

Deep Learning

Dr. Konda Reddy Mopuri
Dept. of Artificial Intelligence
IIT Hyderabad
Jan-May 2023

Recap: Linear classifier

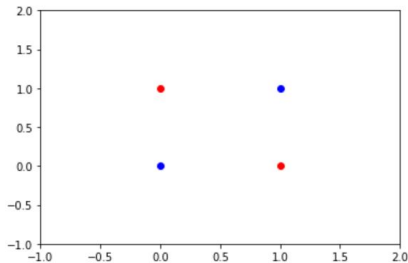
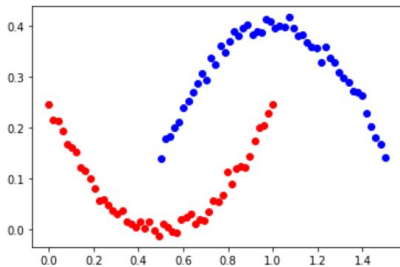
$$\textcircled{1} \quad f(x) = \sigma(\mathbf{w}^T \mathbf{x} + b)$$

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

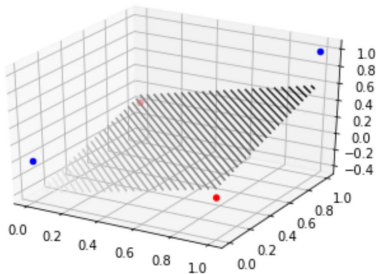
- ① Sometimes, data specific pre-processing makes the data linearly separable

Pre-processing

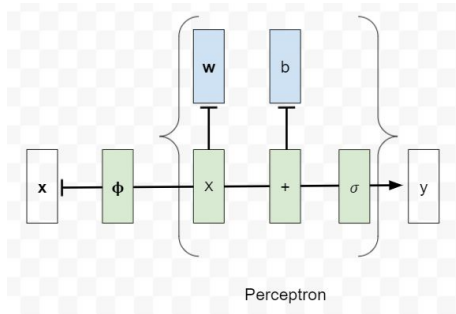
- ① Sometimes, data specific pre-processing makes the data linearly separable
- ② Consider the xor case
$$\phi(\mathbf{x}) = \phi(x_u, x_v) = (x_u, x_v, x_u x_v)$$

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

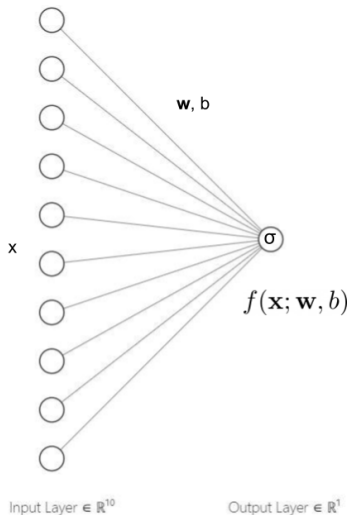
Extending Linear Classifier

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

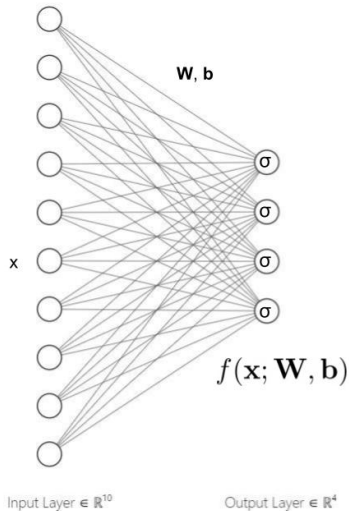
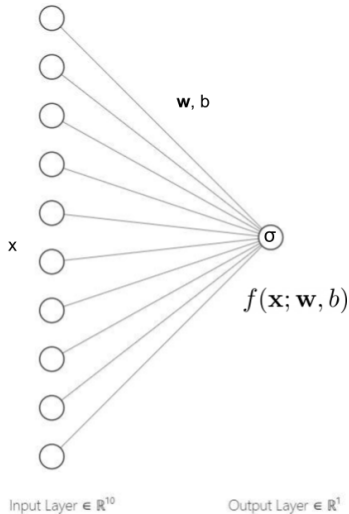
Extending Linear Classifier

- 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



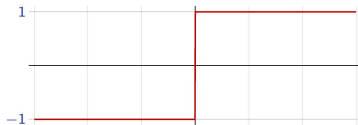
Single unit to a layer of Perceptrons



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

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

