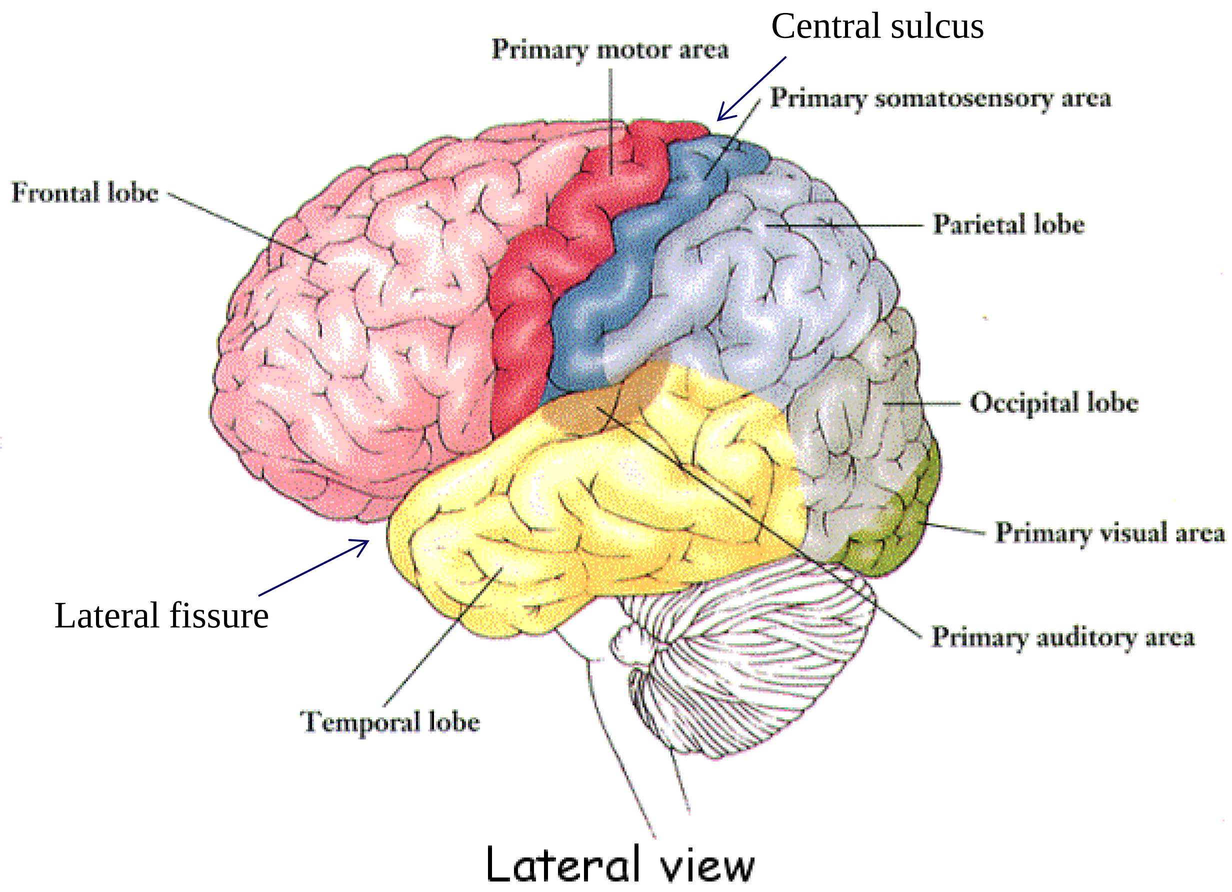
CSE 5526: Introduction to Neural Networks

Instructor: DeLiang Wang

What is this course about?

- AI (artificial intelligence) in the broad sense, in particular learning
- The human brain and its amazing ability, e.g. vision

Human brain


Central sulcus
Primary motor area
Primary somatosensory area
Frontal lobe
Parietal lobe
Occipital lobe
Primary visual area
Lateral fissure
Primary auditory area
Temporal lobe
Lateral view

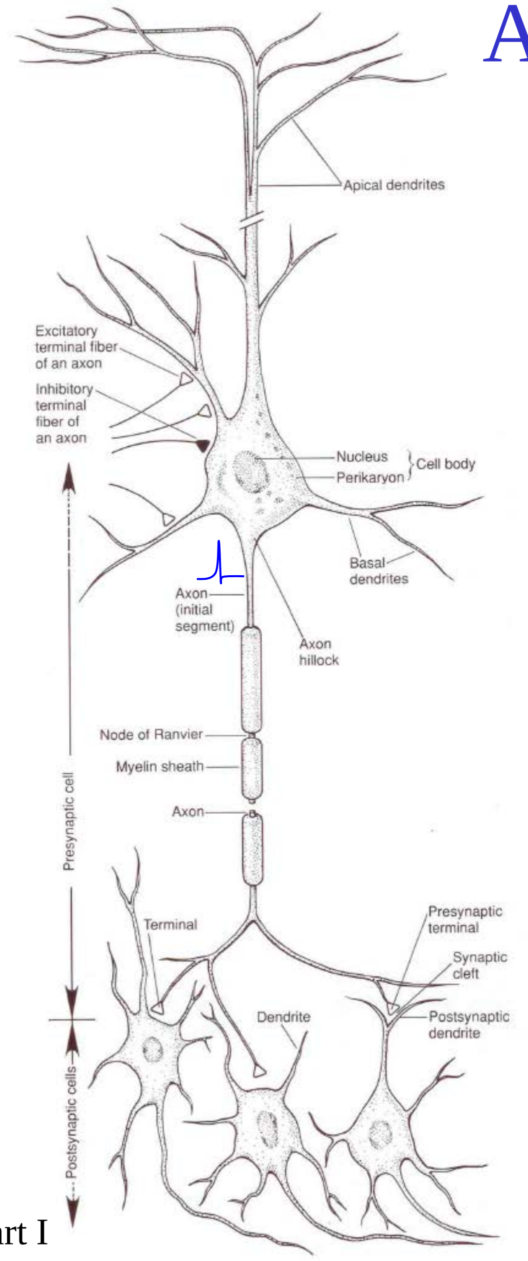
Brain versus computer

Brain	**Computer**
•	•
•	•
•	•

Brain-like computation – Neural networks (NN or ANN) – Neural computation

- Discuss syllabus

A single neuron



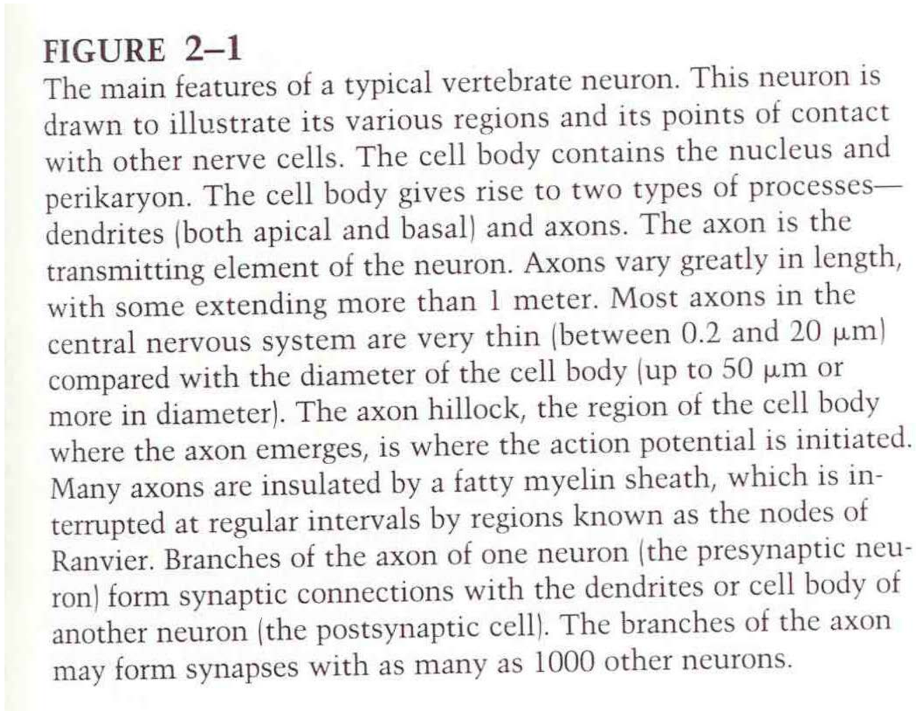


FIGURE 2–1
The main features of a typical vertebrate neuron. This neuron is drawn to illustrate its various regions and its points of contact with other nerve cells. The cell body contains the nucleus and perikaryon. The cell body gives rise to two types of processes—dendrites (both apical and basal) and axons. The axon is the transmitting element of the neuron. Axons vary greatly in length, with some extending more than 1 meter. Most axons in the central nervous system are very thin (between 0.2 and 20 μm) compared with the diameter of the cell body (up to 50 μm or more in diameter). The axon hillock, the region of the cell body where the axon emerges, is where the action potential is initiated. Many axons are insulated by a fatty myelin sheath, which is interrupted at regular intervals by regions known as the nodes of Ranvier. Branches of the axon of one neuron (the presynaptic neuron) form synaptic connections with the dendrites or cell body of another neuron (the postsynaptic cell). The branches of the axon may form synapses with as many as 1000 other neurons.

A single neuron

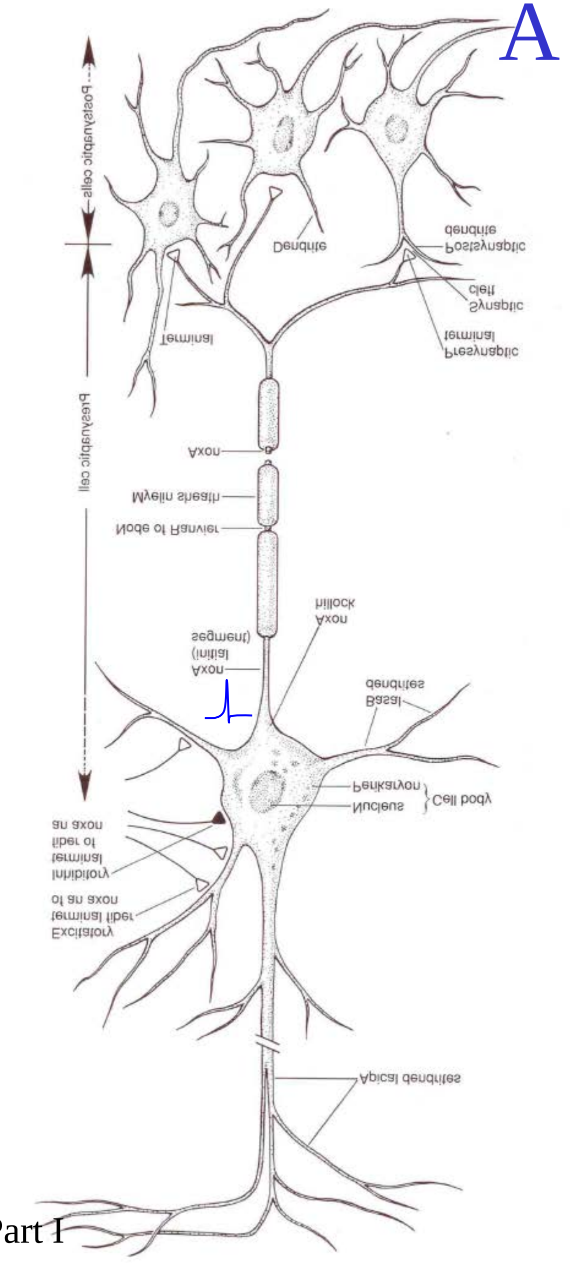
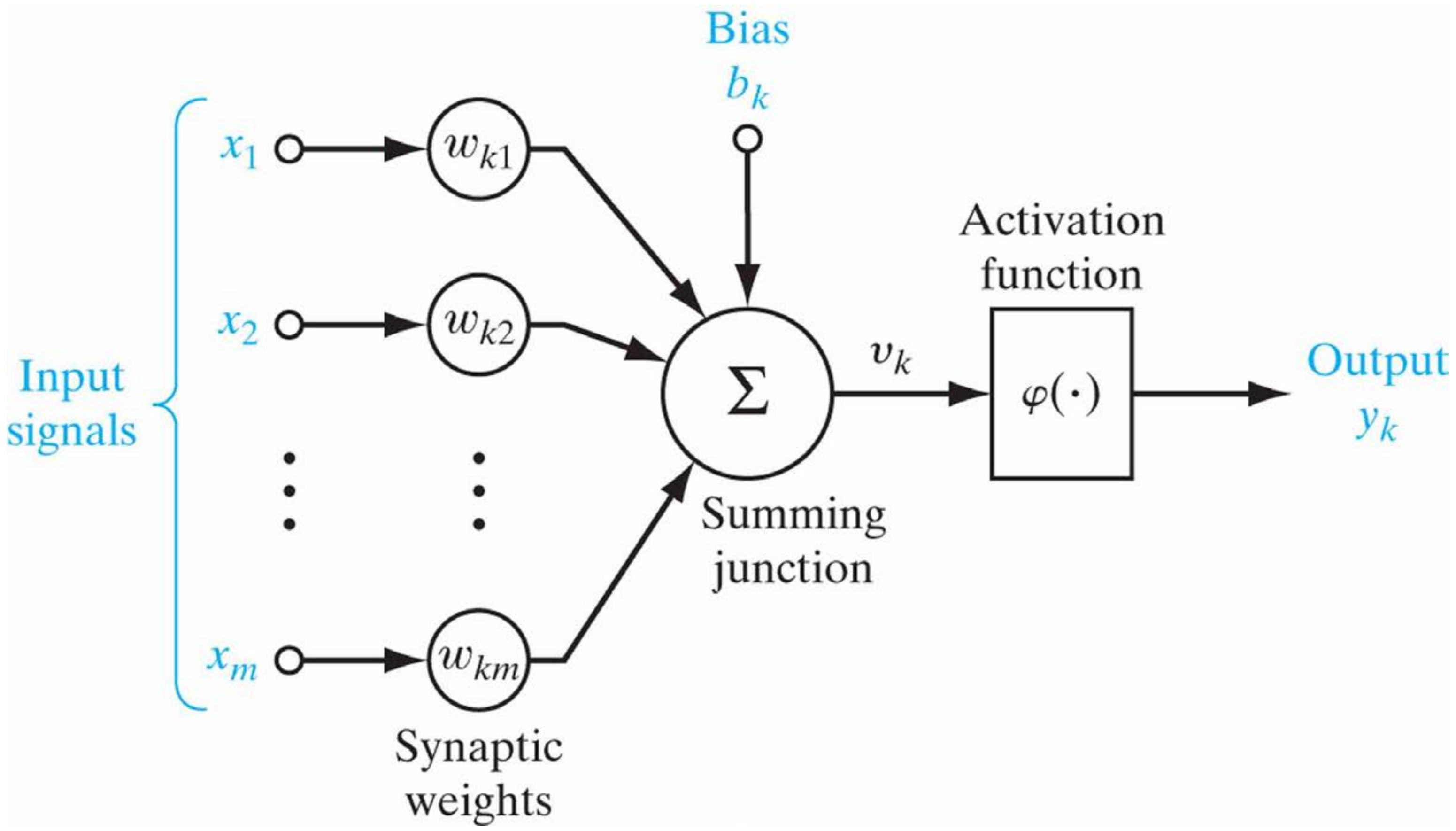


FIGURE 2–1
The main features of a typical vertebrate neuron. This neuron is drawn to illustrate its various regions and its points of contact with other nerve cells. The cell body contains the nucleus and perikaryon. The cell body gives rise to two types of processes—dendrites (both apical and basal) and axons. The axon is the transmitting element of the neuron. Axons vary greatly in length, with some extending more than 1 meter. Most axons in the central nervous system are very thin (between 0.2 and 20 μm) compared with the diameter of the cell body (up to 50 μm or more in diameter). The axon hillock, the region of the cell body where the axon emerges, is where the action potential is initiated. Many axons are insulated by a fatty myelin sheath, which is interrupted at regular intervals by regions known as the nodes of Ranvier. Branches of the axon of one neuron (the presynaptic neuron) form synaptic connections with the dendrites or cell body of another neuron (the postsynaptic cell). The branches of the axon may form synapses with as many as 1000 other neurons.

Real neurons, real synapses

- **Properties**
  - Action potential (impulse) generation
  - Impulse propagation
  - Synaptic transmission & plasticity
  - Spatial summation
- **Terminology**
  - Neurons – units – nodes
  - Synapses – connections – architecture
  - Synaptic weight – connection strength (either positive or negative)

Model of a single neuron



Neuronal model

$$u_k = \sum_{j=1}^{m} w_{kj} x_j$$

Adder, weighted sum, linear combiner

$$v_k = u_k + b_k$$

Activation potential; b_k: bias

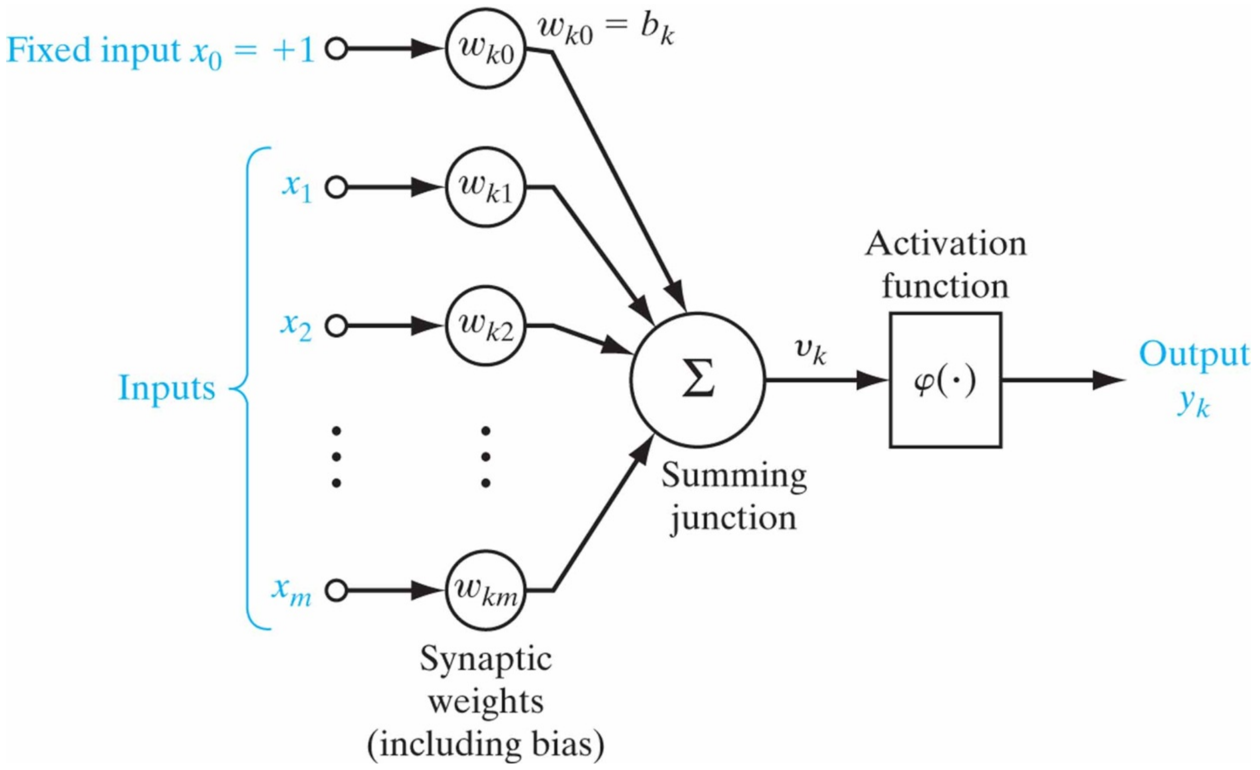
$$y_k = \varphi(v_k)$$

Output; φ: activation function

Another way of including bias

Set $x_0 = +1$ and $w_{k0} = b_k$

So we have $v_k = \sum_{j=0}^{m} w_{kj} x_j$



McCulloch-Pitts model

$x_i \in \{-1, 1\}$ Bipolar input

$$y = \varphi\left(\sum_{i=1}^{m} w_i x_i + b\right)$$

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ -1 & \text{if } v < 0 \end{cases}$$

A form of signum (sign) function

- Note difference from textbook; also number of neurons in the brain

McCulloch-Pitts model (cont.)

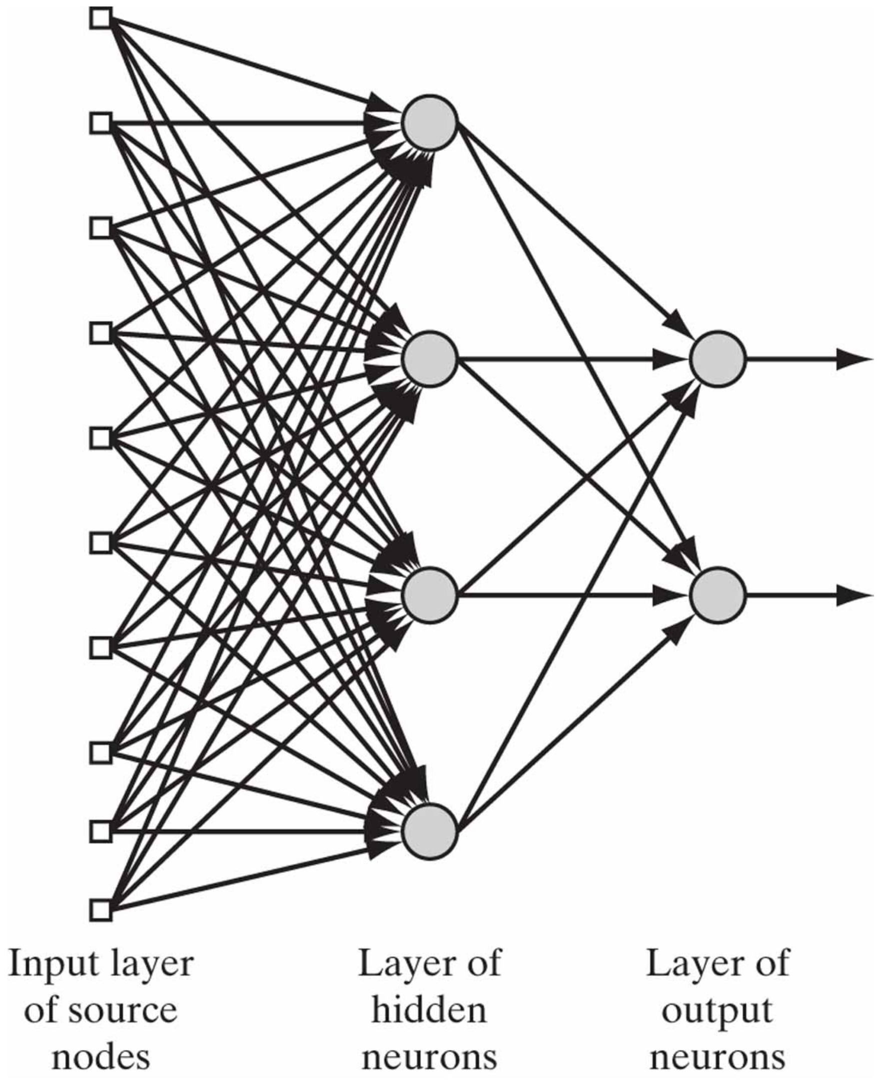
- Example logic gates (see blackboard)

- McCulloch-Pitts networks (introduced in 1943) are the first class of abstract computing machines: finite-state automata
  - Finite-state automata can compute any logic (Boolean) function

Network architecture

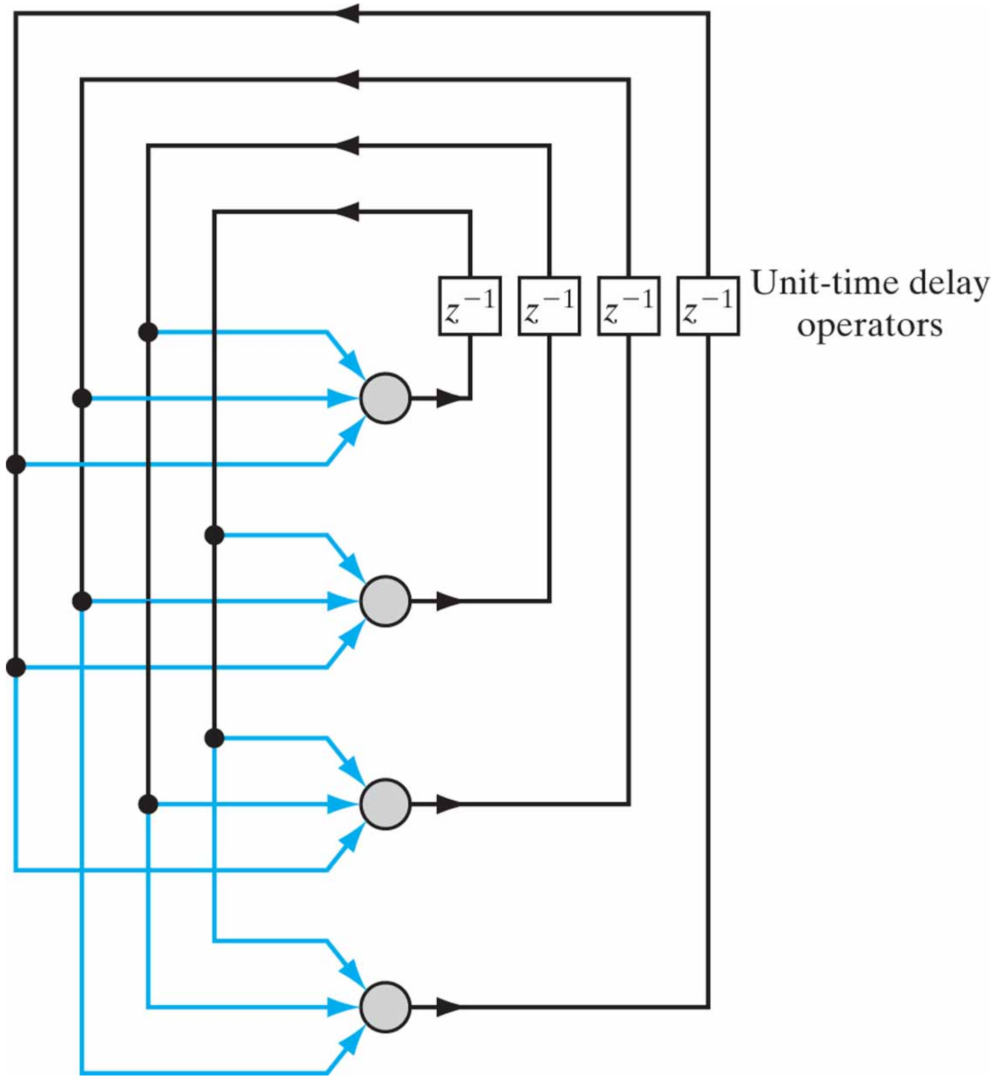
- View an NN as a connected, directed graph, which defines its architecture
  - Feedforward nets: loop-free graph
  - Recurrent nets: with loops

Feedforward net



- Since the input layer consists of source nodes, it is typically not counted when we talk about the number of layers in a feedforward net
- For example, the architecture of 10-4-2 counts as a two-layer net

A one-layer recurrent net



In this net, the input typically sets the initial condition of the output layer

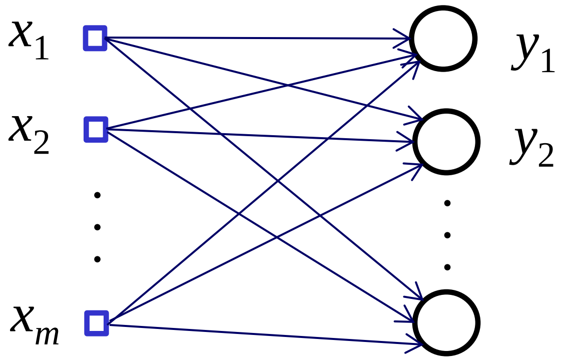
Network components

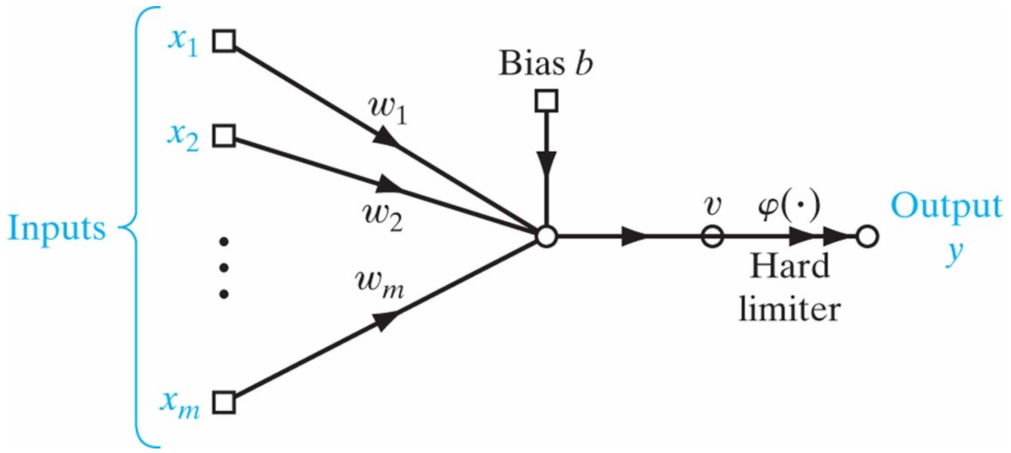
- **Three components characterize a neural net**
  - Architecture
  - Activation function
  - Learning rule (algorithm)

Perceptrons

Perceptrons

- Architecture: **one-layer feedforward net**
  - Without loss of generality, consider a single-neuron perceptron





Definition

$$y = \varphi(v)$$

$$v = \sum_{i=1}^{m} w_i x_i + b$$

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

Hence a McCulloch-Pitts neuron, but with real-valued inputs

Pattern recognition

- With a bipolar output, the perceptron performs a 2-class classification problem
  - Apples vs. oranges
- How do we learn to perform a classification problem?
- Task: The perceptron is given pairs of input $\mathbf{x}_p$ and desired output d_p. How to find **w** (with b incorporated) so that

$$y_p = d_p, \quad \text{for all } p?$$

Decision boundary

- The decision boundary for a given **w**:

$$g(\mathbf{x}) = \sum_{i=1}^{m} w_i x_i + b = 0$$

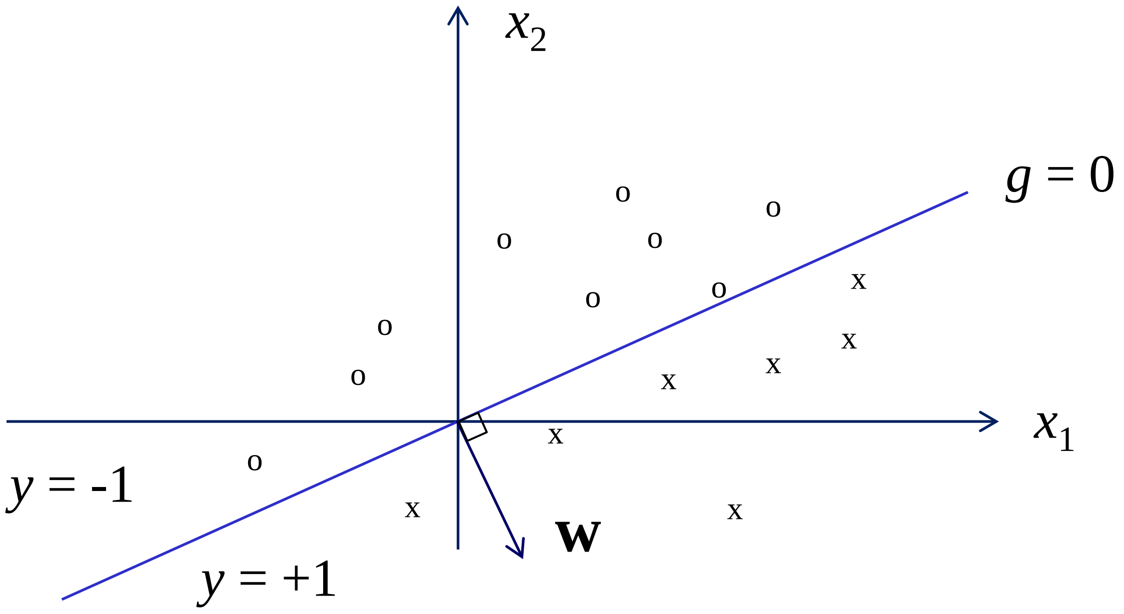
  - *g* is also called the discriminant function for the perceptron, and it is a linear function of **x**. Hence it is a linear discriminant function

Example

- See blackboard

Decision boundary (cont.)

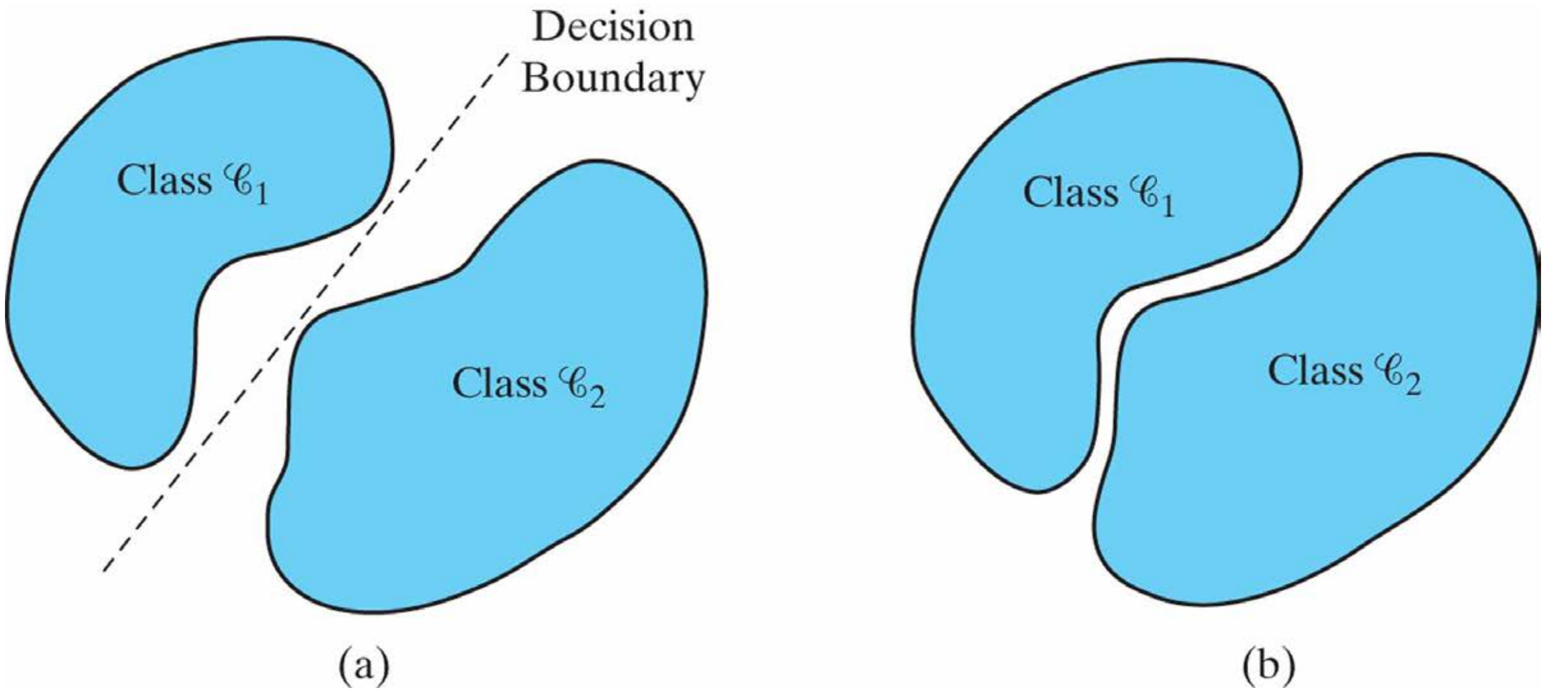
- For an m-dimensional input space, the decision boundary is an $(m - 1)$-dimensional hyperplane perpendicular to $\mathbf{w}$. The hyperplane separates the input space into two halves, with one half having $y = 1$, and the other half having $y = -1$
  - When $b = 0$, the hyperplane goes through the origin



Linear separability

- For a set of input patterns $\mathbf{x}_p$, if there exists one $\mathbf{w}$ that separates $d = 1$ patterns from $d = -1$ patterns, then the classification problem is linearly separable
  - In other words, there exists a linear discriminant function that produces no classification error
  - Examples: AND, OR, XOR (see blackboard)
- A very important concept

Linear separability: a more general illustration



Perceptron learning rule

- Strengthen an active synapse if the postsynaptic neuron fails to fire when it should have fired; weaken an active synapse if the neuron fires when it should not have fired
  - Formulated by Rosenblatt based on biological intuition

Quantitatively

$$w(n+1) = w(n) + \Delta w(n)$$

$$= w(n) + \eta[d(n) - y(n)]x(n)$$

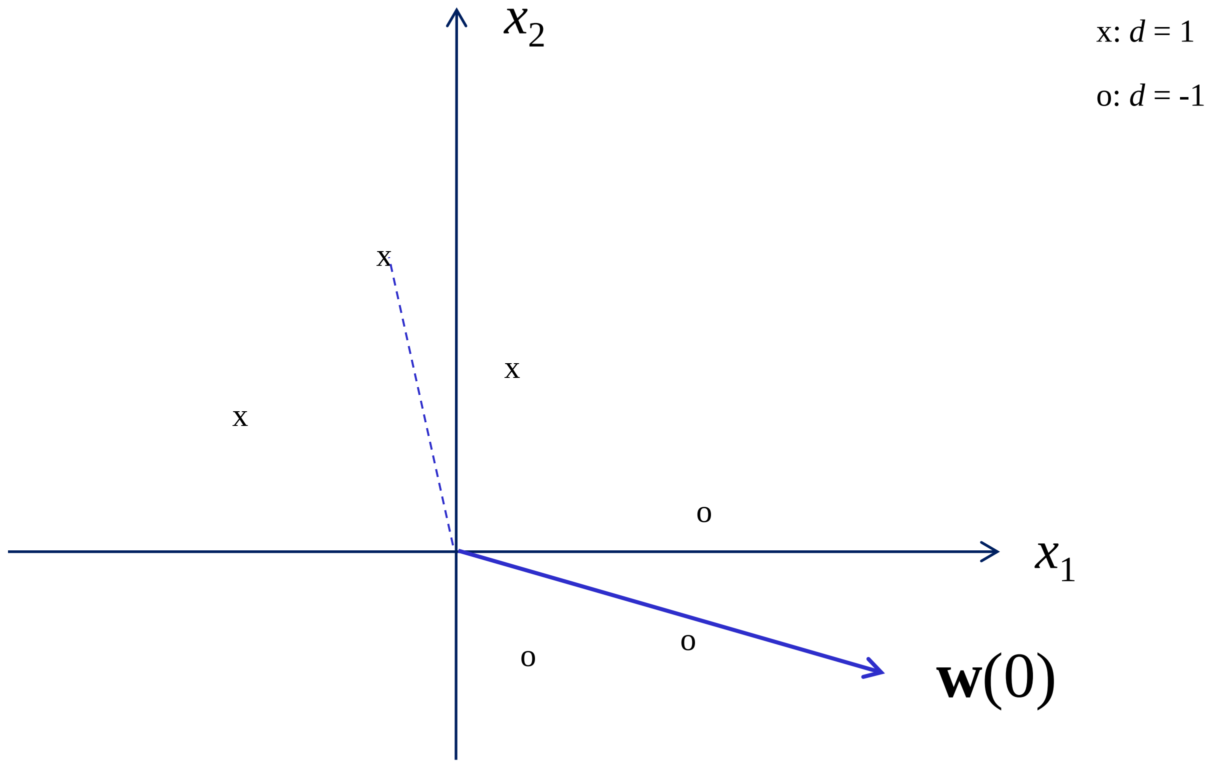
- n: iteration number
- η: step size or learning rate

In vector form

$$\Delta \mathbf{w} = \eta[d - y]\mathbf{x}$$

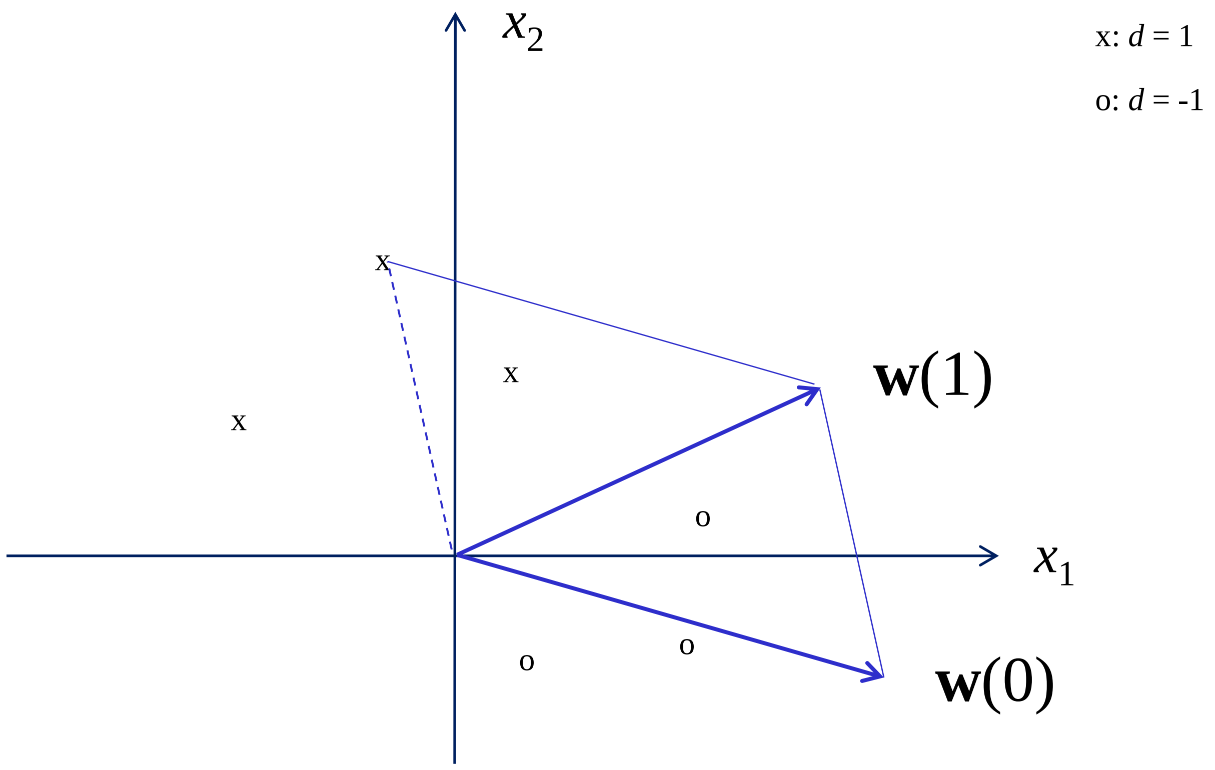
Geometric interpretation

- Assume $\eta = 1/2$



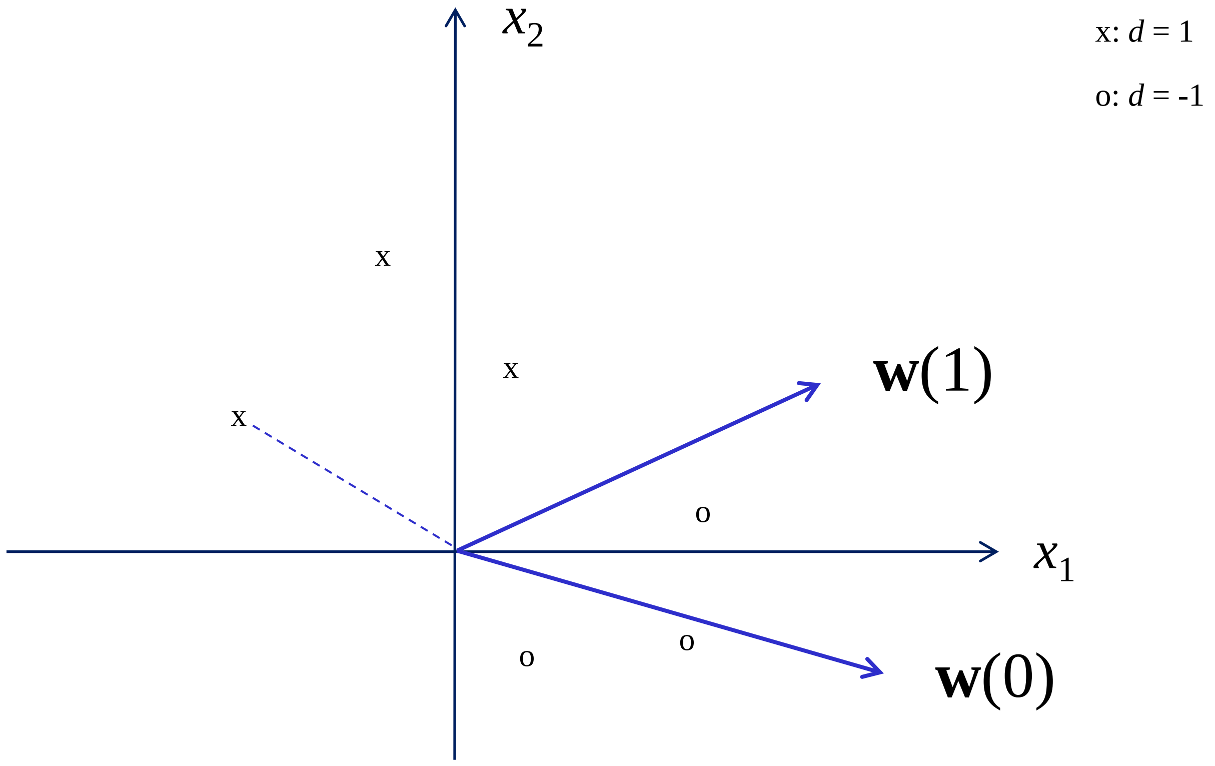
Geometric interpretation

- Assume η = 1/2



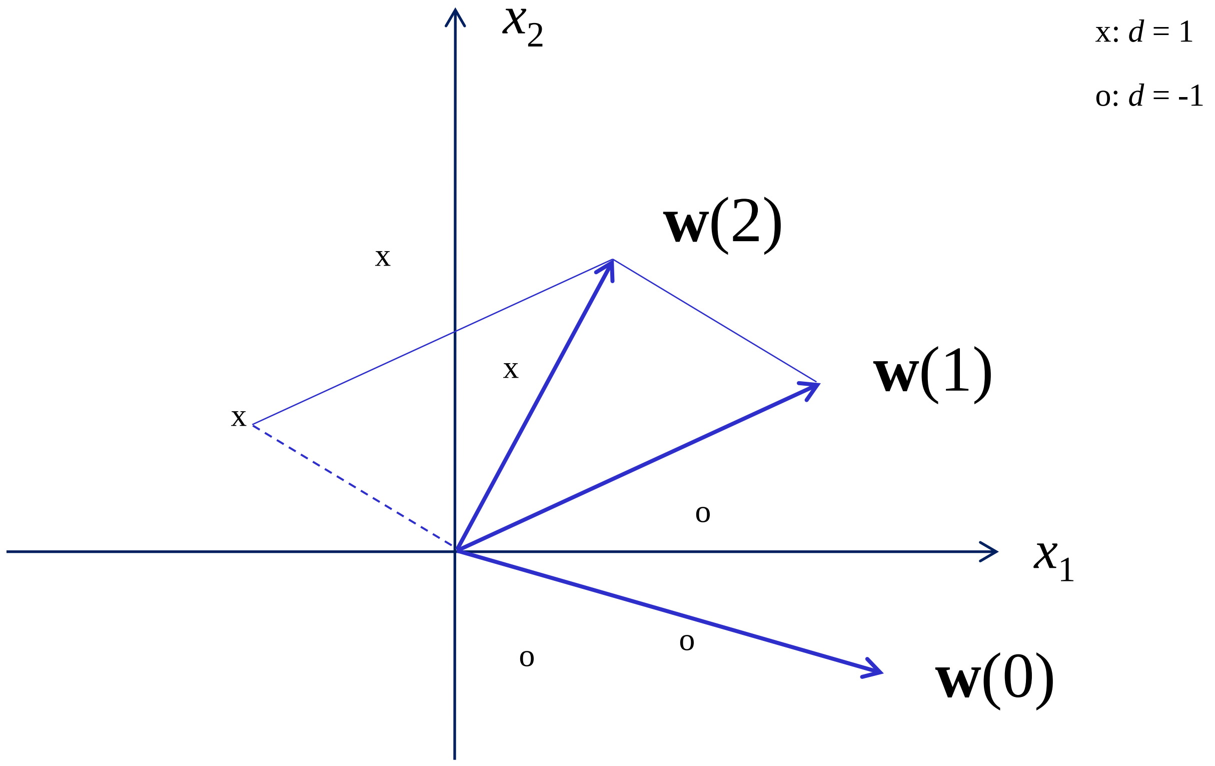
Geometric interpretation

- Assume $\eta = 1/2$



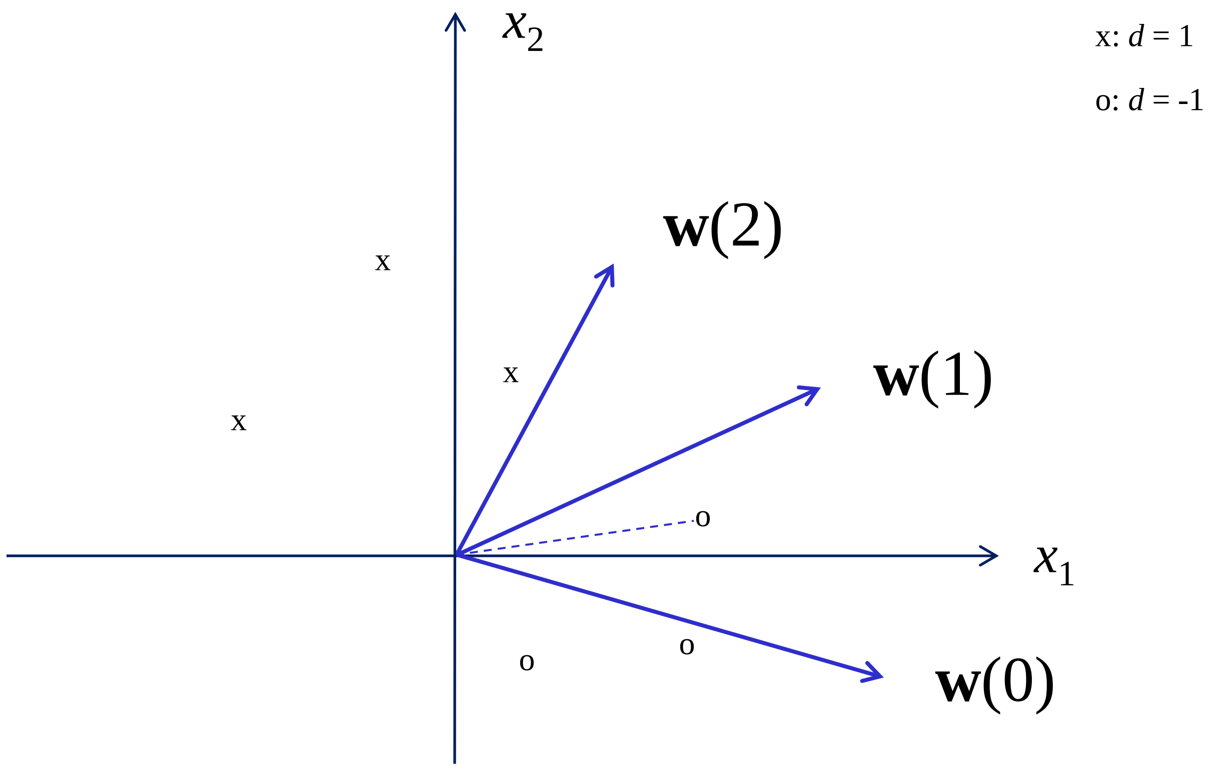
Geometric interpretation

- Assume $\eta = 1/2$



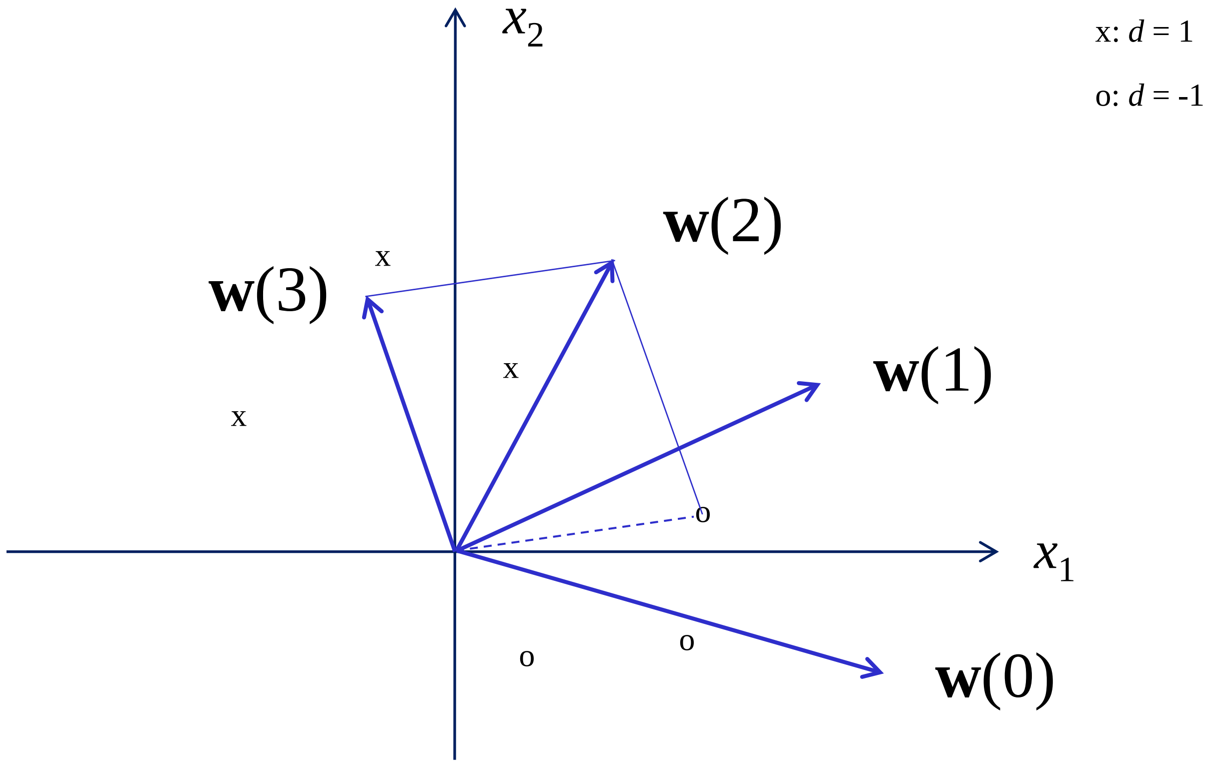
Geometric interpretation

- Assume $\eta = 1/2$

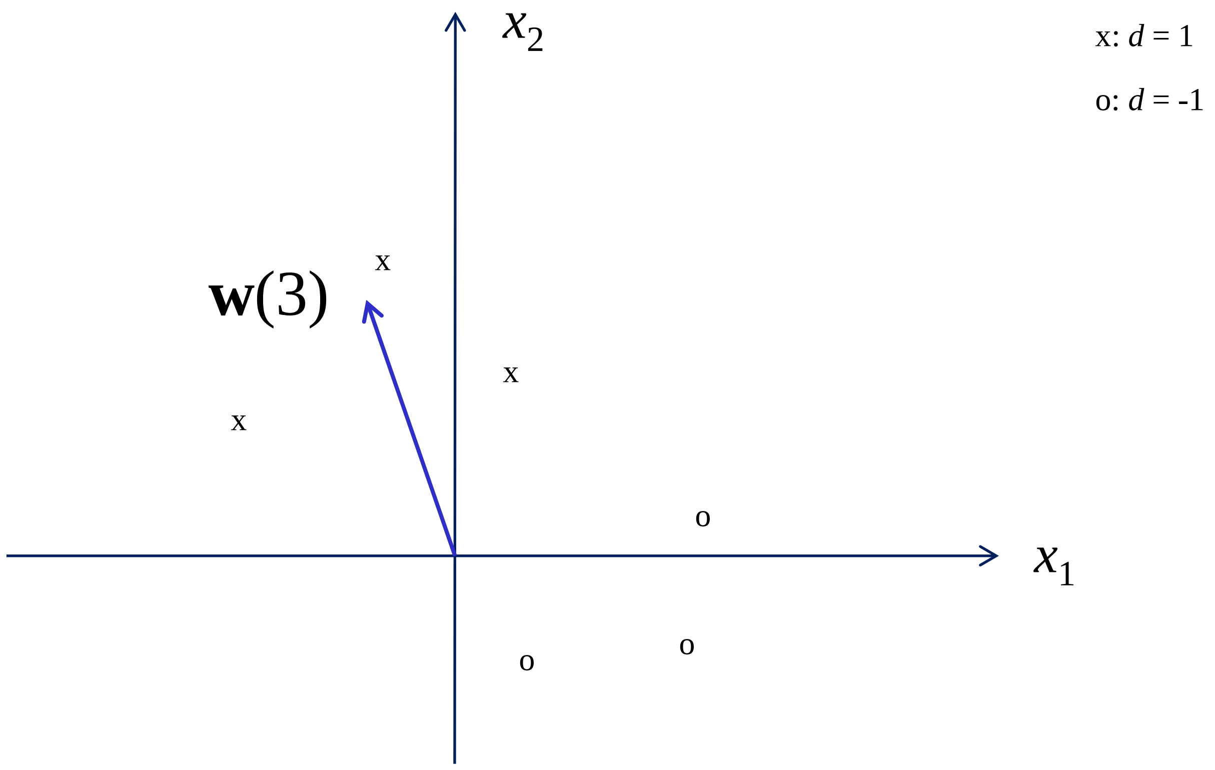


Geometric interpretation

- Assume $\eta = 1/2$



Geometric interpretation



Each weight update moves **w** closer to d = 1 patterns, or away from d = -1 patterns. **w**(3) solves the classification problem

Perceptron convergence theorem

- **Theorem**: *If a classification problem is linearly separable, a perceptron will reach a solution in a finite number of iterations*
- **[Proof]**

Given a finite number of training patterns, because of linear separability, there exists a weight vector $\mathbf{w}_o$ so that

$$d_p \mathbf{w}_o^T \mathbf{x}_p \geq \alpha > 0 \qquad (1)$$

where $\alpha = \min_p (d_p \mathbf{w}_o^T \mathbf{x}_p)$

Proof (cont.)

- We assume that the initial weights are all zero. Let N_p denote the number of times $\mathbf{x}_p$ has been used for *actually* updating the weight vector at some point in learning

At that time:

$$\mathbf{w} = \sum_p N_p [\eta (d_p - y_p) \mathbf{x}_p]$$

$$= 2\eta \sum_p N_p d_p \mathbf{x}_p$$

Proof (cont.)

- Consider $\mathbf{w}_o^T\mathbf{w}$ first

$$\mathbf{w}_o^T\mathbf{w} = 2\eta\sum_p N_p d_p \mathbf{w}_o^T\mathbf{x}_p$$

due to (1) $$\geq 2\eta\alpha\sum_p N_p$$

$$= 2\eta\alpha P \qquad (2)$$

where $P = \sum_p N_p$

Proof (cont.)

- Now consider the change in square length $\|\mathbf{w}\|^2$ after a single update by **x**:

$$\Delta\|\mathbf{w}\|^2 = \|\mathbf{w}+\Delta\mathbf{w}\|^2 - \|\mathbf{w}\|^2 = \|\mathbf{w}+2\eta d\mathbf{x}\|^2 - \|\mathbf{w}\|^2$$

$$= 4\eta^2\|\mathbf{x}\|^2 + 4\eta d\mathbf{w}^T\mathbf{x}$$

$$\le 4\eta^2\beta$$

where $\beta = \max_p \|\mathbf{x}_p\|^2$

Since upon an update, $d(\mathbf{w}^T\mathbf{x}) \le 0$

Proof (cont.)

- By summing $\|\mathbf{w}\|^2$ for *P* steps we have the bound:

$$\|\mathbf{w}\|^2 \leq 4\eta^2 \beta P \qquad (3)$$

- Now square the cosine of the angle between $\mathbf{w}_o$ and **w**, we have by (2) and (3)

$$1 \geq \frac{(\mathbf{w}_o^T \mathbf{w})^2}{\|\mathbf{w}_o\|^2 \|\mathbf{w}\|^2} \geq \frac{4\eta^2 \alpha^2 P^2}{4\eta^2 \beta P \|\mathbf{w}_o\|^2} = \frac{\alpha^2 P}{\beta \|\mathbf{w}_o\|^2}$$

$\uparrow$ Cauchy-Schwarz inequality

Proof (cont.)

- Thus, P must be finite to satisfy the above inequality. This complete the proof
- **Remarks**
  - In the case of $\mathbf{w}(0) = \mathbf{0}$, the learning rate has no effect on the proof. That is, the theorem holds no matter what η ($\eta > 0$) is
  - The solution weight vector is not unique

Generalization

- Performance of a learning machine on test patterns not used during training
- Example: Class 1: handwritten “*m*”: class 2: handwritten “*n*”
  - See blackboard
- Perceptrons generalize by deriving a decision boundary in the input space. Selection of training patterns is thus important for generalization