Artificial Intelligence Foundation – JC3001

Lecture 42: Neural Networks -II

Prof. Aladdin Ayesh (aladdin.ayesh@abdn.ac.uk)

Dr. Binod Bhattarai (binod.bhattarai@abdn.ac.uk)

Dr. Gideon Ogunniye, (g.ogunniye@abdn.ac.uk)

November 2025

Material adapted from:
Russell and Norvig (AIMA Book): Chapter 21
Andrew Ng (Stanford University / Coursera)

Course Progression

- Part 1: Introduction
  1. Introduction to AI ✓
  2. Agents ✓
- Part 2: Problem-solving
  1. Search 1: Uninformed Search ✓
  2. Search 2: Heuristic Search ✓
  3. Search 3: Local Search ✓
  4. Search 4: Adversarial Search ✓
- Part 3: Reasoning and Uncertainty
  1. Reasoning 1: Constraint Satisfaction ✓
  2. Reasoning 2: Logic and Inference ✓
  3. Probabilistic Reasoning 1: BNs ✓
  4. Probabilistic Reasoning 2: HMMs ✓
- Part 4: Planning
  1. Planning 1: Intro and Formalism ✓
  2. Planning 2: Algos and Heuristics ✓
  3. Planning 3: Hierarchical Planning ✓
  4. Planning 4: Stochastic Planning ✓
- Part 5: Learning
  1. Learning 1: Intro to ML ✓
  2. Learning 2: Regression ✓
  3. **Learning 3: Neural Networks**
  4. Learning 4: Reinforcement Learning
- Part 6: Conclusion
  1. Ethical Issues in AI
  2. Conclusions and Discussion

Objectives

- From regression to perceptrons ✓
- Neural Networks
- Deep Learning basics

Outline

- Neural Networks - Training Stage

- Perceptron Rule (threshold)
  - $X \Rightarrow$ input vector
  - $Y \Rightarrow$ target vector
  - $y^{'} \Rightarrow$ perceptron output
  - $\boldsymbol{W} \Rightarrow$ weights vector
  - $\alpha \Rightarrow$ learning rate

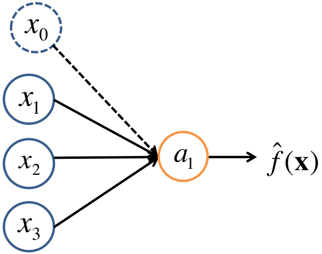
Algorithm 1 Perceptron learning algorithm.

Perceptron Training

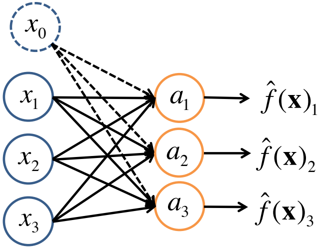
- Single neuron layer
- Output neurons are binary: $\{-1, +1\}$ (threshold function)

$$g(z) = \begin{cases} +1 & \text{if } z \geq 0 \\ -1 & \text{if } z < 0 \end{cases}$$

Binary



3-class

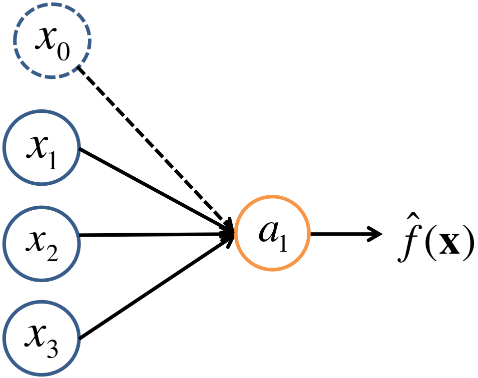


Perceptron Training Example

Given a perceptron network with:

- 3 input features
- 2 classes (thus single output neuron)
- Initial weights:

$$w_0 = -0.5$$
$$w_1 = 0.4$$
$$w_2 = -0.6$$
$$w_3 = 0.6$$



- Learning rate $\alpha = 0.4$

1. Train network with examples $(001, -1)$ and $(110, +1)$
2. Compute output for test examples 111, 000, 100, and 011

Training the Network (Training Example 1)

Instance $x^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}^T$ $\quad f(x^{(1)}) = -1$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_0 & = & -0.5 \\ \boldsymbol{w}_1 & = & 0.4 \\ \boldsymbol{w}_2 & = & -0.6 \\ \boldsymbol{w}_3 & = & 0.6 \end{bmatrix}$$

Training the Network (Training Example 1)

Instance $x^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}^T \quad f(x^{(1)}) = -1$

1. Compute network output:

$$\boldsymbol{W}^T x^{(1)} = (1 \times -0.5) + (0 \times 0.4) + (0 \times -0.6) + (1 \times 0.6) = 0.1$$

$$\hat{f}(x^{(1)}) = threshold(\boldsymbol{W}^T x^{(1)}) = +1 \text{ since } 0.1 \geq 0$$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_0 & = & -0.5 \\ \boldsymbol{w}_1 & = & 0.4 \\ \boldsymbol{w}_2 & = & -0.6 \\ \boldsymbol{w}_3 & = & 0.6 \end{bmatrix}$$

Training the Network (Training Example 1)

Instance $x^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}^T \qquad f(x^{(1)}) = -1$

1. Compute network output:

$$\boldsymbol{W}^T x^{(1)} = (1 \times -0.5) + (0 \times 0.4) + (0 \times -0.6) + (1 \times 0.6) = 0.1$$

$$\hat{f}(x^{(1)}) = threshold(\boldsymbol{W}^T x^{(1)}) = +1 \text{ since } 0.1 \geq 0$$

2. Update weighs: since $\hat{f}(x^{(1)}) \neq f(x^{(1)})$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_0 & = & -0.5 \\ \boldsymbol{w}_1 & = & 0.4 \\ \boldsymbol{w}_2 & = & -0.6 \\ \boldsymbol{w}_3 & = & 0.6 \end{bmatrix}$$

Training the Network (Training Example 1)

Instance $x^{(1)} = \begin{bmatrix}1 & 0 & 0 & 1\end{bmatrix}^T$ $\quad f(x^{(1)}) = -1$

1. Compute network output:

$$\boldsymbol{W}^T x^{(1)} = (1 \times -0.5) + (0 \times 0.4) + (0 \times -0.6) + (1 \times 0.6) = 0.1$$

$$\hat{f}(x^{(1)}) = threshold(\boldsymbol{W}^T x^{(1)}) = +1 \text{ since } 0.1 \geq 0$$

2. Update weighs: since $\hat{f}(x^{(1)}) \neq f(x^{(1)})$

$\Rightarrow \boldsymbol{w}_j \leftarrow \boldsymbol{w}_j + \alpha(y_i - y') * x_i$

$$\begin{aligned}
\boldsymbol{w}_0 &= -0.5 + 0.4(-1 - (+1))1 &= -1.3\\
\boldsymbol{w}_1 &= 0.4 + 0.4(-1 - (+1))0 &= 0.4\\
\boldsymbol{w}_2 &= -0.6 + 0.4(-1 - (+1))0 &= -0.6\\
\boldsymbol{w}_3 &= 0.6 + 0.4(-1 - (+1))1 &= -0.2
\end{aligned}$$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_0 & = & -0.5 \\ \boldsymbol{w}_1 & = & 0.4 \\ \boldsymbol{w}_2 & = & -0.6 \\ \boldsymbol{w}_3 & = & 0.6 \end{bmatrix}$$

Training the Network (Training Example 2)

Instance $x^{(2)} = \begin{bmatrix} 1 & 1 & 1 & 0 \end{bmatrix}^T$ $\quad f(x^{(2)}) = +1$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_0 & = & -1.3 \\ \boldsymbol{w}_1 & = & 0.4 \\ \boldsymbol{w}_2 & = & -0.6 \\ \boldsymbol{w}_3 & = & -0.2 \end{bmatrix}$$

Training the Network (Training Example 2)

Instance $x^{(2)} = [1 \quad 1 \quad 1 \quad 0]^T \qquad f(x^{(2)}) = +1$

1. Compute network output:

$$\boldsymbol{W}^T x^{(2)} = (1 \times -1.3) + (1 \times 0.4) + (1 \times -0.6) + (0 \times -0.2) = -1.5$$

$$\hat{f}(x^{(2)}) = threshold(\boldsymbol{W}^T x^{(2)}) = -1 \text{ since } -1.5 < 0$$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_0 & = & -1.3 \\ \boldsymbol{w}_1 & = & 0.4 \\ \boldsymbol{w}_2 & = & -0.6 \\ \boldsymbol{w}_3 & = & -0.2 \end{bmatrix}$$

Training the Network (Training Example 2)

Instance $x^{(2)} = \begin{bmatrix}1 & 1 & 1 & 0\end{bmatrix}^T \qquad f(x^{(2)}) = +1$

1. Compute network output:

$$\boldsymbol{W}^T x^{(2)} = (1 \times -1.3) + (1 \times 0.4) + (1 \times -0.6) + (0 \times -0.2) = -1.5$$

$$\hat{f}(x^{(2)}) = threshold(\boldsymbol{W}^T x^{(2)}) = -1 \text{ since } -1.5 < 0$$

2. Update weighs: since $\hat{f}(x^{(2)}) \neq f(x^{(2)})$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_0 & = & -1.3 \\ \boldsymbol{w}_1 & = & 0.4 \\ \boldsymbol{w}_2 & = & -0.6 \\ \boldsymbol{w}_3 & = & -0.2 \end{bmatrix}$$

Training the Network (Training Example 2)

Instance $x^{(2)} = \begin{bmatrix}1 & 1 & 1 & 0\end{bmatrix}^T$ $\quad f(x^{(2)}) = +1$

1. Compute network output:

$$\boldsymbol{W}^T x^{(2)} = (1 \times -1.3) + (1 \times 0.4) + (1 \times -0.6) + (0 \times -0.2) = -1.5$$

$$\hat{f}(x^{(2)}) = threshold(\boldsymbol{W}^T x^{(2)}) = -1 \text{ since } -1.5 < 0$$

2. Update weighs: since $\hat{f}(x^{(2)}) \neq f(x^{(2)})$
$\Rightarrow \boldsymbol{w}_j \leftarrow \boldsymbol{w}_j - \alpha(y_i - y^{'}) * x_i$

$$\begin{aligned}
\boldsymbol{w}_0 &= -1.3 + 0.4(+1-(-1))1 &&= -0.5\\
\boldsymbol{w}_1 &= 0.4 + 0.4(+1-(-1))1 &&= 1.2\\
\boldsymbol{w}_2 &= -0.6 + 0.4(+1-(-1))1 &&= 0.2\\
\boldsymbol{w}_3 &= -0.2 + 0.4(+1-(-1))0 &&= -0.2
\end{aligned}$$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_0 & = & -1.3\\ \boldsymbol{w}_1 & = & 0.4\\ \boldsymbol{w}_2 & = & -0.6\\ \boldsymbol{w}_3 & = & -0.2 \end{bmatrix}$$

Training the Network (Training Example 1 - Round 2)

Instance $x^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}^T \quad f(x^{(1)}) = -1$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_0 & = & -0.5 \\ \boldsymbol{w}_1 & = & 1.2 \\ \boldsymbol{w}_2 & = & 0.2 \\ \boldsymbol{w}_3 & = & -0.2 \end{bmatrix}$$

Instance $x^{(1)} = \begin{bmatrix}1 & 0 & 0 & 1\end{bmatrix}^T \qquad f(x^{(1)}) = -1$

1. Compute network output:

$$\boldsymbol{W}^T x^{(1)} = (1 \times -0.5) + (0 \times 1.2) + (0 \times 0.2) + (1 \times -0.2) = -0.7$$

$$\hat{f}(x^{(1)}) = threshold(\boldsymbol{W}^T x^{(1)}) = -1 \text{ since } -0.7 < 0$$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_0 & = & -0.5 \\ \boldsymbol{w}_1 & = & 1.2 \\ \boldsymbol{w}_2 & = & 0.2 \\ \boldsymbol{w}_3 & = & -0.2 \end{bmatrix}$$

Instance $x^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}^T \quad f(x^{(1)}) = -1$

1. Compute network output:

$$\boldsymbol{W}^T x^{(1)} = (1 \times -0.5) + (0 \times 1.2) + (0 \times 0.2) + (1 \times -0.2) = -0.7$$

$$\hat{f}(x^{(1)}) = threshold(\boldsymbol{W}^T x^{(1)}) = -1 \text{ since } -0.7 < 0$$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_0 & = & -0.5 \\ \boldsymbol{w}_1 & = & 1.2 \\ \boldsymbol{w}_2 & = & 0.2 \\ \boldsymbol{w}_3 & = & -0.2 \end{bmatrix}$$

2. Update weights
   No update necessary since $\hat{f}(x^{(1)}) = f(x^{(1)})$

Training the Network (Training Example 2 - Round 2)

Instance $x^{(2)} = \begin{bmatrix}1 & 1 & 1 & 0\end{bmatrix}^T$ $\quad f(x^{(2)}) = +1$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_0 & = & -0.5 \\ \boldsymbol{w}_1 & = & 1.2 \\ \boldsymbol{w}_2 & = & 0.2 \\ \boldsymbol{w}_3 & = & -0.2 \end{bmatrix}$$

Training the Network (Training Example 2 - Round 2)

Instance $x^{(2)} = \begin{bmatrix} 1 & 1 & 1 & 0 \end{bmatrix}^T \quad f(x^{(2)}) = +1$

1. Compute network output:

$$\boldsymbol{W}^T x^{(2)} = (1 \times -0.5) + (1 \times 1.2) + (1 \times 0.2) + (0 \times -0.2) = 0.9$$

$$\hat{f}(x^{(2)}) = threshold(\boldsymbol{W}^T x^{(2)}) = +1 \text{ since } 0.9 \geq 0$$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_0 & = & -0.5 \\ \boldsymbol{w}_1 & = & 1.2 \\ \boldsymbol{w}_2 & = & 0.2 \\ \boldsymbol{w}_3 & = & -0.2 \end{bmatrix}$$

Training the Network (Training Example 2 - Round 2)

Instance $x^{(2)} = \begin{bmatrix}1 & 1 & 1 & 0\end{bmatrix}^T \qquad f(x^{(2)}) = +1$

1. Compute network output:

$$\boldsymbol{W}^T x^{(2)} = (1 \times -0.5) + (1 \times 1.2) + (1 \times 0.2) + (0 \times -0.2) = 0.9$$

$$\hat{f}(x^{(2)}) = threshold(\boldsymbol{W}^T x^{(2)}) = +1 \text{ since } 0.9 \geq 0$$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_0 & = & -0.5 \\ \boldsymbol{w}_1 & = & 1.2 \\ \boldsymbol{w}_2 & = & 0.2 \\ \boldsymbol{w}_3 & = & -0.2 \end{bmatrix}$$

2. Update weights

   No update necessary since $\hat{f}(x^{(2)}) = f(x^{(2)})$

Training the Network (Training Example 2 - Round 2)

Instance $x^{(2)} = \begin{bmatrix}1 & 1 & 1 & 0\end{bmatrix}^T \qquad f(x^{(2)}) = +1$

1. Compute network output:

$$\boldsymbol{W}^T x^{(2)} = (1 \times -0.5) + (1 \times 1.2) + (1 \times 0.2) + (0 \times -0.2) = 0.9$$

$$\hat{f}(x^{(2)}) = threshold(\boldsymbol{W}^T x^{(2)}) = +1 \text{ since } 0.9 \geq 0$$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_0 & = & -0.5 \\ \boldsymbol{w}_1 & = & 1.2 \\ \boldsymbol{w}_2 & = & 0.2 \\ \boldsymbol{w}_3 & = & -0.2 \end{bmatrix}$$

2. Update weights
No update necessary since $\hat{f}(x^{(2)}) = f(x^{(2)})$

We have converged

Testing the Network (Example 3)

Instance $x^{(3)} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_0 & = & -0.5 \\ \boldsymbol{w}_1 & = & 1.2 \\ \boldsymbol{w}_2 & = & 0.2 \\ \boldsymbol{w}_3 & = & -0.2 \end{bmatrix}$$

Testing the Network (Example 3)

Instance $x^{(3)} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T$

1. Compute network output:

$$\boldsymbol{W}^T x^{(3)} = (1 \times -0.5) + (1 \times 1.2) + (1 \times 0.2) + (1 \times -0.2) = 0.7$$

$$\hat{f}(x^{(3)}) = threshold(\boldsymbol{W}^T x^{(3)}) = +1 \text{ since } 0.7 > 0$$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_0 & = & -0.5 \\ \boldsymbol{w}_1 & = & 1.2 \\ \boldsymbol{w}_2 & = & 0.2 \\ \boldsymbol{w}_3 & = & -0.2 \end{bmatrix}$$

Testing the Network (Example 4)

Instance $x^{(4)} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_0 & = & -0.5 \\ \boldsymbol{w}_1 & = & 1.2 \\ \boldsymbol{w}_2 & = & 0.2 \\ \boldsymbol{w}_3 & = & -0.2 \end{bmatrix}$$

Testing the Network (Example 4)

Instance $x^{(4)} = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^T$

1. Compute network output:

$$\boldsymbol{W}^T x^{(4)} = (1 \times -0.5) + (0 \times 1.2) + (0 \times 0.2) + (0 \times -0.2) = -0.5$$

$$\hat{f}(x^{(4)}) = threshold(\boldsymbol{W}^T x^{(4)}) = -1 \text{ since } -0.5 < 0$$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_0 & = & -0.5 \\ \boldsymbol{w}_1 & = & 1.2 \\ \boldsymbol{w}_2 & = & 0.2 \\ \boldsymbol{w}_3 & = & -0.2 \end{bmatrix}$$

Testing the Network (Example 5)

Instance $x^{(5)} = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}^T$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_0 & = & -0.5 \\ \boldsymbol{w}_1 & = & 1.2 \\ \boldsymbol{w}_2 & = & 0.2 \\ \boldsymbol{w}_3 & = & -0.2 \end{bmatrix}$$

Testing the Network (Example 5)

Instance $x^{(5)} = \begin{bmatrix}1 & 1 & 0 & 0\end{bmatrix}^T$

1. Compute network output:

$$\boldsymbol{W}^T x^{(5)} = (1 \times -0.5) + (1 \times 1.2) + (0 \times 0.2) + (0 \times -0.2) = 0.7$$

$$\hat{f}(x^{(5)}) = threshold(\boldsymbol{W}^T x^{(5)}) = +1 \text{ since } 0.7 > 0$$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_0 & = & -0.5 \\ \boldsymbol{w}_1 & = & 1.2 \\ \boldsymbol{w}_2 & = & 0.2 \\ \boldsymbol{w}_3 & = & -0.2 \end{bmatrix}$$

Testing the Network (Example 6)

Instance $x^{(6)} = \begin{bmatrix} 1 & 0 & 1 & 1 \end{bmatrix}^T$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_0 & = & -0.5 \\ \boldsymbol{w}_1 & = & 1.2 \\ \boldsymbol{w}_2 & = & 0.2 \\ \boldsymbol{w}_3 & = & -0.2 \end{bmatrix}$$

Instance $x^{(6)} = \begin{bmatrix}1 & 0 & 1 & 1\end{bmatrix}^T$

1. Compute network output:

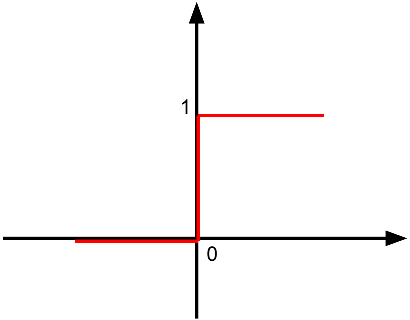
$$\boldsymbol{W}^T x^{(6)} = (1 \times -0.5) + (0 \times 1.2) + (1 \times 0.2) + (1 \times -0.2) = -0.5$$

$$\hat{f}(x^{(6)}) = threshold(\boldsymbol{W}^T x^{(6)}) = -1 \text{ since } -0.5 < 0$$

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_0 & = & -0.5 \\ \boldsymbol{w}_1 & = & 1.2 \\ \boldsymbol{w}_2 & = & 0.2 \\ \boldsymbol{w}_3 & = & -0.2 \end{bmatrix}$$

Perceptron Training

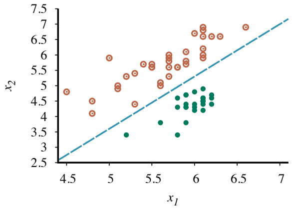
- Perceptron works for any linearly separable problem
- Its activation function is non-differentiable, returning 0 or 1

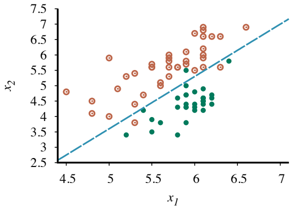


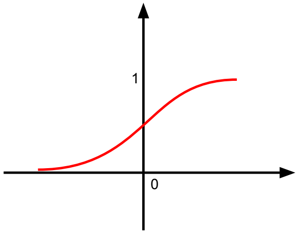
Sigmoid Function

- Although perceptron works fine for linearly separable problems, in non-linear problems it fails
- A solution is to add more layers and use a differentiable function, such as the sigmoid function

$$g(z) = \frac{1}{1+e^{-z}}$$





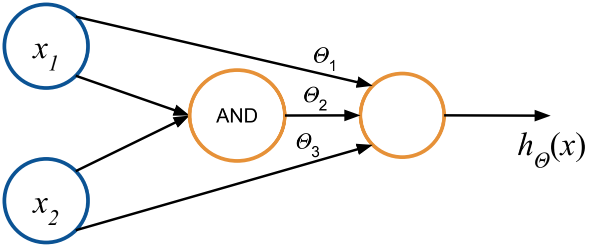


How powerful is a perceptron unit?

Quiz

Given:

- $w_1 = ?;$
- $w_2 = ?;$
- $w_3 = ?;$
- $\sigma = ?;$ and
- $x_1, x_2 \in \{0, 1\}$
- $h_{w}(x)$ = XOR

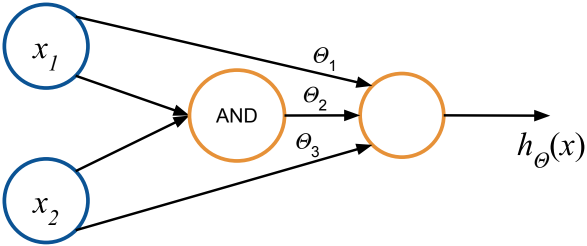


How powerful is a perceptron unit?

Quiz

Given:

- $w_1 = 1;$
- $w_2 = -3;$
- $w_3 = 1;$
- $\sigma = 0.5;$ and
- $x_1, x_2 \in \{0, 1\}$
- $h_w(x)$ = XOR



To continue in the next session.