Introduction

11-785 Introduction to Deep Learning

Heeji Won

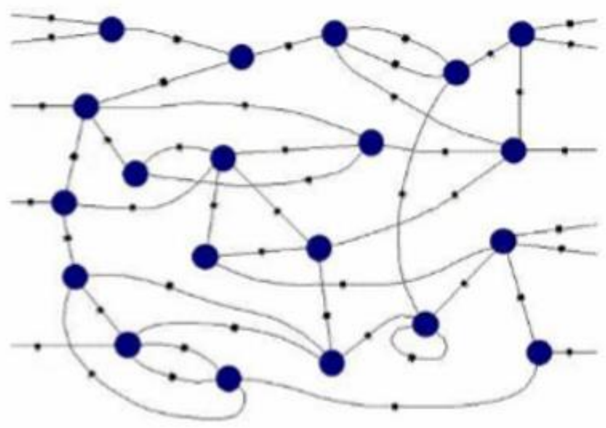
Contents

01. Connectionism

- The Earliest model of cognition was **associationism**
  -> But how to store them?

=> The answer was **Connectionism!**

- Neurons connect to neurons
- The workings of the brain are encoded in these connections

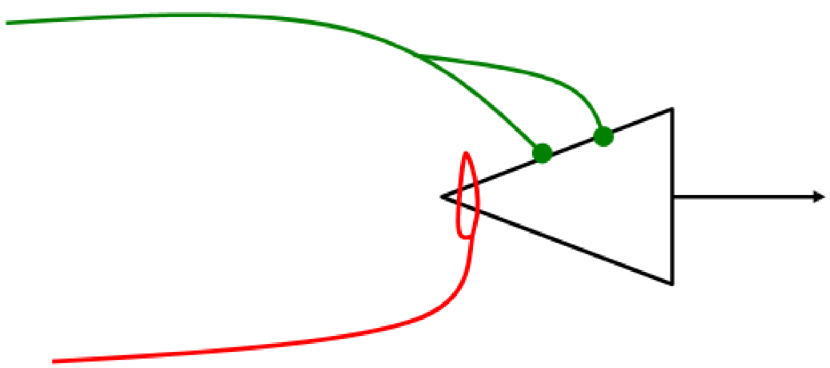


Connectionist Machine

✓ Neural network models are connectionist machines!

02. pre-Perceptron Models

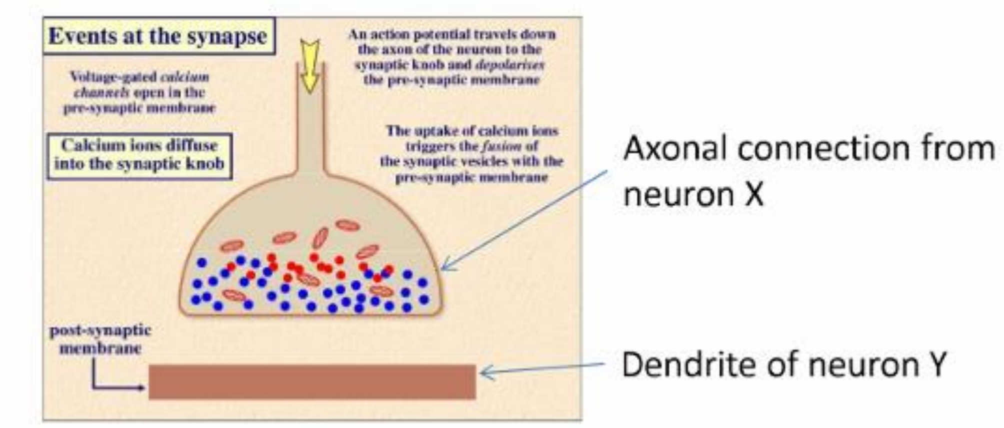
> The McCulloch and Pitts model



- Excitatory synapse : Transmits weighted input to the neuron
- Inhibitory synapse : Any signal from an inhibitory synapse prevents neuron from firing

– The activity of any inhibitory synapse absolutely prevents excitation of the neuron at that time.

> The McCulloch and Pitts model

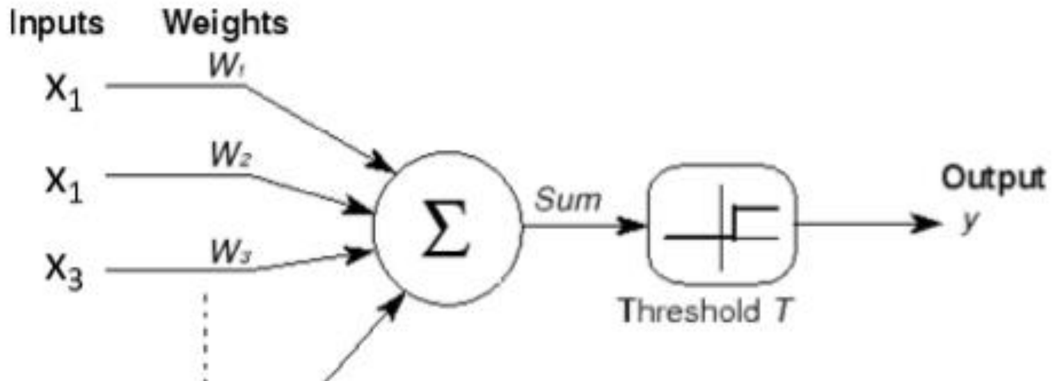


– If neuron repeatedly triggers neuron , the synaptic knob connecting(weight of the connection) to gets larger

$$w_{xy} = w_{xy} + \eta xy$$

✓ Unstable weights (only increase)

03. Perceptron



$$Y = \begin{cases} 1 \ \ if \ \sum_i w_i x_i - T \geq 0 \\ 0 \qquad \qquad else \end{cases}$$

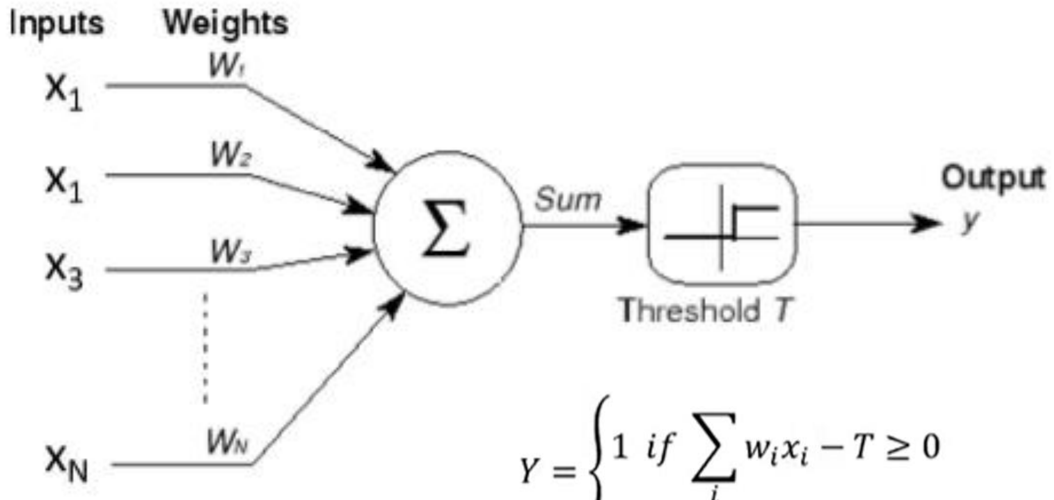
- Number of inputs combine linearly
  - Threshold logic: Fire if combined input exceeds threshold

- Also provided a learning algorithm

$$\mathbf{w} = \mathbf{w} + \eta(d(\mathbf{x}) - y(\mathbf{x}))\mathbf{x}$$

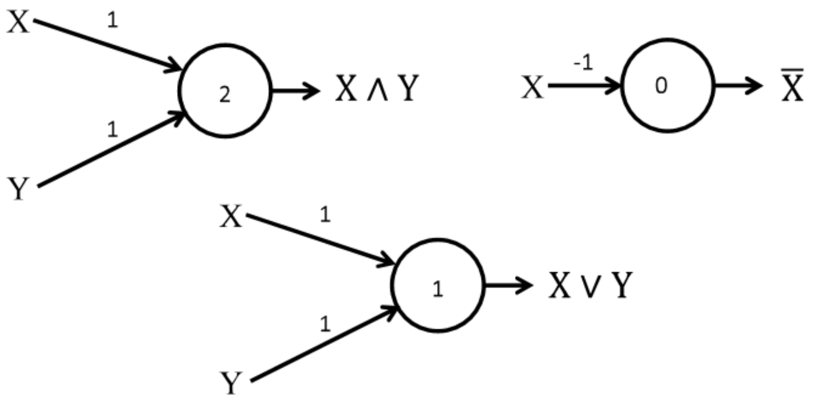
  - d(x) is the desired output in response to input x
  - y(x) is the actual output in response to x

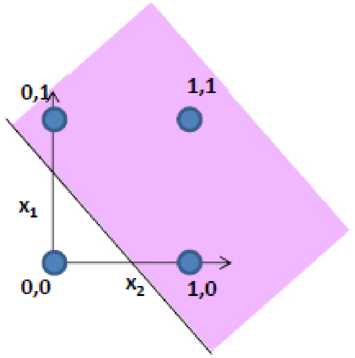
03. Perceptron



$$Y = \begin{cases} 1 \ \ if \ \sum_i w_i x_i - T \geq 0 \\ 0 \qquad\qquad else \end{cases}$$

- Easy to mimic any Boolean gate



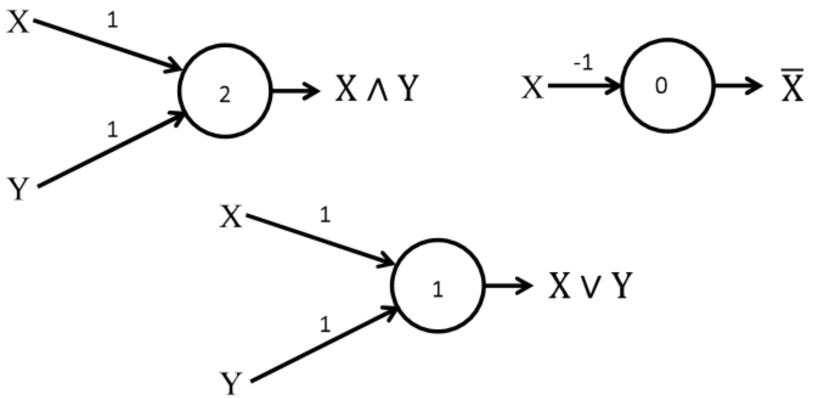


A perceptron can be
a linear classifier

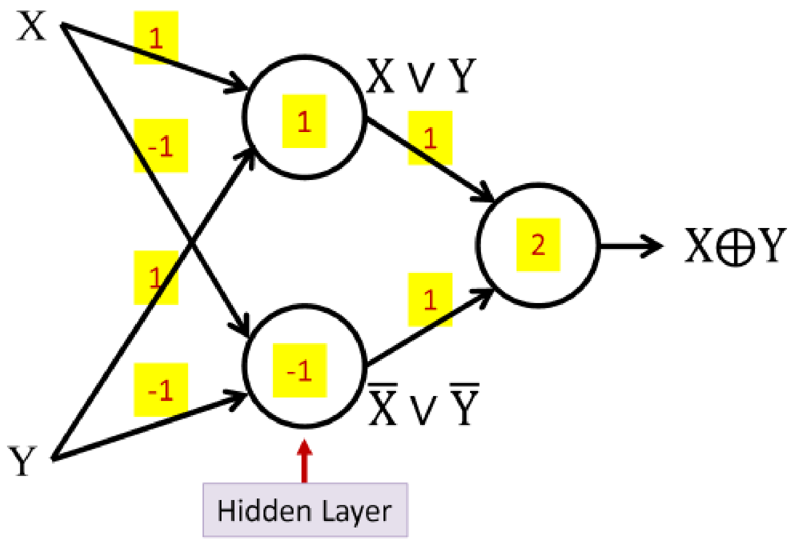
✓ But, no solution for XOR

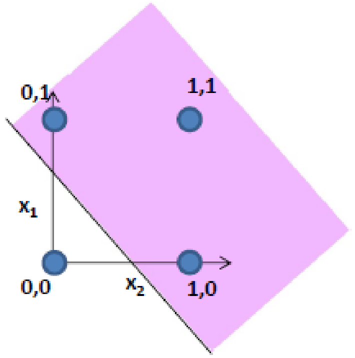
03. Perceptron

- Easy to mimic any Boolean gate



Multi-later Perceptron!





A perceptron can be
a linear classifier

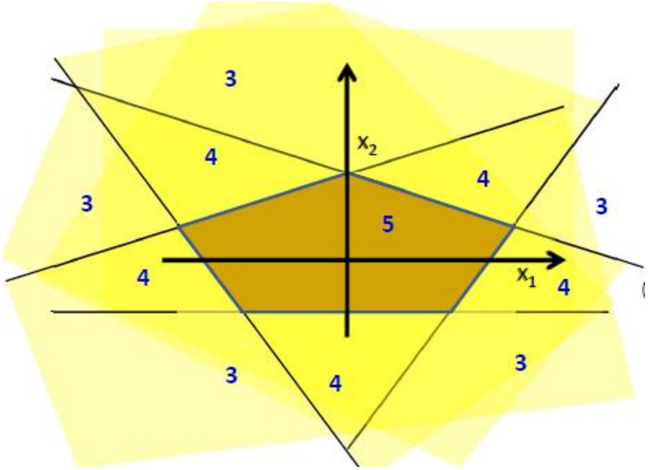
The first layer is a "**hidden**" layer

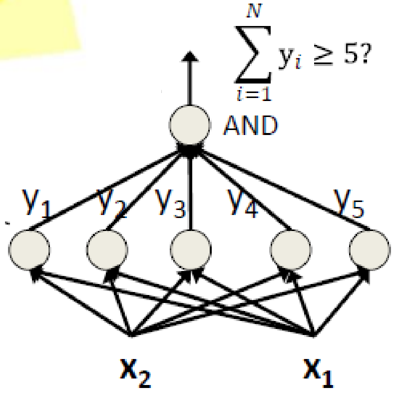
✓ But, no solution for XOR

03. Perceptron

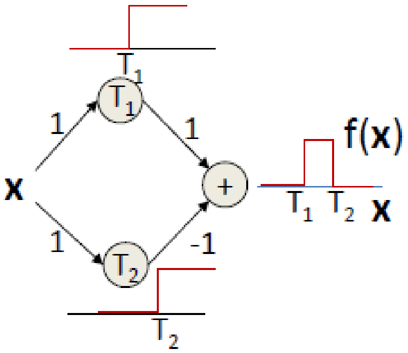
- A perceptron can …

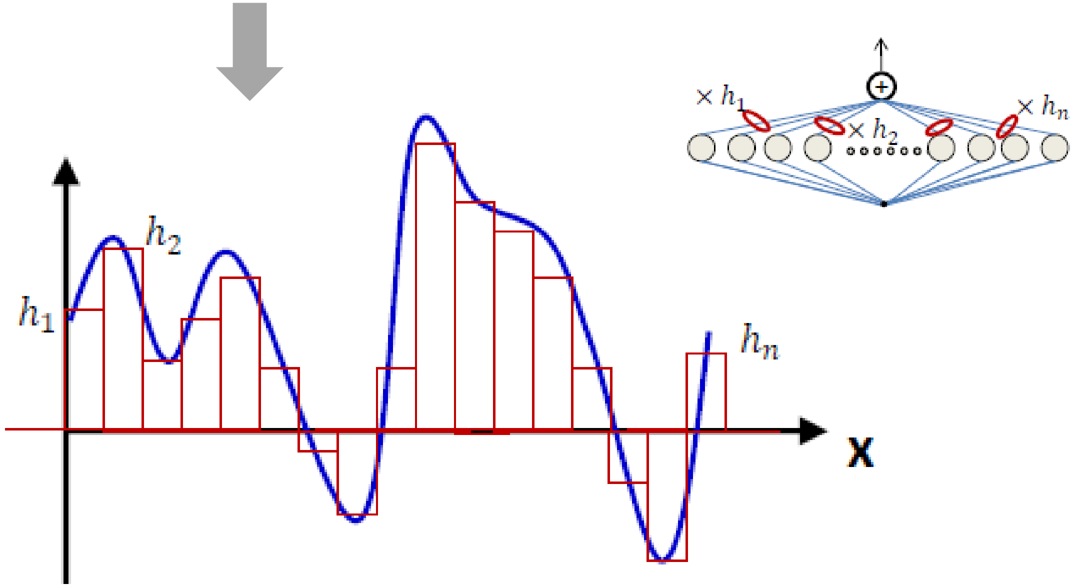
– composing complicated "decision" boundaries





– modeling an arbitrary function





Thank you