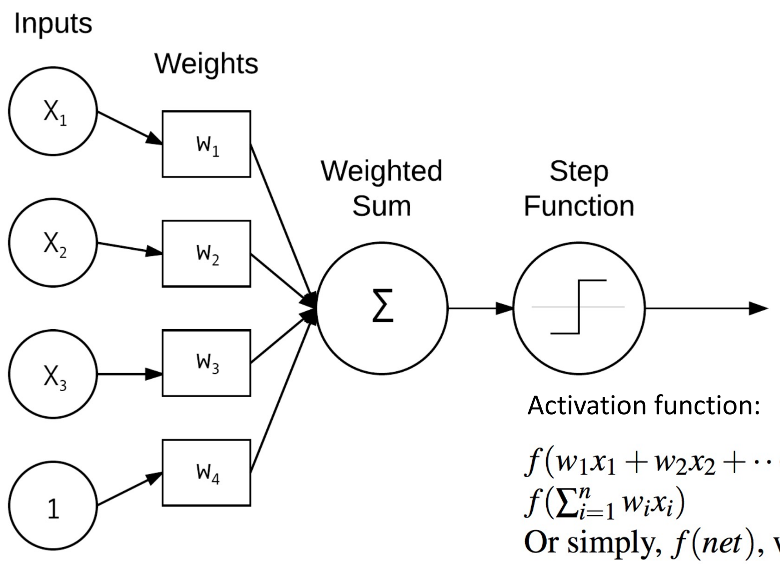
Neural Network Basics

- Multiple inputs
- Multiple outputs
- Multiple layers (hidden layers)



Neural Network Basics

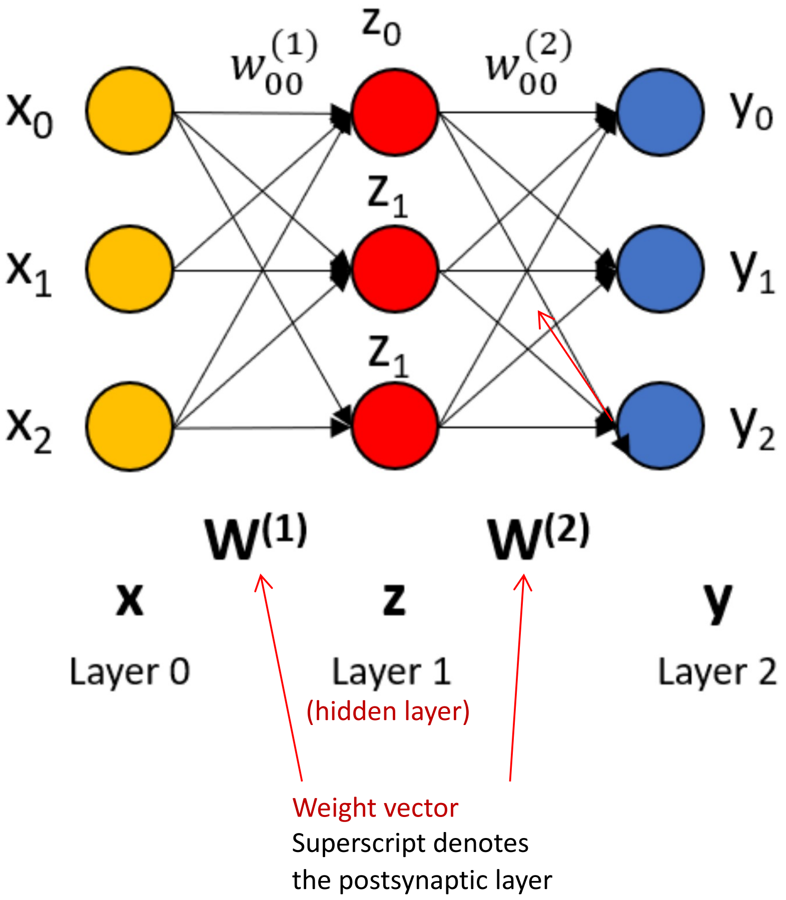
Appendix 1 specifies how the value of y_0 and z_0 are computed



Activation function:

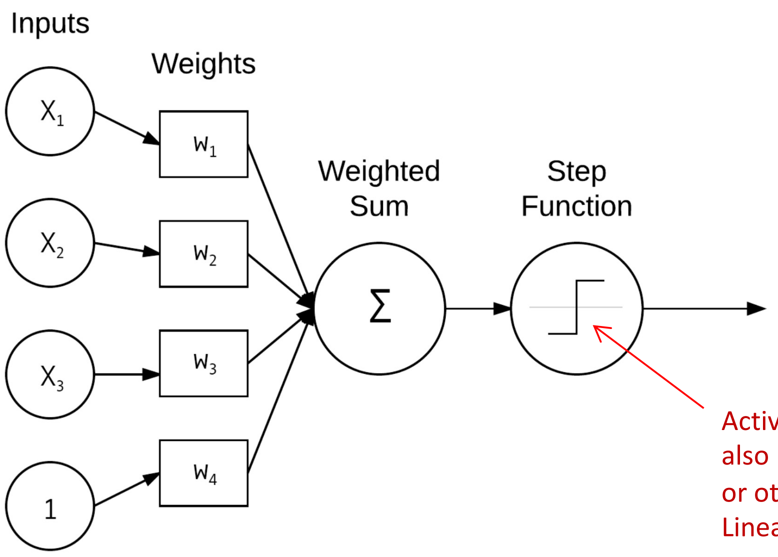
$f(w_1x_1 + w_2x_2 + \cdots + w_nx_n)$
$f(\sum_{i=1}^{n} w_ix_i)$
Or simply, $f(net)$, where $net = \sum_{i=1}^{n} w_ix_i$

Credit: Adrian Rosebrock, Deep Learning for Computer Vision, PyImageSearch, 2017



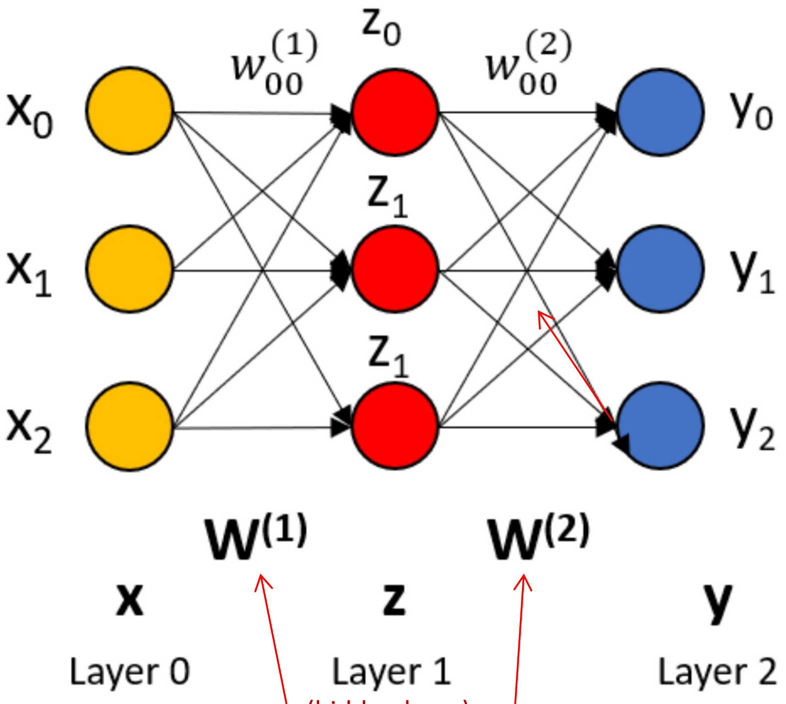
Neural Network Basics

Appendix 1 specifies how the value of y_0 and z_0 are computed



Activation function can also be sigmoid, tanh, or other, e.g. Rectified Linear Unit (ReLU)

Credit: Adrian Rosebrock, Deep Learning for Computer Vision, PyImageSearch, 2017

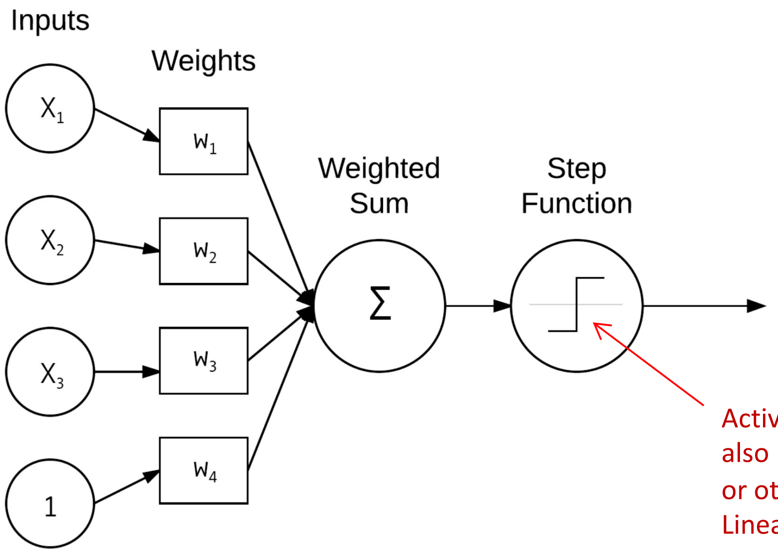


(hidden layer)

Weight vector
Superscript denotes the postsynaptic layer

Neural Network Basics

Appendix 1 specifies how the value of y_0 and z_0 are computed



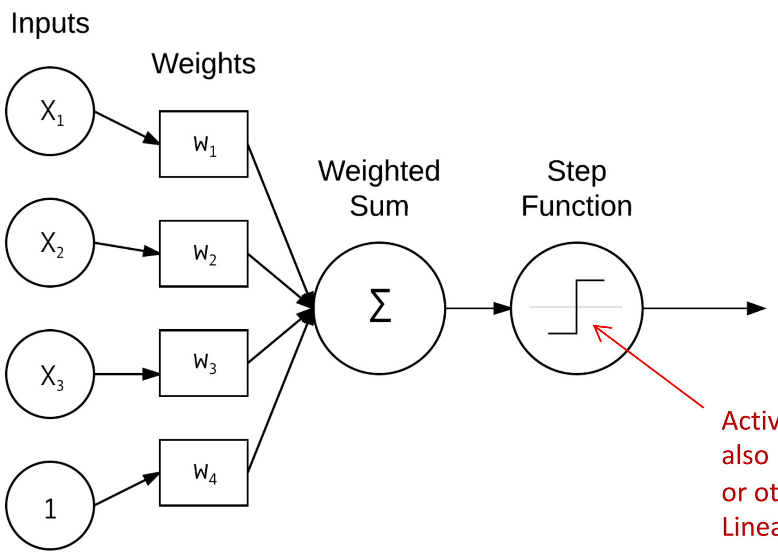
Activation function can also be sigmoid, tanh, or other, e.g. Rectified Linear Unit (ReLU)

Credit: Adrian Rosebrock, Deep Learning for Computer Vision, PyImageSearch, 2017

Type of function	Graphical represent.	Mathematical formula	MATLAB implementation
Linear		$g^{\text{lin}}(x) = x$	`x`
Step		$g^{\text{step}}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{elsewhere} \end{cases}$	`floor(0.5*(1+sign(x)))`
Threshold-linear		$g^{\text{theta}}(x) = x\,\Theta(x)$	`x.*floor(0.5*(1+sign(x)))`
Sigmoid		$g^{\text{sig}}(x) = \frac{1}{1+\exp(-x)}$	`1./(1+exp(-x))`
Radial-basis		$g^{\text{gauss}}(x) = \exp(-x^2)$	`exp(-x.^2)`

Neural Network Basics

Appendix 1 specifies how the value of y_0 and z_0 are computed

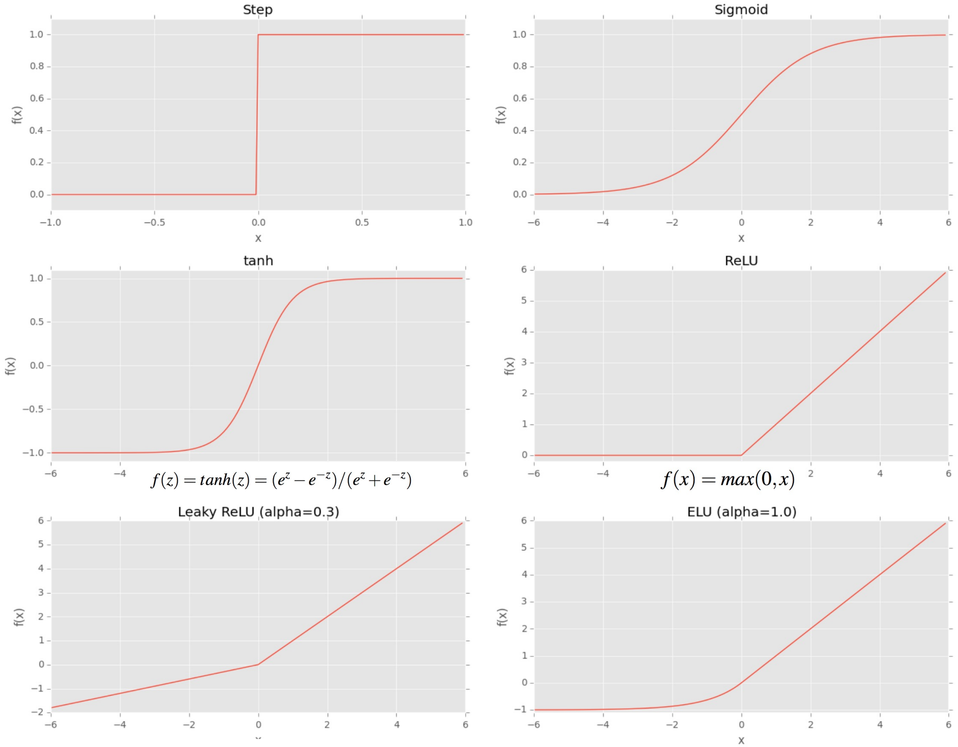


Activation function can also be sigmoid, tanh, or other, e.g. Rectified Linear Unit (ReLU)

Credit: Adrian Rosebrock, Deep Learning for Computer Vision, PyImageSearch, 2017

$$f(net) = \begin{cases} 1 & \text{if } net > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$s(t) = 1/(1+e^{-t}) \qquad t = \sum_{i=1}^{n} w_i x_i$$



$$f(net) = \begin{cases} net & \text{if } net >= 0 \\ \alpha \times net & \text{otherwise} \end{cases}$$

$$f(net) = \begin{cases} net & \text{if } net >= 0 \\ \alpha \times (exp(net) - 1) & \text{otherwise} \end{cases}$$

Hebbian Learning

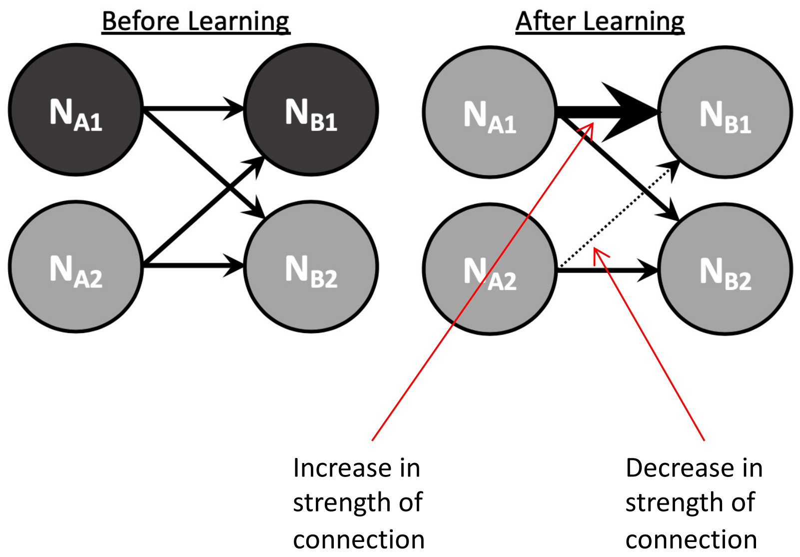
(Hebb, 1949)

- Unsupervised learning
- Captures associations between two stimuli
- The synaptic strength — the bond between connecting neurons — is increased if both neurons are active at the same time
- Neurons that fire together, wire together

Hebbian Learning

(Hebb, 1949)

- Assuming two groups of neurons, A and B
  - Two neurons per group
- If N_{A1} is co-active with N_{B1}
  - This will cause the strength of the connection between N_{A1} and N_{B1} to increase
  - It may also cause a decrease in the strength of the connection between N_{A2} and N_{B1} to increase
- The next time N_{A1} , there is a better chance that N_{B1} is more active



Hebbian Learning

(Hebb, 1949)

Unsupervised learning

- Driven by intrinsic co-activity rather than some extrinsic supervisory signal
- Can find consistencies in environmental stimuli that are important for the organism
- Sometimes referred to as experience-dependent learning
- In neurorobotics, it can result in learning categories or classes of information based on exploration
- In cognitive science, this classification is called perceptual categorization

Hebbian Learning

(Hebb, 1949)

Neuromodulated Hebbian learning

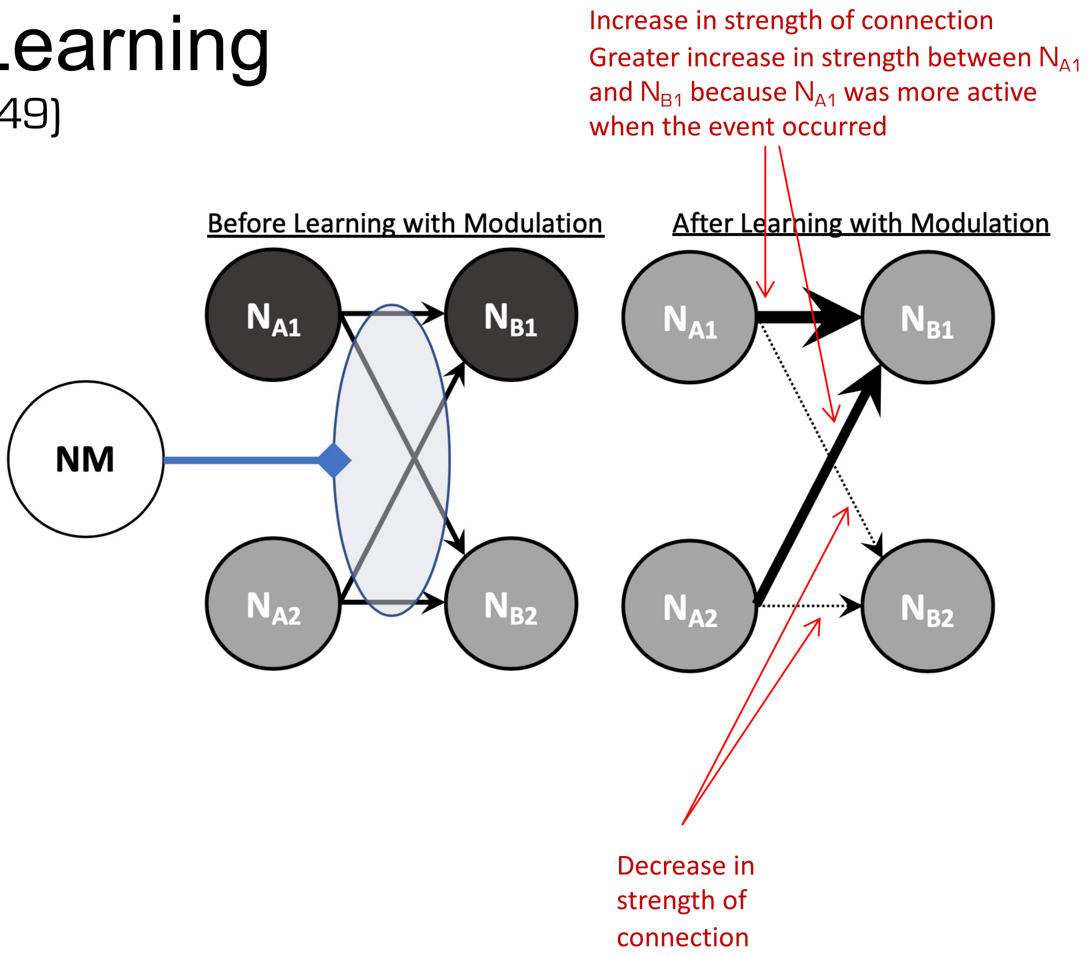
- Learning significant events or associations
- A modulation signal can boost learning in the brain
  - Neuromodulation is effected through the release of neurotransmitters by neurons
- Neuromodulators
  - Dopamine
  - Seratonin
  - Noradrenaline
  - Acetylcholine

Hebbian Learning

(Hebb, 1949)

Neuromodulated Hebbian learning

- Neuron NM is a neuromodulator that signals an important event and releases its neurotransmitter to all neurons
    - Connections between active neurons are potentiated
    - Other connections are depressed
- The next time the event occurs, N_{B1} will be active

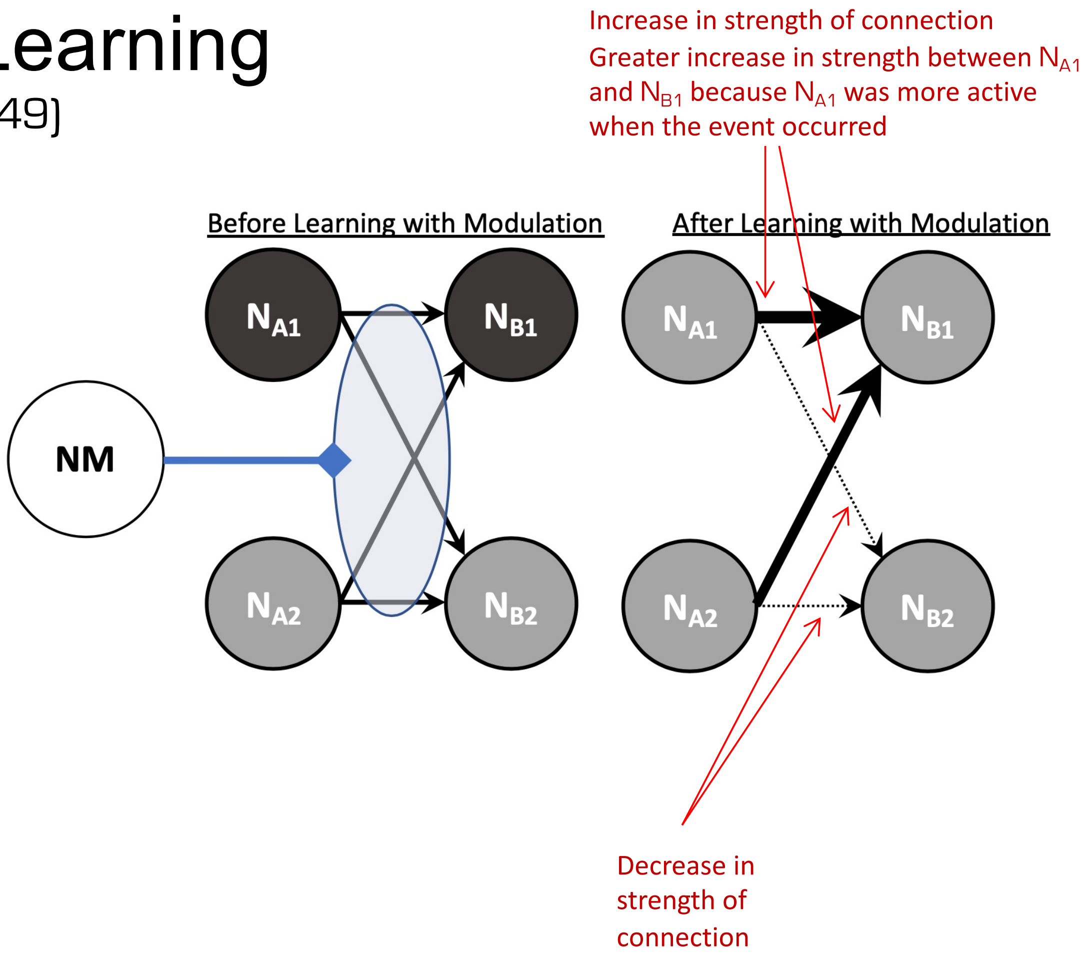


Hebbian Learning

(Hebb, 1949)

Neuromodulated Hebbian learning

- The next time the event occurs, N_{B1} will be active
- For example, if the event was seeing of a predator
  - Signalled by N_{A1} or N_{A2}
  - N_{B1} might trigger an escape response



Hebbian Learning

(Hebb, 1949)

Hebbian learning rule

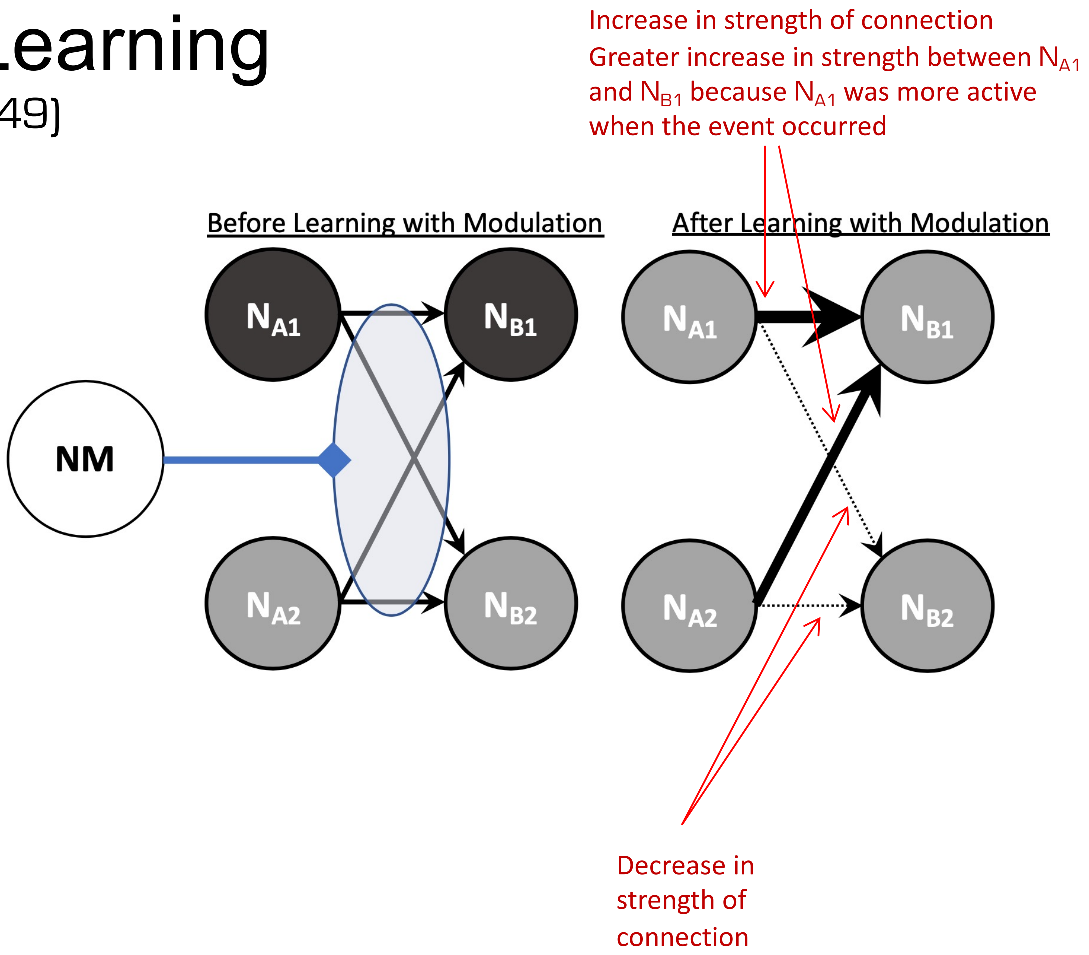
- Weight update function

$$\Delta w_{ij} = \alpha \, x_i x_j$$

- α is the learning rate (between 0 and 1)

- Weight will increase in proportion to the co-activity of the presynaptic and postsynaptic neurons

$$x_j = w_{ij} x_i$$



Hebbian Learning

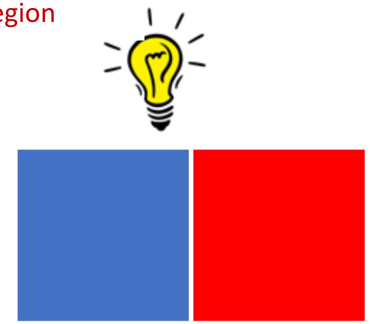
(Hebb, 1949)

Example

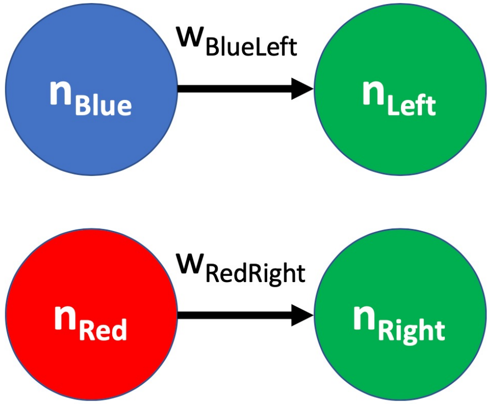
– Neurorobot task

- Explore an area colored blue on one side and red on the other
- Determine which side is brighter
- Input neurons represent the brightness of the blue and red sides
- Output neurons represent the robot's movement: move to the left or move the the right

Note: the light is closer to the blue region



B.



Hebbian Learning

(Hebb, 1949)

Example

– Initial values of the network

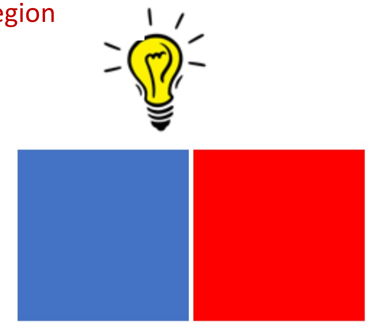
Learning rate: $\alpha = 0.25$;

Output ~~Input~~ neurons: $n_{left} = n_{right} = 0.5$;

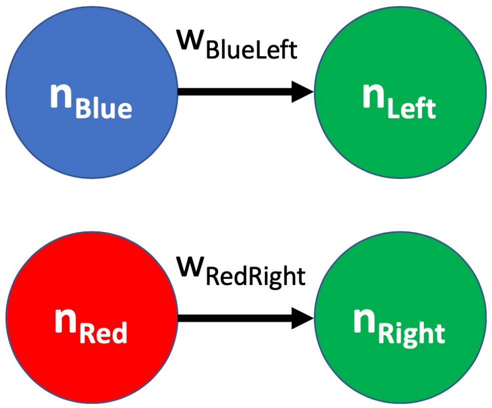
Input ~~Output~~ neurons: $n_{blue} = 0.75$; $n_{red} = 0.25$;

Weights $w_{BlueLeft} = w_{RedRight} = 0.5$;

Note: the light is closer to the blue region



B.



Hebbian Learning

(Hebb, 1949)

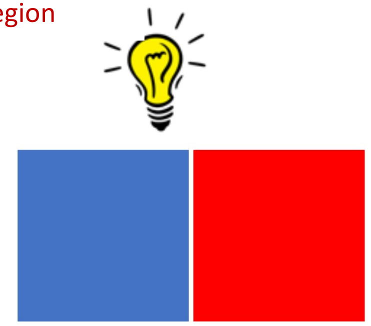
Example

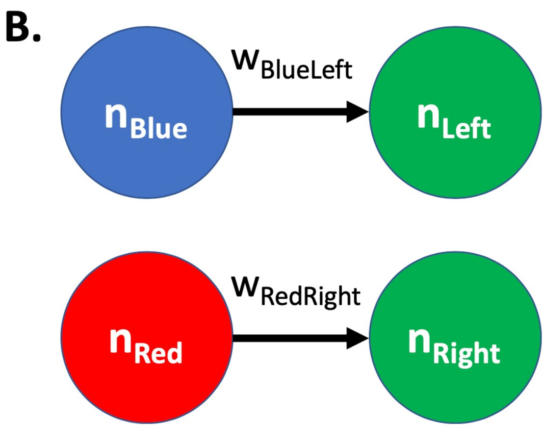
– Iterate weight update ten times

1: n_{Left}=0.445, $w_{BlueLeft}$=0.594, n_{Right}=0.133, $w_{RedRight}$t=0.531
2: n_{Left}=0.508, $w_{BlueLeft}$=0.677, n_{Right}=0.135, $w_{RedRight}$=0.54
3: n_{Left}=0.579, $w_{BlueLeft}$=0.772, n_{Right}=0.137, $w_{RedRight}$=0.548
4: n_{Left}=0.661, $w_{BlueLeft}$=0.881, n_{Right}=0.139, $w_{RedRight}$=0.557
5: n_{Left}=0.754, $w_{BlueLeft}$=1.01, n_{Right}=0.141, $w_{RedRight}$=0.565
6: n_{Left}=0.86, $w_{BlueLeft}$=1.15, n_{Right}=0.144, $w_{RedRight}$=0.574
7: n_{Left}=0.981, $w_{BlueLeft}$=1.31, n_{Right}=0.146, $w_{RedRight}$=0.583
8: n_{Left}=1.12, $w_{BlueLeft}$=1.49, n_{Right}=0.148, $w_{RedRight}$=0.592
9: n_{Left}=1.28, $w_{BlueLeft}$=1.7, n_{Right}=0.15, $w_{RedRight}$=0.601
10: n_{Left}=1.46, $w_{BlueLeft}$=1.94, n_{Right}=0.153, $w_{RedRight}$=0.611

Motor control to move left is stronger than to move right

Note: the light is closer to the blue region





Hebbian Learning

(Hebb, 1949)

More realistic Hebbian learning scenarios

- Increase the number of input
    - For example, place cells (neurons) in the brain (hippocampus) can be modelled as the association of many inputs (e.g. landmarks) with one output (location)
    - When the landmarks are. activated in the brain, a place cell corresponding to that place activates
- Increase the number of layers
- Modify the learning rule: many variants of the Hebbian learning rule exist

Reading

Hwu, T. and Krichmar, J. (2022). Neurorobotics: Connecting the Brain, Body and Environment, MIT Press.

Chapter 3, Sections 3.1 - 3.3, pp. 45 - 52

References

Hebb, D. O. (1949). The Organization of Behavior, John Wiley & Sons.

The Organization of Behavior

A NEUROPSYCHOLOGICAL THEORY

D. O. HEBB
McGill University

1949
New York · JOHN WILEY & SONS, *Inc.*
London · CHAPMAN & HALL, *Limited*

Videos

G. Sanderson, Neural Networks, 3Blue1Brown.

https://www.youtube.com/playlist?list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi