Deep Learning

3.1 Perceptron and MLP

Dr. Konda Reddy Mopuri
kmopuri@iittp.ac.in
Dept. of CSE, IIT Tirupati
Aug-Dec 2021

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

$$f(x) = \mathbb{1}(w \sum_i x_i + b \geq 0)$$

Threshold Logic Unit

1. let's implement simple functions

Threshold Logic Unit

1. let's implement simple functions
2. NOT

Threshold Logic Unit

1. let's implement simple functions
2. NOT
   - NOT$(x) = \mathbb{1}(-x + 0.5 \geq 0)$

Threshold Logic Unit

1. let's implement simple functions
2. NOT
    - $\text{NOT}(x) = \mathbb{1}(-x + 0.5 \geq 0)$
3. OR

Threshold Logic Unit

1. let's implement simple functions
2. NOT
   - NOT$(x) = \mathbb{1}(-x + 0.5 \geq 0)$
3. OR
   - OR$(x, y) = \mathbb{1}(x + b - 0.5 \geq 0)$

Threshold Logic Unit

1. let's implement simple functions
2. NOT
   - NOT$(x) = \mathbb{1}(-x + 0.5 \geq 0)$
3. OR
   - OR$(x, y) = \mathbb{1}(x + b - 0.5 \geq 0)$
4. AND

Threshold Logic Unit

1. let's implement simple functions
2. NOT
   - $\text{NOT}(x) = \mathbb{1}(-x + 0.5 \geq 0)$
3. OR
   - $\text{OR}(x, y) = \mathbb{1}(x + b - 0.5 \geq 0)$
4. AND
   - $\text{AND}(x, y) = \mathbb{1}(x + y - 1.5 \geq 0)$

Threshold Logic Unit

1. Can realize any Boolean function using TLUs

Threshold Logic Unit

1. Can realize any Boolean function using TLUs
2. What one unit does? - Learn linear separation

Threshold Logic Unit

1. Can realize any Boolean function using TLUs
2. What one unit does? - Learn linear separation
3. No learning; heuristics approach

Perceptron

1. Rosenblatt 1957 (American Psychologist)

Perceptron

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

Perceptron

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

Perceptron

1. 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

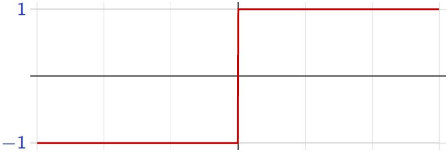
Perceptron

1. 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 consier +1 and -1 responses

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

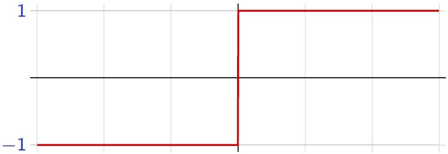


$$f(\mathbf{x}) = \sigma(\mathbf{w^T} \cdot \mathbf{x} + \mathbf{b})$$

Perceptron

1. For simplicity we consier +1 and -1 responses

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



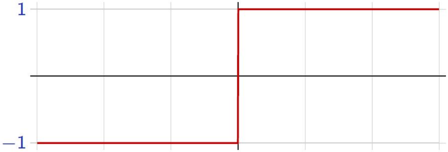
$$f(\mathbf{x}) = \sigma(\mathbf{w^T} \cdot \mathbf{x} + \mathbf{b})$$

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

Perceptron

1. For simplicity we consier +1 and -1 responses

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



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

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 separate

Perceptron vs. MP neuron

1. Perceptron is more general computational model
2. Inputs can be real
3. Weights are separate
4. Mechanism for learning weights

Weights and Bias

1. Why are the weights important?

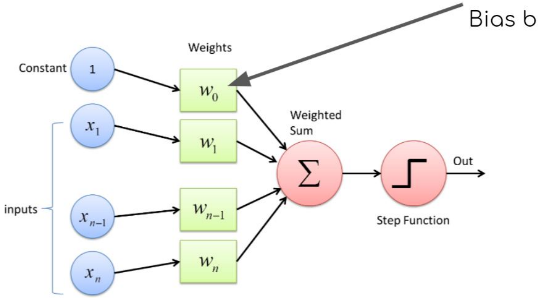


Figure credits: DeepAI

Weights and Bias

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

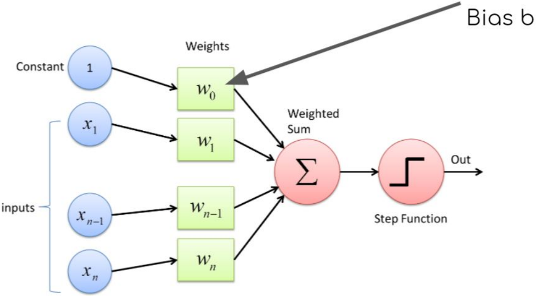


Figure credits: DeepAI

Perceptron

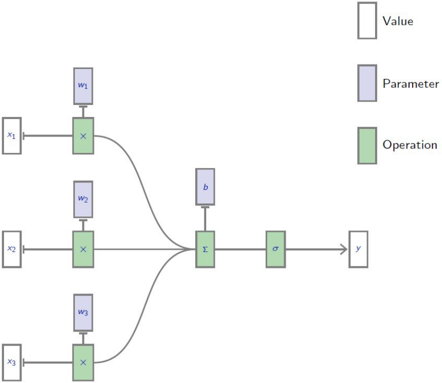


Figure credits: François Fleuret

Perceptron

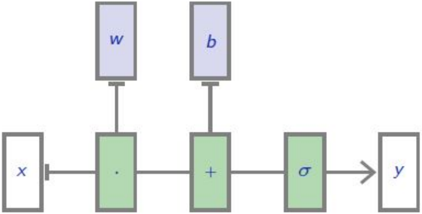


Figure credits: François Fleuret

Perceptron Learning algorithm

1. Training data $(x_n, y_n) \in \mathcal{R}^D \times -1, 1, n = 1, \dots, N$

Perceptron Learning algorithm

1. Training data $(x_n, y_n) \in \mathcal{R}^D \times -1, 1, n = 1, \dots, N$
2. Start with $\mathbf{w} = \mathbf{0}$

Perceptron Learning algorithm

1. Training data $(x_n, y_n) \in \mathcal{R}^D \times -1, 1, n = 1, \dots, N$
2. Start with $\mathbf{w} = \mathbf{0}$
3. While $\exists n_k$ such that $y_{nk}(\mathbf{w_k^T} \cdot \mathbf{x_{nk}} \leq \mathbf{0})$, update $\mathbf{w_{k+1}} = \mathbf{w_k} + \mathbf{y_{nk}} \cdot \mathbf{x_{nk}}$

Perceptron Learning algorithm

1. Training data $(x_n, y_n) \in \mathcal{R}^D \times -1, 1, n = 1, \ldots, N$
2. Start with $\mathbf{w} = \mathbf{0}$
3. While $\exists n_k$ such that $y_{nk}(\mathbf{w_k^T} \cdot \mathbf{x_{nk}} \leq \mathbf{0})$, update $\mathbf{w_{k+1}} = \mathbf{w_k} + \mathbf{y_{nk}} \cdot \mathbf{x_{nk}}$
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

Perceptron Learning Algorithm

1. Convergence result: Can show that after some iterations, no training sample gets misclassified

Perceptron Learning Algorithm

1. Convergence result: Can show that after some iterations, no training sample gets misclassified
2. Stops as soon as it finds a separating boundary

Perceptron Learning Algorithm

1. Convergence result: Can show that after some iterations, no training sample gets misclassified
2. Stops as soon as it finds a separating boundary
3. Other algorithms maximize the margin from boundary to the samples