Informatics 1 Cognitive Science

Lecture 5: The Perceptron

Frank Keller

25 January 2024

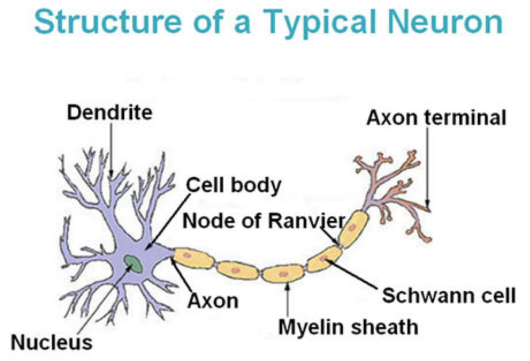
School of Informatics
University of Edinburgh
keller@inf.ed.ac.uk

Slide credits: Frank Mollica, Chris Lucas, Mirella Lapata

Overview

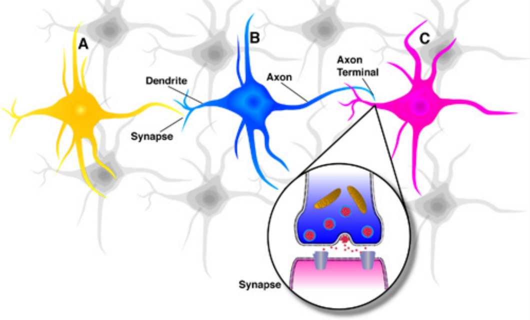
Neural Networks

A Single Neuron



- Neuron receives inputs and combines these in the cell body.
- If the input reaches a threshold, then the neuron may fire (produce an output).
- Some inputs are excitatory, while others are inhibitory.

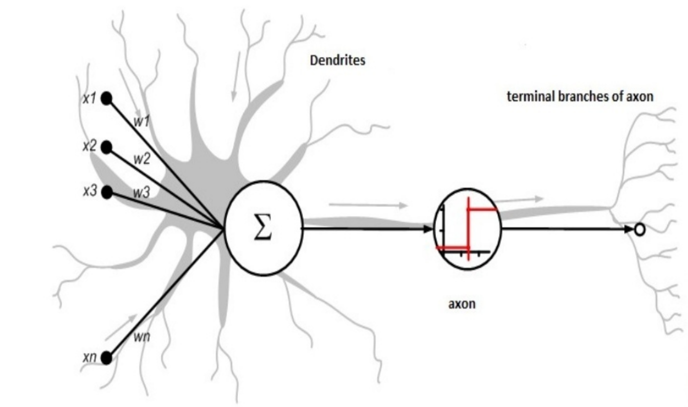
Biological Neural Networks



- In biological neural networks, connections are synapses.
- Input connection is a conduit through which a member of a network receives information (INPUT)
- Output connection is a conduit through which a member of a network sends information (OUTPUT).

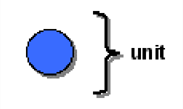
Neural Networks

Neural networks (aka deep learning) is the name for a computer modeling approach inspired by information processing in networks of biological neurons.



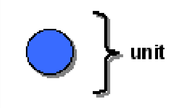
Anatomy of a Neural Network

Units are to a neural net model what neurons are to a biological neural network — the basic information processing structures.

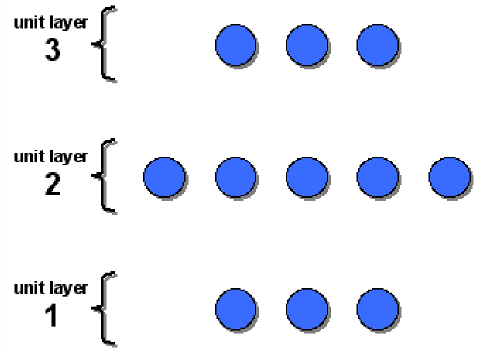


Anatomy of a Neural Network

Units are to a neural net model what neurons are to a biological neural network — the basic information processing structures.

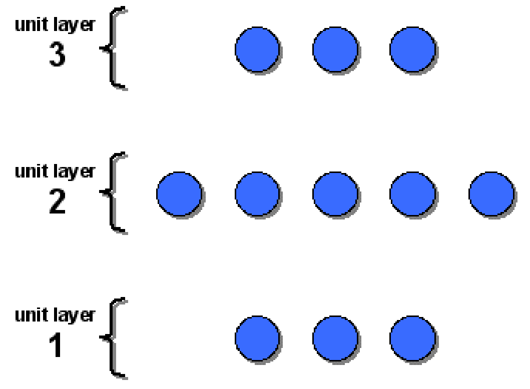


Biological neural networks are organized in layers of neurons. Neural net models are organized in layers of units, not random clusters.



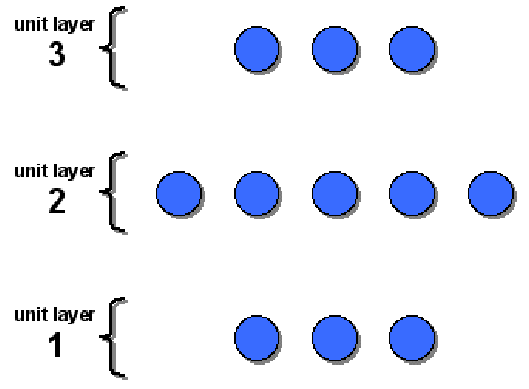
Anatomy of a Neural Network

But what you see here still isn't a network. Something is missing.

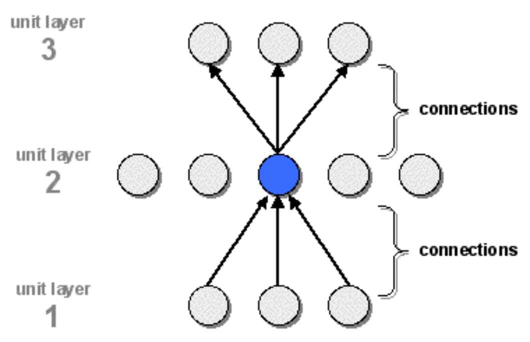


Anatomy of a Neural Network

But what you see here still isn't a network. Something is missing. Network connections are conduits through which information flows between members of a network.



Anatomy of a Neural Network

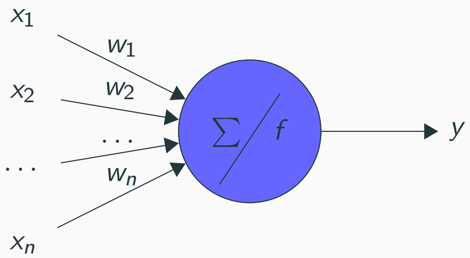


- Connections are represented with lines
- Arrows in a neural net indicate the flow of information from one unit to the next.

The Perceptron

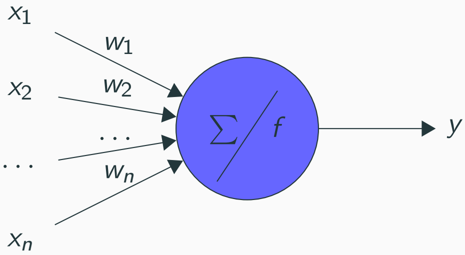
Perceptron: An Artificial Neuron

The perceptron was developed by Frank Rosenblatt in 1957. It's the simplest kind of artificial neural network.



Perceptron: An Artificial Neuron

The perceptron was developed by Frank Rosenblatt in 1957. It's the simplest kind of artificial neural network.

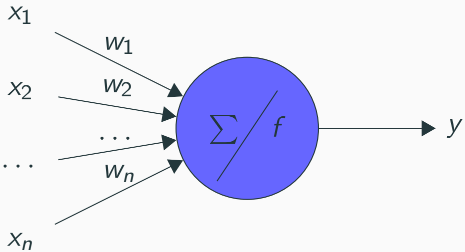


Input function:

$$u(\mathrm{x}) = \sum_{i=1}^{n} w_i x_i$$

Perceptron: An Artificial Neuron

The perceptron was developed by Frank Rosenblatt in 1957. It's the simplest kind of artificial neural network.



Activation function: threshold

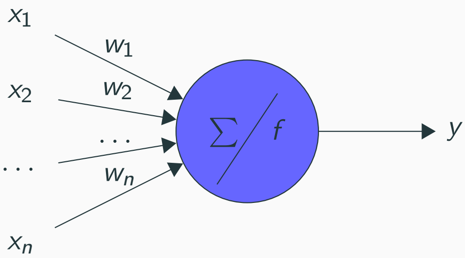
$$y = f(u(x)) = \begin{cases} 1, & \text{if } u(x) > \theta \\ 0, & \text{otherwise} \end{cases}$$

Input function:

$$u(x) = \sum_{i=1}^{n} w_i x_i$$

Perceptron: An Artificial Neuron

The perceptron was developed by Frank Rosenblatt in 1957. It's the simplest kind of artificial neural network.



Input function:

$$u(\mathrm{x}) = \sum_{i=1}^{n} w_i x_i$$

Activation function: threshold

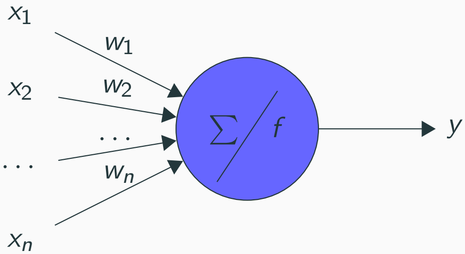
$$y = f(u(\mathrm{x})) = \begin{cases} 1, & \text{if } u(\mathrm{x}) > \theta \\ 0, & \text{otherwise} \end{cases}$$

Activation state:

0 or 1 (-1 or 1)

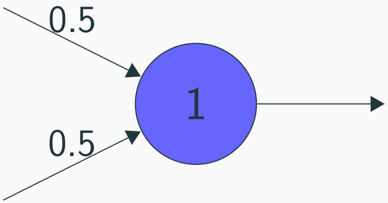
Perceptron: An Artificial Neuron

The perceptron was developed by Frank Rosenblatt in 1957. It's the simplest kind of artificial neural network.



- Inputs are in the range [0, 1], where 0 is "off" and 1 is "on".
- Weights can be any real number (positive or negative).

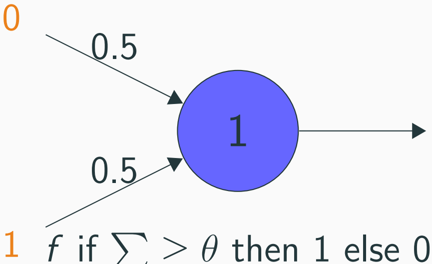
Perceptron for AND



f if $\sum \geq \theta$ then 1 else 0

x_1	x_2	x_1 AND x_2
0	0	0
0	1	0
1	0	0
1	1	1

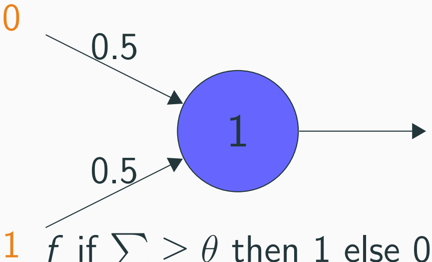
Perceptron for AND



f if $\sum \geq \theta$ then 1 else 0

x_1	x_2	x_1 AND x_2
0	0	0
0	1	0
1	0	0
1	1	1

Perceptron for AND

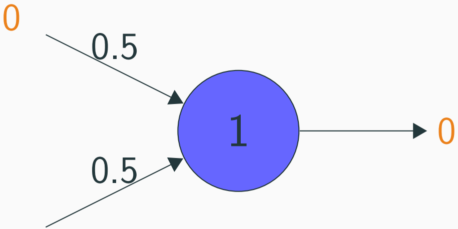


f if $\sum \geq \theta$ then 1 else 0

$$0 \cdot 0.5 + 1 \cdot 0.5 = 0.5$$

x_1	x_2	x_1 AND x_2
0	0	0
0	1	0
1	0	0
1	1	1

Perceptron for AND



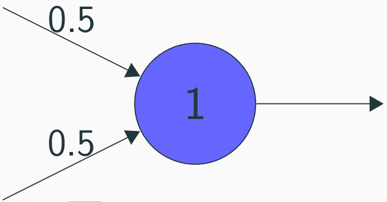
f if $\sum \geq \theta$ then 1 else 0

$$0 \cdot 0.5 + 1 \cdot 0.5 = 0.5$$

x_1	x_2	x_1 AND x_2
0	0	0
0	1	0
1	0	0
1	1	1

Perceptrons for Logic

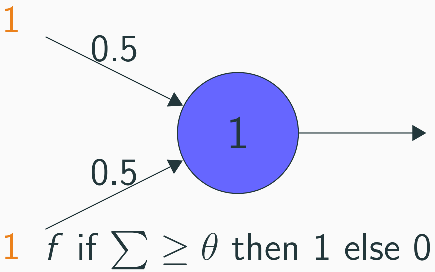
Perceptron for AND



f if $\sum \geq \theta$ then 1 else 0

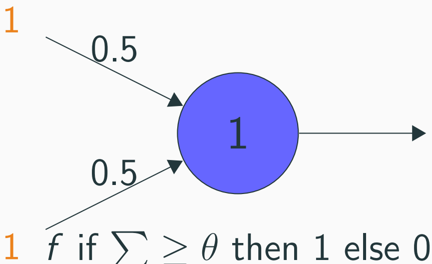
x_1	x_2	x_1 AND x_2
0	0	0
0	1	0
1	0	0
1	1	1

Perceptron for AND



x_1	x_2	x_1 AND x_2
0	0	0
0	1	0
1	0	0
1	1	1

Perceptron for AND



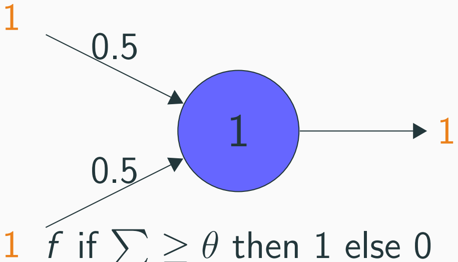
f if $\sum \geq \theta$ then 1 else 0

$$1 \cdot 0.5 + 1 \cdot 0.5 = 1$$

x_1	x_2	x_1 AND x_2
0	0	0
0	1	0
1	0	0
1	1	1

Perceptrons for Logic

Perceptron for AND



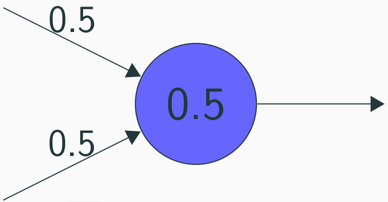
f if $\sum \geq \theta$ then 1 else 0

$$1 \cdot 0.5 + 1 \cdot 0.5 = 1$$

x_1	x_2	x_1 AND x_2
0	0	0
0	1	0
1	0	0
1	1	1

Perceptrons for Logic

Perceptron for OR

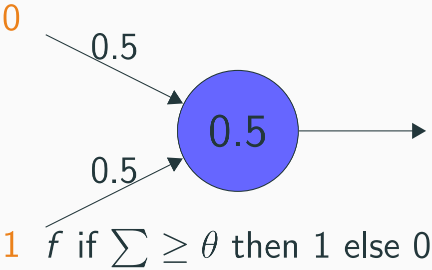


f if $\sum \geq \theta$ then 1 else 0

x_1	x_2	x_1 OR x_2
0	0	0
0	1	1
1	0	1
1	1	1

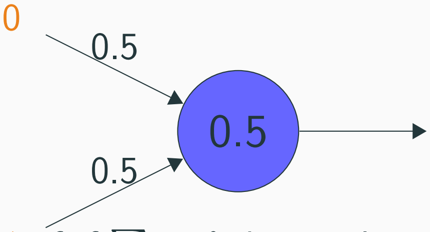
Perceptrons for Logic

Perceptron for OR



x_1	x_2	x_1 OR x_2
0	0	0
0	1	1
1	0	1
1	1	1

Perceptron for OR

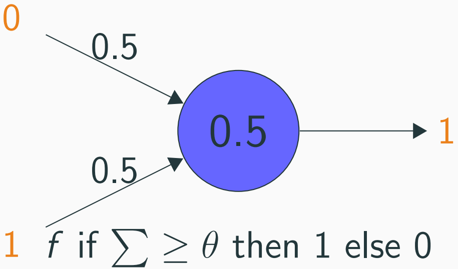


f if $\sum \geq \theta$ then 1 else 0

$$0 \cdot 0.5 + 1 \cdot 0.5 = 0.5$$

x_1	x_2	x_1 OR x_2
0	0	0
0	1	1
1	0	1
1	1	1

Perceptron for OR



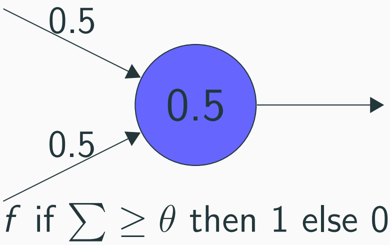
f if $\sum \geq \theta$ then 1 else 0

$$0 \cdot 0.5 + 1 \cdot 0.5 = 0.5$$

x_1	x_2	x_1 OR x_2
0	0	0
0	1	1
1	0	1
1	1	1

Perceptrons for Logic

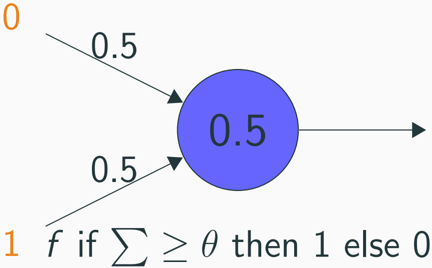
Perceptron for XOR



x_1	x_2	x_1 XOR x_2
0	0	0
0	1	1
1	0	1
1	1	0

XOR is an exclusive OR because it only returns a true value of 1 if the two values are exclusive, i.e., they are both different.

Perceptron for XOR

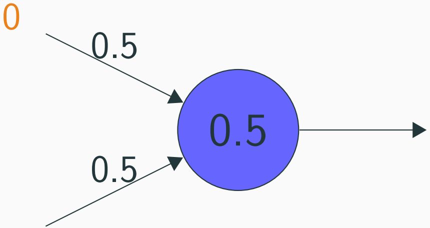


x_1	x_2	x_1 XOR x_2
0	0	0
0	1	1
1	0	1
1	1	0

XOR is an exclusive OR because it only returns a true value of 1 if the two values are exclusive, i.e., they are both different.

Perceptrons for Logic

Perceptron for XOR



f if $\sum \geq \theta$ then 1 else 0

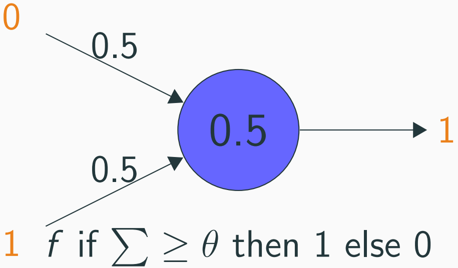
$$0 \cdot 0.5 + 1 \cdot 0.5 = 0.5$$

x_1	x_2	x_1 XOR x_2
0	0	0
0	1	1
1	0	1
1	1	0

XOR is an exclusive OR because it only returns a true value of 1 if the two values are exclusive, i.e., they are both different.

Perceptrons for Logic

Perceptron for XOR



f if $\sum \geq \theta$ then 1 else 0

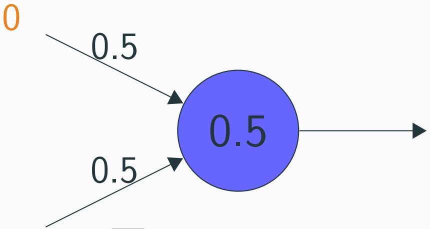
$$0 \cdot 0.5 + 1 \cdot 0.5 = 0.5$$

x_1	x_2	x_1 XOR x_2
0	0	0
0	1	1
1	0	1
1	1	0

XOR is an exclusive OR because it only returns a true value of 1 if the two values are exclusive, i.e., they are both different.

Perceptrons for Logic

Perceptron for XOR

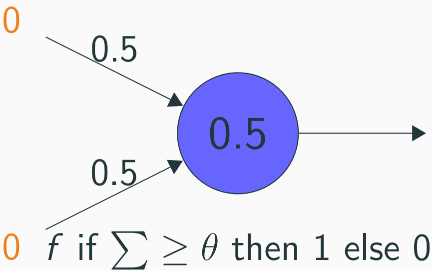


f if $\sum \geq \theta$ then 1 else 0

x_1	x_2	x_1 XOR x_2
0	0	0
0	1	1
1	0	1
1	1	0

Perceptrons for Logic

Perceptron for XOR



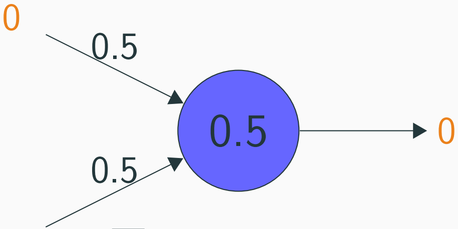
f if $\sum \geq \theta$ then 1 else 0

$$0 \cdot 0.5 + 0 \cdot 0.5 = 0$$

x_1	x_2	x_1 XOR x_2
0	0	0
0	1	1
1	0	1
1	1	0

Perceptrons for Logic

Perceptron for XOR

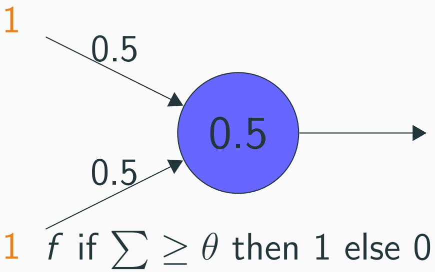


f if $\sum \geq \theta$ then 1 else 0

$$0 \cdot 0.5 + 0 \cdot 0.5 = 0$$

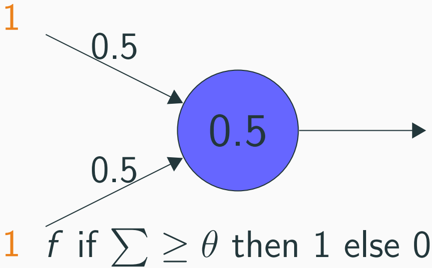
x_1	x_2	x_1 XOR x_2
0	0	0
0	1	1
1	0	1
1	1	0

Perceptron for XOR



x_1	x_2	x_1 XOR x_2
0	0	0
0	1	1
1	0	1
1	1	0

Perceptron for XOR

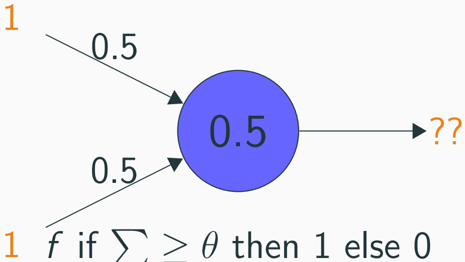


f if $\sum \geq \theta$ then 1 else 0

$$1 \cdot 0.5 + 1 \cdot 0.5 = 1$$

x_1	x_2	x_1 XOR x_2
0	0	0
0	1	1
1	0	1
1	1	0

Perceptron for XOR



f if $\sum \geq \theta$ then 1 else 0

$$1 \cdot 0.5 + 1 \cdot 0.5 = 1$$

x_1	x_2	x_1 XOR x_2
0	0	0
0	1	1
1	0	1
1	1	0

Perceptrons for Logic

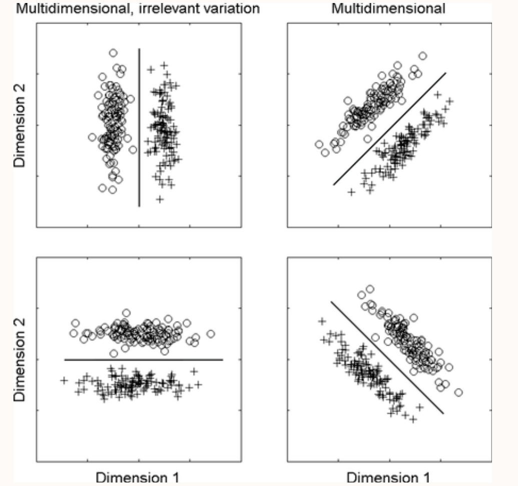
Time for a short quiz on Wooclap!

https://app.wooclap.com/GEKKBD

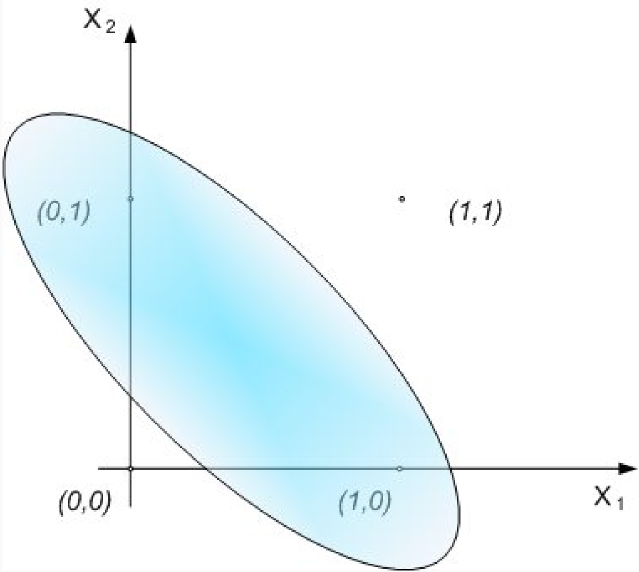
Perceptrons as Classifiers

Perceptrons as Classifiers

Perceptrons are linear classifiers, i.e., they can only separate points with a hyperplane (a straight line in two dimensions).



The XOR problem again



What is the Perceptron Really Seeing?

Sequence of exemplars presented to the Perceptron:

N	**input** x	**target** t
1	(0,1,0,0)	1
2	(1,0,0,0)	0
3	(0,1,1,1)	0
4	(1,0,1,0)	0
5	(1,1,1,1)	1
6	(0,1,0,0)	1

- This Perceptron has 4 inputs (binary) $\approx$ feature vector representing exemplars
- The Perceptron sees 6 exemplars or training items
- We know what the right answer is: target

What is the Perceptron Really Seeing?

Sequence of exemplars presented to the Perceptron:

N	**input** x	**target** t
1	(0,1,0,0)	1
2	(1,0,0,0)	0
3	(0,1,1,1)	0
4	(1,0,1,0)	0
5	(1,1,1,1)	1
6	(0,1,0,0)	1

- This Perceptron has 4 inputs (binary) $\approx$ feature vector representing exemplars
- The Perceptron sees 6 exemplars or training items
- We know what the right answer is: target
- But we don't know the weights/threshold!

What is the Perceptron Really Seeing?

Sequence of exemplars presented to the Perceptron:

N	**input** x	**target** t	**output** o
1	(0,1,0,0)	1	0
2	(1,0,0,0)	0	0
3	(0,1,1,1)	0	1
4	(1,0,1,0)	0	1
5	(1,1,1,1)	1	0
6	(0,1,0,0)	1	1

- This Perceptron has 4 inputs (binary) $\approx$ feature vector representing exemplars
- The Perceptron sees 6 exemplars or training items
- We know what the right answer is: target
- But we don't know the weights/threshold!
- What would happen if we used random weights/threshold?

Learning in Perceptrons

Learning

Q_1: But... choosing weights and threshold θ for the perceptron is not easy! How to learn the weights and threshold from examples?

A_1: We can use a learning algorithm that adjusts the weights and threshold based on examples.

http://www.youtube.com/watch?v=vGwemZhPlsA&feature=youtu.be

Learning: A trick to learn θ

$$\sum_{i=1}^{n} w_i x_i > \theta$$

Learning: A trick to learn θ

$$\sum_{i=1}^{n} w_i x_i > \theta$$

$$\sum_{i=1}^{n} w_i x_i - \theta > 0$$

Learning: A trick to learn θ

$$\sum_{i=1}^{n} w_i x_i > \theta$$

$$\sum_{i=1}^{n} w_i x_i - \theta > 0$$

$$w_1 x_1 + w_2 x_2 + \ldots w_n x_n - \theta > 0$$

Learning: A trick to learn θ

$$\sum_{i=1}^{n} w_i x_i > \theta$$

$$\sum_{i=1}^{n} w_i x_i - \theta > 0$$

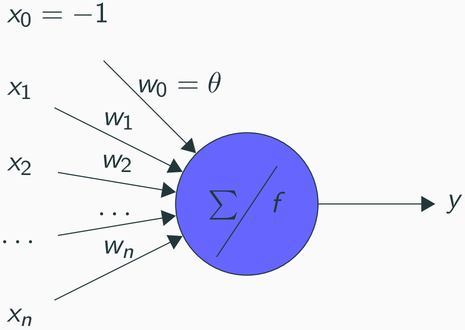
$$w_1 x_1 + w_2 x_2 + \ldots w_n x_n - \theta > 0$$

$$w_1 x_1 + w_2 x_2 + \ldots w_n x_n + \theta(-1) > 0$$

Learning: A trick to learn θ

$$\sum_{i=1}^{n} w_i x_i > \theta$$

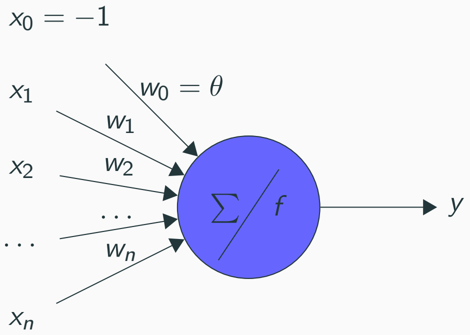
$$\sum_{i=1}^{n} w_i x_i - \theta > 0$$



$$w_1 x_1 + w_2 x_2 + \dots w_n x_n - \theta > 0$$

$$w_1 x_1 + w_2 x_2 + \dots w_n x_n + \theta(-1) > 0$$

Learning: A trick to learn θ



- We can consider θ as a weight to be learned!
- The input is fixed as -1. The activation function is then:

$$y = f(u(\mathrm{x})) = \begin{cases} 1, & \text{if } u(\mathrm{x}) > 0 \\ 0, & \text{otherwise} \end{cases}$$

The Perceptron Learning Rule

Learning Rule

Learning happens by adjusting weights. The threshold can be considered as a weight.

Perceptron Learning Rule

$$w_i \leftarrow w_i + \Delta w_i$$

$$\Delta w_i = \eta(t - o)x_i$$

- η, $0 < \eta \leq 1$ is a constant called learning rate.
- t is the target output of the current example.
- o is the output obtained by the Perceptron.

Perceptron Learning Rule

$$w_i \leftarrow w_i + \Delta w_i$$

$$\Delta w_i = \eta(t - o)x_i$$

$$o = 1 \text{ and } t = 1$$
$$o = 0 \text{ and } t = 1$$

- Learning rate η is positive; controls how big changes Δw_i are.
- If $x_i > 0$, $\Delta w_i > 0$ then w_i increases in an attempt to make $w_i x_i$ become larger than θ.
- If $x_i < 0$, $\Delta w_i < 0$ then w_i reduces.

Learning Rule

Perceptron Learning Rule

$$w_i \leftarrow w_i + \Delta w_i$$

$$\Delta w_i = \eta(t - o)x_i$$

$$\begin{aligned} o = 1 \text{ and } t = 1 \quad \Delta w_i &= \eta(t - o)x_i = \eta(1 - 1)x_i = 0 \\ o = 0 \text{ and } t = 1 \quad \Delta w_i &= \eta(t - o)x_i = \eta(1 - 0)x_i = \eta x_i \end{aligned}$$

- Learning rate η is positive; controls how big changes Δw_i are.
- If $x_i > 0$, $\Delta w_i > 0$ then w_i increases in an attempt to make $w_i x_i$ become larger than θ.
- If $x_i < 0$, $\Delta w_i < 0$ then w_i reduces.

Time for a short quiz on Wooclap!

https://app.wooclap.com/GEKKBD

Learning Rule: Exercise

Perceptron Learning Rule

$$w_i \leftarrow w_i + \Delta w_i$$

$$\Delta w_i \ = \eta(t - o)x_i$$

Consider a Perceptron with only one input x_1, weight $w_1 = 0.5$, threshold $\theta = 0$ and learning rate $\eta = 0.6$. Consider also the training example $\{x_1 = -1, t = 1\}$. For now, let's temporarily ignore the learning of the threshold and consider it fixed.

Learning Rule: Exercise

Perceptron Learning Rule

$$w_i \leftarrow w_i + \Delta w_i$$

$$\Delta w_i = \eta(t - o)x_i$$

Consider a Perceptron with only one input x_1, weight $w_1 = 0.5$, threshold $\theta = 0$ and learning rate $\eta = 0.6$. Consider also the training example $\{x_1 = -1, t = 1\}$. For now, let's temporarily ignore the learning of the threshold and consider it fixed.

- Determine the output of the Perceptron for the input -1:

Learning Rule: Exercise

Perceptron Learning Rule

$$w_i \leftarrow w_i + \Delta w_i$$

$$\Delta w_i \; = \eta(t - o)x_i$$

Consider a Perceptron with only one input x_1, weight $w_1 = 0.5$, threshold $\theta = 0$ and learning rate $\eta = 0.6$. Consider also the training example $\{x_1 = -1, t = 1\}$. For now, let's temporarily ignore the learning of the threshold and consider it fixed.

- Determine the output of the Perceptron for the input -1:
  $w_1 x_1 = 0.5(-1) = -0.5 \leq \theta \rightarrow o = 0$
- The new weight w_1 after applying the learning rule:

Learning Rule: Exercise

Perceptron Learning Rule

$$w_i \leftarrow w_i + \Delta w_i$$

$$\Delta w_i \ = \eta(t - o)x_i$$

Consider a Perceptron with only one input x_1, weight $w_1 = 0.5$, threshold $\theta = 0$ and learning rate $\eta = 0.6$. Consider also the training example $\{x_1 = -1, t = 1\}$. For now, let's temporarily ignore the learning of the threshold and consider it fixed.

- Determine the output of the Perceptron for the input -1:
  $w_1 x_1 = 0.5(-1) = -0.5 \leq \theta \rightarrow o = 0$
- The new weight w_1 after applying the learning rule:
  $\Delta w_1 = 0.6(1 - 0)(-1) = -0.6 \rightarrow w_1 = 0.5 - 0.6 = -0.1$

Learning Rule: Exercise

Perceptron Learning Rule

$$w_i \leftarrow w_i + \Delta w_i$$

$$\Delta w_i \; = \eta(t - o)x_i$$

Consider a Perceptron with only one input x_1, weight $w_1 = 0.5$, threshold $\theta = 0$ and learning rate $\eta = 0.6$. Consider also the training example $\{x_1 = -1, t = 1\}$. For now, let's temporarily ignore the learning of the threshold and consider it fixed.

- Determine the output of the Perceptron for the input -1:
  $w_1 x_1 = 0.5(-1) = -0.5 \leq \theta \rightarrow o = 0$
- The new weight w_1 after applying the learning rule:
  $\Delta w_1 = 0.6(1 - 0)(-1) = -0.6 \rightarrow w_1 = 0.5 - 0.6 = -0.1$
- The new output of the Perceptron for the input -1:

Learning Rule: Exercise

Perceptron Learning Rule

$$w_i \leftarrow w_i + \Delta w_i$$

$$\Delta w_i \ = \eta(t - o)x_i$$

Consider a Perceptron with only one input x_1, weight $w_1 = 0.5$, threshold $\theta = 0$ and learning rate $\eta = 0.6$. Consider also the training example $\{x_1 = -1, t = 1\}$. For now, let's temporarily ignore the learning of the threshold and consider it fixed.

- Determine the output of the Perceptron for the input -1:
  $w_1 x_1 = 0.5(-1) = -0.5 \leq \theta \rightarrow o = 0$
- The new weight w_1 after applying the learning rule:
  $\Delta w_1 = 0.6(1 - 0)(-1) = -0.6 \rightarrow w_1 = 0.5 - 0.6 = -0.1$
- The new output of the Perceptron for the input -1:
  $w_1 x_1 = -0.1(-1) = 0.1 \geq \theta \rightarrow o = 1$

Learning Algorithm

```
Initialize all weights randomly.
repeat
    for each training example do
        Apply the learning rule.
    end for
until the error is acceptable or a certain number
of iterations is reached
```

This algorithm is guaranteed to find a solution with zero error in a limited number of iterations as long as the examples are linearly separable.

Summary

- Neural networks (aka deep learning) is a computer modeling approach inspired by networks of biological neurons.
- A neural net consists of units and connections.
- The perceptron is the simplest neural network model; it is a linear classifier.
- A learning algorithm for perceptrons exists.
- **Key limitation:** only works for linearly separable data.