Learning Neural Networks - Part 1

Alice Gao

Lecture 8

Readings: RN 18.7, PM 7.5.

Outline

Learning Goals

By the end of the lecture, you should be able to

- Describe motivations for using a neural network model.
- Describe the simple mathematical model of a neuron.
- Describe desirable properties of an activation function. Give examples of activation functions and their properties.
- Distinguish feed-forward and recurrent neural networks.
- Learn a perceptron that represents a simple logical function.
- Determine the logical function represented by a perceptron.
- Explain why a perceptron cannot represent the XOR function.
- Construct a 3-layer neural network that represents the XOR function.

Learning Goals

Introduction to Artificial Neural Networks

Introduction to Perceptrons

Limitations of Perceptrons

Revisiting the Learning goals

History and Background

- Artificial Intelligence : building machines that can behave intelligently.
- Machine Learning let computers learn w/o being explicitly programmed. a branch of AI.
- Deep Learning a branch of ML.
  hierarchical network that mimics the human brain.

  - ImageNet AlexNet won the challenge. in 2012.
    (supervised learning)
  - The Cat Experiment Google Brain
    (unsupervised learning)
    recognize cats in YouTube videos.

Learning complex relationships

- Image interpretation, speech recognition, and translation.
- The relationship between inputs and outputs can be extremely complex.
- How can we build a model to learn such complex relationships?

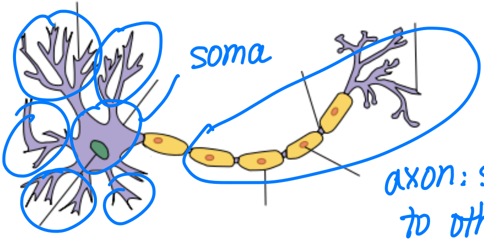
Humans can learn complex relationships well.
Can we build a model that mimics the human brain?

Human brains

- A brain is a set of densely connected neurons.
- Components of a neuron: dendrites, soma, axon, synapse
- Depending on the input signals, the neuron performs computations and decides to fire or not.

dendrites: receive inputs from other neurons

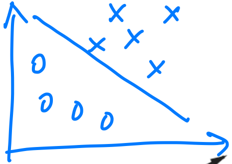
synapse: links between neurons.

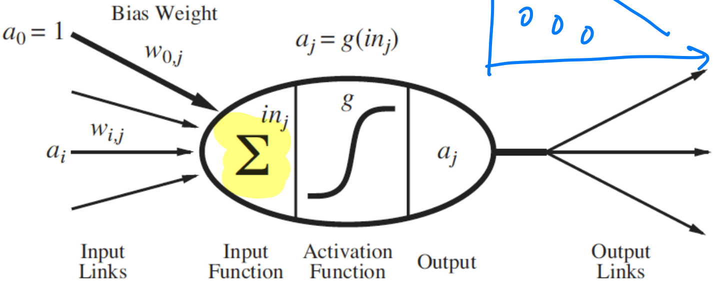


axon: sends outputs to other neurons.

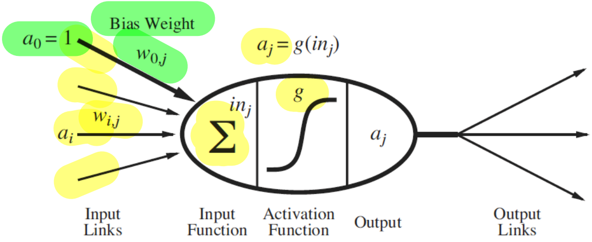
A simple mathematical model of a neuron

- McCulloch and Pitts 1943.
- A linear classfier — it “fires” when a linear combination of its inputs exceeds some threshold.





A simple mathematical model of a neuron



- Neuron j computes a weighted sum of its input signals. $in_j = \sum_{i=0}^{n} w_{ij} a_i$.
- Neuron j applies an activation function g to the weighted sum to derive the output. $a_j = g(in_j) = g\left(\sum_{i=0}^{n} w_{i,j} a_i\right)$.

Desirable Properties of The Activation Function

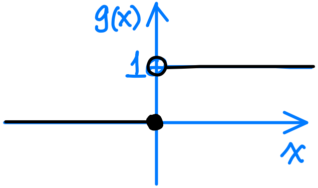
What are some desirable properties of the activation function?

- It should be non-linear.
  combining linear functions will not produce a non-linear function.
  complex relationships are often non-linear.
- It should mimic the behaviour of real neurons.
  If the weighted sum of input signals is large enough, the neuron fires. Otherwise, it does not fire.
- It should be differentiable almost everywhere.
  learn a neural network using gradient descent, which requires the activation function to be differentiable.

Common activation functions

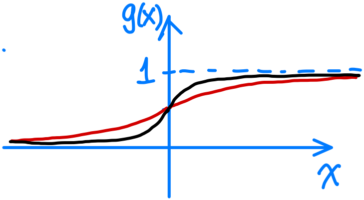
- Step function: $g(x) = 1$ if $x > 0$. $g(x) = 0$ if $x \leq 0$.

simple to use, not differentiable
not used in practice, but useful
to explain concepts.



- Sigmoid function: $g(x) = \dfrac{1}{1 + e^{-kx}}$.

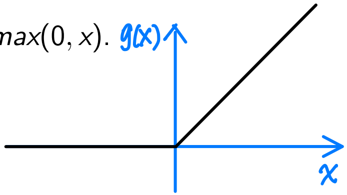
can approximate the step function.
clear, bounded prediction.
differentiable.
"vanishing gradient" problem.
computationally expensive.



Common activation functions (continued)

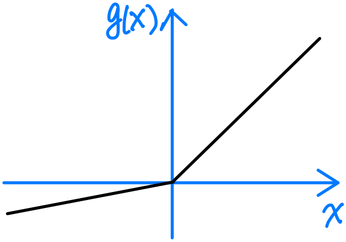
- Rectified linear unit (ReLu): $g(x) = max(0, x)$.

  computationally efficient.
  non-linear and differentiable.
  the dying ReLu problem.

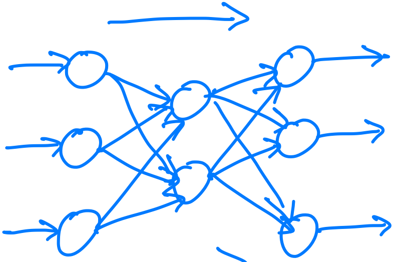


- Leaky ReLU: $g(x) = (0.1 * x, x)$.

  enables learning for
  negative input values.

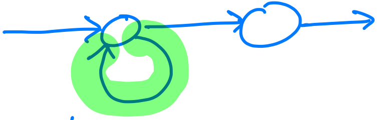


Connecting the neurons together into a network



- Feed-forward network

directed acyclic graph
no loops.
a function of its inputs.

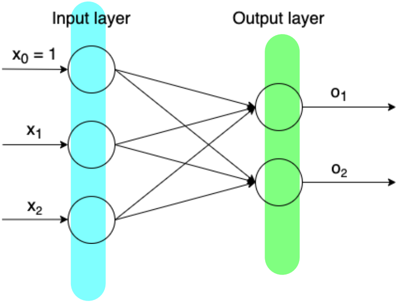


- Recurrent network

has loops.
has memory.
better model of human brain,
but more difficult to interpret and train.

Perceptrons

- Single-layer feed-forward neural network
- The inputs are connected directly to the outputs.
- Can represent logical functions, e.g. AND, OR, and NOT.



CQ: What does the perceptron compute?

CQ: Consider the following perceptron, where the activation function is the step function. ($g(x) = 1$ if $x > 0$. $g(x) = 0$ if $x \leq 0$.). Which of the following logical function does the perceptron compute?

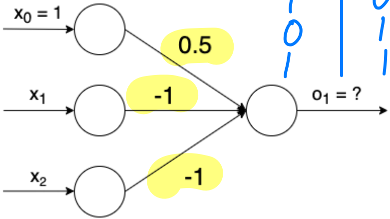
x_1	x_2	o_1
0	0	1
1	0	0
0	1	0
1	1	0

(A) $x_1 \land x_2$

(B) $\neg(x_1 \land x_2)$

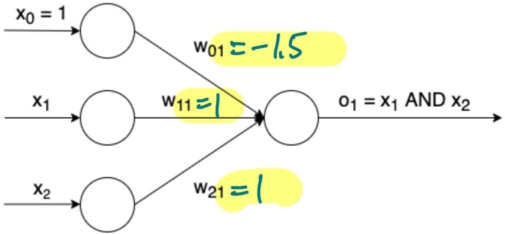
(C) $x_1 \lor x_2$

(D) $\neg(x_1 \lor x_2)$

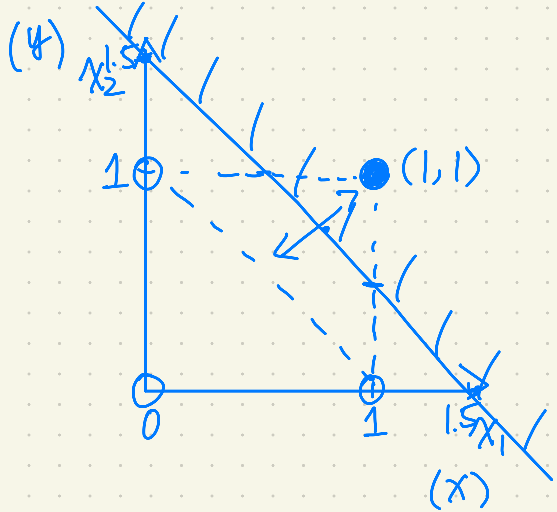


CQ: Learning a perceptron for the AND function

CQ: Consider the perceptron below where the activation function is the step function ($g(x) = 1$ if $x > 0$. $g(x) = 0$ if $x \leq 0$.). What should the weights w_{01}, w_{11} and w_{21} be such that the perceptron represents an AND function?



x_1	x_2	o_1
0	0	0
0	1	0
1	0	0
1	1	1



$y = -x + 1.5$

$x_2 = -x_1 + 1.5$

$x_1 + x_2 - 1.5 = 0.$

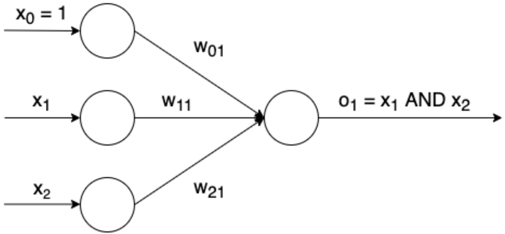
$\rightarrow 1 + 1 - 1.5 = 0.5$

$x_1 + x_2 - 1.5 > 0$

$W_{11} = 1 \quad W_{21} = 1 \quad W_{01} = -1.5$

CQ: Learning a perceptron for the AND function

CQ: Consider the perceptron below where the activation function is the step function ($g(x) = 1$ if $x > 0$. $g(x) = 0$ if $x \leq 0$.). How do we learn the weights w_{01}, w_{11} and w_{21} such that the perceptron represents an AND function?



x_1	x_2	o_1
0	0	0
0	1	0
1	0	0
1	1	1

x_0	x_1	x_2	W_{01}	W_{11}	W_{21}	O_{actual}	$O_{expected}$
1	1	1	0	0	0	0	1
1	1	0	0.2	0.2	0.2	1	0
1	0	1	0	0	0.2	1	0
1	1	1	-0.2	0	0	0	1
1	1	0	0	0.2	0.2	1	0
1	1	1	-0.2	0	0.2	0	1
1	0	1	0	0.2	0.4	1	0
			-0.2	0.2	0.2		

A perceptron representing OR

A question for you:

Consider a perceptron with three inputs x_0, x_1 and x_2 where the activation function is the step function ($g(x) = 1$ if $x > 0$. $g(x) = 0$ if $x \leq 0$.).

- What should the weights w_{01}, w_{11} and w_{21} be such that the perceptron represents an OR function?
- How do we learn these weights?

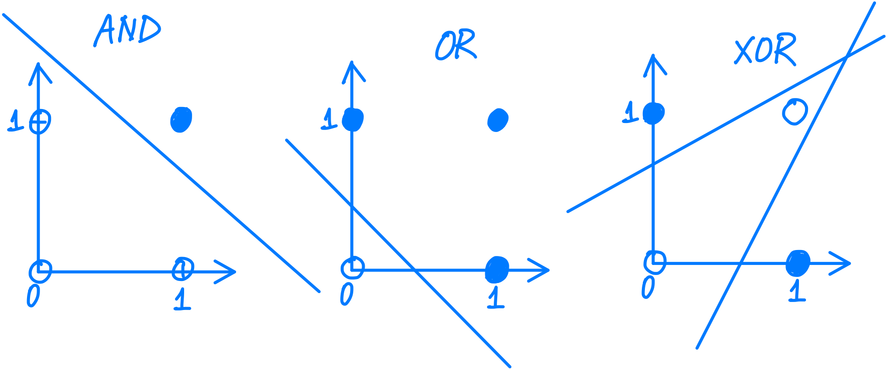
Limitations of perceptrons

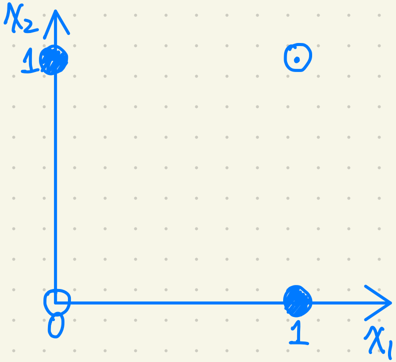
- Perceptrons: An introduction to computational geometry. Minsky and Papert. MIT Press. Cambridge MA 1969.
- Results:
  - XOR cannot be represented using perceptrons. We need a deeper network.
  - No one knew how to train deeper networks.
- Led to the first AI winter.

CQ: Why can't a perceptron represent XOR?

a perceptron is a linear classifier.

XOR is not linearly separable.



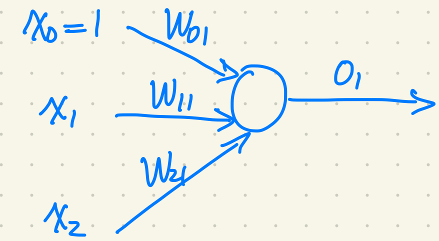


Proof:

Assume that we can represent the XOR function using a perceptron.

The activation function is the step function.

$$g(x) = \begin{cases} 1, & \text{if } x > 0, \\ 0 & \text{if } x \leq 0. \end{cases}$$



$$\left.\begin{aligned} W_{21}\cdot 1 + W_{11}\cdot 0 + W_{01} &> 0 \\ W_{21}\cdot 0 + W_{11}\cdot 1 + W_{01} &> 0 \\ W_{21}\cdot 1 + W_{11}\cdot 1 + W_{01} &\leq 0 \\ W_{21}\cdot 0 + W_{11}\cdot 0 + W_{01} &\leq 0 \end{aligned}\right. \Longrightarrow \begin{aligned} &W_{21} + W_{01} > 0 \Longrightarrow W_{21} > -W_{01} \quad ① \\ &W_{11} + W_{01} > 0 \Longrightarrow W_{11} > -W_{01} \quad ② \\ &W_{21} + W_{11} + W_{01} \leq 0 \quad ③ \\ &W_{01} \leq 0 \quad ⑥ \end{aligned}$$

$$① + ②$$

$$W_{21} + W_{11} > -2W_{01} \quad ④$$

$$③ \Longrightarrow W_{21} + W_{11} \leq -W_{01} \quad ⑤$$

$$W_{21} + W_{11} > -2W_{01} \geq -W_{01} \geq W_{21} + W_{11}$$

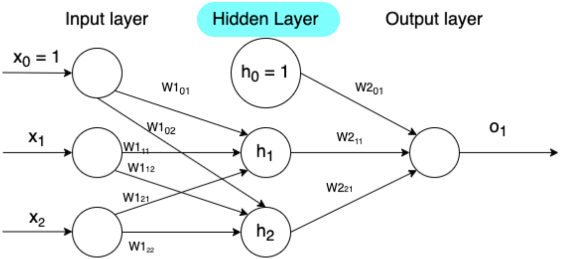
$$W_{21} + W_{11} > W_{21} + W_{11}$$

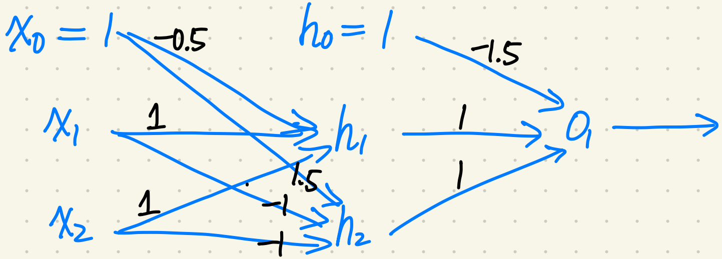
This is a contradiction. $\blacksquare$

XOR as a 3-Layer Neural Network

$$O_1 = ((x_1 \vee x_2) \wedge (\neg(x_1 \wedge x_2)))$$

Can you come up with the weights such that
the following network represents the XOR function?





$$o_1 = ((x_1 \lor x_2) \land (\neg (x_1 \land x_2)))$$
$$= (x_1 \text{ XOR } x_2)$$

$$h_1 = g(x_1 + x_2 - 0.5)$$

x_1	x_2	h_1
0	0	0
0	1	1
1	0	1
1	1	1

$$h_1 = (x_1 \lor x_2)$$

$$h_2 = g(-x_1 - x_2 + 1.5)$$

x_1	x_2	h_2
0	0	1
0	1	1
1	0	1
1	1	0

$$h_2 = \neg(x_1 \land x_2)$$

$$o_1 = g(h_1 + h_2 - 1.5)$$

h_1	h_2	o_1
0	0	0
0	1	0
1	0	0
1	1	1

$$o_1 = (h_1 \land h_2)$$

Revisiting the Learning Goals

By the end of the lecture, you should be able to

- Describe motivations for using a neural network model.
- Describe the simple mathematical model of a neuron.
- Describe desirable properties of an activation function. Give examples of activation functions and their properties.
- Distinguish feed-forward and recurrent neural networks.
- Learn a perceptron that represents a simple logical function.
- Determine the logical function represented by a perceptron.
- Explain why a perceptron cannot represent the XOR function.
- Construct a 3-layer neural network that represents the XOR function.