EENG582
Artificial Neural Networks

Learning Process

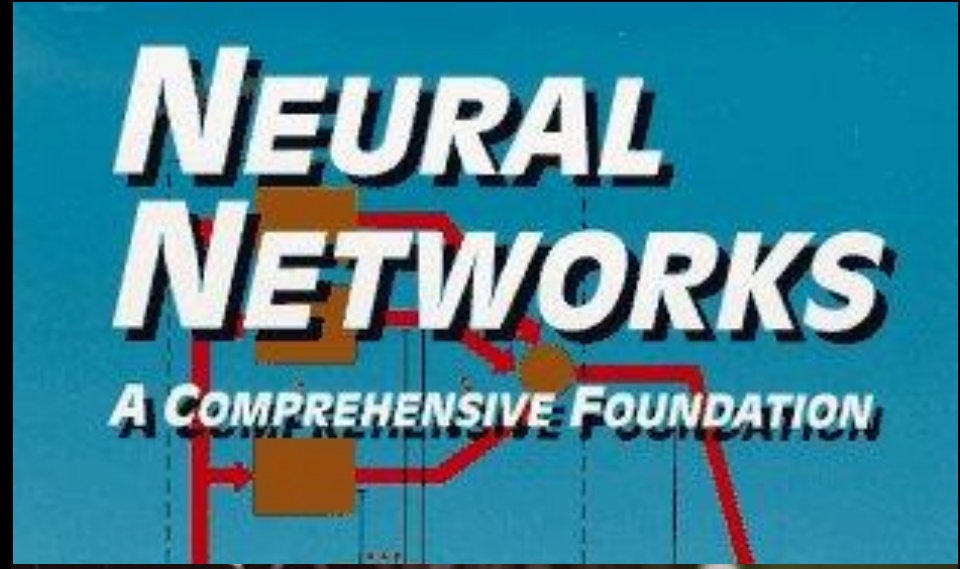
Sections 2.1 – 2.7
The Learning Process

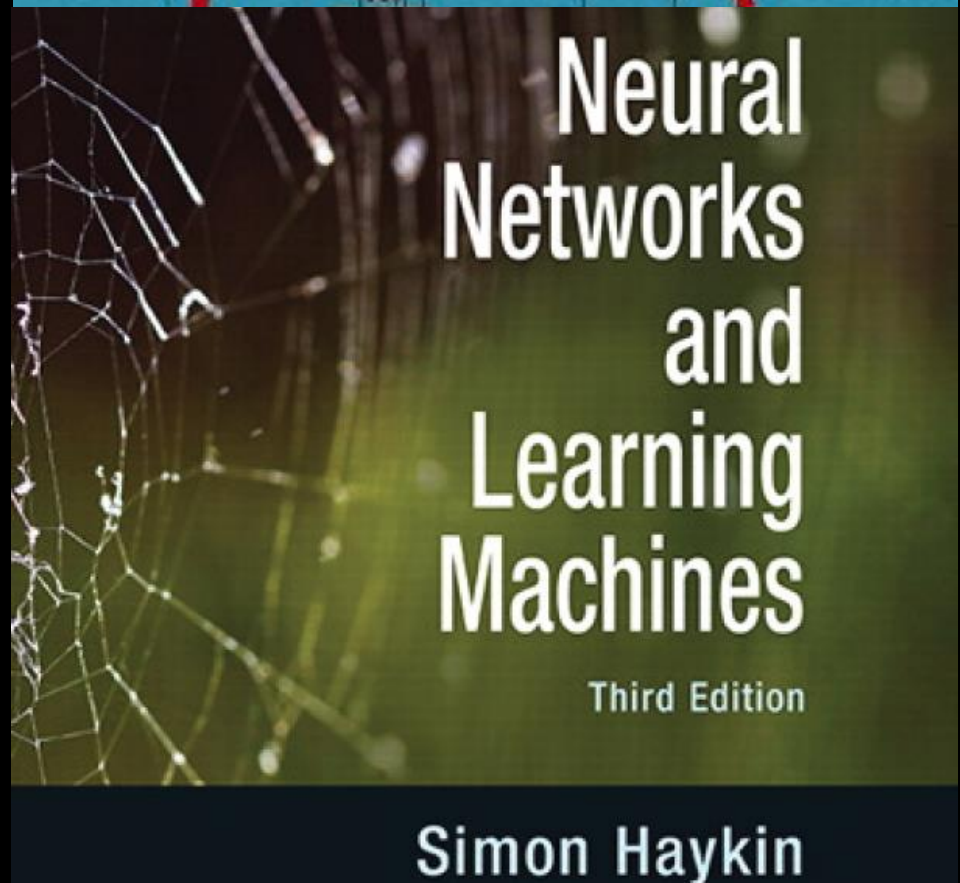
Artificial Neural Networks

Prof. Dr. Hasan AMCA

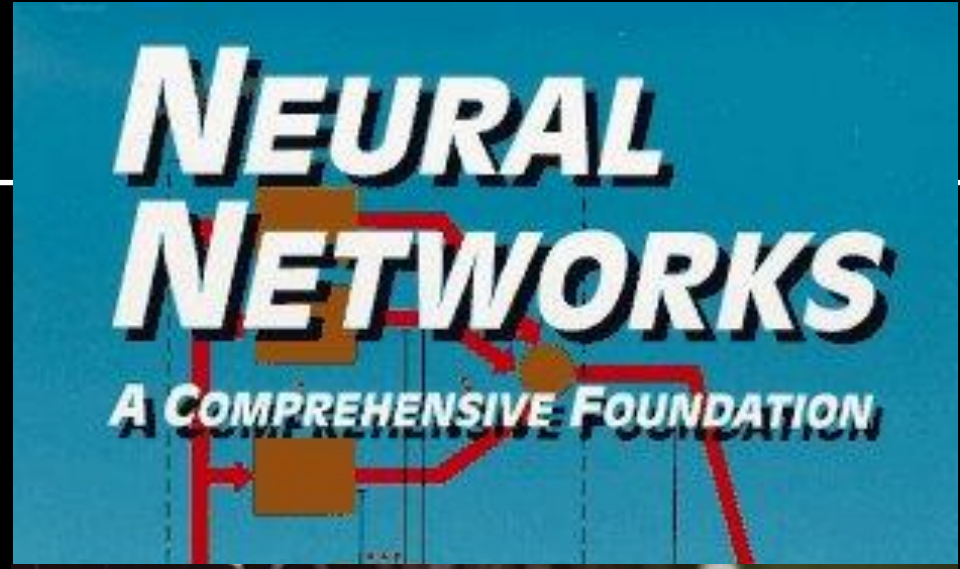
Electrical and Electronic Engineering Department
(ee.emu.edu.tr)

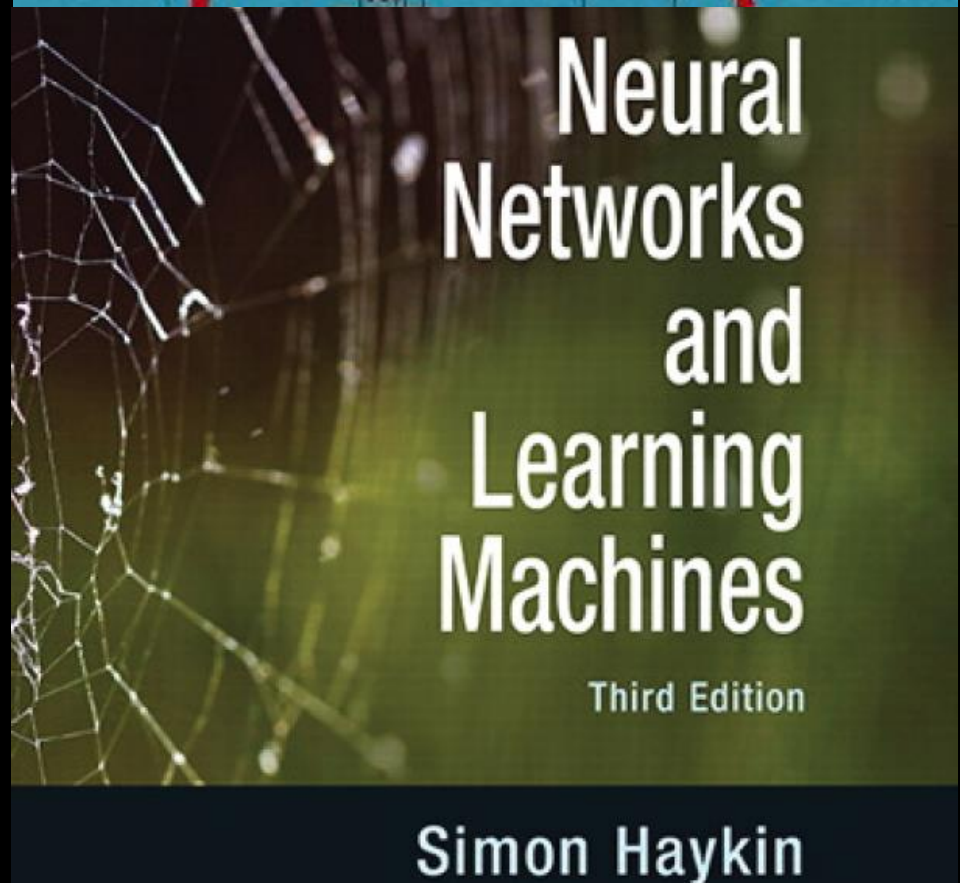
Eastern Mediterranean University
(emu.edu.tr)





2 The Learning Process





2.1 Introduction

- Neural networks learn from its environment and improve its performance.
- A neural network learns its environment through an interactive process of adjustments applied to its synaptic weights and bias.
- We define learning in the context of neural networks as:

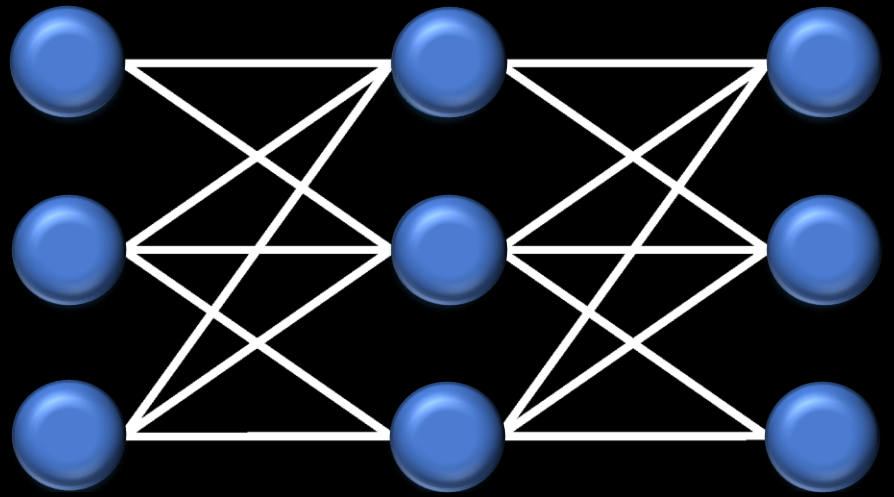
> *Learning is a process by which the free parameters of a neural network are adapted through a process of stimulation by the environment in which the network is embedded.*

- The type of learning is determined by how the parameter changes take place.

2.1 Introduction

Reinforcement Learning
Decision process
Reward system
Learn series of actions

Supervised Learning
Labeled data
Direct feedback
Predict outcome



Unsupervised Learning
No label
No feedback
Find hidden structures

Deep Learning
Feature extraction
Classification

https://www.mdpi.com/1424-8220/20/1/109/htm

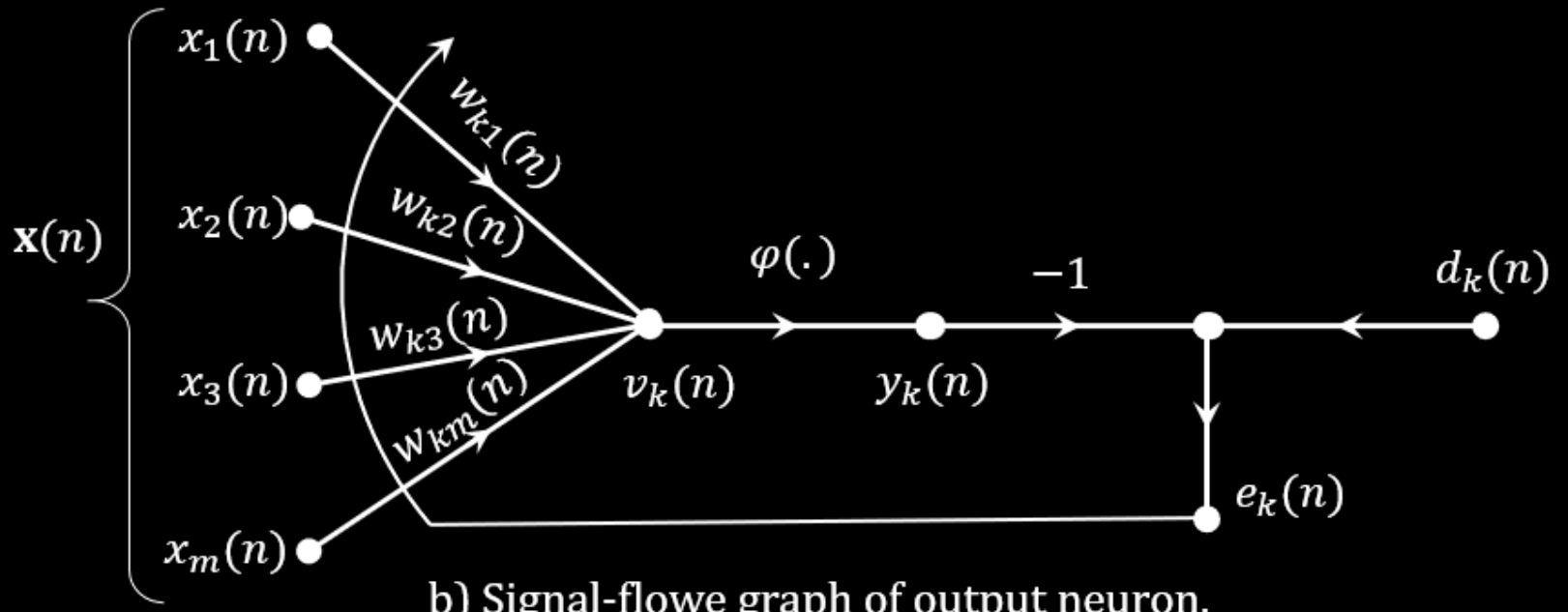
2.1 Introduction

- Definition of learning process implies the following sequence of events:
  1. The neural network is stimulated by its environment.
  2. NN undergoes changes in its free parameters as a result of this stimulation.
  3. NN responds in a new way to the environment because of the changes that have occurred in its internal Structure (i.e. weights).

Learning is a process by which the free parameters of a neural network are adapted through a process of stimulation by the environment in which the network is embedded.

2.2 Error-Correction Learning

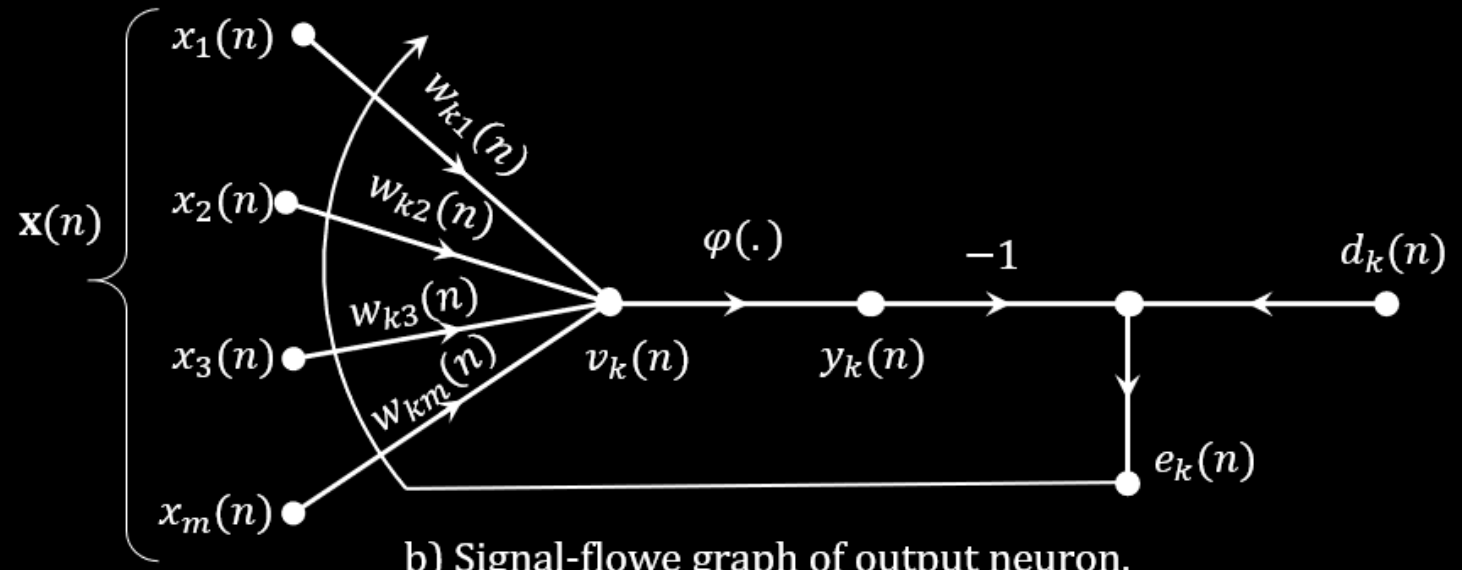
- For the first learning rule, consider simple case of a neuron k constituting the only computational node in the output layer of a feedforward neural network, as depicted in Fig. 2.1a.
- Neuron k is driven by a signal vector $x(n)$ produced by one or more layers of hidden neurons, which are driven by an input vector (stimulus) applied to the source nodes (i.e., input layer) of the neural network.



b) Signal-flowe graph of output neuron.

2.2 Error-Correction Learning

- The argument n denotes discrete time step of an iterative process involved in adjusting the synaptic weights of neuron k.
- The output signal of neuron k is denoted by $y_k(n)$.
- This output signal is compared to a desired response or target output, $d_k(n)$.
- Consequently, an error signal, $e_k(n)$, is produced.



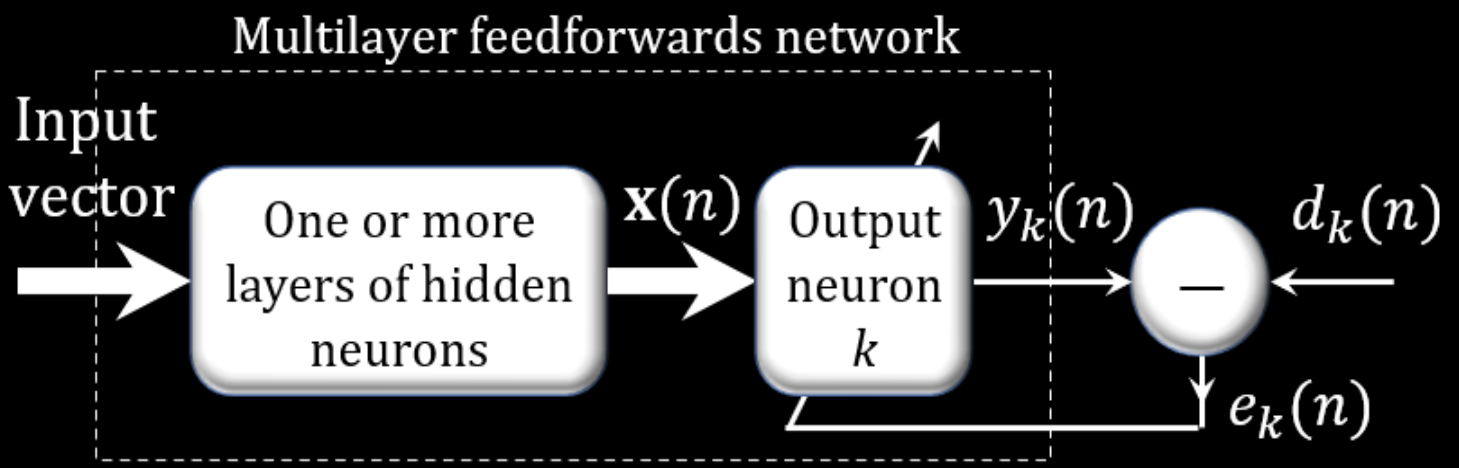
b) Signal-flowe graph of output neuron.

2.2 Error-Correction Learning

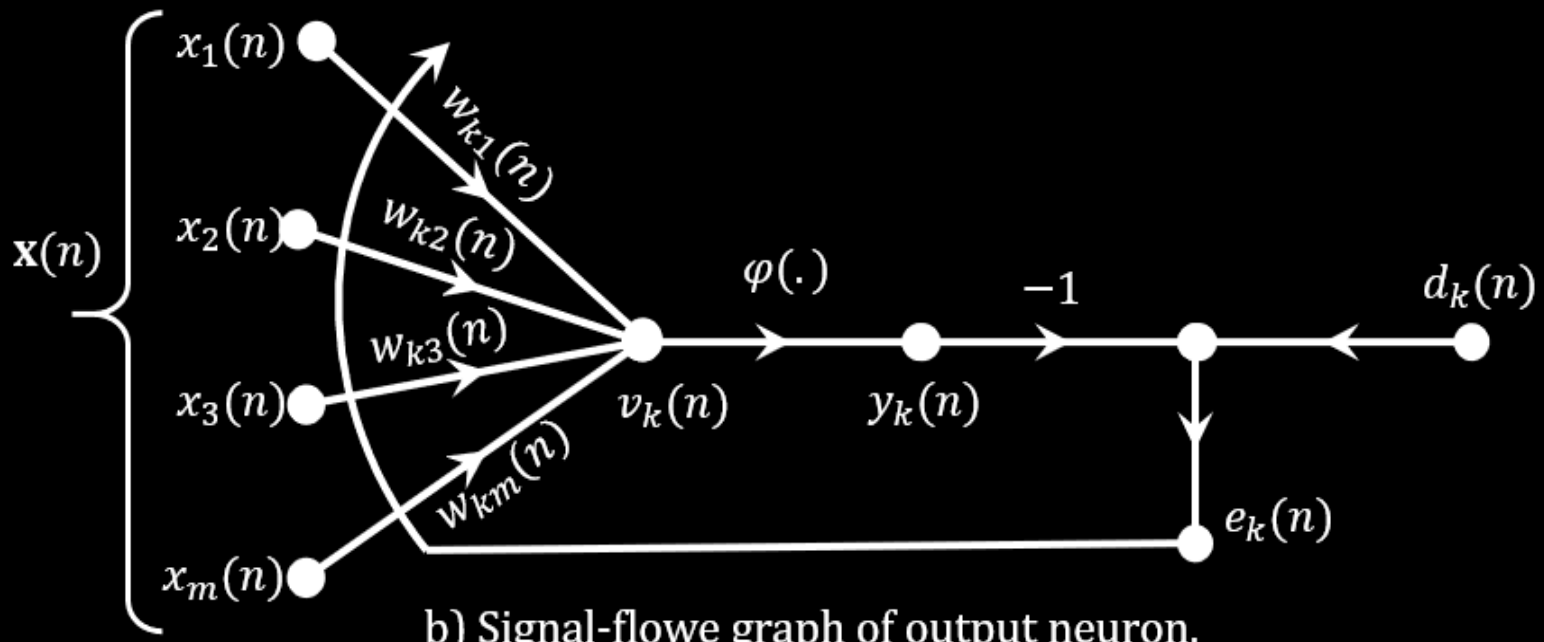
- The error signal

$$e_k(n) = d_k(n) - y_k(n) \quad (2.1)$$

- $e_k(n)$ actuates a control mechanism to apply a sequence of corrective adjustments to the synaptic weights of neuron k.
- The NN aims to make the output signal $y_k(n)$ come closer to the desired response $d_k(n)$ in a step-by-step manner.

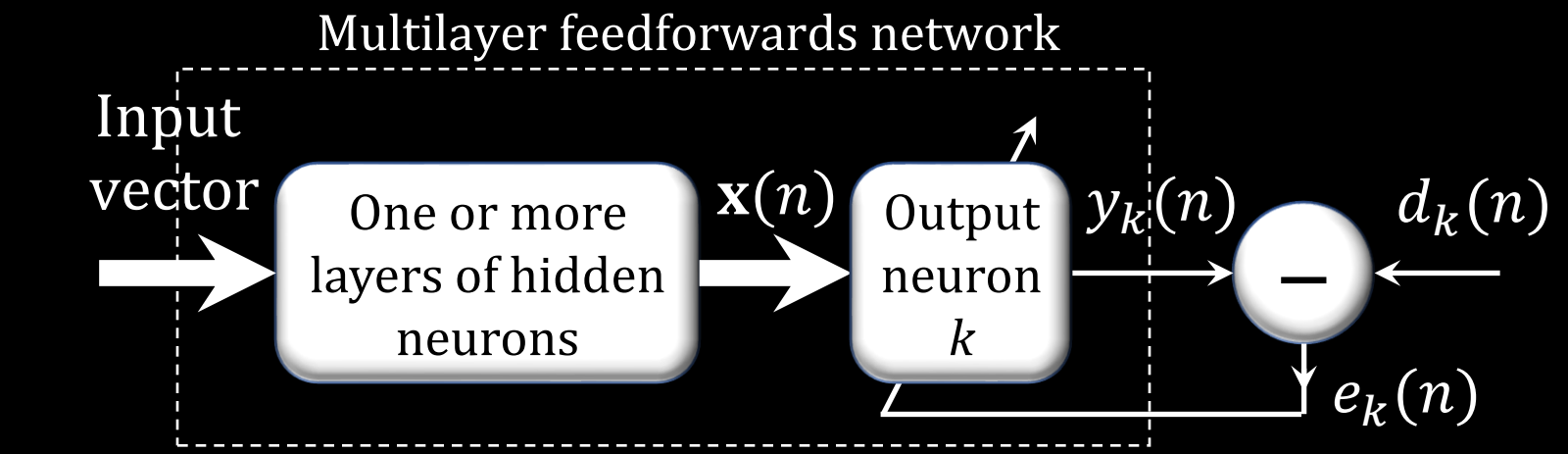


a) Block diagram of a neural network highlighting the only neuron in the output layer.

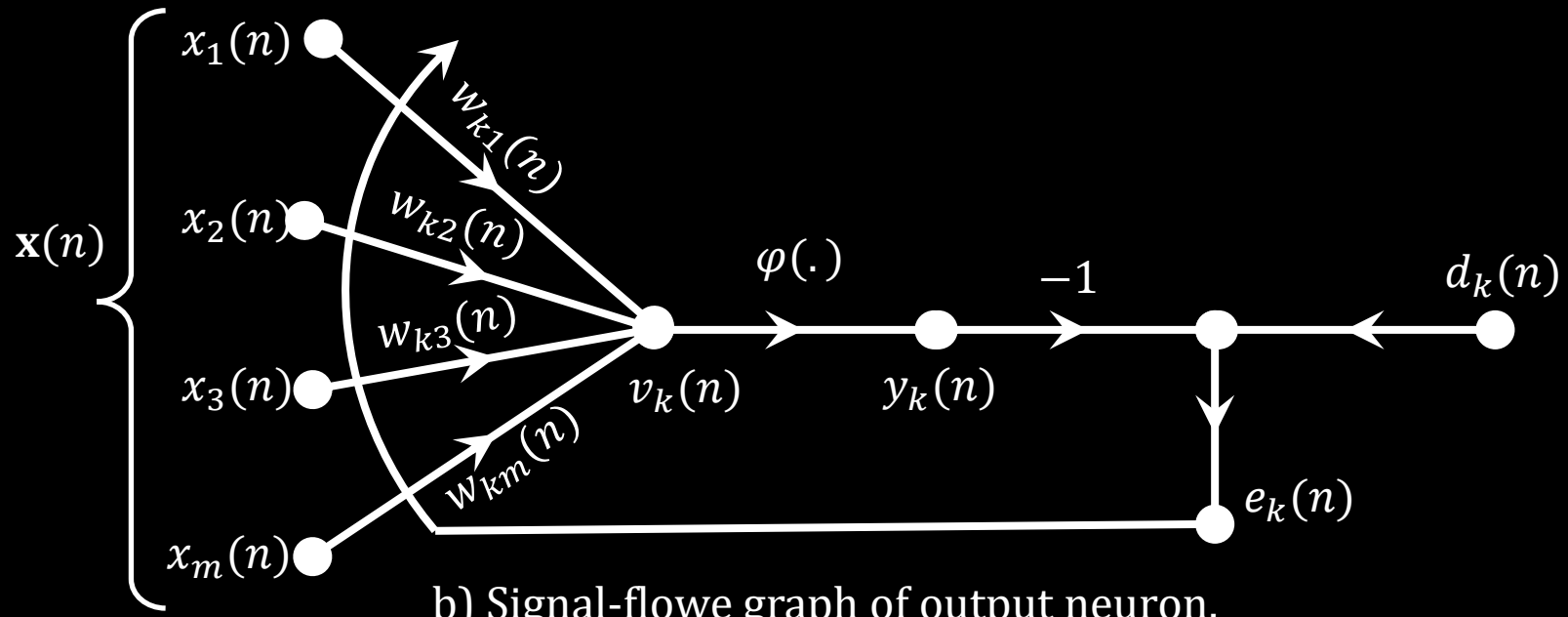


b) Signal-flowe graph of output neuron.

Fig. 2.1 Illustrating error-correction learning.



a) Block diagram of a neural network highlighting the only neuron in the output layer.



b) Signal-flowe graph of output neuron.

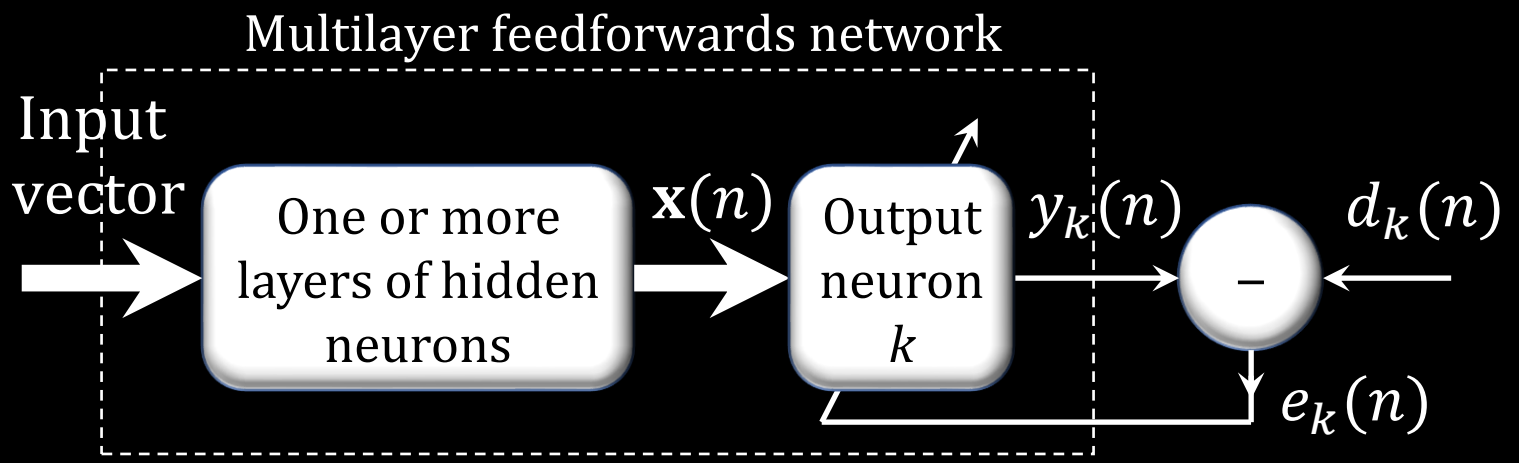
Fig. 2.1 Illustrating error-correction learning.

2.2 Error-Correction Learning

- This objective is achieved by minimizing a cost function or index of performance, $E(n)$, defined in terms of the error signal $e_k(n)$ as:

$$E(n) = \frac{1}{2} e_k^2(n) \tag{2.2}$$

That is, $E(n)$ is the instantaneous value of the error energy. The step-by-step adjustments to the synaptic weights of neuron k are continued until the system reaches a steady state.



a) Block diagram of a neural network highlighting the only neuron in the output layer.

2.2 Error-Correction Learning

- Minimization of the cost function $E(n)$ leads to a learning rule referred to as the delta rule.
- Let $w_{kj}(n)$ denote the value of synaptic weight w_{kj} of neuron k excited by element $x_j(n)$ of the signal vector $\mathbf{x}(n)$ at time step n.
- According to the delta rule, the adjustment of $\Delta w_{kj}(n)$ applied to the synaptic weight w_{kj} at time step n is defined by

$$\Delta w_{kj}(n) = \eta e_k(n) x_j(n) \tag{2.3}$$

- Where η is the *learning rate parameter* which is carefully selected to ensure stability or convergence of the iterative learning process.

2.2 Error-Correction Learning

- The updated value of synaptic weight w_{kj} is determined by

$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n) \tag{2.4}$$

In effect, $w_{kj}(n)$ and $w_{kj}(n+1)$ may be viewed as the old and new values of synaptic weight w_{kj} respectively. This can also be written as

$$w_{kj}(n) = z^{-1}\left[w_{kj}(n+1)\right] \tag{2.5}$$

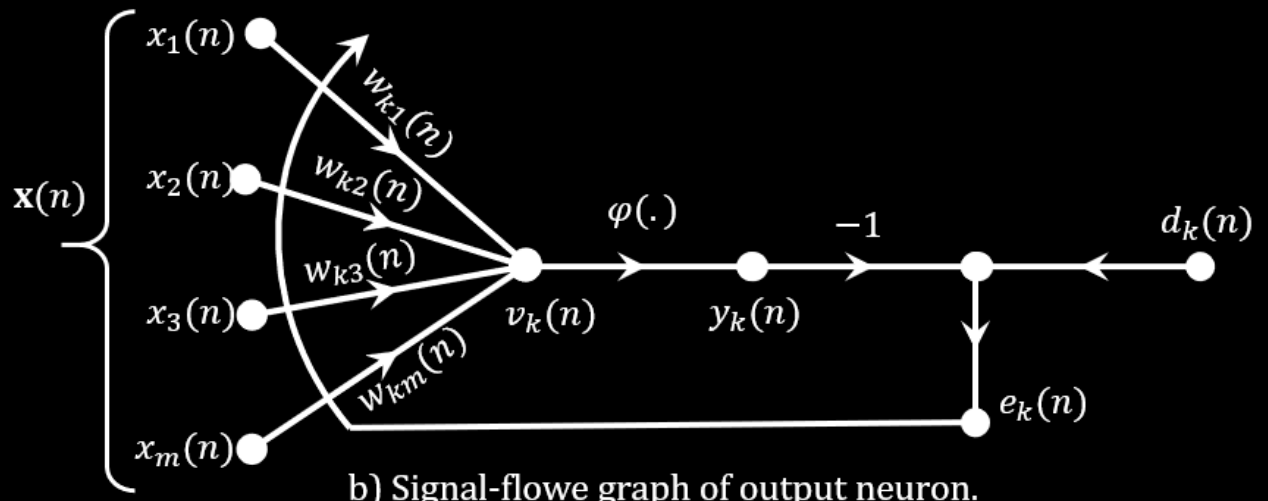
where z^{-1} is the unit-delay operator representing a storage element.

2.3 Memory-Based Learning

- In memory-based learning, all (or most) of the past experiences are explicitly stored in a large memory of correctly classified input-output examples:

$$\{\mathbf{x}_i, d_i\}_{i=1}^{N}$$

- where $\mathbf{x}_i$ is the input vector and d_i denotes corresponding desired response
- All memory-based learning algorithms involve two essential ingredients:
  - ~ Criterion used for defining the local neighborhood of the test vector $\mathbf{x}_{test}$.
  - ~ Learning rule applied to training examples in the local neighborhood of $\mathbf{x}_{test}$.



b) Signal-flowe graph of output neuron.

2.3 Memory-Based Learning

- In an effective type of memory-based learning known as the *nearest neighbor rule*, the local neighborhood is defined as the training example that lies in the immediate neighborhood of the test vector $\mathbf{x}_{\text{test}}$. In particular, the vector

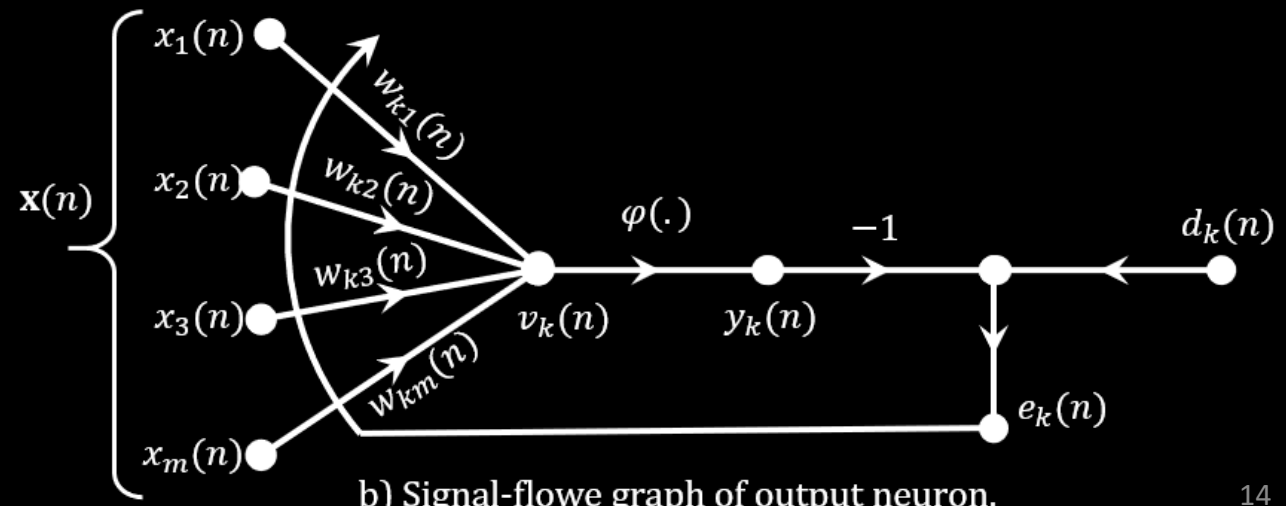
$$\mathbf{x}'_N \in \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \tag{2.6}$$

- is said to be the nearest neighbor of $\mathbf{x}_{\text{test}}$ if

$$\min_t d(\mathbf{x}_i, \mathbf{x}_{test}) = d(\mathbf{x}'_N, \mathbf{x}_{\text{test}}) \tag{2.7}$$

Where $d(\mathbf{x}_i, \mathbf{x}_{test})$ is the Euclidean distance between the vectors $\mathbf{x}_i$ and $\mathbf{x}_{test}$.

The class associated with the minimum distance, that is, vector $\mathbf{x}'_N$, is reported as the classification of $\mathbf{x}_{test}$. This rule is independent of the underlying distribution responsible for generating the training examples.



b) Signal-flowe graph of output neuron.

2.3 Memory-Based Learning

- A variant of the nearest neighbor classifier is the *k*-nearest neighbor classifier, which suggests to:
  - ~ Identify the *k* classified patterns that lie nearest to the test vector $\mathbf{x}_{test}$ for some integer *k*.
  - ~ Assign $\mathbf{x}_{test}$ to the class (hypothesis) that is most frequently represented in the *k*-nearest neighbors to $\mathbf{x}_{test}$.
  - ~ i.e. use majority vote to make classification.
- Thus, the *k*-nearest neighbor classifier acts like an averaging device.
- It discriminates against a single outlier, as illustrated in Fig. 2.2 for $k = 3$.
- An outlier is an observation that is improbably large for a nominal model of interest.

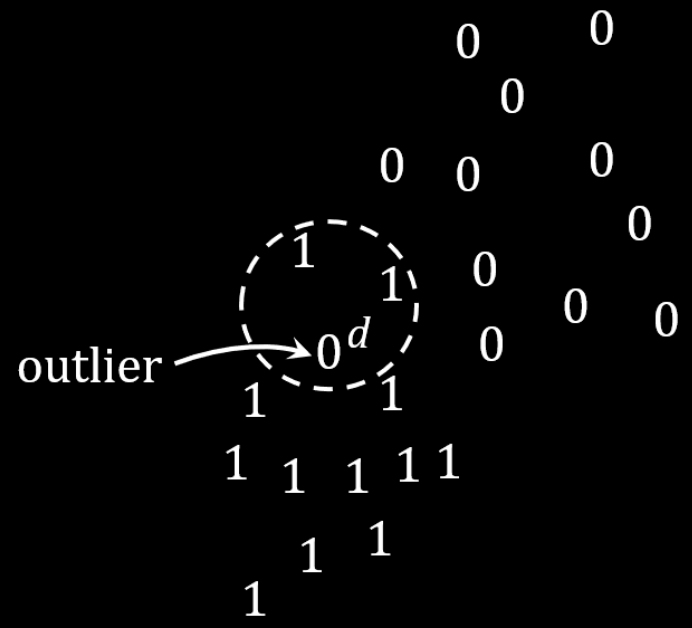


Fig. 2.2

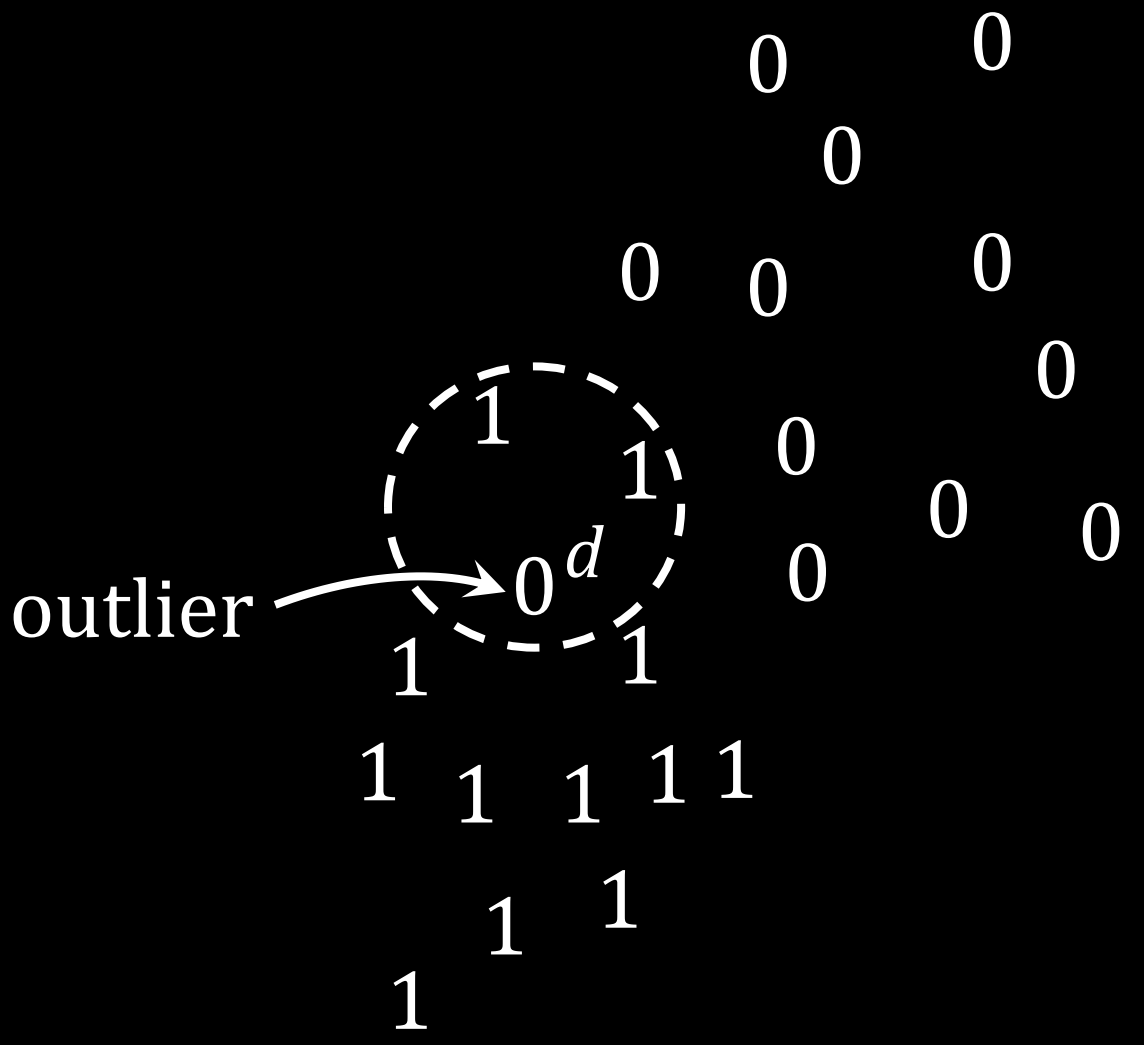
2.3 Memory-Based Learning

Fig. 2.2

The area lying inside the dashed circle includes two points pertaining to class 1 and an outlier from class 0.

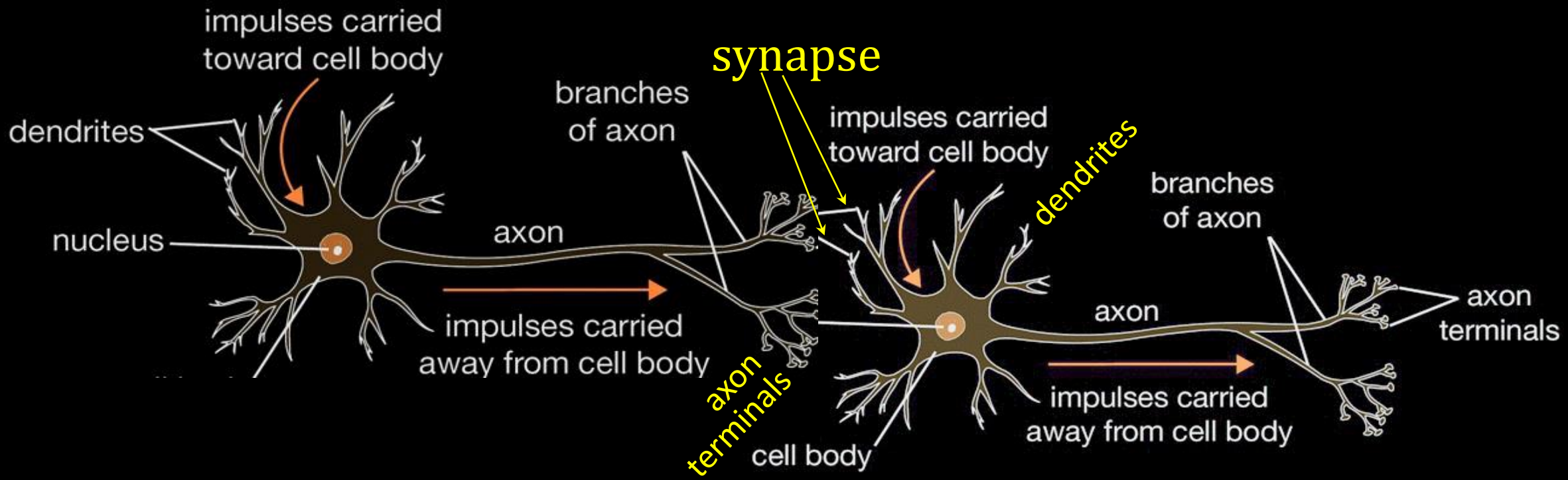
The point d corresponds to the test vector $\mathbf{x}_{\text{test}}$.

With $k = 3$, the k-nearest neighbor classifier assigns class 1 to point d even though it lies closest to the outlier.



2.4 Hebbian Learning

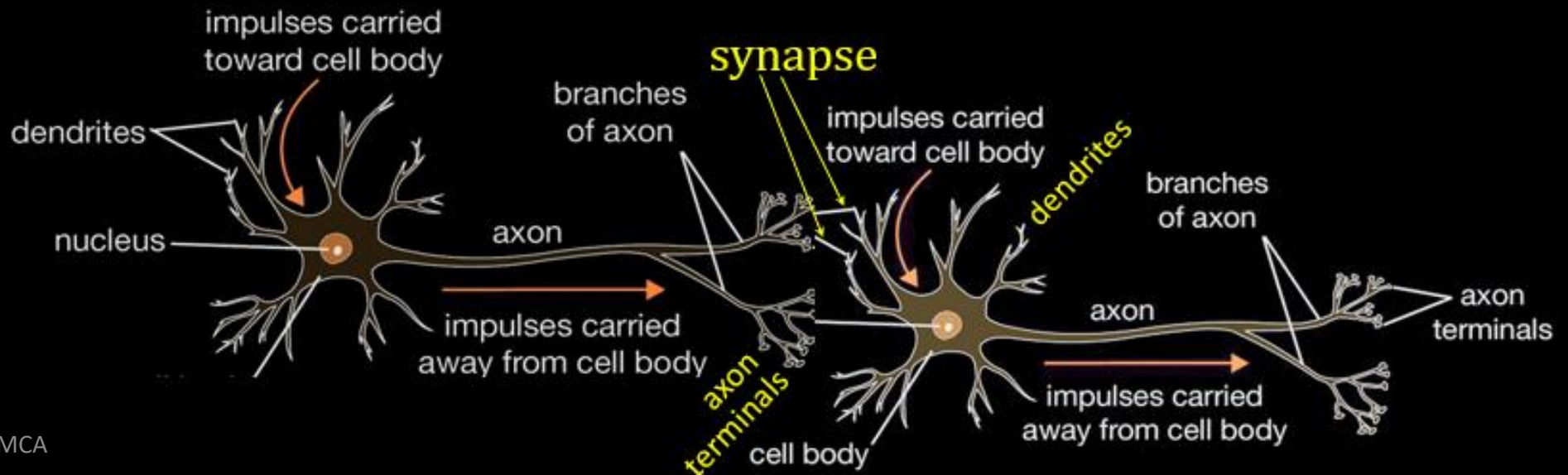
- When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it,
- some growth process or metabolic changes take place in one or both cells such that A's efficiency as one of the cells firing B, is increased.



Source: https://www.cs.toronto.edu/~lczhang/360/lec/w02/terms.html

2.4 Hebbian Learning

- Hebb proposed that associative learning would result in a modification in the activity pattern of a spatially distributed "assembly of nerve cells" such that:
  1) If two neurons on either side of a synapse (connection) are activated simultaneously, (i.e., synchronously), then the strength of that synapse is selectively increased.
  2) If two neurons on either side of a synapse are activated asynchronously, then that synapse is selectively weakened or eliminated.



2.4 Hebbian Learning

- **The four key mechanisms that characterize a Hebbian synapse**

1. **Time-dependent mechanism**

   The modifications in a Hebbian synapse depend on the exact time of occurrence of the presynaptic and postsynaptic signals.

2. **Local mechanism**

   A synapse is the transmission site where information-bearing signals (representing ongoing activity in the presynaptic and postsynaptic units) are in spatio-temporal closeness.

2.4 Hebbian Learning

3. Interactive mechanism

Occurrence of a change in a Hebbian synapse depends on signals on both sides of the synapse.

That is, a Hebbian form of learning depends on a "true interaction" between presynaptic and postsynaptic signals

4. Conjunctional or correlational mechanism

The co-occurrence of presynaptic and postsynaptic signals is sufficient to produce the synaptic modification.

The correlation over time between presynaptic and postsynaptic signals is viewed as being responsible for a synaptic change.

Accordingly, a Hebbian synapse is also referred to as a correlational synapse. Correlation is indeed the basis of learning.

2.4 Hebbian Learning

- The Hebbian traces in the forward pass are updated using Algorithm 1.
- Hebbian update for corresponding class, c is performed (line 6).
- Across the model's lifetime, Hebb is only updated during training and test time.

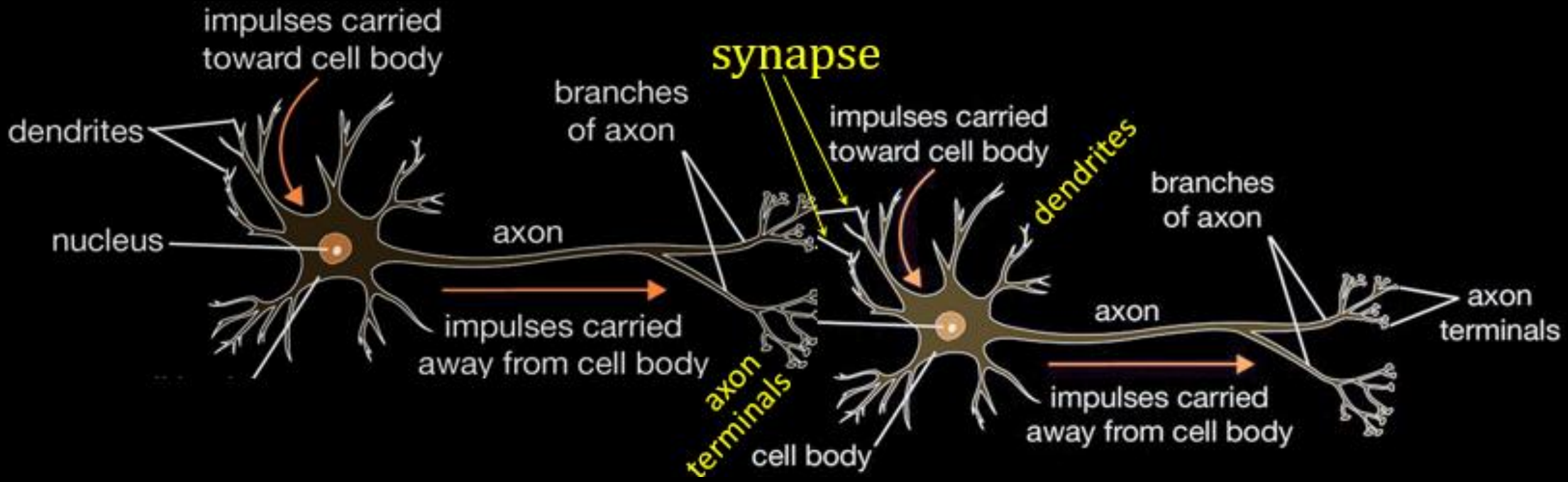
Algorithm 1. Batch update Hebbian traces.

Input: $h_{1:B}$ (hidden activations of penultimate layer)
$y_{1:B}$ (target labels),
Hebb (Hebbian trace)
Output: $z_{1:B}$ (softmax pre-activations)
for each target label $c \in \{y_{1:B}\}$ **do**
  $s \leftarrow \sum_{b=1}^{B} [y_b = c]$
  if $s > 0$ **then**
    $\tilde{h} \leftarrow \frac{1}{s} \sum_{b=1}^{B} h[y_b = c]$
    $\text{Hebb}_{:,c} \leftarrow (1-\eta)\text{Hebb}_{:,c} + \eta\tilde{h}$
  end if
end for
$z \leftarrow (\theta + \alpha \text{Hebb})h$

Vithursan -- Personal Site – ICML AMTL 2019 - Differentiable Hebbian Plasticity for Continual Learning (vithursant.com)

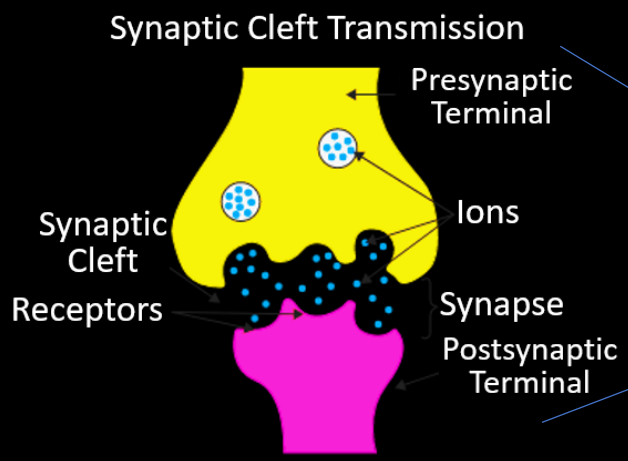
2.4 Hebbian Learning

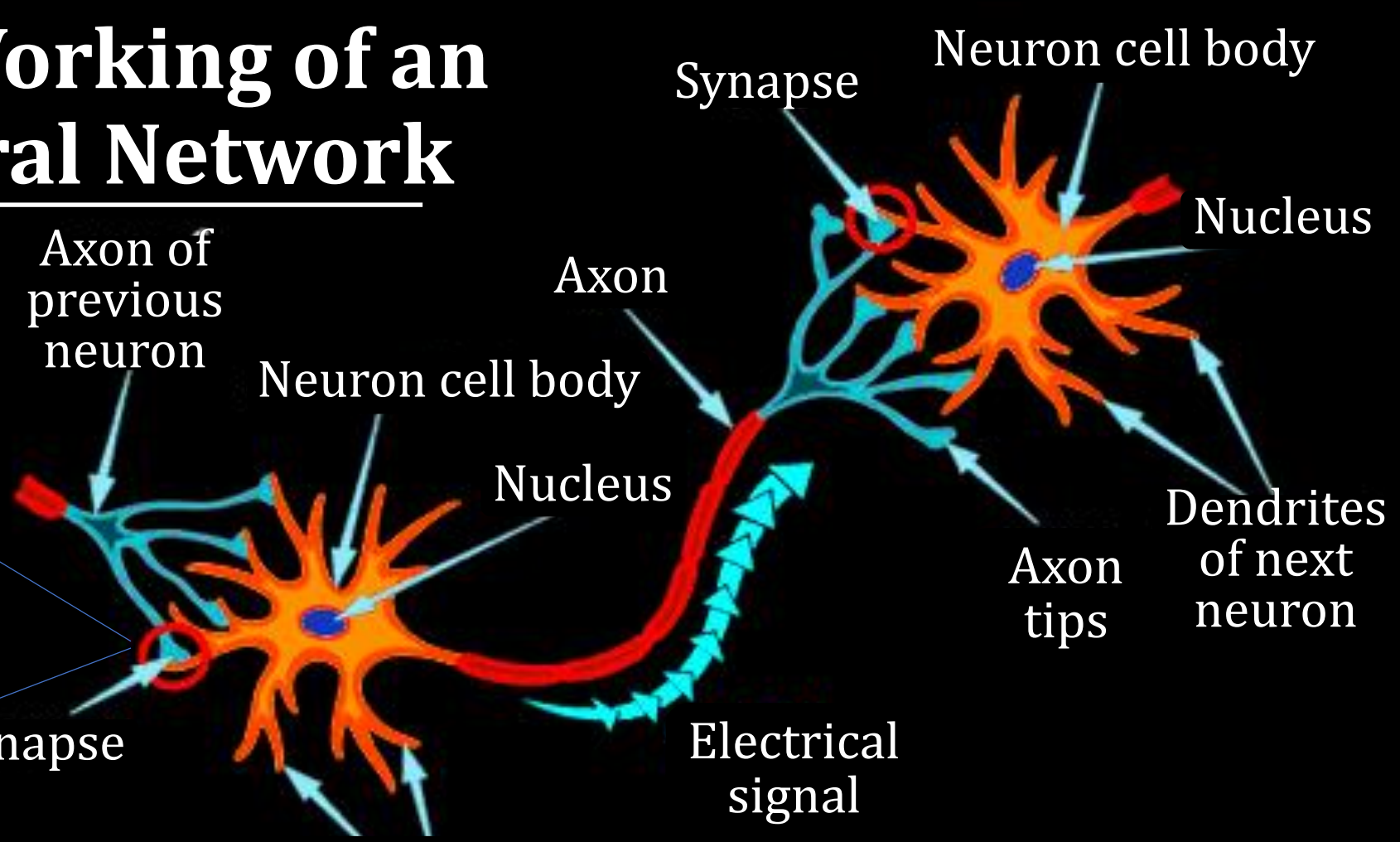
Learning Associations Through the Strengthening of Connections



Hebbian Learning: When two joining cells fire simultaneously, the connection between them strengthens (Hebb, 1949).

Step-by-Step Working of an Artificial Neural Network





Step 1: External signal is received by dendrites

Step 2: External signal is processed in the neuron cell body

Step 3: Processed signal converted to an output signal and transmitted through the axon

Step 4: Output signal is received by dendrites of the next neuron through the synapse

2.4 Hebbian Learning

Synaptic Enhancement and Depression

- A Hebbian synapse increases its strength with positively correlated presynaptic and postsynaptic signals and decreases its strength when these signals are either uncorrelated or negatively correlated.
- In both Hebbian and anti-Hebbian synapses, however, the modification of synaptic efficiency relies on a mechanism that is time-dependent, highly local, and strongly interactive in nature.

2.4 Hebbian Learning

Mathematical Models of Hebbian Modifications

- To formulate Hebbian learning in mathematical terms, consider a synaptic weight w_{kj} of neuron k with presynaptic and postsynaptic signals denoted by x_j and y_k, respectively.
- Adjustment applied to synaptic weight w_{kj} at time step n is expressed as

$$\Delta w_{kj}(n) = F\left(y_k(n), x_j(n)\right) \tag{2.8}$$

where $F(\cdot, \cdot)$ is a function of both postsynaptic and presynaptic signals.

2.4 Hebbian Learning

Hebb's Hypothesis

- The simplest form of Hebbian learning is described by

$$\Delta w_{kj}(n) = \eta y_k(n) x_j(n) \qquad (2.9)$$

- Where $\eta > 0$ determines the rate of learning.
- The top curve of Fig. 2.3 shows a graphical representation of Eq. (2.9).
- The repeated application of the input signal (presynaptic activity) x_j leads to an increase in (postsynaptic activity) y_k
- Therefore, exponential growth drives synaptic connection into saturation.
- At that point no information will be stored in the synapse and selectivity is lost.

2.4 Hebbian Learning

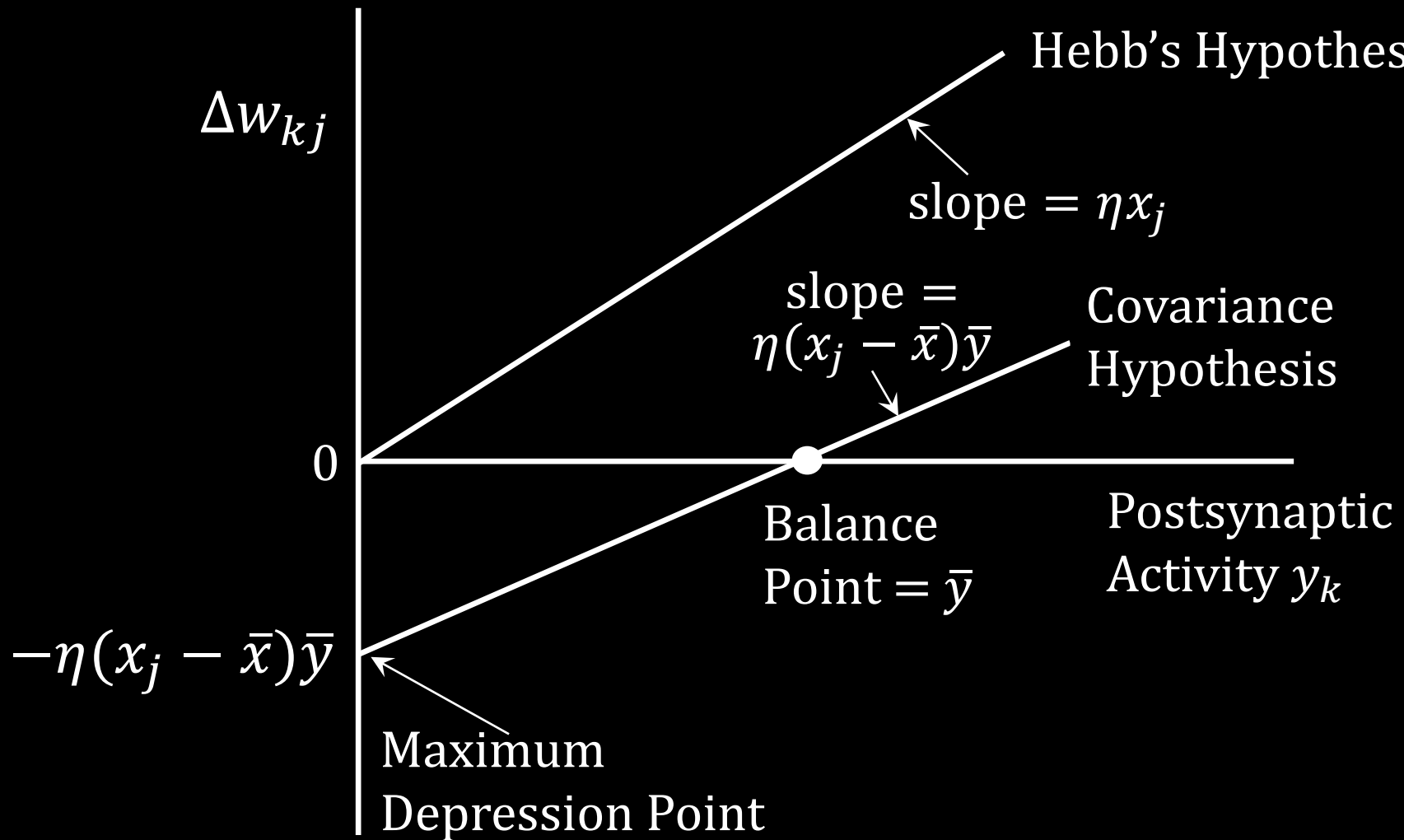


Fig. 2.3 Illustration of Hebb's hypothesis and the covariance hypothesis $\Delta w_{kj}(n)$ plotted versus the output signal (postsynaptic activity) y_k.

2.4 Hebbian Learning

Covariance Hypothesis

- One way of overcoming the limitation of Hebb's hypothesis is to use the covariance hypothesis
  - where the presynaptic (x_j) and postsynaptic (y_k) signals in Eq. (2.9) are replaced by the departure of presynaptic and postsynaptic signals from their respective average values over a certain time interval.
- According to the covariance hypothesis, the adjustment applied to the synaptic weight w_{kj} is defined by

$$\Delta w_{kj} = \eta(x_j - \bar{x})(y_k - \bar{y}) \qquad (2.10)$$

2.4 Hebbian Learning

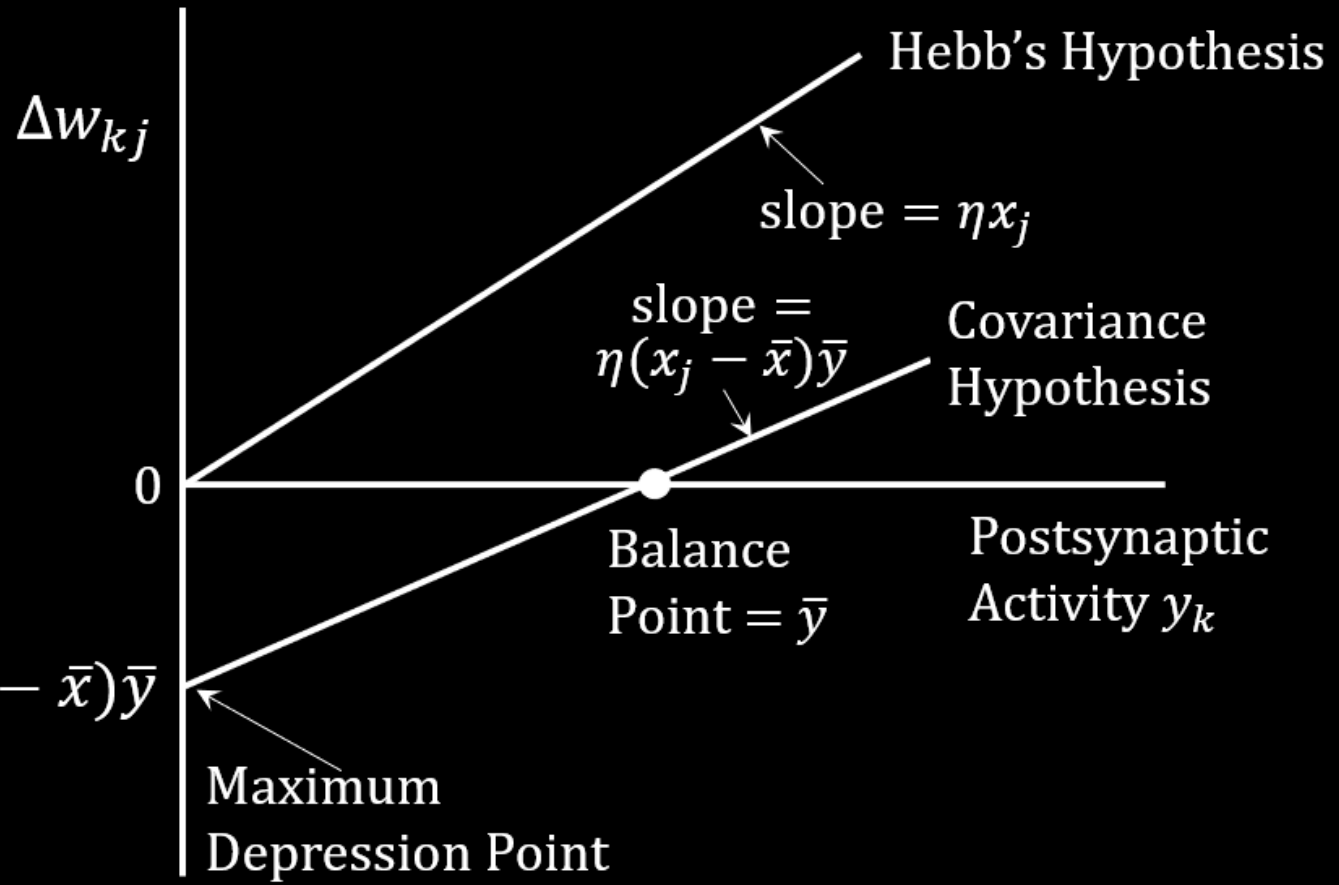
Covariance Hypothesis

- The covariance hypothesis allows for

1. Convergence to a nontrivial state, which is reached when

$$x_k = \bar{x} \text{ and } y_j = \bar{y}$$

1. Prediction of both synaptic potentiation (i.e., increase in synaptic strength) and synaptic depression (i.e., decrease in synaptic strength).



2.4 Hebbian Learning

Covariance Hypothesis

- Recalling (2.10) $\Delta w_{kj} = \eta(x_j - \bar{x})(y_k - \bar{y})$, we make the following important observations

1. Synaptic weight w_{kj} is enhanced if there are sufficient levels of presynaptic and postsynaptic activities, that is, the conditions $x_j > \bar{x}$ and $y_k - \bar{y}$ are both satisfied.
2. Synaptic weight w_{kj} is depressed if there is either
   - a presynaptic activation (i.e. $x_j > \bar{x}$) in the absence of sufficient postsynaptic activation (i.e. $y_k < \bar{y}$), or
   - a postsynaptic activation (i.e. $y_k > \bar{y}$) in the absence of sufficient presynaptic activation (i.e. $x_j < \bar{x}$).

2.5 Competitive Learning

- In competitive learning, the output neurons of a neural network compete among themselves to become active (fired).
  - ~ In neural network based on Hebbian learning, several output neurons may be active simultaneously,
  - ~ In **competitive learning** only a single output neuron is active at any time.

2.5 Competitive Learning

- There are three basic elements to a competitive learning rule

1) A set of neurons that are all the same except for some randomly distributed synaptic weights, and which therefore respond differently to a given set of input patterns.
2) A limit imposed on the "strength" of each neuron.
3) A mechanism that permits the neurons to compete for the right to respond to a given subset of inputs, only one output neuron is active (i.e., "on") at a time.
4) The neuron that wins the competition is called a *winner-takes-all* neuron.

2.5 Competitive Learning

- In *competitive learning*, the neural network has a single layer of output neurons, each of which is fully connected to the input nodes.
- The network may include feedback connections among the neurons, as indicated in Fig. 2.4.

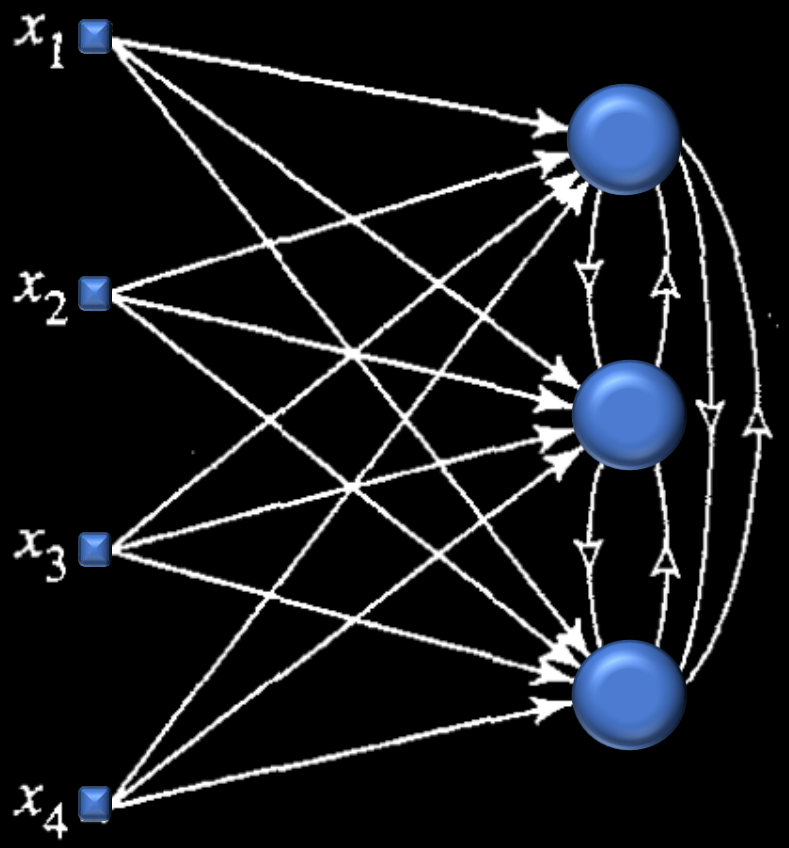


Fig. 2.4 Architectural graph of a simple competitive learning network with feedforward (excitatory) connections from the source nodes to the neurons, and lateral (inhibitory) connections among the neurons; the lateral connections are signified by open arrows.

2.5 Competitive Learning

- For a neuron k to be the winning neuron, its induced local field v_k for a specified input pattern **x** must be the largest among all the neurons in the network.
- The output signal y_k of winning neuron k is set equal to 1; the losers are set equal to 0.
- We thus write

$$y_k = \begin{cases} 1 & if\ v_k > v_j \quad for\ all\ j, j \neq k \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

where v_k represents the combined action of all the forward and feedback inputs to neuron k.

2.5 Competitive Learning

- Let w_{kj} denote the synaptic weight connecting input node j to neuron k.
- Suppose that each neuron is allotted a fixed amount of synaptic weight, which is distributed among its input nodes; that is,

$$\sum_j w_{kj} = 1 \qquad \text{for all } k \tag{2.12}$$

- If a neuron does not respond to a particular input, no learning takes place in that neuron.
- If a particular neuron wins the competition, each input node of that neuron abandons some proportion of its synaptic weight, and
- the weight abandoned is then distributed equally among the active input nodes.

2.5 Competitive Learning

- According to the competitive learning rule, change Δw_{kj} applied to synaptic weight w_{kj} is defined by

$$\Delta w_{kj} = \begin{cases} \eta(x_j - w_{kj}) & \text{if neuron } j \text{ wins the competition} \\ 0 & \text{if neuron } j \text{ loses the competition} \end{cases} \quad (2.13)$$

where η is the learning-rate parameter.

- This rule has the overall effect of moving the synaptic weight vector $\mathbf{w_k}$ of winning neuron k toward the input pattern $\mathbf{x}$.

2.5 Competitive Learning

- We may use the geometric analogy depicted in Fig. 2.5 to illustrate the essence of competitive learning.
- It is assumed that each input pattern (vector) $\mathbf{x}$ has some constant Euclidean length so that we may view it as a point on an N-dimensional sphere where N is number of input nodes.
- N also represents the dimension of each synaptic weight vector $\mathbf{w}_k$.

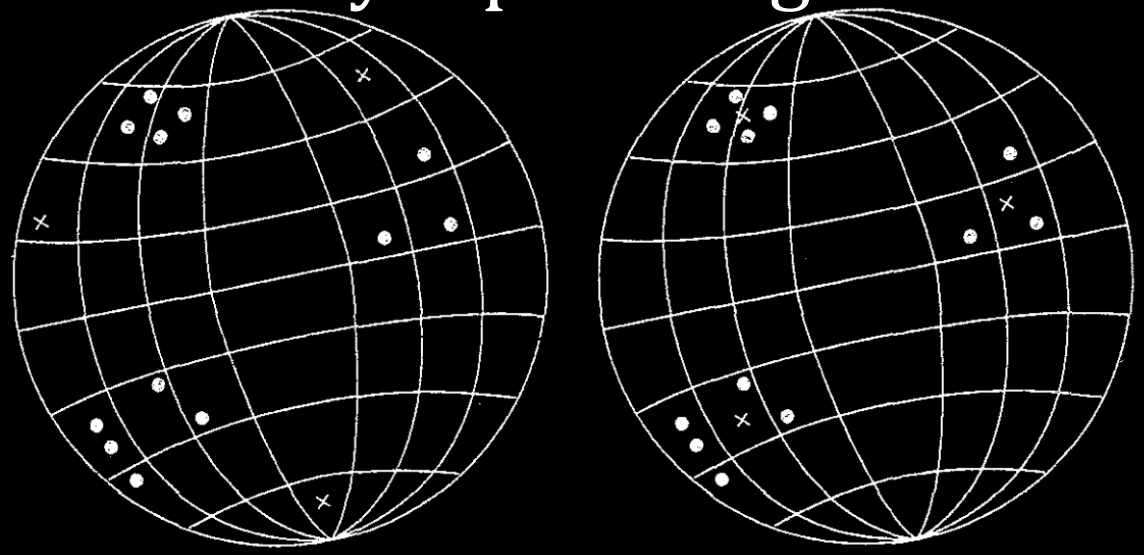


Fig. 2.5 Geometric interpretation of the competitive learning process. The dots represent the input vectors, and the crosses represent the synaptic weight vectors of three output neurons. (a) Initial state of the network. (b) Final state of the network.

2.5 Competitive Learning

- It is assumed that all neurons in the network are constrained to have the same Euclidean length (norm), as shown by

$$\sum_j w_{kj}^2 = 1 \qquad \text{for all } k \tag{2.14}$$

- Figure 2.5b shows a typical final state of the network that results from the use of competitive learning.
- This figure illustrates the ability of a neural network to perform clustering through competitive learning.

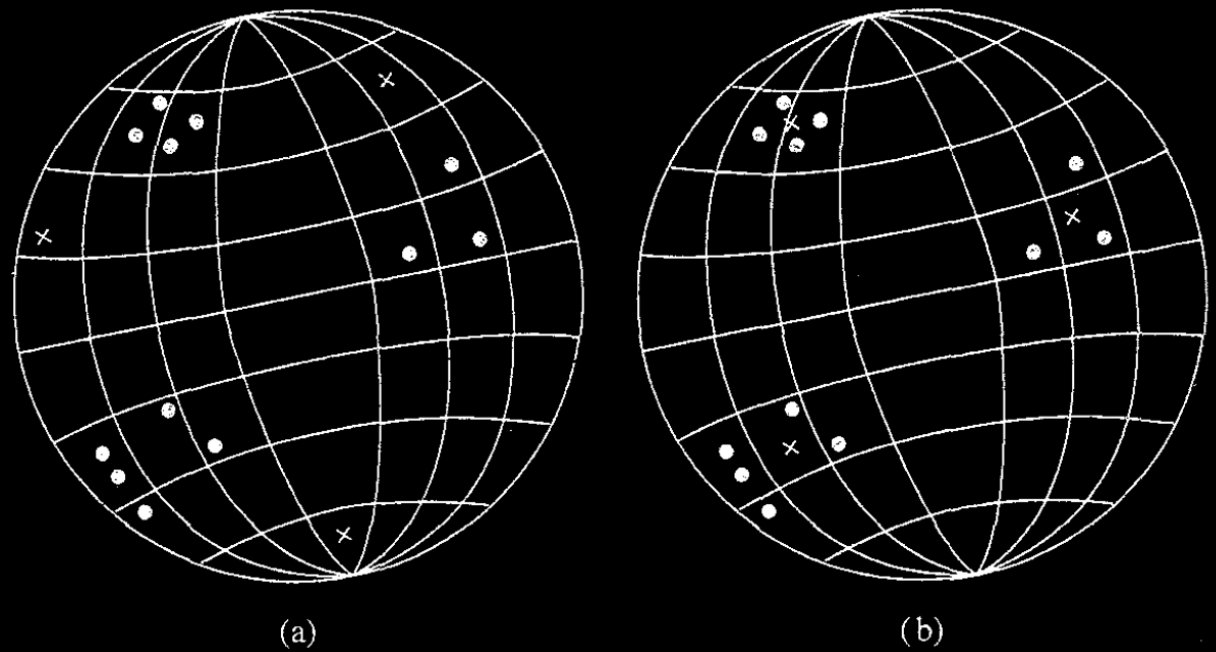


Fig. 2.5 Geometric interpretation of the competitive learning process. The dots represent the input vectors, and the crosses represent the synaptic weight vectors of three output neurons. (a) Initial state of the network. (b) Final state of the network.

2.5 Competitive Learning

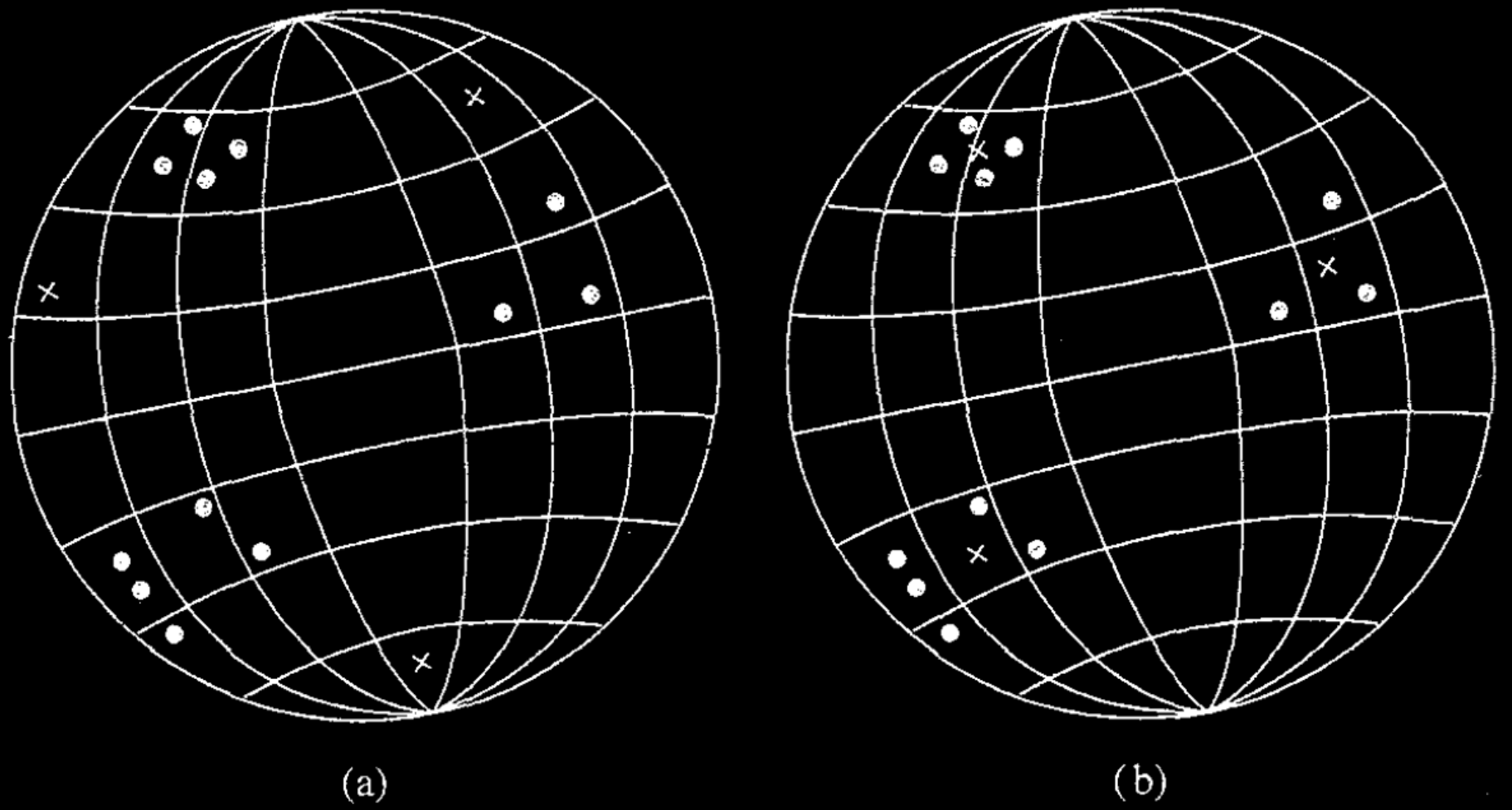


Fig. 2.5 Geometric interpretation of the competitive learning process. The dots represent the input vectors, and the crosses represent the synaptic weight vectors of three output neurons. (a) Initial state of the network. (b) Final state of the network.

2.6 Boltzmann Learning

- In a Boltzmann machine the neurons constitute a recurrent structure, and they operate in a binary manner (either "on" state or in "off" state)
- The machine is characterized by an energy function, E, the value of which is determined by the particular states occupied by the individual neurons of the machine, as shown by

$$E = -\frac{1}{2}\sum_j \sum_{\substack{k \\ j \neq k}} w_{kj} x_k x_j \qquad (2.15)$$

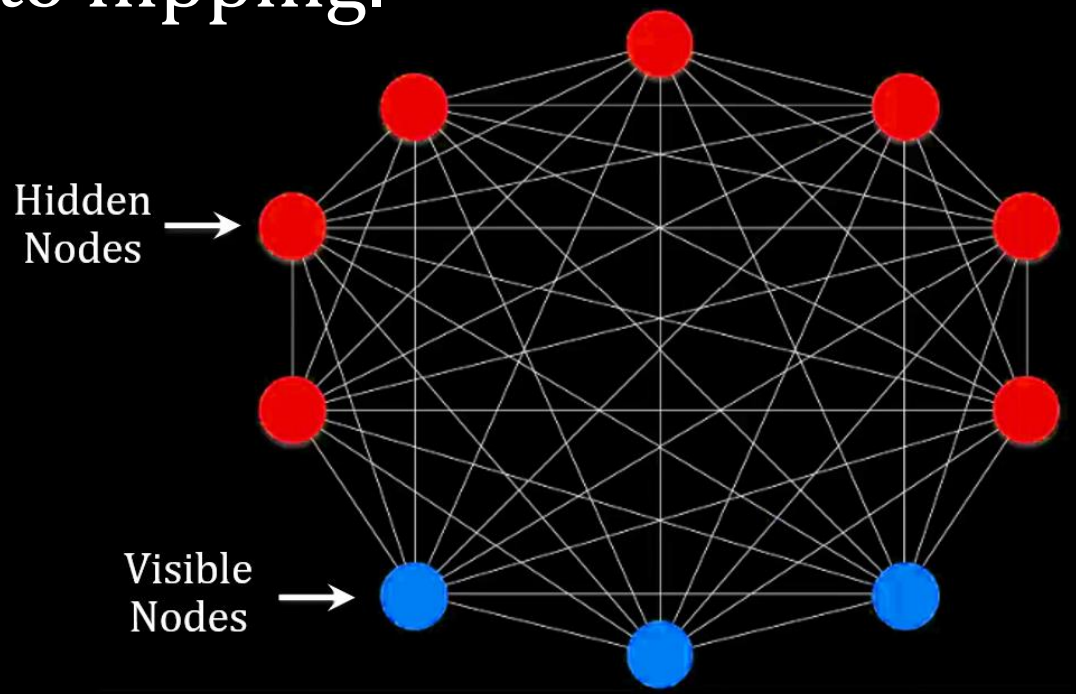
where x_j is the state of neuron j and w_{kj} is the synaptic weight connecting neuron j to neuron k and $j \neq k$ means no self-feedback.

2.6 Boltzmann Learning

- The machine operates by choosing a neuron k at some step of the learning process, then flipping the state of neuron k from state x_k state to $-x_k$ at some temperature T with probability

$$P(x_k \rightarrow -x_k) = \frac{1}{1+\exp(-\Delta E_k/T)} \qquad (2.16)$$

Where ΔE_k is the energy change due to flipping.



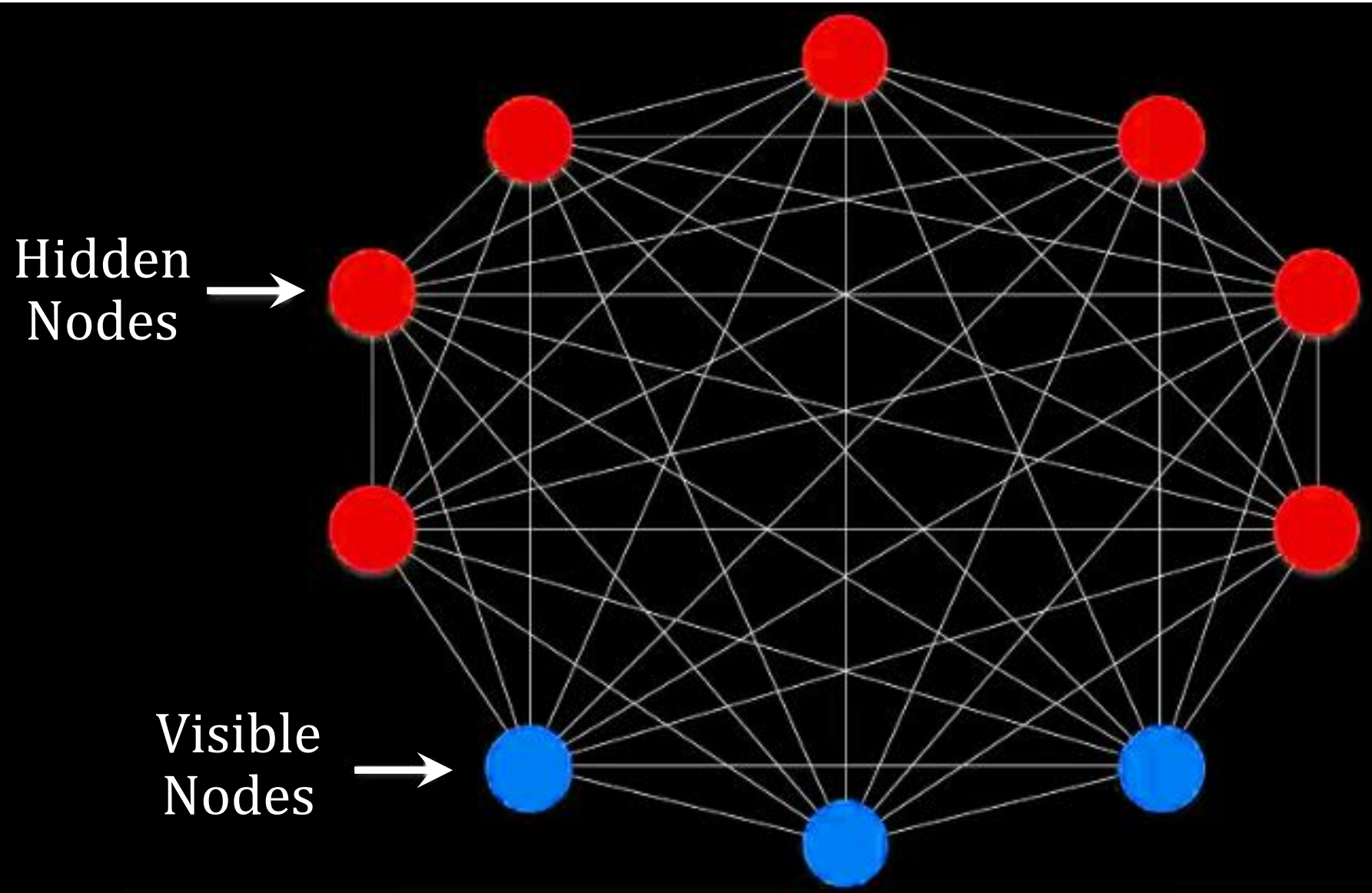
2.6 Boltzmann Learning

- Let ρ_{kj}^{+} denote the correlation between the states of neurons j and k, with the network in its clamped condition and ρ_{kj}^{-} denote the correlation between the states of neurons j and k with the network in its free-running condition.
- The change Δw_{kj} applied to the synaptic weight w_{kj} from neuron j to neuron k is defined by

$$\Delta \mathrm{w_{kj}} = \eta\left(\rho_{kj}^{+} - \rho_{kj}^{-}\right) \quad j \neq k \tag{2.17}$$

where ρ is a learning-rate parameter.

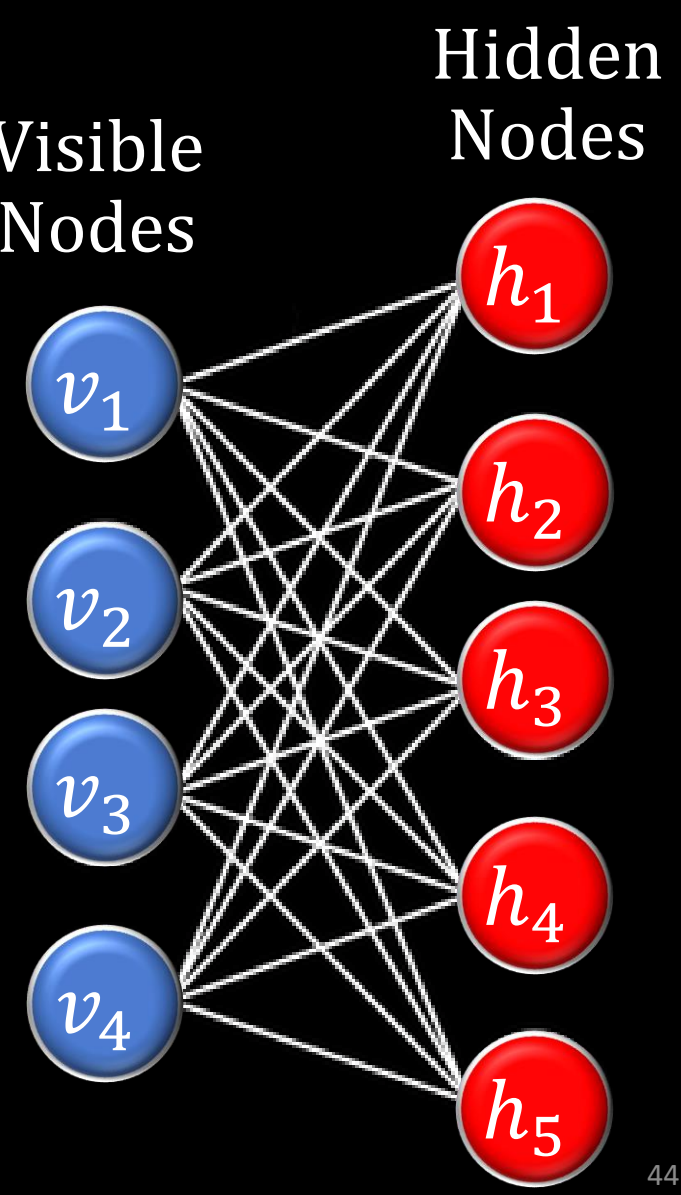
Boltzmann Machine



An Intuitive Introduction Of Boltzmann Machine | by Manish Nayak | Data Driven Investor | Medium

Restricted Boltzmann Machine (RBM)

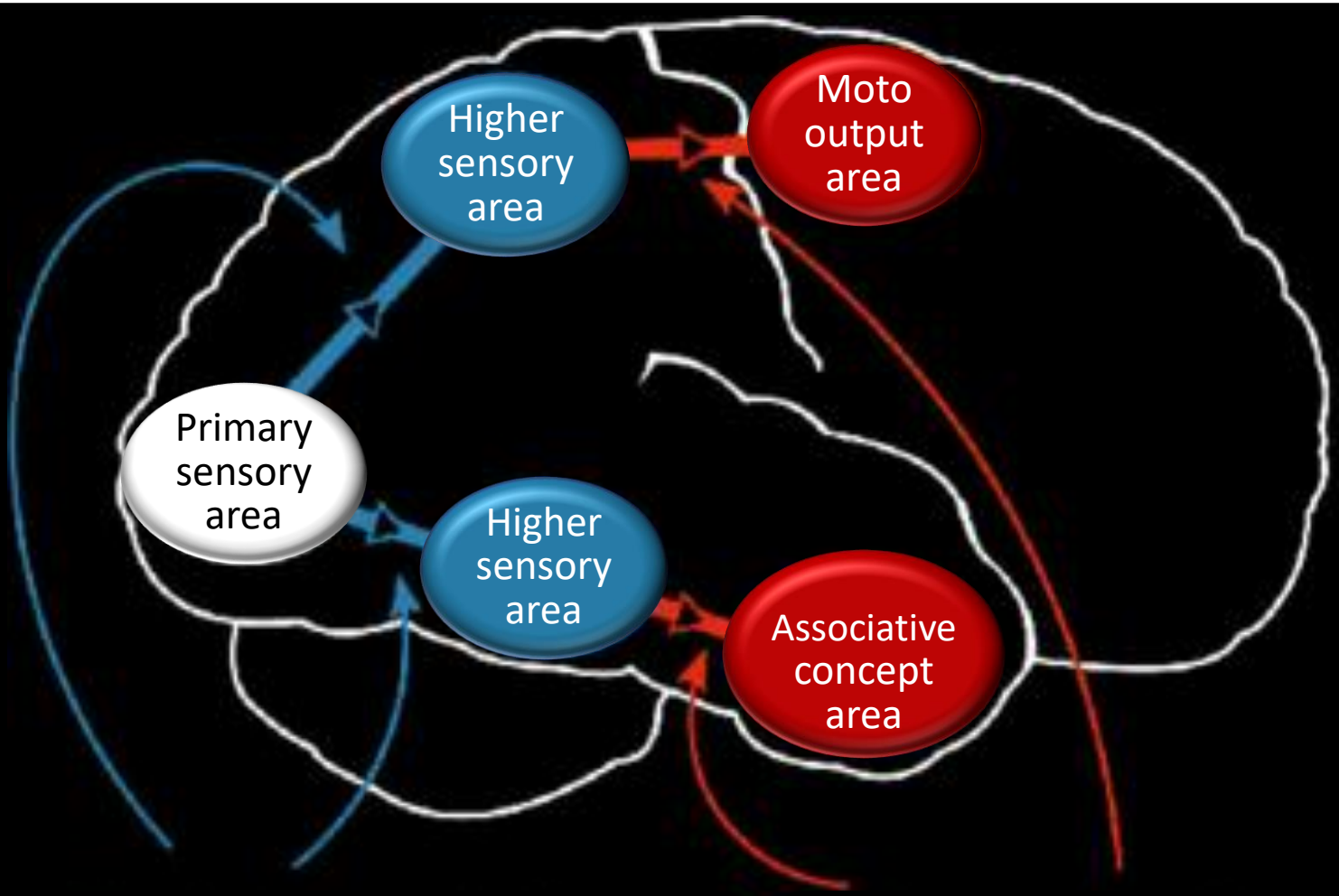
- Since there are no connections between the hidden neurons in the Restricted Boltzmann Machine,
- and since the connections between the visible layer and the hidden layer are undirected,
- it follows that the states of the hidden neurons are conditionally independent of each other, given the visible states.
- Hence, the RBM is capable of extracting an unbiased sample from the posterior distribution, given a data vector clamped onto the visible neurons.



2.7 Credit-Assignment Problem

- Credit-assignment (loading problem) is the problem of assigning credit or blame for overall outcomes to each of the internal decisions made by a learning machine and which contributed to those outcomes.
- We may decompose the credit-assignment problem into two subproblems:
  1. The assignment of credit for outcomes to actions. The temporal credit-assignment problem involves the instants of time when the actions that deserve credit were actually taken.
  2. The assignment of credit for actions to internal decisions. This structural credit-assignment problem involves assigning credit to the internal structures of actions generated by the system.

2.7 Credit-Assignment Problem



The behavioral effects of changes to these synaptic connections…

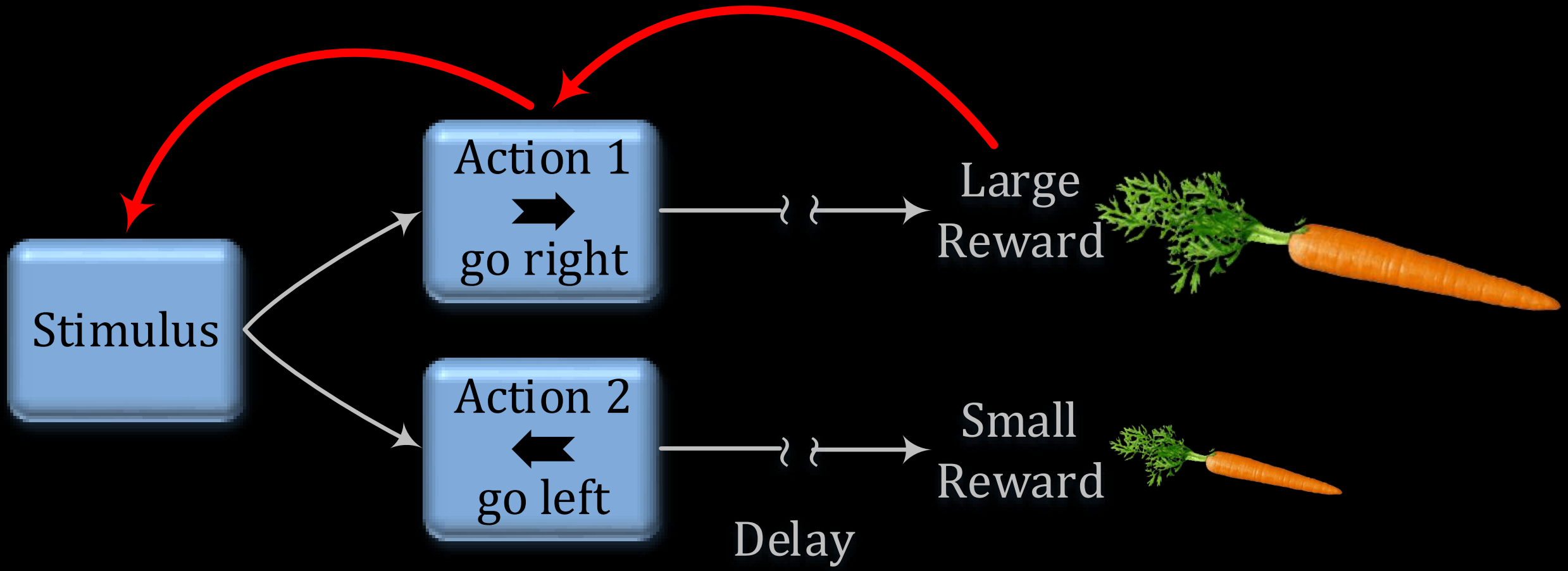
… depends on the status of these synaptic connections…

(19) (PDF) Towards deep learning with segregated dendrites (researchgate.net)

2.7 Credit-Assignment Problem

- The **structural credit-assignment** suggests which component of the multicomponent learning machine should have its behavior altered and by how much in order to improve system performance
- The **temporal credit-assignment** is relevant when there are many actions taken by a learning machine that result in certain outcomes.
- Since the output neuron *k* is visible to the outside world, it is possible to supply a desired response to this neuron
  - by adjusting the synaptic weights of the output neuron in accordance with error-correction learning.

2.7 Credit-Assignment Problem - Example



https://www.researchgate.net/publication/324052774_Solving_the_Credit_Assignment_Problem_With_the_Prefrontal_Cortex

2.7 Credit-Assignment Problem - Example

- Example tasks highlighting the challenge of credit assignment and learning strategies enabling animals to solve a problem.
- An example of a distal reward task that can be successfully learned with eligibility traces and TD rules, where intermediate choices can acquire motivational significance and subsequently reinforce preceding decisions

- note that in some experimental variants of the behavioral paradigm, stimuli disappeared before an animal revealed its choice,
- whereas in others the cues remained on the screen until the trial outcome was revealed