Deep Learning

2 Perceptron is not perfect!

Dr. Konda Reddy Mopuri
Dept. of Artificial Intelligence
IIT Hyderabad

Jan-May 2023

Recap: Linear classifier

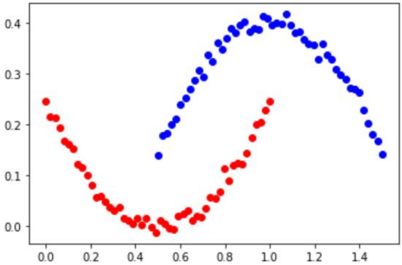
1. $f(x) = \sigma(\mathbf{w}^T\mathbf{x} + b)$

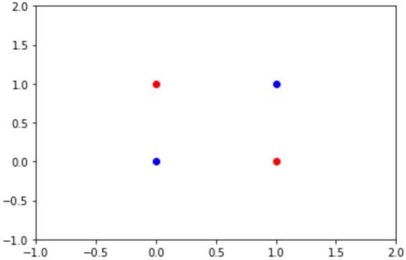
Recap: Linear classifier

1. $f(x) = \sigma(\mathbf{w}^T\mathbf{x} + b)$
2. Seen a couple of simple examples: MP neuron and Perceptron

Linear Classifiers: Shortcomings

- Lower capacity: data has to be linearly separable
- Some times no hyper-plane can separate the data (e.g. XOR data)





Pre-processing

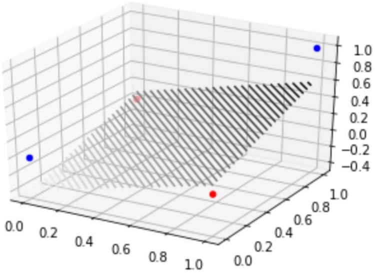
1. Sometimes, data specific pre-processing makes the data linearly separable

Pre-processing

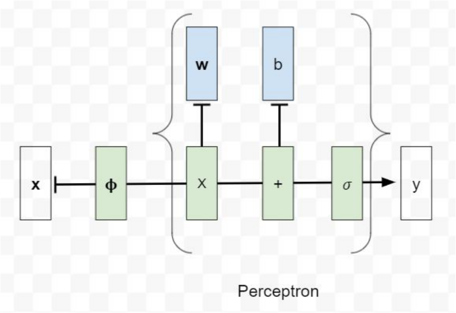
1. Sometimes, data specific pre-processing makes the data linearly separable
2. Consider the xor case
   $\phi(\mathbf{x}) = \phi(x_u, x_v) = (x_u, x_v, x_u x_v)$

Pre-processing

1. Sometimes, data specific pre-processing makes the data linearly separable
2. Consider the xor case
   $\phi(\mathbf{x}) = \phi(x_u, x_v) = (x_u, x_v, x_u x_v)$
3. Consider the perceptron in the new space $f(\mathbf{x}) = \sigma(\mathbf{w}^T \phi(\mathbf{x}) + b)$



Pre-processing



Pre-processing

1. Feature design (or pre-processing) may also be another way to reduce the capacity without affecting (or improving) the bias

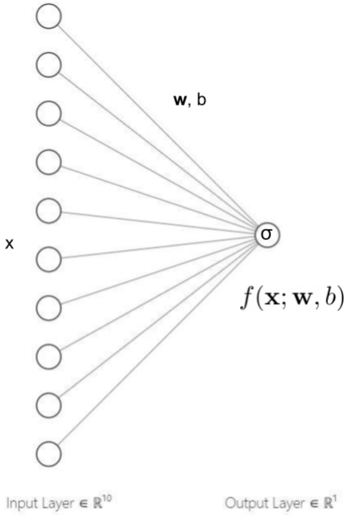
Extending Linear Classifier

1. Single class: $f(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x} + b)$ from $\mathcal{R}^D \rightarrow \mathcal{R}$ where $\mathbf{w}$ and $\mathbf{x} \in \mathcal{R}^D$

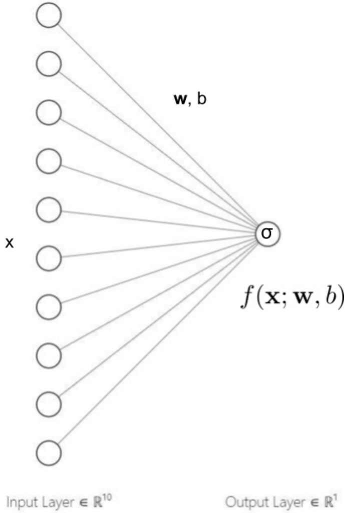
Extending Linear Classifier

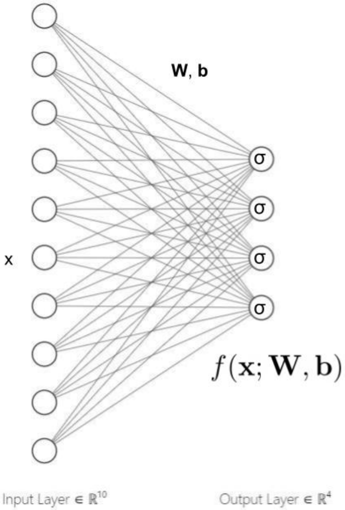
1. Single class: $f(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x} + b)$ from $\mathcal{R}^D \rightarrow \mathcal{R}$ where $\mathbf{w}$ and $\mathbf{x} \in \mathcal{R}^D$
2. Multi-class: $f(\mathbf{x}) = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$ from $\mathcal{R}^D \rightarrow \mathcal{R}^C$ where $\mathbf{W} \in \mathcal{R}^{C \times D}$ and $\mathbf{b} \in \mathcal{R}^C$

Single unit to a layer of Perceptrons


w, b
x
σ
$f(\mathbf{x}; \mathbf{w}, b)$
Input Layer $\in \mathbb{R}^{10}$
Output Layer $\in \mathbb{R}^{1}$

Single unit to a layer of Perceptrons

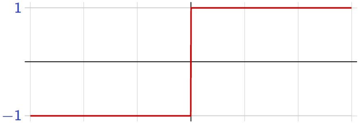

w, b
x
σ
$f(\mathbf{x}; \mathbf{w}, b)$
Input Layer ∈ $\mathbb{R}^{10}$
Output Layer ∈ $\mathbb{R}^{1}$


W, b
x
σ
σ
σ
σ
$f(\mathbf{x}; \mathbf{W}, \mathbf{b})$
Input Layer ∈ $\mathbb{R}^{10}$
Output Layer ∈ $\mathbb{R}^{4}$

Threshold-ing is very harsh!

1. Perceptron's o/p is discontinuous!

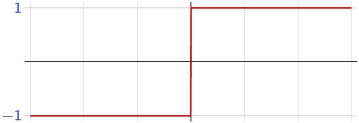
$$\sigma(x) = \begin{cases} 1 & \text{when } x \geq 0 \\ -1 & \text{else} \end{cases}$$



Threshold-ing is very harsh!

1. Perceptron's o/p is discontinuous!

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



2. Think of inputs -0.0001 and 0

Enough of Boolean functions!

1. Many real world problems have non-binary outputs

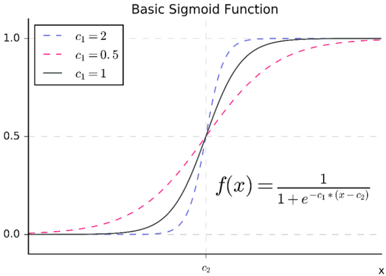
Enough of Boolean functions!

1. Many real world problems have non-binary outputs
2. Perceptron only gives two outputs!

Enough of Boolean functions!

1. Many real world problems have non-binary outputs
2. Perceptron only gives two outputs!
3. Sigmoid neuron

$$f(\mathbf{x}) = \frac{\mathbf{1}}{\mathbf{1} + \mathbf{e}^{-\mathbf{w}^{\mathbf{T}}\mathbf{x}}}$$


Basic Sigmoid Function
$c_1 = 2$
$c_1 = 0.5$
$c_1 = 1$
$f(x) = \frac{1}{1 + e^{-c_1 * (x - c_2)}}$
c_2
x