# Deep Learning

## 01 The Artificial Neuron (MP Neuron and Perceptron)

Dr. Konda Reddy Mopuri
Dept. of AI, IIT Hyderabad
Jan-May 2025

# Previously



- Course organization
- Brief history - connectionist philosophy

# The Neuron



- About $90$ billion neurons in human brain

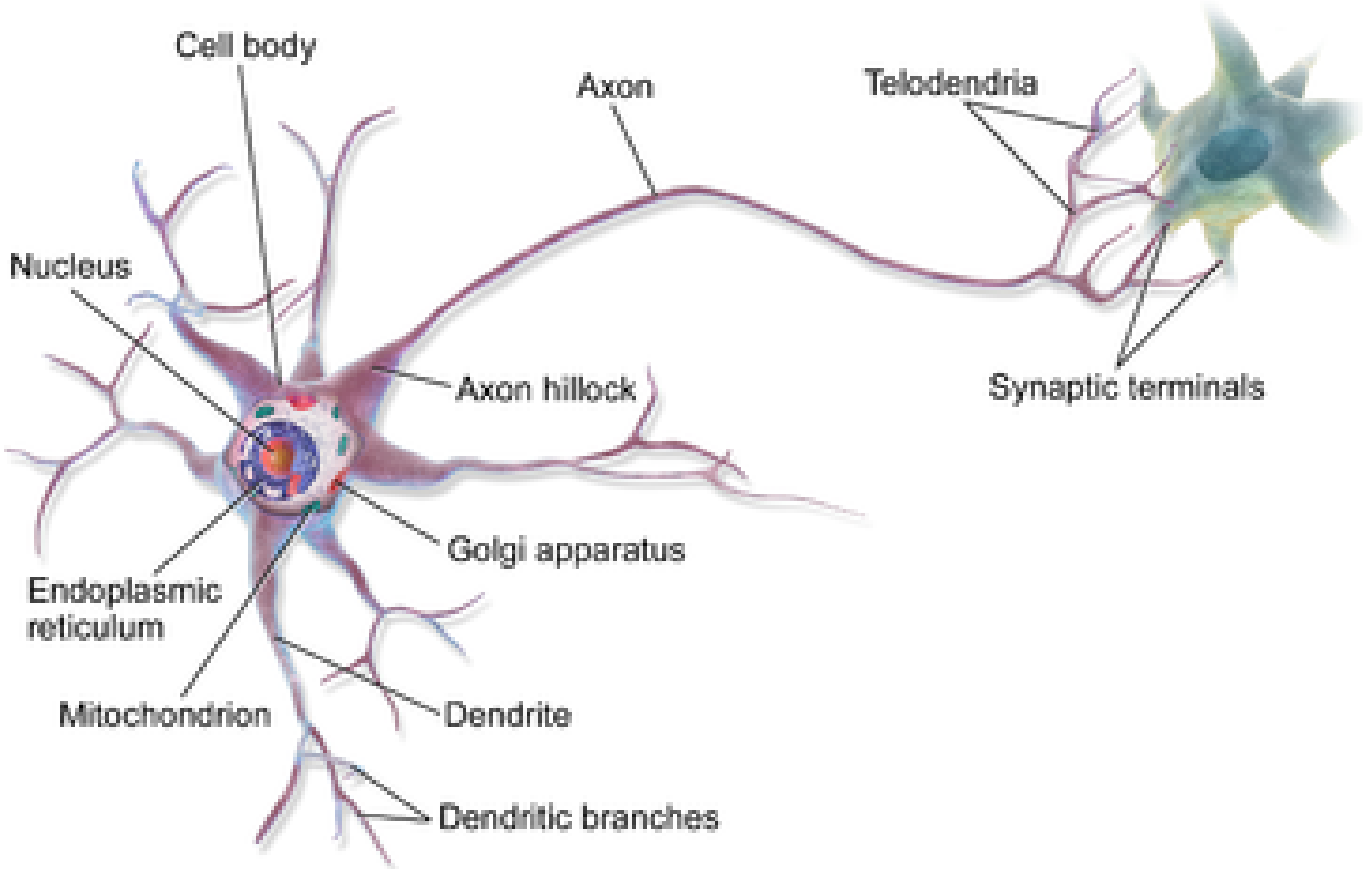
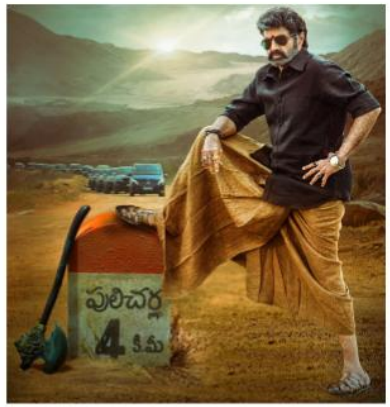


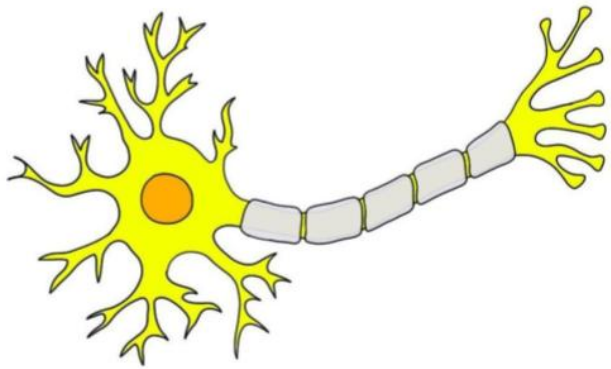

Figure credits: Wikipedia

# The dilemma: To watch or not to watch?



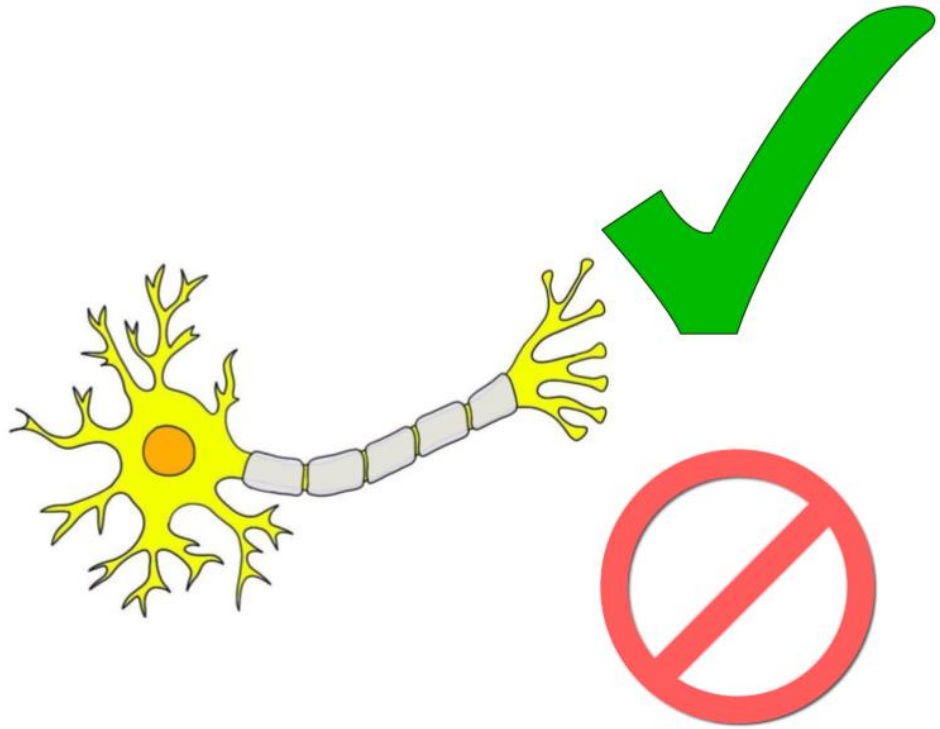

# Let's use our brain



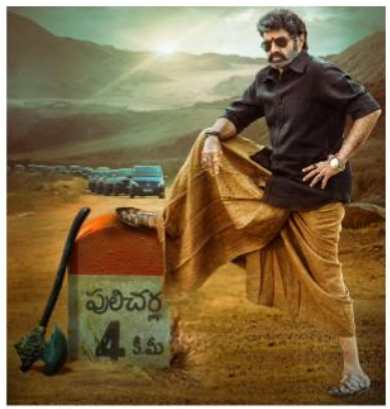
# Let's use our brain



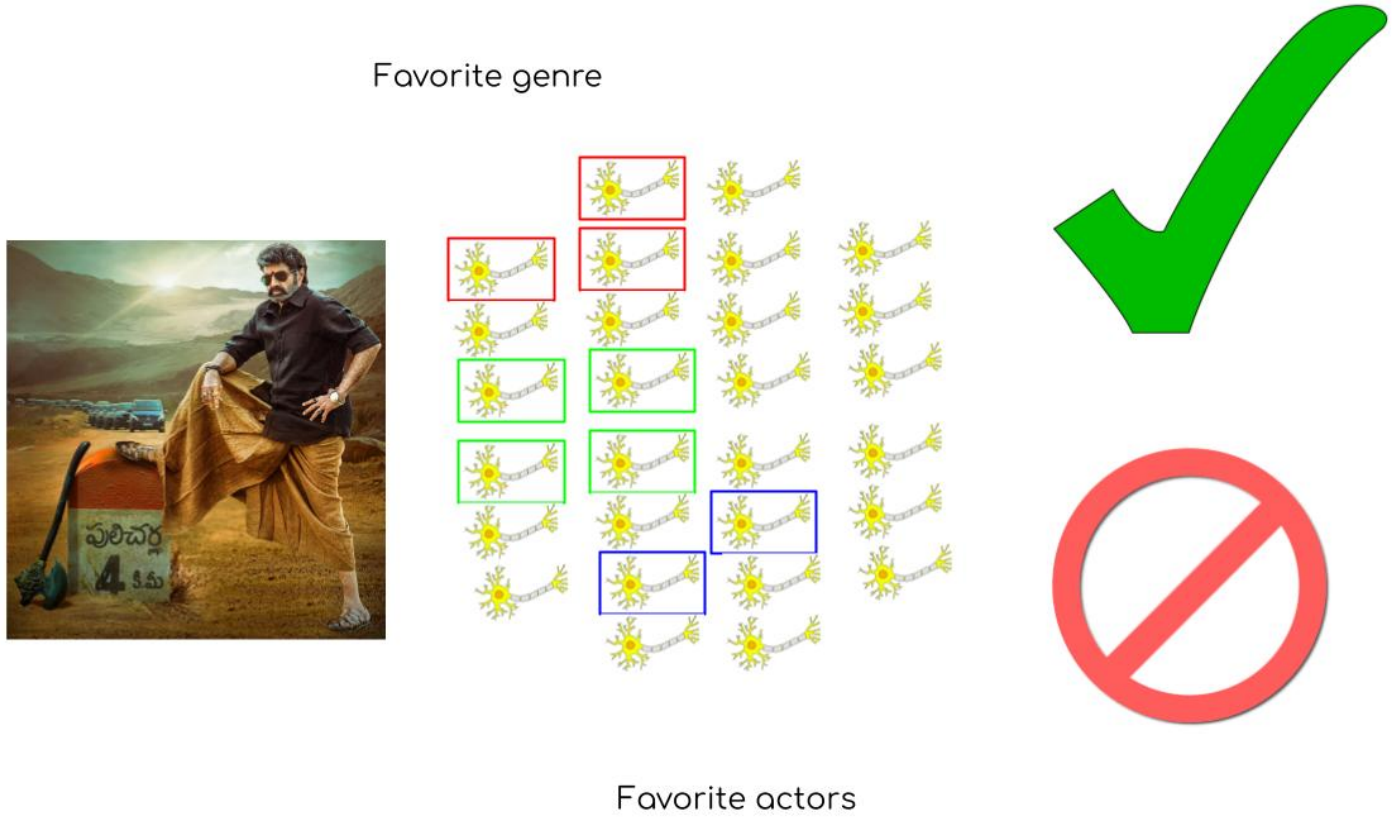
# It's a network of many neurons





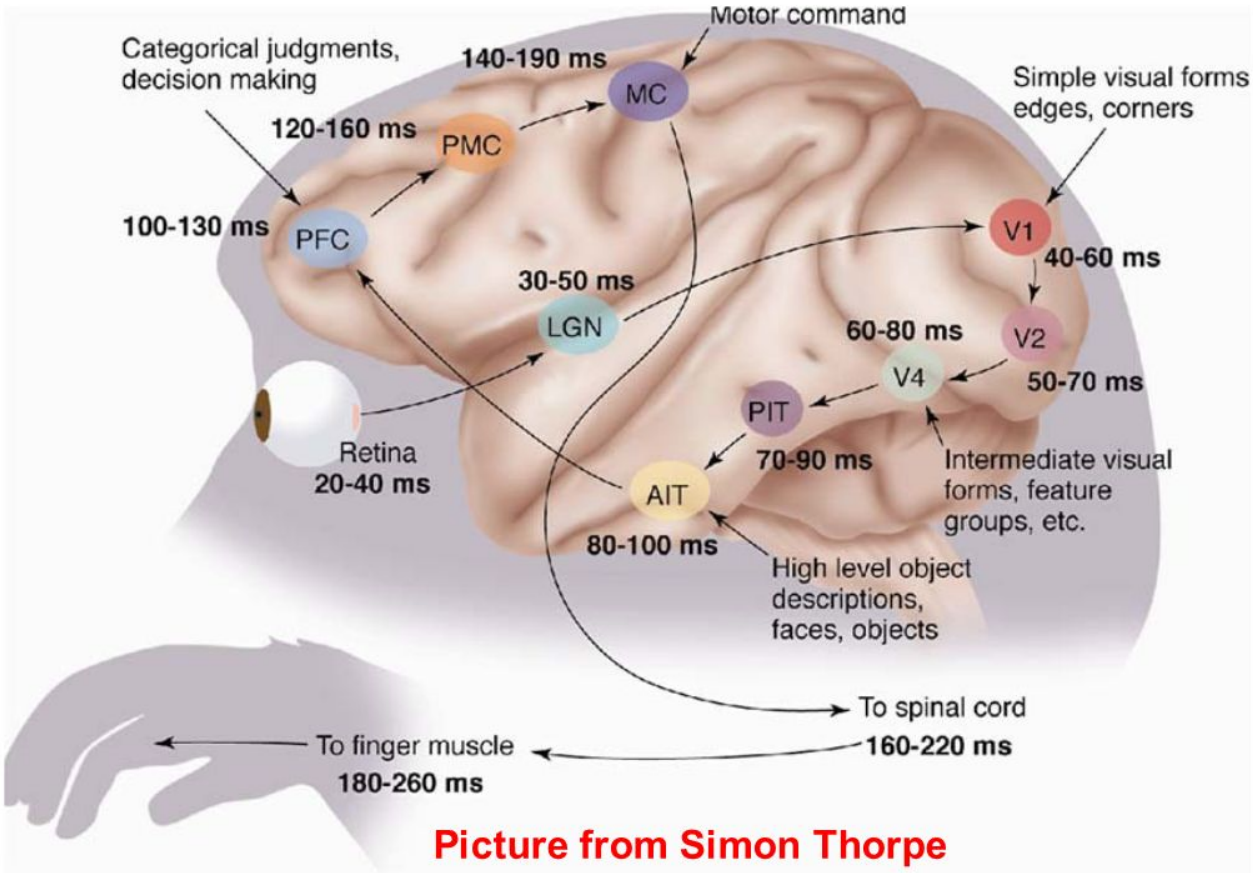

# There is a division of responsibilities





# Neurons in the brain have a hierarchy





# Threshold Logic Unit

1. First Mathematical Model for a neuron

# Threshold Logic Unit

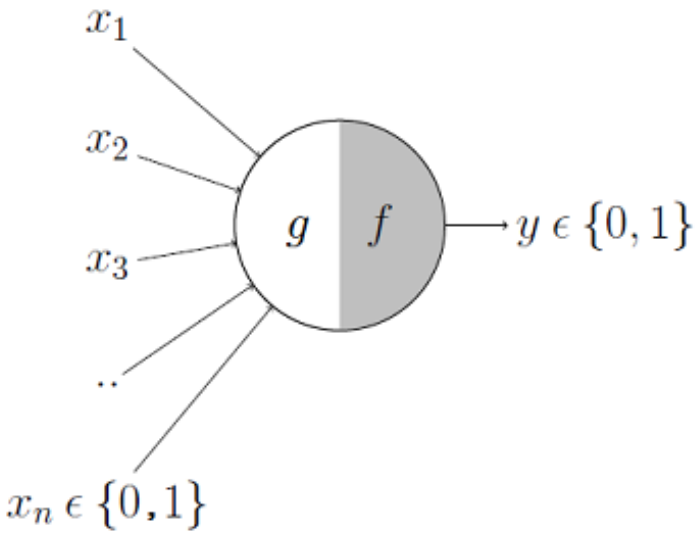


1. First Mathematical Model for a neuron
2. McCulloch and Pitts, $1943 \rightarrow$ MP neuron

# Threshold Logic Unit



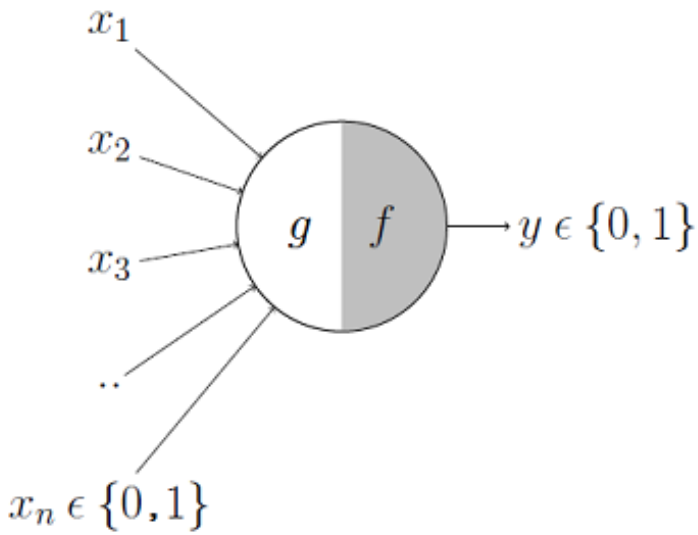
1. First Mathematical Model for a neuron
2. McCulloch and Pitts, $1943 \rightarrow$ MP neuron
3. Boolean inputs and output



# Threshold Logic Unit



1. First Mathematical Model for a neuron
2. McCulloch and Pitts, $1943 \rightarrow$ MP neuron
3. Boolean inputs and output




4. $$f(x) = \mathbb{1}(\sum_i x_i \geq \theta)$$

# Threshold Logic Unit



1. Inputs can be of excitatory or inhibitory nature

# Threshold Logic Unit



1. Inputs can be of excitatory or inhibitory nature
2. When an inhibitory input is set ($=1$) output $\rightarrow 0$

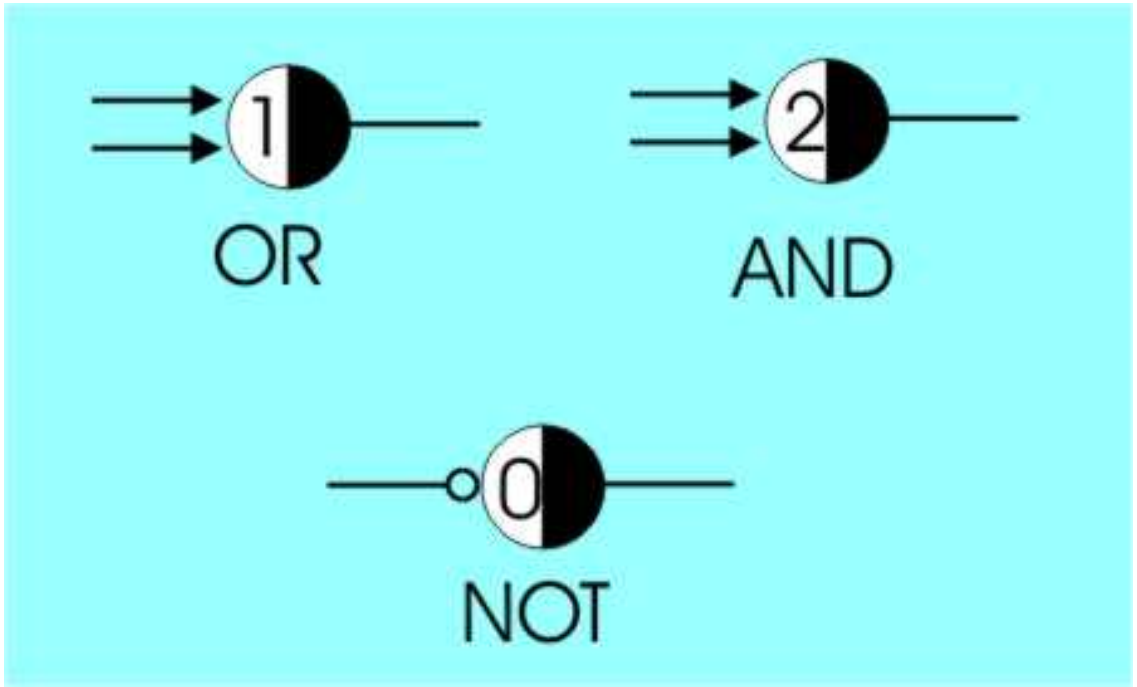
# Threshold Logic Unit



1. Inputs can be of excitatory or inhibitory nature
2. When an inhibitory input is set ($=1$) output $\rightarrow 0$
3. Counts the number of 'ON' signals on the excitatory inputs versus the inhibitory

# Threshold Logic Unit






Example Boolean functions

# Threshold Logic Unit

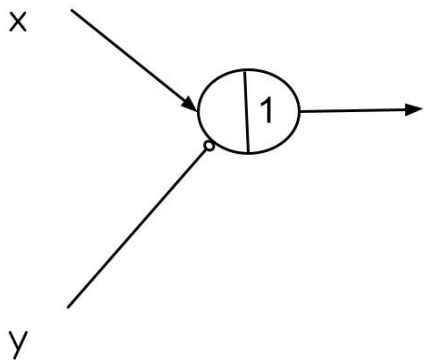


1. let's implement simple functions

# Threshold Logic Unit



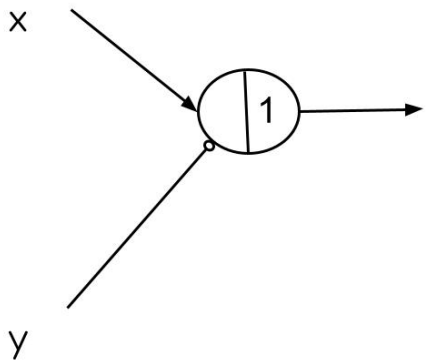
1. let's implement simple functions
2. xy'



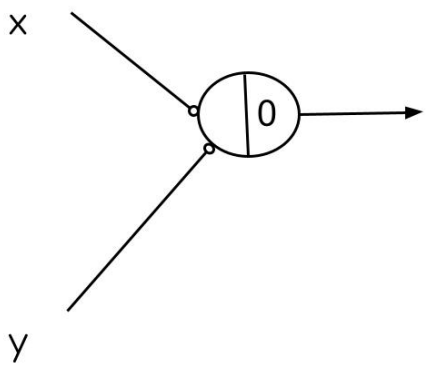
# Threshold Logic Unit



1. let's implement simple functions
2. xy'




3. NOR



# Threshold Logic Unit



1. What does one unit do? - Learn linear separation

# Threshold Logic Unit



1. What does one unit do? - Learn linear separation
   - line in 2D, plane in 3D, hyperplane in higher dimensions

# Threshold Logic Unit



1. What does one unit do? - Learn linear separation
   - line in 2D, plane in 3D, hyperplane in higher dimensions
2. No learning; heuristic approach

# Perceptron



1. Frank Rosenblatt 1957 (American Psychologist)

# Perceptron



1. Frank Rosenblatt 1957 (American Psychologist)
2. Very crude biological model

# Perceptron



1. Frank Rosenblatt 1957 (American Psychologist)
2. Very crude biological model
3. Similar to MP neuron - Performs linear classification

# Perceptron



1. Frank Rosenblatt 1957 (American Psychologist)
2. Very crude biological model
3. Similar to MP neuron - Performs linear classification
4. Inputs can be real, weights can be different for different i/p components

# Perceptron

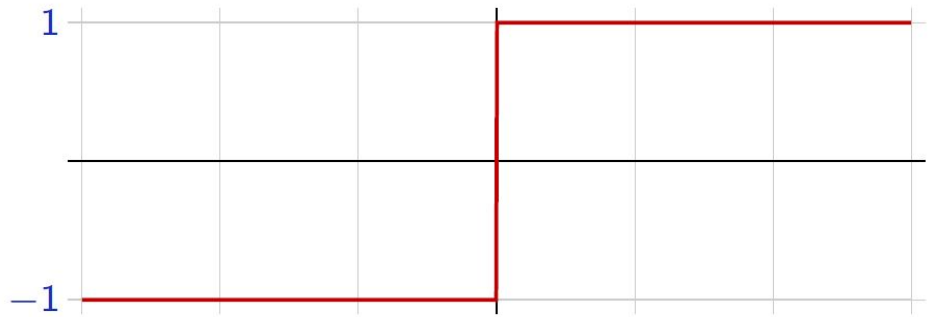


1. Frank Rosenblatt 1957 (American Psychologist)
2. Very crude biological model
3. Similar to MP neuron - Performs linear classification
4. Inputs can be real, weights can be different for different i/p components
5. $$f(x) = \begin{cases} 1 & \text{when } \sum_i w_i x_i + b \geq 0 \\ 0 & \text{else} \end{cases}$$

# Perceptron

1. For simplicity we consider $+1$ and -1 responses

$$\sigma(x) = \begin{cases} 1 & \text{when } x \geq 0 \\ -1 & \text{else} \end{cases}$$

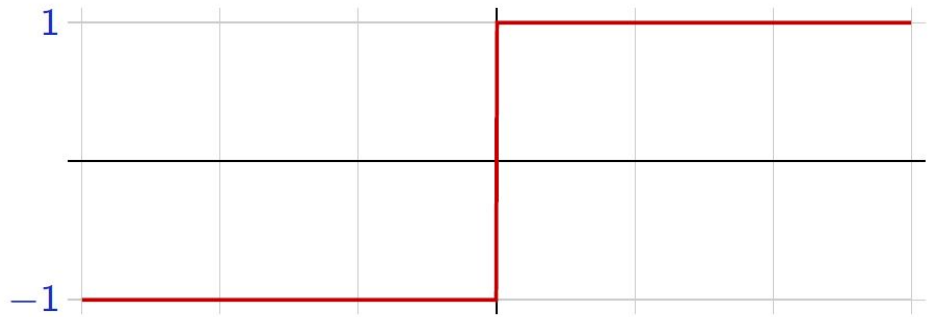



$$f(\mathbf{x}) = \sigma(\mathbf{w^T} \cdot \mathbf{x} + \mathbf{b})$$

# Perceptron

1. For simplicity we consider $+1$ and -1 responses

$$\sigma(x) = \begin{cases} 1 & \text{when } x \geq 0 \\ -1 & \text{else} \end{cases}$$



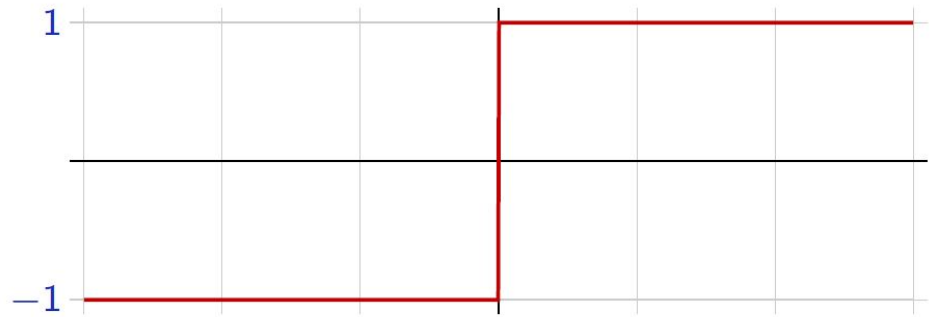

$$f(\mathbf{x}) = \sigma(\mathbf{w^T} \cdot \mathbf{x} + \mathbf{b})$$

2. In general, $\sigma(\cdot)$ that follows a linear operation is called an activation function

# Perceptron

1. For simplicity we consider $+1$ and -1 responses

$$\sigma(x) = \begin{cases} 1 & \text{when } x \geq 0 \\ -1 & \text{else} \end{cases}$$




$$f(\mathbf{x}) = \sigma(\mathbf{w^T} \cdot \mathbf{x} + \mathbf{b})$$

2. In general, $\sigma(\cdot)$ that follows a linear operation is called an activation function
3. $\mathbf{w}$ are referred to as weights and $b$ as the bias

# Perceptron vs. MP neuron



1. Perceptron is more general computational model

# Perceptron vs. MP neuron



1. Perceptron is more general computational model
2. Inputs can be real

# Perceptron vs. MP neuron



1. Perceptron is more general computational model
2. Inputs can be real
3. Weights are different on the input components

# Perceptron vs. MP neuron



1. Perceptron is more general computational model
2. Inputs can be real
3. Weights are different on the input components
4. Mechanism for learning the weights

# Weights and Bias



1. Why are the weights important?

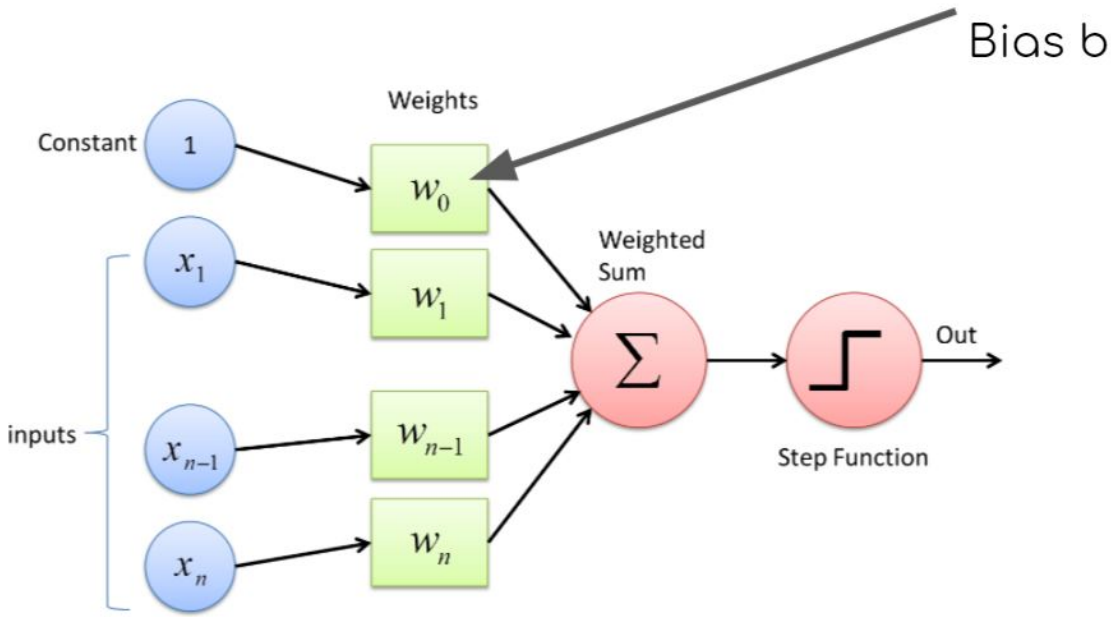



Figure credits: DeepAI

# Weights and Bias



1. Why are the weights important?
2. Why is it called 'bias'? What does it capture?

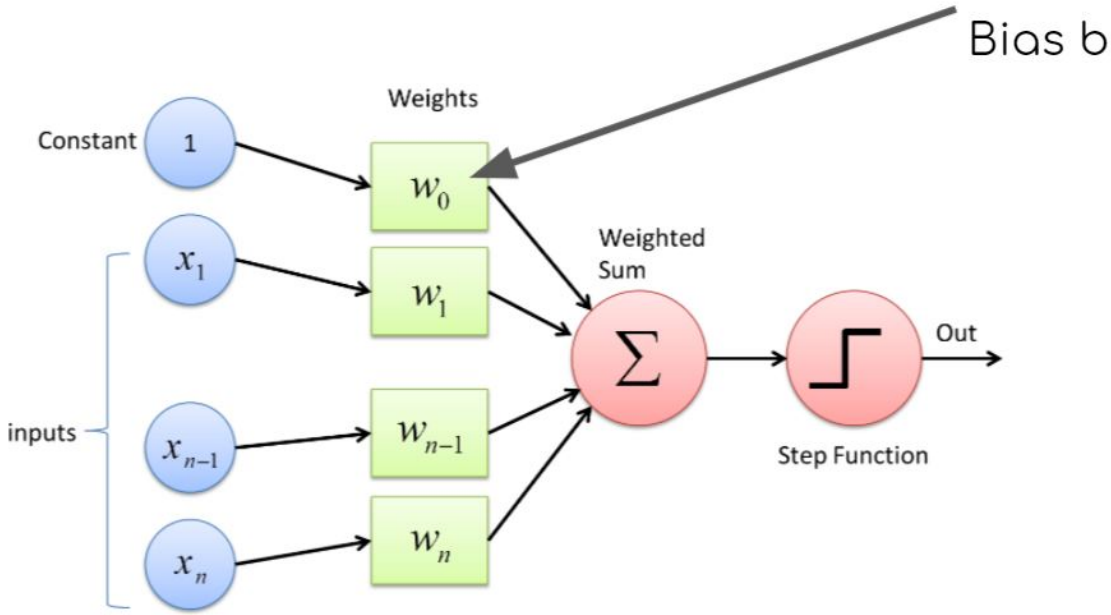



Figure credits: DeepAI

# Perceptron

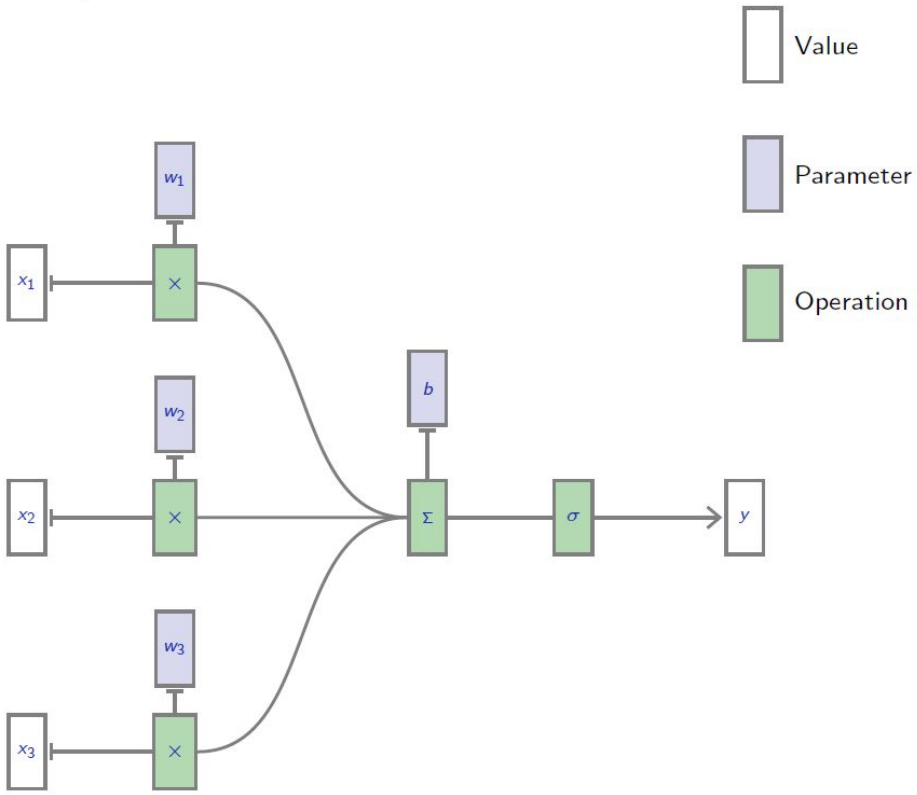



Figure credits: François Fleuret

# Perceptron

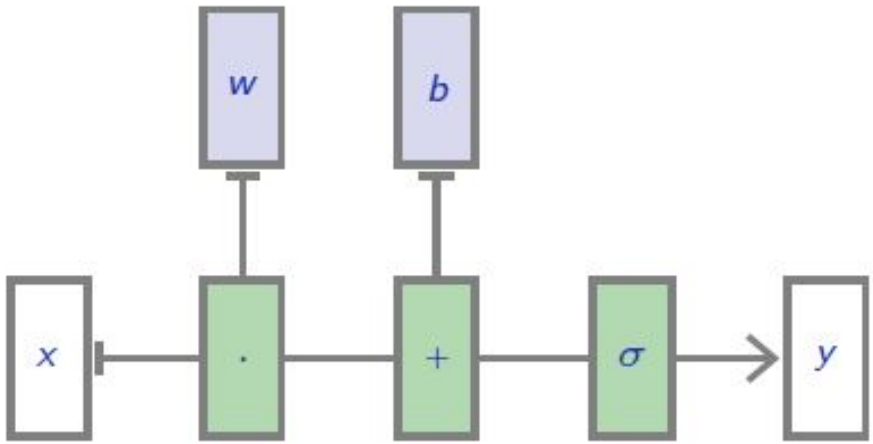



Figure credits: François Fleuret

# Perceptron Learning algorithm



1. Training data $(x^i, y^i) \in \mathcal{R}^D \times \{-1, 1\}, i = 1, \ldots, N$

# Perceptron Learning algorithm



1. Training data $(x^i, y^i) \in \mathcal{R}^D \times \{-1, 1\}, i = 1, \ldots, N$
2. Start with $k \leftarrow 1$ and $\mathbf{w_k} = \mathbf{0}$

# Perceptron Learning algorithm



1. Training data $(x^i, y^i) \in \mathcal{R}^D \times \{-1, 1\}, i = 1, \ldots, N$
2. Start with $k \leftarrow 1$ and $\mathbf{w_k} = \mathbf{0}$
3. While $\exists\ i \in \{1, 2 \ldots N\}$ such that $y^i(\mathbf{w_k^T} \cdot \mathbf{x^i}) \leq \mathbf{0}$, update
   $\mathbf{w_{k+1}} = \mathbf{w_k} + \mathbf{y^i} \cdot \mathbf{x^i}$
   $k \leftarrow k + 1$

# Perceptron Learning algorithm



1. Training data $(x^i, y^i) \in \mathcal{R}^D \times \{-1, 1\}, i = 1, \ldots, N$
2. Start with $k \leftarrow 1$ and $\mathbf{w_k} = \mathbf{0}$
3. While $\exists\ i \in \{1, 2 \ldots N\}$ such that $y^i(\mathbf{w_k^T} \cdot \mathbf{x^i}) \leq \mathbf{0}$, update
   $\mathbf{w_{k+1}} = \mathbf{w_k} + \mathbf{y^i} \cdot \mathbf{x^i}$
   $k \leftarrow k + 1$
4. Note that the bias $b$ is absorbed as a component of $\mathbf{w}$ and $\mathbf{x}$ is appended with $1$ suitably

# Perceptron Learning Algorithm



▸ Colab Notebook: Perceptron-learning

# Perceptron Learning Algorithm



1. Convergence result: For linearly separable dataset, the algorithm converges after finite iterations (refer to suggested readings)

# Perceptron Learning Algorithm



1. Convergence result: For linearly separable dataset, the algorithm converges after finite iterations (refer to suggested readings)
2. Stops as soon as it finds a separating boundary

# Perceptron Learning Algorithm



1. Convergence result: For linearly separable dataset, the algorithm converges after finite iterations (refer to suggested readings)
2. Stops as soon as it finds a separating boundary
3. Other (some) algorithms maximize the margin from the boundary to the samples

# So far...



1. Generalized to non-binary (real) input

# So far...



1. Generalized to non-binary (real) input
2. Considered unequal importance to the inputs

# So far...



1. Generalized to non-binary (real) input
2. Considered unequal importance to the inputs
3. Avoided heuristics with 'learning' the Threshold

# So far...



1. Generalized to non-binary (real) input
2. Considered unequal importance to the inputs
3. Avoided heuristics with 'learning' the Threshold
4. What if the data is not linearly separable?

# Coming..



1. Single layer Neural Networks - Regression & Classification