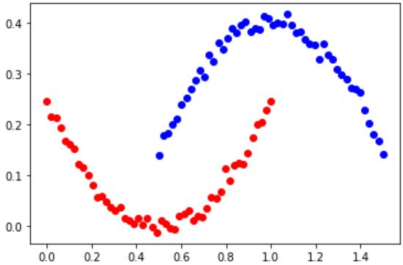
# Deep Learning

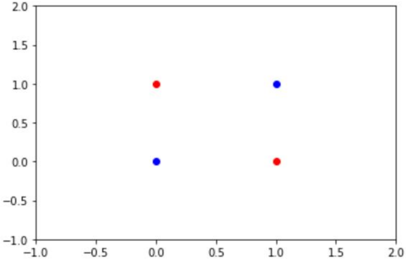
## 02 Network of Perceptrons

Dr. Konda Reddy Mopuri
Dept. of Artificial Intelligence
IIT Hyderabad
Jan-May 2024

# 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

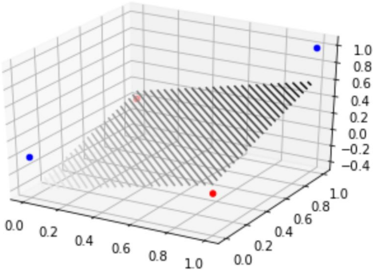
1. Sometimes, data specific pre-processing makes the data linearly separable

# 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

1. (Recap: Polynomial regression): increasing the degree $\rightarrow$ increase the model capacity

# Pre-processing

1. (Recap: Polynomial regression): increasing the degree $\rightarrow$ increase the model capacity
2. (Recap: Bias-Variance decomposition): to reduce the bias error, we increased the model capacity

# Pre-processing

1. (Recap: Polynomial regression): increasing the degree $\rightarrow$ increase the model capacity
2. (Recap: Bias-Variance decomposition): to reduce the bias error, we increased the model capacity
3. Feature design (or pre-processing) may also be another way to reduce the capacity without affecting (or improving) the bias

# Consider the XOR function

| $x_1$ | $x_2$ | XOR |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

# Consider the XOR function

- If we attempt to realize XOR function with a single perceptron

| $x_1$ | $x_2$ | XOR | |
|---|---|---|---|
| 0 | 0 | 0 | $w_0 < 0$ |
| 0 | 1 | 1 | $w_2 + w_0 \geq 0$ |
| 1 | 0 | 1 | $w_1 + w_0 \geq 0$ |
| 1 | 1 | 0 | $w_1 + w_2 + w_0 < 0$ |

# Consider the XOR function

- If we attempt to realize XOR function with a single perceptron

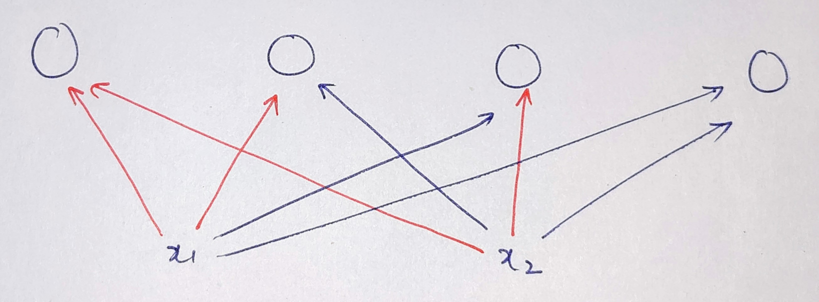
| $x_1$ | $x_2$ | XOR | |
|---|---|---|---|
| 0 | 0 | 0 | $w_0 < 0$ |
| 0 | 1 | 1 | $w_2 + w_0 \geq 0 \Rightarrow w_2 \geq -w_0$ |
| 1 | 0 | 1 | $w_1 + w_0 \geq 0 \Rightarrow w_1 \geq -w_0$ |
| 1 | 1 | 0 | $w_1 + w_2 + w_0 < 0 \Rightarrow w_1 + w_2 < -w_0$ |

# Consider the XOR function

- Clearly, a single perception cannot represent the XOR function!

# Let's see if multiple perceptions can do this

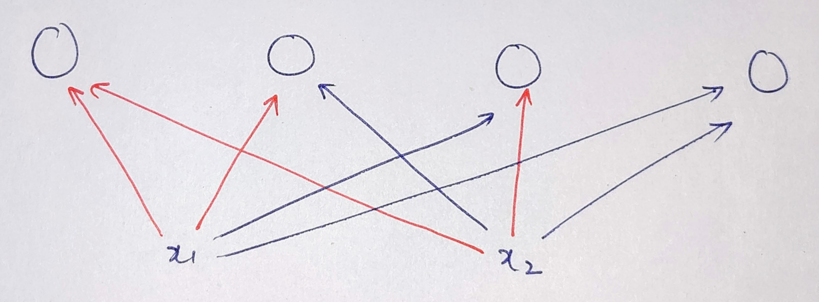
- Consider 4 perceptions



# Let's see if multiple perceptions can do this

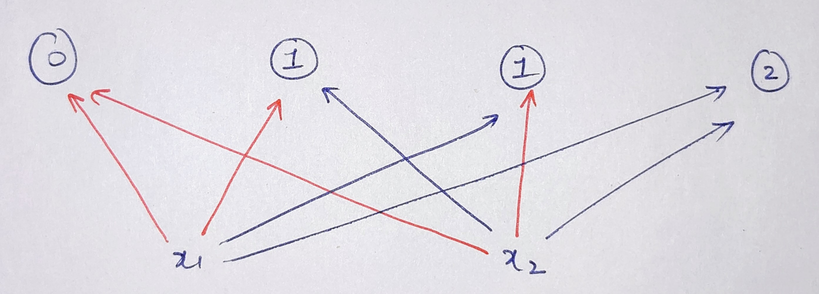


- Consider 4 perceptions
- $\color{red}{\rightarrow} = -1$ and $\color{blue}{\rightarrow} = +1$



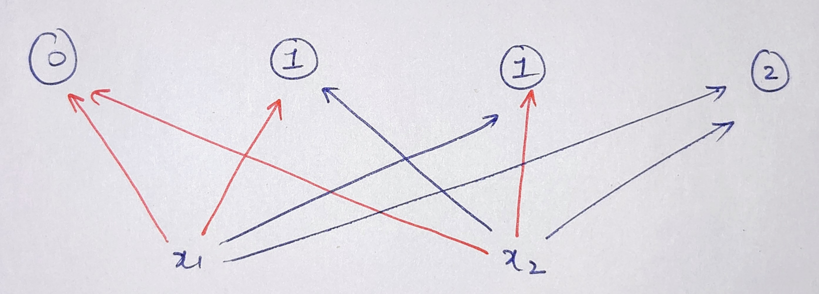
# Let's see if multiple perceptions can do this

- Let's have these thresholds



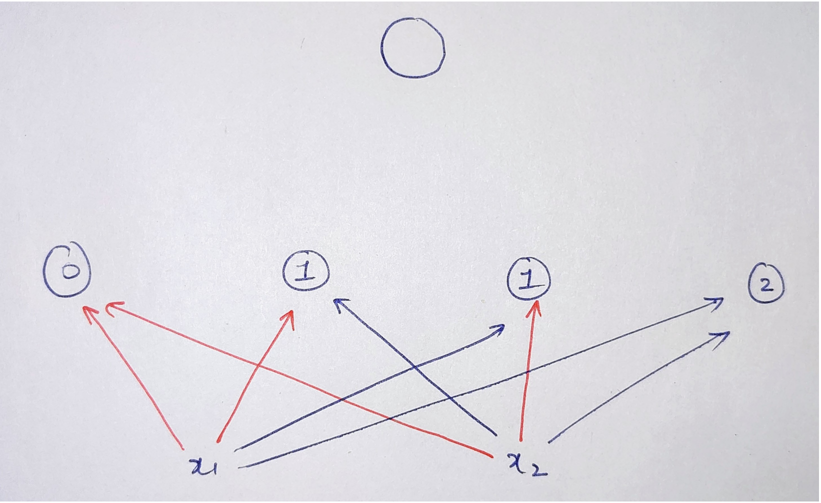
# Let's see if multiple perceptions can do this

- Let's have these thresholds
- Notice, each of them fire for exactly one specific input pattern



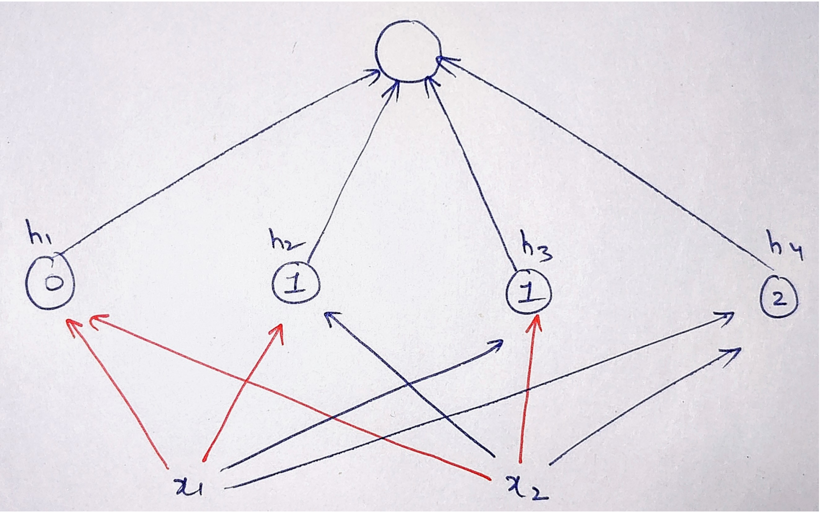
# Let's see if multiple perceptions can do this

- Let's now add another perceptron



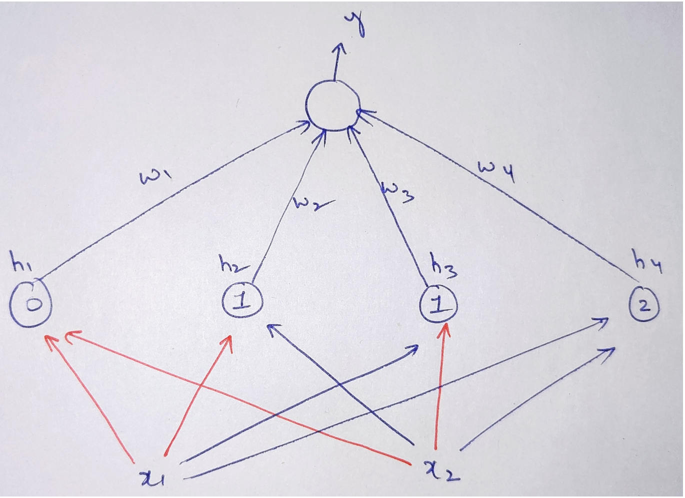
# Let's see if multiple perceptions can do this

- Connect the previous (hidden) ones and call it the output perceptron



# Let's see if multiple perceptions can do this

- See if we can find a set of weights $(W_i)$ to represent the XOR function



# Let's see if multiple perceptions can do this

| $x_1$ | $x_2$ | $h_1$ | $h_2$ | $h_3$ | $h_4$ | $y$ |
|---|---|---|---|---|---|---|
| 0 | 0 | | | | | |
| 0 | 1 | | | | | |
| 1 | 0 | | | | | |
| 1 | 1 | | | | | |

# Let's see if multiple perceptions can do this

| $x_1$ | $x_2$ | $h_1$ | $h_2$ | $h_3$ | $h_4$ | $y$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 | |
| 0 | 1 | | | | | |
| 1 | 0 | | | | | |
| 1 | 1 | | | | | |

# Let’s see if multiple perceptions can do this

| $x_1$ | $x_2$ | $h_1$ | $h_2$ | $h_3$ | $h_4$ | $y$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 | |
| 0 | 1 | 0 | 1 | 0 | 0 | |
| 1 | 0 | 0 | 0 | 1 | 0 | |
| 1 | 1 | 0 | 0 | 0 | 1 | |

# Let's see if multiple perceptions can do this

| $x_1$ | $x_2$ | $h_1$ | $h_2$ | $h_3$ | $h_4$ | $y$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 | $w_1$ |
| 0 | 1 | 0 | 1 | 0 | 0 | $w_2$ |
| 1 | 0 | 0 | 0 | 1 | 0 | $w_3$ |
| 1 | 1 | 0 | 0 | 0 | 1 | $w_4$ |

# Let's see if multiple perceptions can do this

| $x_1$ | $x_2$ | $h_1$ | $h_2$ | $h_3$ | $h_4$ | $y$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 | $w_1 + w_0$ |
| 0 | 1 | 0 | 1 | 0 | 0 | $w_2 + w_0$ |
| 1 | 0 | 0 | 0 | 1 | 0 | $w_3 + w_0$ |
| 1 | 1 | 0 | 0 | 0 | 1 | $w_4 + w_0$ |

# Let's see if multiple perceptions can do this

| $x_1$ | $x_2$ | $h_1$ | $h_2$ | $h_3$ | $h_4$ | $y$ | XOR |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 | $w_1 + w_0$ | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | $w_2 + w_0$ | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | $w_3 + w_0$ | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | $w_4 + w_0$ | 0 |

# Let's see if multiple perceptions can do this

| $x_1$ | $x_2$ | $h_1$ | $h_2$ | $h_3$ | $h_4$ | $y$ | XOR |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 | $w_1 + w_0$ | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | $w_2 + w_0$ | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | $w_3 + w_0$ | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | $w_4 + w_0$ | 0 |

$w_1 < -w_0 \quad w_2 \geq -w_0 \quad w_3 \geq -w_0 \quad w_4 < -w_0$

# Let's see if multiple perceptions can do this

- Clearly possible to find such weights → represent the XOR function!

| $x_1$ | $x_2$ | $h_1$ | $h_2$ | $h_3$ | $h_4$ | $y$ | XOR |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 | $w_1 + w_0$ | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | $w_2 + w_0$ | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | $w_3 + w_0$ | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | $w_4 + w_0$ | 0 |

$w_1 < -w_0 \quad w_2 \geq -w_0 \quad w_3 \geq -w_0 \quad w_4 < -w_0$

# What about other 2-input Boolean functions?

- Possible to represent!

# What about other 2-input Boolean functions?

- Possible to represent!
- Leads to finding a different set of non-contradicting weights

# What if there are more inputs?

- Can do the same with $2^n$ perceptions in the hidden layer and $1$ in the output layer!

# What did we just find?



- Any Boolean function of $n$ inputs can be exactly represented with $2^n$ perceptions in the hidden layer and $1$ in the output layer!

# What did we just find?

- Any Boolean function of $n$ inputs can be exactly represented with $2^n$ perceptions in the hidden layer and $1$ in the output layer!
- Note that $2^n + 1$ is a sufficient but not necessary

# What did we just find?

- Any Boolean function of $n$ inputs can be exactly represented with $2^n$ perceptions in the hidden layer and $1$ in the output layer!
- Note that $2^n + 1$ is a sufficient but not necessary
- Caveat: the size of the hidden layer grows exponentially!

# Network of Perceptrons

- Generally referred to as MLP (Multi-Layered Network of Perceptrons)

# Moving on from Boolean functions

- $y = f(x)$, where $x \in \mathcal{R}^n$ and $y \in \mathcal{R}$

# Moving on from Boolean functions

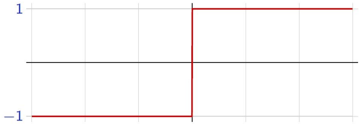


- $y = f(x)$, where $x \in \mathcal{R}^n$ and $y \in \mathcal{R}$
- Can MLPs represent such functions?

# Threshold-ing is very harsh!

1. Perceptron's o/p is discontinuous!

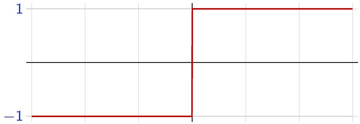
$$\sigma(x) = \begin{cases} 1 & \text{when } x \geq 0 \\ -1 & \text{else} \end{cases}$$



# 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

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