CS 4700: Foundations of Artificial Intelligence

Spring 2020
Prof. Haym Hirsh

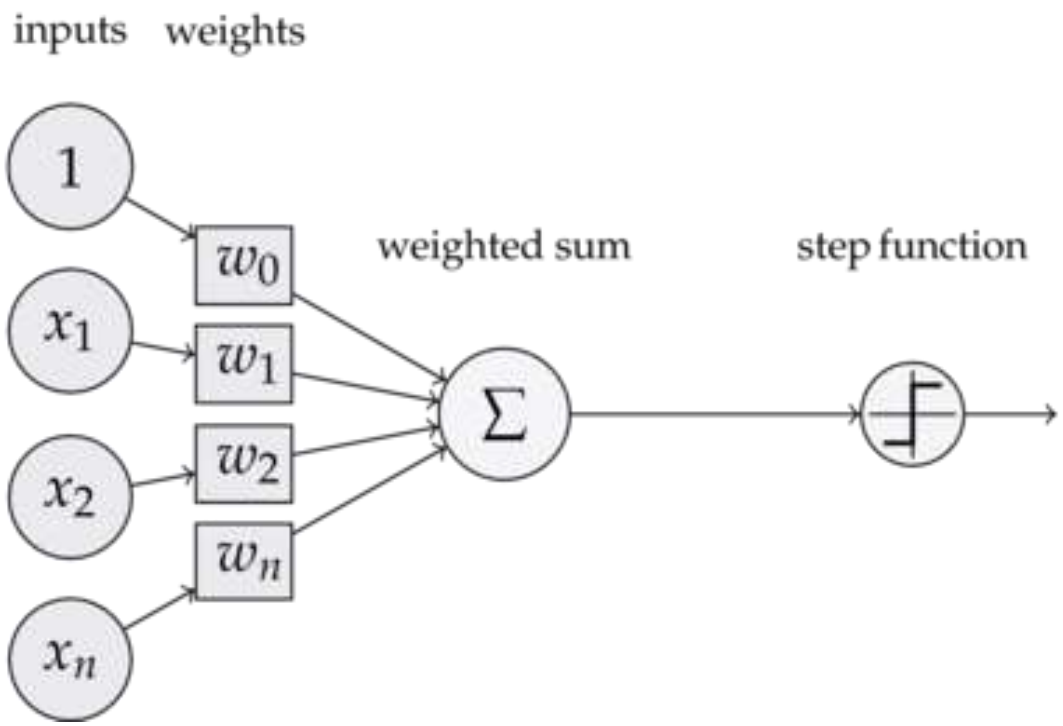
Lecture 27
April 17, 2020

ONLINE

Backup plans:

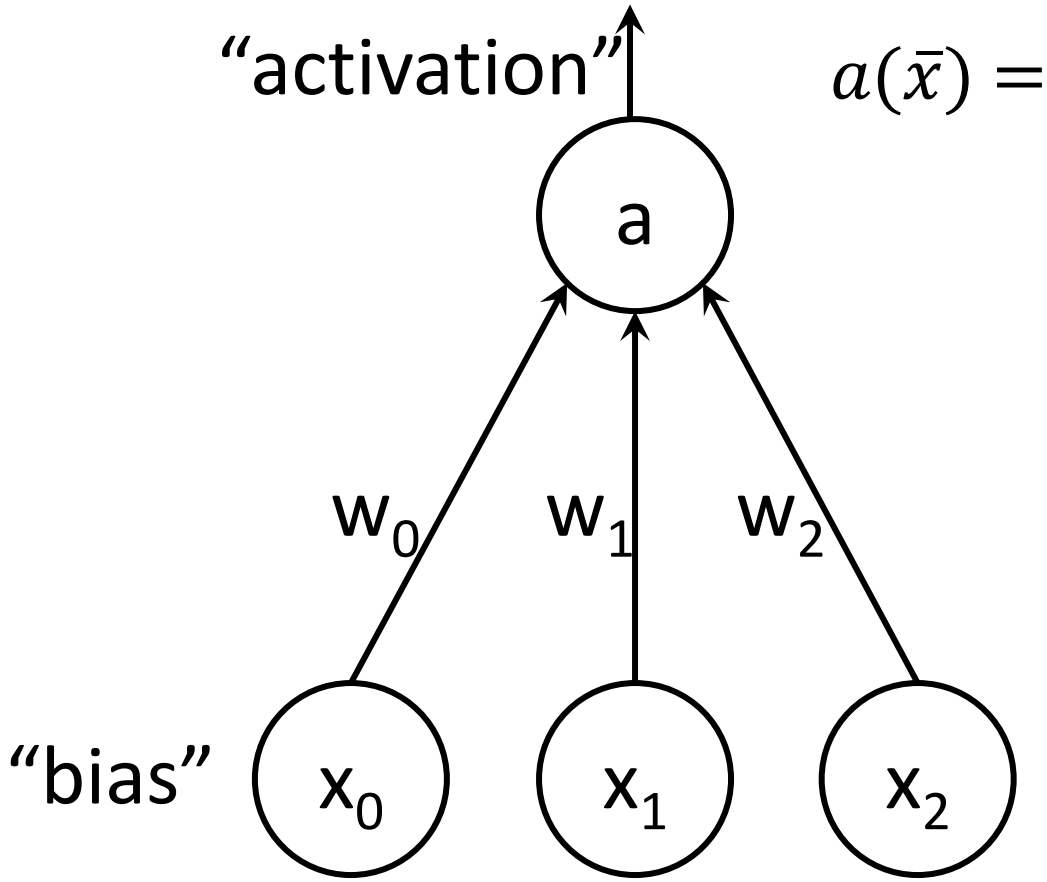
If this Zoom meeting ends prematurely,
five-minute break, check Piazza

Perceptrons



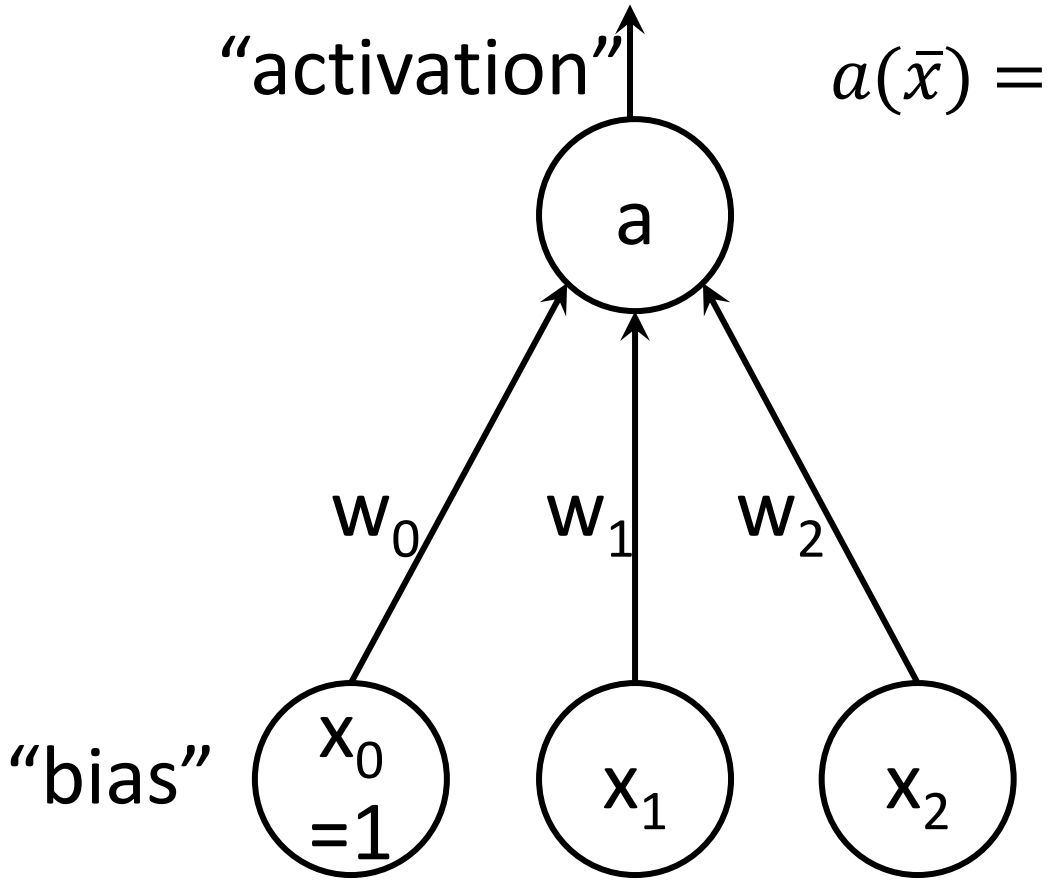
https://blog.dbrgn.ch/2013/3/26/perceptrons-in-python/

Abstract Representation of a Perceptron



$$a(\bar{x}) = +1 \text{ if } \sum_{i=0}^{n} w_i x_i = \bar{w} \cdot \bar{x} \geq 0$$
$$0 \quad \text{otherwise}$$

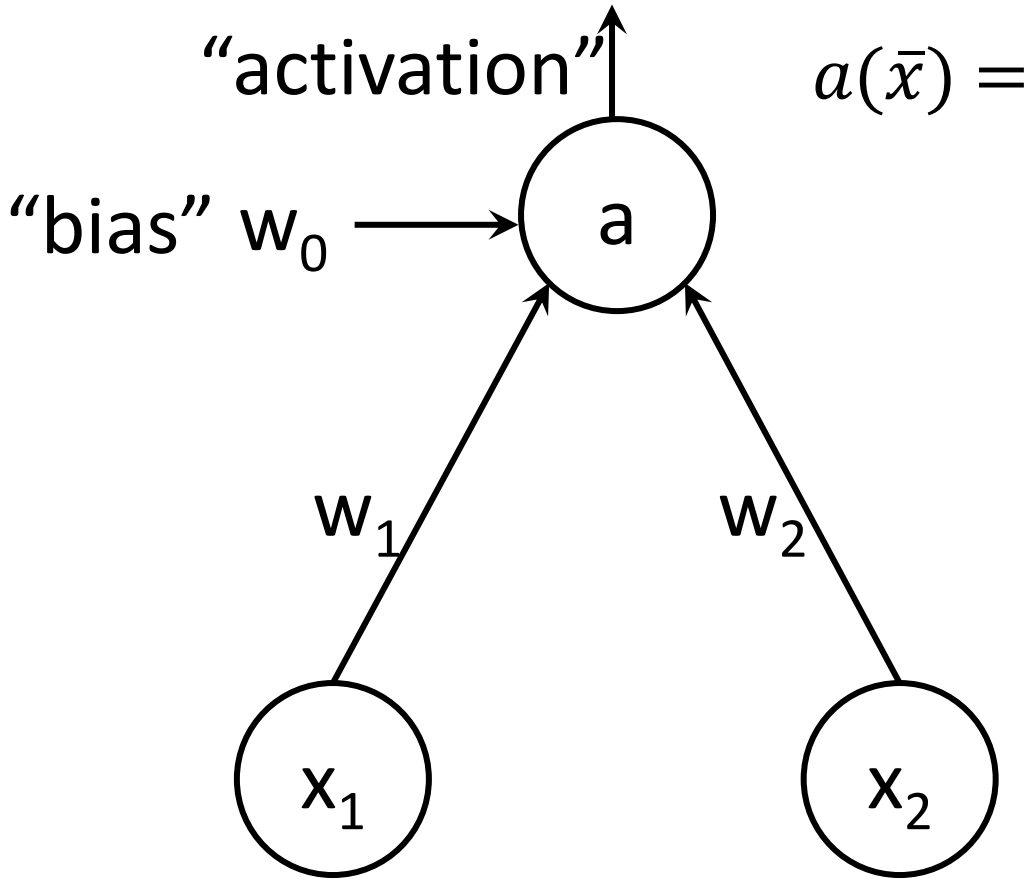
Abstract Representation of a Perceptron



$$a(\bar{x}) = +1 \text{ if } \sum_{i=0}^{n} w_i x_i = \bar{w} \cdot \bar{x} \geq 0$$

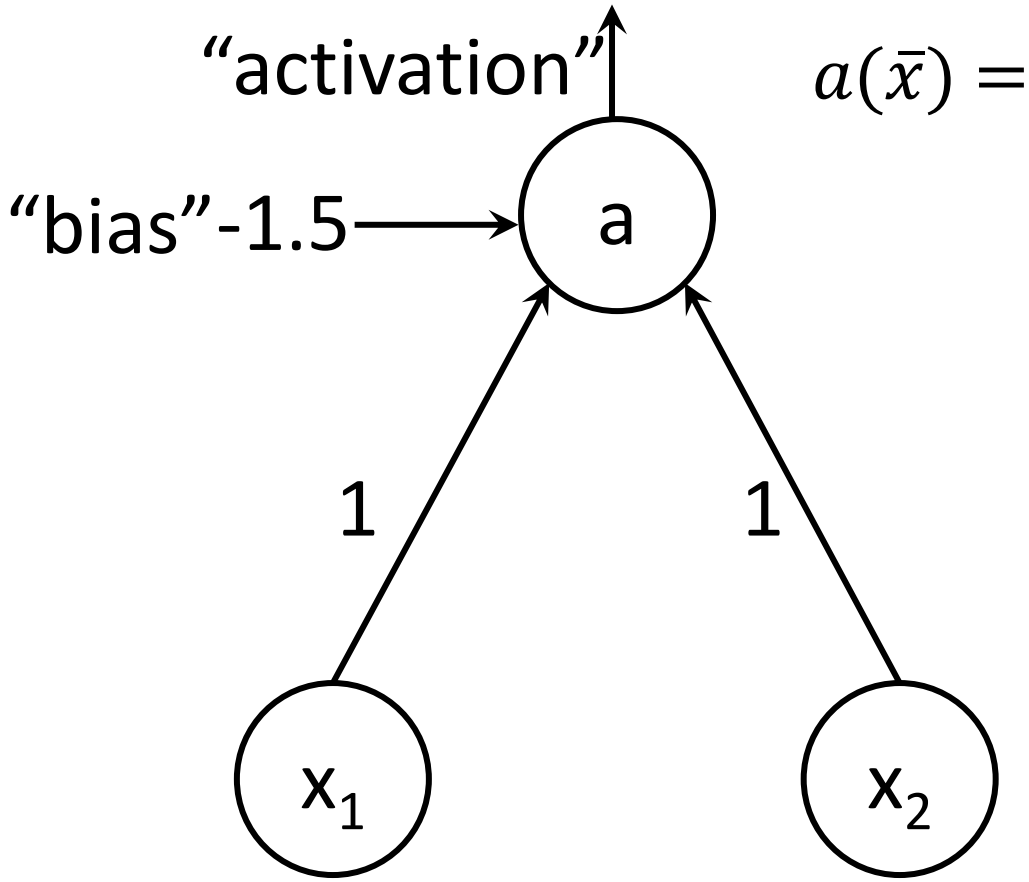
0 otherwise

Abstract Representation of a Perceptron



$$a(\bar{x}) = +1 \text{ if } \sum_{i=0}^{n} w_i x_i = \bar{w} \cdot \bar{x} \geq 0$$
$$0 \quad \text{otherwise}$$

Abstract Representation of a Perceptron



$$a(\bar{x}) = +1 \text{ if } \sum_{i=0}^{n} w_i x_i = \bar{w} \cdot \bar{x} \geq 0$$
$$0 \quad \text{otherwise}$$

Linear Classification

General form of a linear classifier

$$\mathrm{h}_{\overline{w}}(\bar{x}) = \begin{cases} +1 & \text{if } \sum_{i=0}^{n} w_i x_i = \overline{w} \cdot \bar{x} \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Finding a classifier given a set of data means finding a $\overline{w}$ for which $\mathrm{h}_{\overline{w}}(\overline{\mathrm{x}_i})$ comes close to $y_i = \mathrm{f}_{\overline{w}}(\overline{\mathrm{x}_i})$ for the training data

Linear Classification

General form of a linear classifier

$$\mathrm{h}_{\overline{w}}(\bar{x}) = \begin{cases} +1 & \text{if } \sum_{i=0}^{n} w_i x_i = \overline{w} \cdot \bar{x} \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Finding a classifier given a set of data means finding a $\overline{w}$ for which $\mathrm{h}_{\overline{w}}(\overline{\mathrm{x}}_i)$ comes close to $y_i = \mathrm{f}_{\overline{w}}(\overline{\mathrm{x}}_i)$ for the training data

Linear Classification

General form of a linear classifier

$$\mathrm{h}_{\overline{w}}(\bar{x}) = \begin{cases} +1 & \text{if } \sum_{i=0}^{n} w_i x_i = \overline{w} \cdot \bar{x} \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Finding a classifier given a set of data means finding a $\overline{w}$ for which $\mathrm{h}_{\overline{w}}(\overline{\mathrm{x}}_i)$ comes close to $y_i = \mathrm{f}_{\overline{w}}(\overline{\mathrm{x}}_i)$ for the training data

TBA

Linear Classification

General form of a linear classifier

$$\mathrm{h}_{\overline{w}}(\bar{x}) = \begin{cases} +1 & \text{if } \sum_{i=0}^{n} w_i x_i = \overline{w} \cdot \bar{x} \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Finding a classifier given a set of data means finding a $\overline{w}$ for which $\mathrm{h}_{\overline{w}}(\overline{\mathrm{x}_i})$ comes close to $y_i = \mathrm{f}_{\overline{w}}(\overline{\mathrm{x}_i})$ for the training data

Linear Classification

General form of a linear classifier

$$\mathrm{h}_{\overline{w}}(\bar{x}) = \begin{cases} +1 & \text{if } \sum_{i=0}^{n} w_i x_i = \overline{w} \cdot \bar{x} \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Finding a classifier given a set of data means finding a $\overline{w}$ for which $\mathrm{h}_{\overline{w}}(\overline{\mathrm{x}_i})$ comes close to $y_i = \mathrm{f}_{\overline{w}}(\overline{\mathrm{x}_i})$ for the training data

How?

Search

Learning with Perceptrons

- Start with an initial $\overline{w}$.

- Take an example <$\overline{x}_i$,y_i>
  - If $h_{\overline{w}}(\overline{x}_i)$=$y_i$ do nothing
  - Otherwise move $\overline{w}$ to make $\overline{w} \cdot \overline{x}$
    bigger (if y_i=1 and $h_{\overline{w}}(\overline{x}_i)$=0)
    smaller (if y_i=0 and $h_{\overline{w}}(\overline{x}_i)$=1)
- Repeat until you get all the labels right

Learning with Perceptrons

- Start with an initial $\overline{w}$. [Often either all w_i=0 or w_i=small number]

- Take an example <$\overline{x}_i$,y_i>
  - If $h_{\overline{w}}(\overline{x}_i)$=$y_i$ do nothing
  - Otherwise move $\overline{w}$ to make $\overline{w} \cdot \overline{x}$
    bigger (if y_i=1 and $h_{\overline{w}}(\overline{x}_i)$=0)
    smaller (if y_i=0 and $h_{\overline{w}}(\overline{x}_i)$=1)
- Repeat until you get all the labels right

Learning with Perceptrons

- Start with an initial $\overline{w}$.

- Take an example <$\overline{x}_i$,y_i>
  - If $h_{\overline{w}}(\overline{x}_i)$=$y_i$ do nothing
  - Otherwise move $\overline{w}$ to make $\overline{w} \cdot \overline{x}$
    bigger (if y_i=1 and $h_{\overline{w}}(\overline{x}_i)$=0)
    smaller (if y_i=0 and $h_{\overline{w}}(\overline{x}_i)$=1)
- Repeat until you get all the labels right

Learning with Perceptrons

- Start with an initial $\overline{w}$.

- Take an example <$\overline{x}_i$,y_i>
  - If $h_{\overline{w}}(\overline{x}_i)$=$y_i$ do nothing
  - Otherwise move $\overline{w}$ to make $\overline{w} \cdot \overline{x}$
    bigger (if y_i=1 and $h_{\overline{w}}(\overline{x}_i)$=0)
    smaller (if y_i=0 and $h_{\overline{w}}(\overline{x}_i)$=1)
- Repeat $\boxed{\text{until you get all the labels right}}$

"comes close to"

Learning with Perceptrons

- Start with an initial $\overline{w}$.

- Take an example <$\overline{x}_i$,y_i>
  - If $h_{\overline{w}}(\overline{x}_i)$=$y_i$ do nothing
  - Otherwise move $\overline{w}$ to make $\overline{w} \cdot \overline{x}$
    bigger (if y_i=1 and $h_{\overline{w}}(\overline{x}_i)$=0)
    smaller (if y_i=0 and $h_{\overline{w}}(\overline{x}_i)$=1)
- Repeat until you get all the labels right

Learning with Perceptrons

- Start with an initial $\overline{w}$.

- Take an example <$\overline{x}_i$,y_i>
  - If $h_{\overline{w}}(\overline{x}_i)$=$y_i$ do nothing
  - Otherwise $\boxed{\text{move } \overline{w}}$ to make $\overline{w} \cdot \overline{x}$
    bigger (if y_i=1 and $h_{\overline{w}}(\overline{x}_i)$=0)
    smaller (if y_i=0 and $h_{\overline{w}}(\overline{x}_i)$=1)
- Repeat until you get all the labels right

Perceptron Learning Rule

Move $\overline{w}$: Perceptron Learning Rule

$$w_j \leftarrow w_j + \alpha x_{ij}(y_i - h_{\overline{w}}(\bar{x}_i))$$

Perceptron Learning Rule

Move $\overline{w}$: Perceptron Learning Rule

$$w_j \leftarrow w_j + \alpha x_{ij}(y_i - h_{\overline{w}}(\bar{x}_i))$$

New value

Old value

Perceptron Learning Rule

Move $\overline{w}$: Perceptron Learning Rule

$$w_j^{t+1} \leftarrow w_j^{t} + \alpha x_{ij}(y_i - h_{\overline{w}^t}(\bar{x}_i))$$

New value

Old value

Perceptron Learning Rule

Move $\overline{w}$: Perceptron Learning Rule

$$w_j \leftarrow w_j + \alpha x_{ij}(y_i - h_{\overline{w}}(\bar{x}_i))$$

New value

Old value

Perceptron Learning Rule

Move $\overline{w}$: Perceptron Learning Rule

$$w_j \leftarrow w_j + \overbrace{\boxed{\alpha x_{ij}(y_i - h_{\overline{w}}(\bar{x}_i))}}^{\Delta w_j}$$

New value

Old value

Perceptron Learning Rule

Move $\overline{w}$: Perceptron Learning Rule

$$w_j \leftarrow w_j + \overset{\Delta w_j}{\boxed{\alpha x_{ij}(y_i - h_{\overline{w}}(\bar{x}_i))}}$$

Perceptron Learning Rule

Move $\overline{w}$: Perceptron Learning Rule

$$w_j \leftarrow w_j + \overset{\Delta w_j}{\boxed{\alpha x_{ij}(y_i - h_{\overline{w}}(\bar{x}_i))}}$$

- If $h_{\overline{w}}(\bar{x}_i)$ is correct ($y_i = h_{\overline{w}}(\bar{x}_i)$), all w_j are unchanged
- If $h_{\overline{w}}(\bar{x}_i)$ is too big (y_i=0 and $h_{\overline{w}}(\bar{x}_i)$=1), w_j decreases
- If $h_{\overline{w}}(\bar{x}_i)$ is too small (y_i=1 and $h_{\overline{w}}(\bar{x}_i)$=0), w_j increases

Perceptron Learning Rule

Move $\overline{w}$: Perceptron Learning Rule

$$w_j \leftarrow w_j + \overset{\Delta w_j}{\boxed{\alpha x_{ij}(y_i - h_{\overline{w}}(\bar{x}_i))}}$$

- x_{ij} gets the sign right and scales the update by the input value

Perceptron Learning Rule

Move $\overline{w}$: Perceptron Learning Rule

$$w_j \leftarrow w_j + \overset{\Delta w_j}{\boxed{\alpha x_{ij}(y_i - h_{\overline{w}}(\bar{x}_i))}}$$

- α is the *learning rate* (sometimes called η)
  - Smaller means smaller increments, more conservative and slower
  - Larger means bigger jumps, can be faster, or can miss solution*

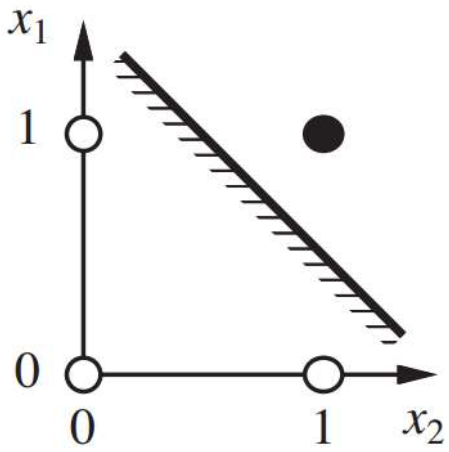
Perceptron Learning Rule

Perceptron($\{(\bar{x}_i, y_i)_{i \in \{1:N\}}\}$)

$w_0 = w_1 = w_2 = \ldots = w_n = 0$
Repeat
    For i = 1 to N [for each example]
        $h \leftarrow h_{\bar{w}}(\bar{x}_i)$
        For j = 0 to n [for each feature]
            $w_j \leftarrow w_j + \alpha x_{ij}(y_i - h)$ [Perceptron learning rule]
Until $h_{\bar{w}}(\bar{x})$ gets all data correct

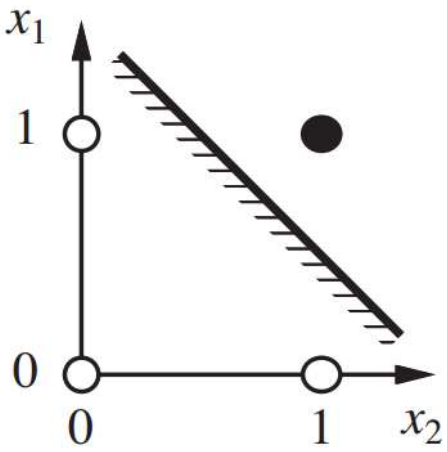
Perceptron Learning Rule: Example

$$w_j \leftarrow w_j + \alpha x_{ij}(y_i - h)$$



Perceptron Learning Rule: AND Example

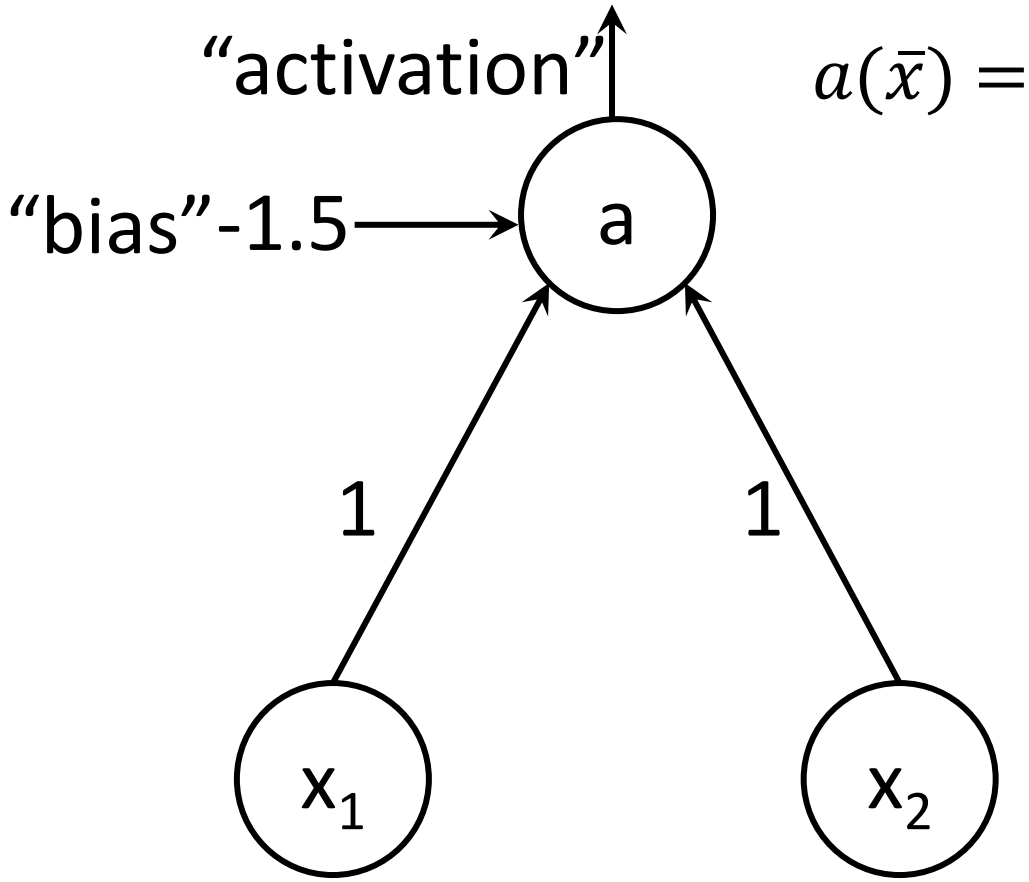
$$w_j \leftarrow w_j + \alpha x_{ij}(y_i - h)$$



And gate

x_1	x_2	$f(x_1,x_2)$
0	0	0
0	1	0
1	0	0
1	1	1

Abstract Representation of a Perceptron



$$a(\bar{x}) = +1 \text{ if } \sum_{i=0}^{n} w_i x_i = \bar{w} \cdot \bar{x} \geq 0$$
$$0 \quad \text{otherwise}$$

Perceptron Learning Rule: AND Example

$$w_j \leftarrow w_j + \alpha x_{ij}(y_i - h)$$

$$\alpha = 0.3, \quad w_0 = w_1 = w_2 = 0$$

Training Data

x_1	x_2	$f(x_1,x_2)$
0	0	0
0	1	0
1	0	0
1	1	1

https://docs.google.com/spreadsheets/d/1BHVKFw1nO1sIIuhs-3Ry4hL64B4Ja15WaE335jmLQXA

Excel Version (Uses Variable Names):

https://drive.google.com/file/d/1Yv1IdsBGKojytTRS97DtZKrJujTogmVb/view

Perceptron Learning Rule: 2nd Example

$$w_j \leftarrow w_j + \alpha x_{ij}(y_i - h)$$

$$\alpha = 0.3, \quad w_0 = w_1 = w_2 = 0$$

Training Data

x_1	x_2	$f(x_1,x_2)$
-1	0.5	1
-1	-1	1
1	0.5	0
0.5	1	0

Perceptron Learning Rule: 3rd Example

$$w_j \leftarrow w_j + \alpha x_{ij}(y_i - h)$$

$$\alpha = 0.3, \quad w_0 = w_1 = w_2 = 0$$

Training Data

x_1	x_2	$f(x_1,x_2)$
0	0	0
0	1	1
1	0	1
1	1	0

Perceptron Learning Rule: 3rd Example

$$w_j \leftarrow w_j + \alpha x_{ij}(y_i - h)$$

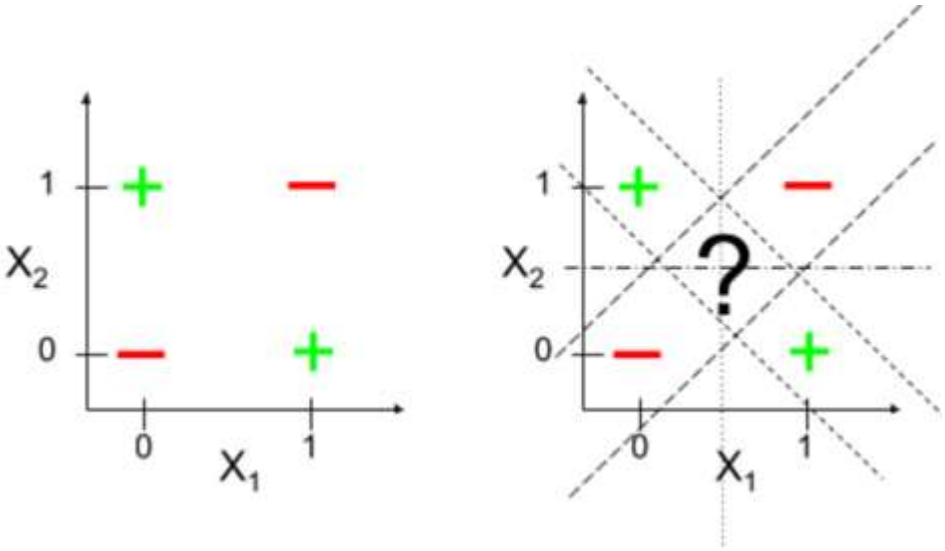
$$\alpha = 0.3, \quad w_0 = w_1 = w_2 = 0$$

Training Data

XOR

x_1	x_2	$f(x_1, x_2)$
0	0	0
0	1	1
1	0	1
1	1	0

Perceptrons: The Problem with XOR



(https://medium.com/@claude.coulombe/the-revenge-of-perceptron-learning-xor-with-tensorflow-eb52cbdf6c60)

Perceptron Convergence Theorem

If the training data are linearly separable (there exists a hyperplane that separates the 0's and 1's) then the perceptron learning rule ($0<\alpha\leq 1$) will find a separating hyperplane in a finite number of steps.

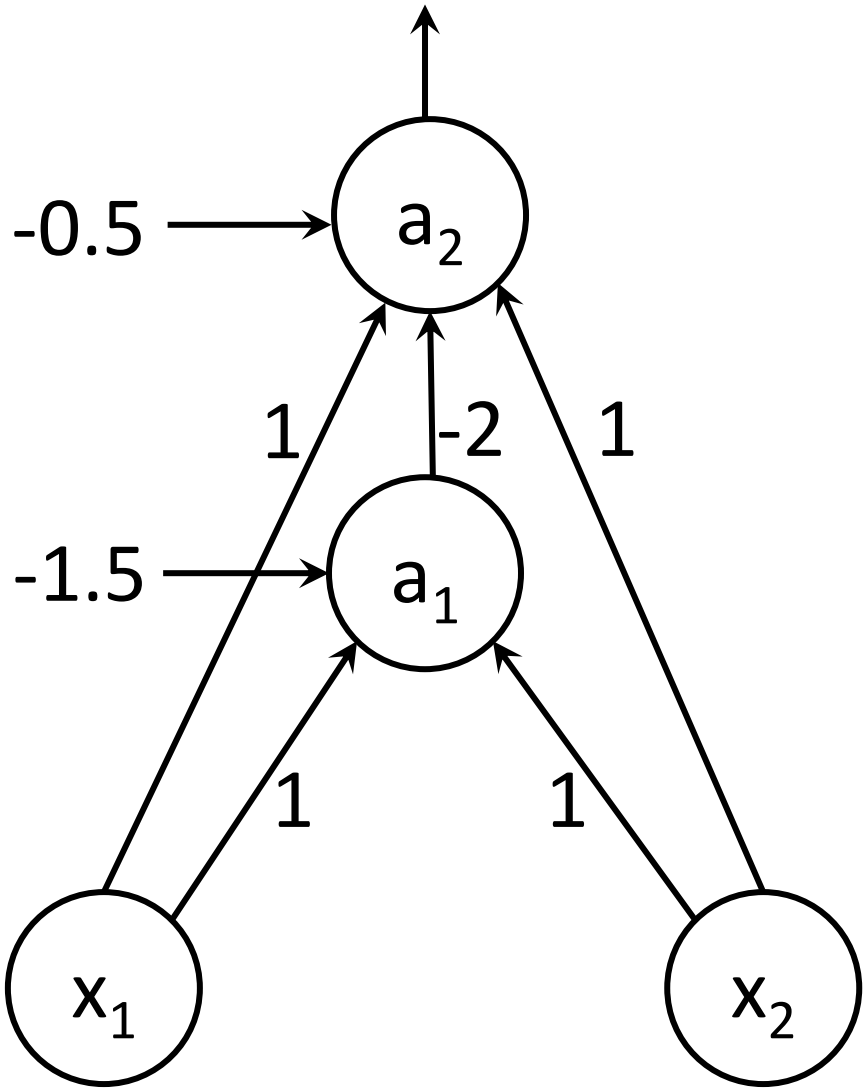
Perceptron Convergence Theorem

If the training data are linearly separable (there exists a hyperplane that separates the 0's and 1's) then the perceptron learning rule ($0<\alpha\leq 1$) will find a separating hyperplane in a finite number of steps.

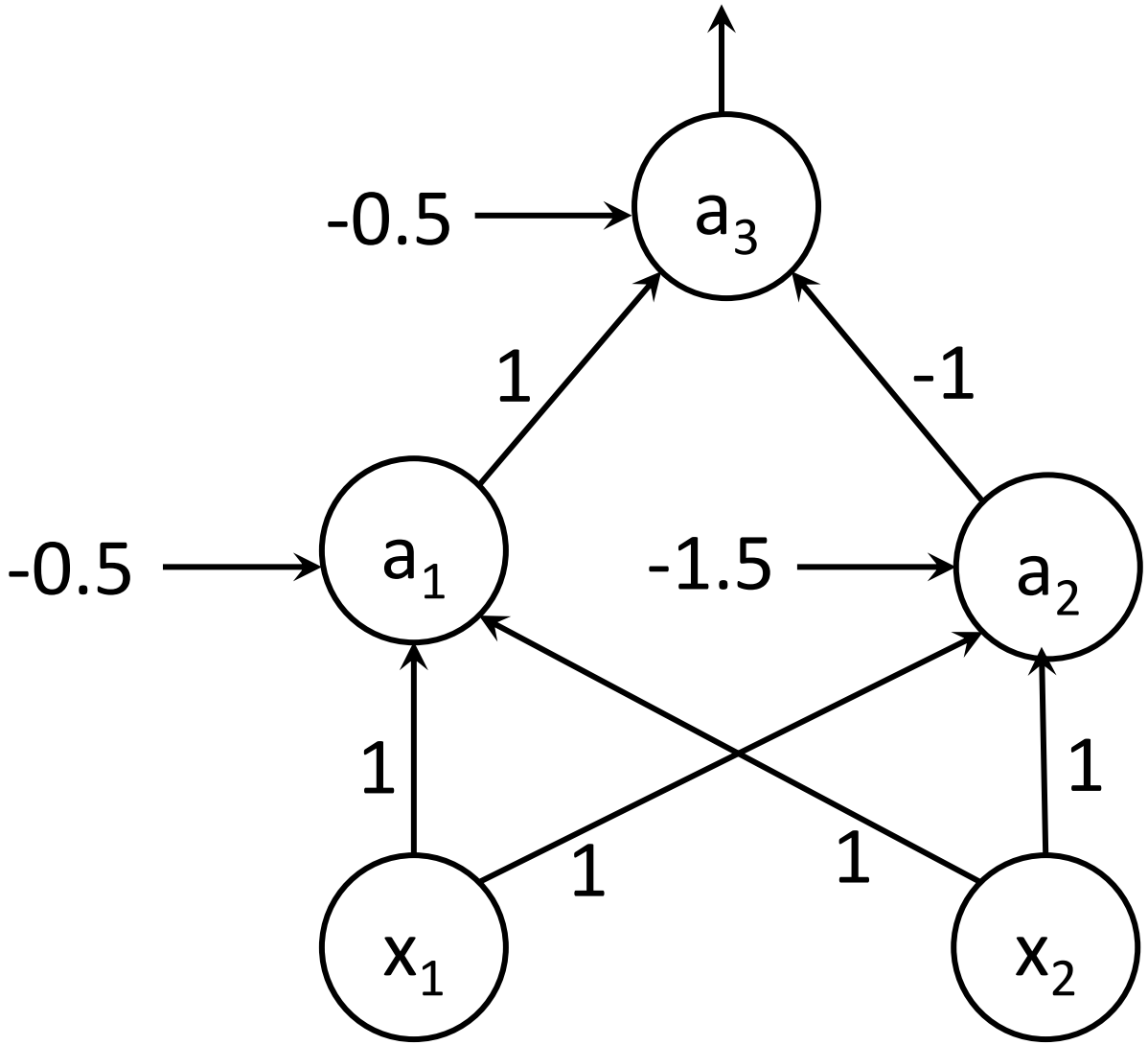
Many other theorems. Examples:

- If not linearly separable weights will eventually cycle. (α constant)
- Yields lowest error on data (α decreases as O(1/iteration-number))

Perceptron: The Solution for XOR?

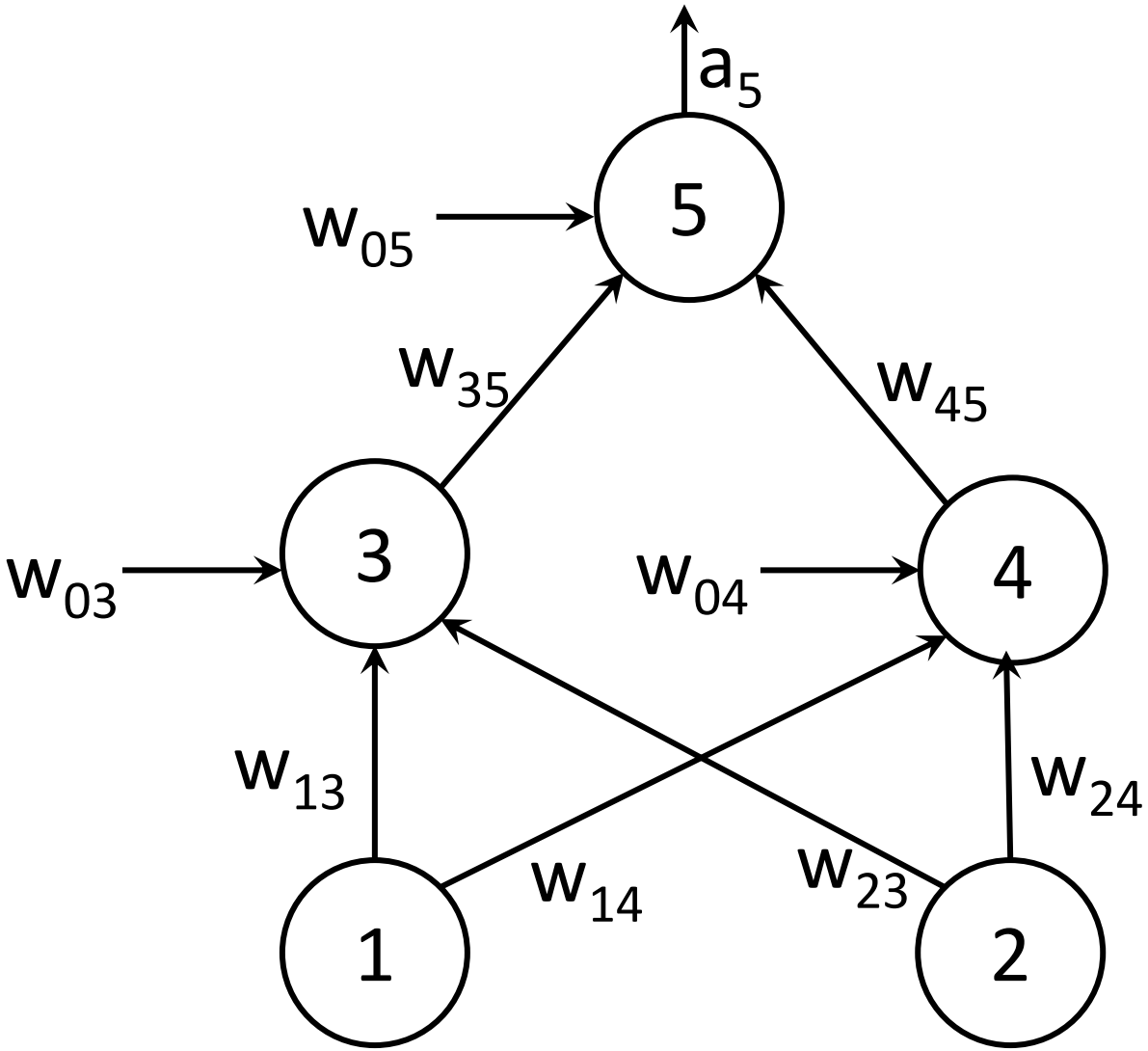


Perceptron: The Solution for XOR?



Multi-layer
Neural Network

Multi-Layer Perceptron



Multi-layer
Neural Network