Deep Learning

Syed Irtaza Muzaffar

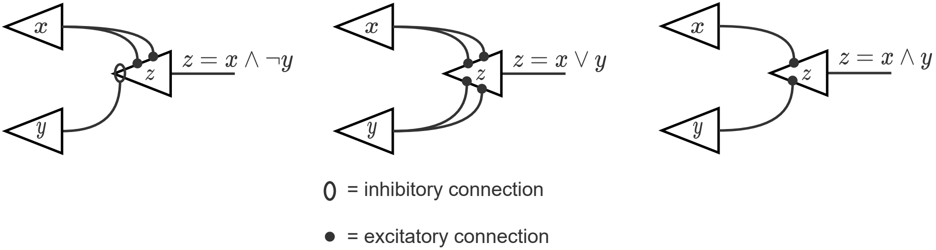
Introduction to Deep Learning

Mimicing the neuron

- Having established the neuron as the central element of animal systems, early research mimiced its function.
- Artificial neuron models were extremely simplified abstractions of the real neuron.
- To this day, they are extremely simplified.

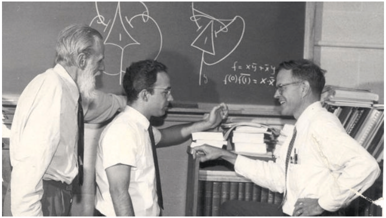
> *All models are wrong, but some are useful. -George Box*

McChulloch & Pitts Model



- For Boolean (0/1) inputs and outputs.
- Assumed neurons to either fire or stay silent.
- Fire if 2 excitatory connections are 1 and inhibitory connection is 0.
- This allowed them to model neurons using propositional logic.
- No mechanism for learning.

Interesting Note
Walter Pitts



- Pitts was an autodidact (self-taught).
- Rough, uneducated family. Ran away at 15. *Wandered* around universities.
- In 1943, at 20 years old, wrote a seminal paper[1].
- Burned years of unpublished research (probabilistic 3D neural networks) when his own paper went against it.
- Died at 46. No official degrees, no titles.

[1] Warren S McCulloch and Walter Pitts. 'A logical calculus of the ideas immanent in nervous activity'. In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133.

Hebbian Learning[2]

- Donald Hebb, Candian neuropsychologist.

> When an axon of cell A is near enough to excite cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased.

- If x triggers y, then increase the weight w_{xy} of their connection as

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

- Association determines strength of connection.
- Problems: Unbounded learning.

[2] DO Hebb. 'The organization of behavior; a neuropsychological theory.'. In: (1949).

Perceptron

In 1958, Frank Rosenblatt used Hebbian learning on McCulloch-Pitts neurons to make an artificial *eye that learned to see*.[3]

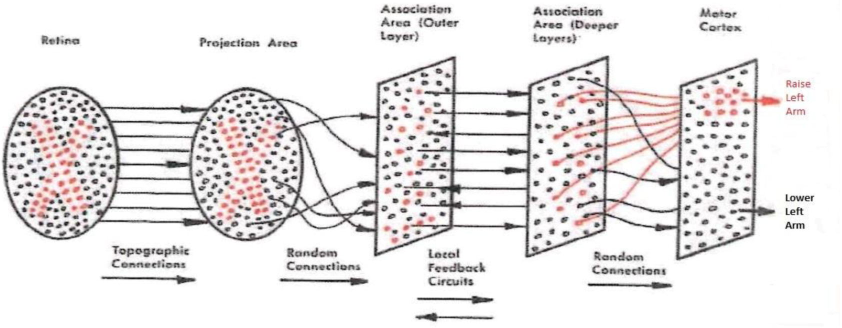


FIG. 1 — Organization of a biological brain. (Red areas indicate active cells, responding to the letter X.)

[3] Frank Rosenblatt. 'The design of an intelligent automaton'. In: *ONR Res. Rev.* (1958), pp. 5–13.

Perceptron

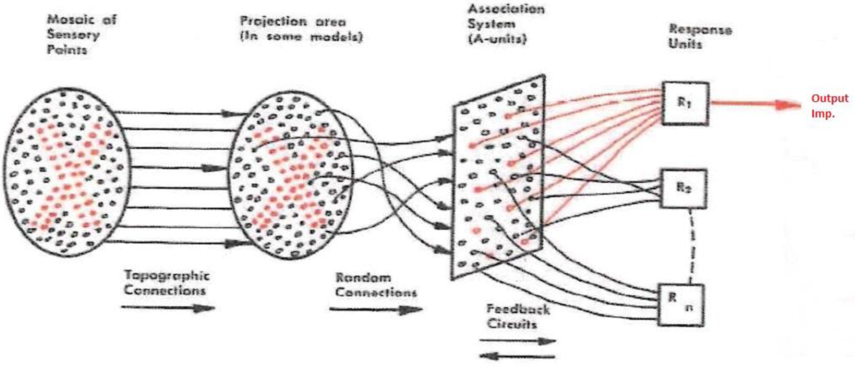


FIG. 2 — Organization of a perceptron.

Perceptron

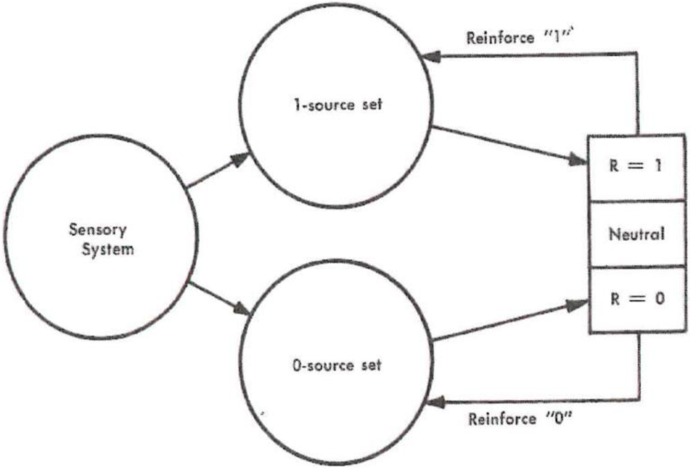
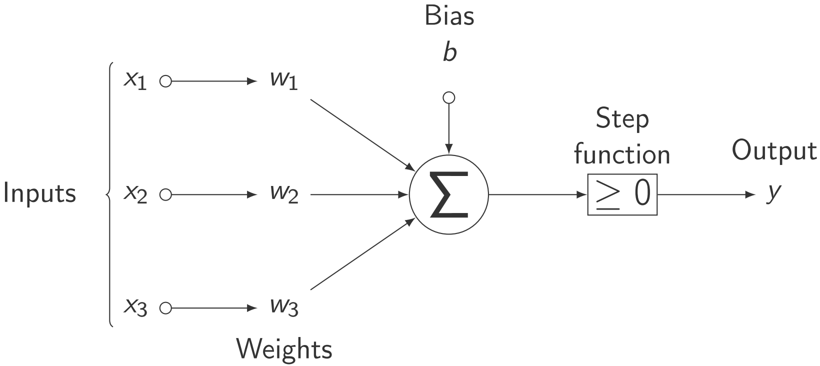


FIG. 3 — Detailed organization of a single perceptron.

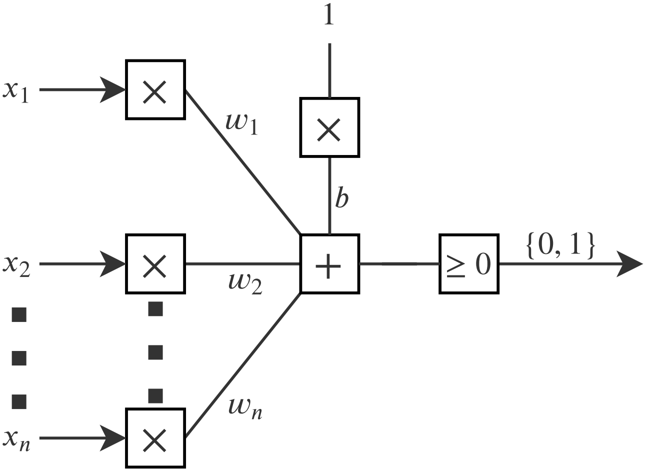
Perceptron

In subsequent years, only the last response unit came to be known as the perceptron[4].



[4] Frank Rosenblatt. 'The perceptron: a probabilistic model for information storage and organization in the brain.'. In: *Psychological review* 65.6 (1958), p. 386.

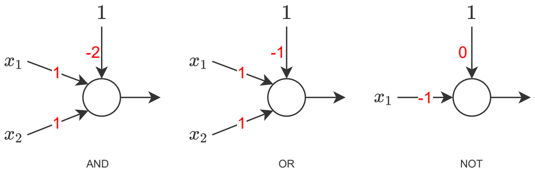
Perceptron



Perceptron

$$f(x_1, x_2, \ldots, x_n) = \begin{cases} 1 & \sum w_i x_i + b \geq 0 \\ 0 & \sum w_i x_i + b < 0 \end{cases}$$

- For real-valued inputs and weights.
- But also models Boolean logic.



  - AND gate: set all w_is to 1 and $b = -n$.
  - OR gate: set all w_is to 1 and $b = -1$.
  - NOT gate: for just a single input x_1, set w_1 to -1 and $b = 0$.
  - Since {AND, OR, NOT} form a basis for all logic gates, *combinations of perceptrons* can model *any* logic function.

Perceptron

- *Importantly*, the weights of a perceptron can be learned.
- Perceptron learning rule:

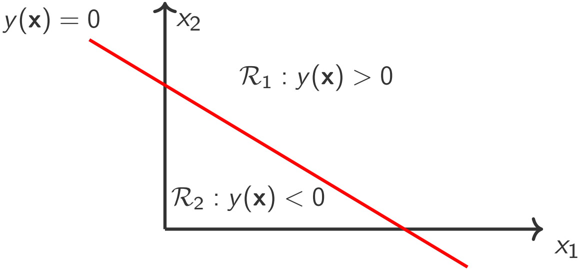
$$w_i \leftarrow w_i + \eta(y - t)x_i$$

  if output y and desired target t are different.

Can you identify a redundant statement on this slide?

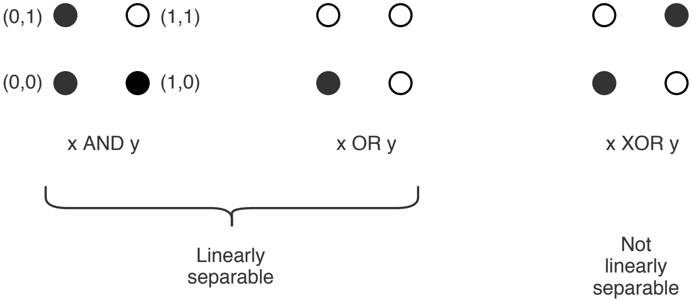
Detour – Linear Classifier

- Consider the function $y(\mathbf{x}) = \sum w_i x_i + w_0 = \mathbf{w}^T\mathbf{x} + w_0$.
- Thresholding $y(\mathbf{x})$ against 0 divides input space into two regions.
  - $\mathcal{R}_1$: where $y(\mathbf{x}) > 0$, and
  - $\mathcal{R}_2$: where $y(\mathbf{x}) < 0$.
- The line $y(\mathbf{x}) = 0$ is called the *linear decision boundary*.



Perceptron

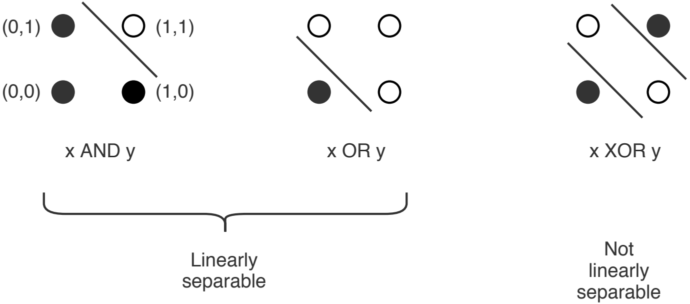
- A perceptron is actually a linear classifier.
- It's weights w_i and b represent a line that divides input space into 2 regions.



A perceptron cannot model the XOR problem because *XOR is not a linear classification problem*. No single line can separate the 0s (black) from the 1s (white).

Perceptron

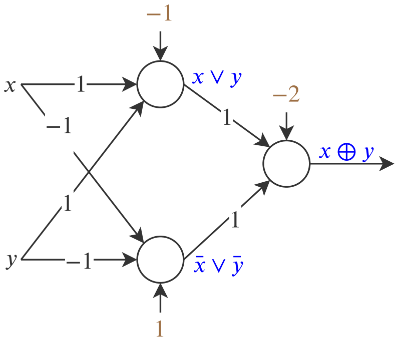
- A perceptron is actually a linear classifier.
- It's weights w_i and b represent a line that divides input space into 2 regions.



A perceptron cannot model the XOR problem because *XOR is not a linear classification problem*. No single line can separate the 0s (black) from the 1s (white). *But combination of two lines can*.

Multilayer Perceptrons

- Combination of perceptrons can solve the XOR problem.
- Two lines can divide the XOR space into 0-region and 1-region.



3 perceptrons can model XOR. Two perceptrons for the two lines and a final perceptron to combine them into a final decision.

A *network* of 3 neurons is more powerful than 1 neuron.
Just like the brain!