



Human-Centered Data & AI



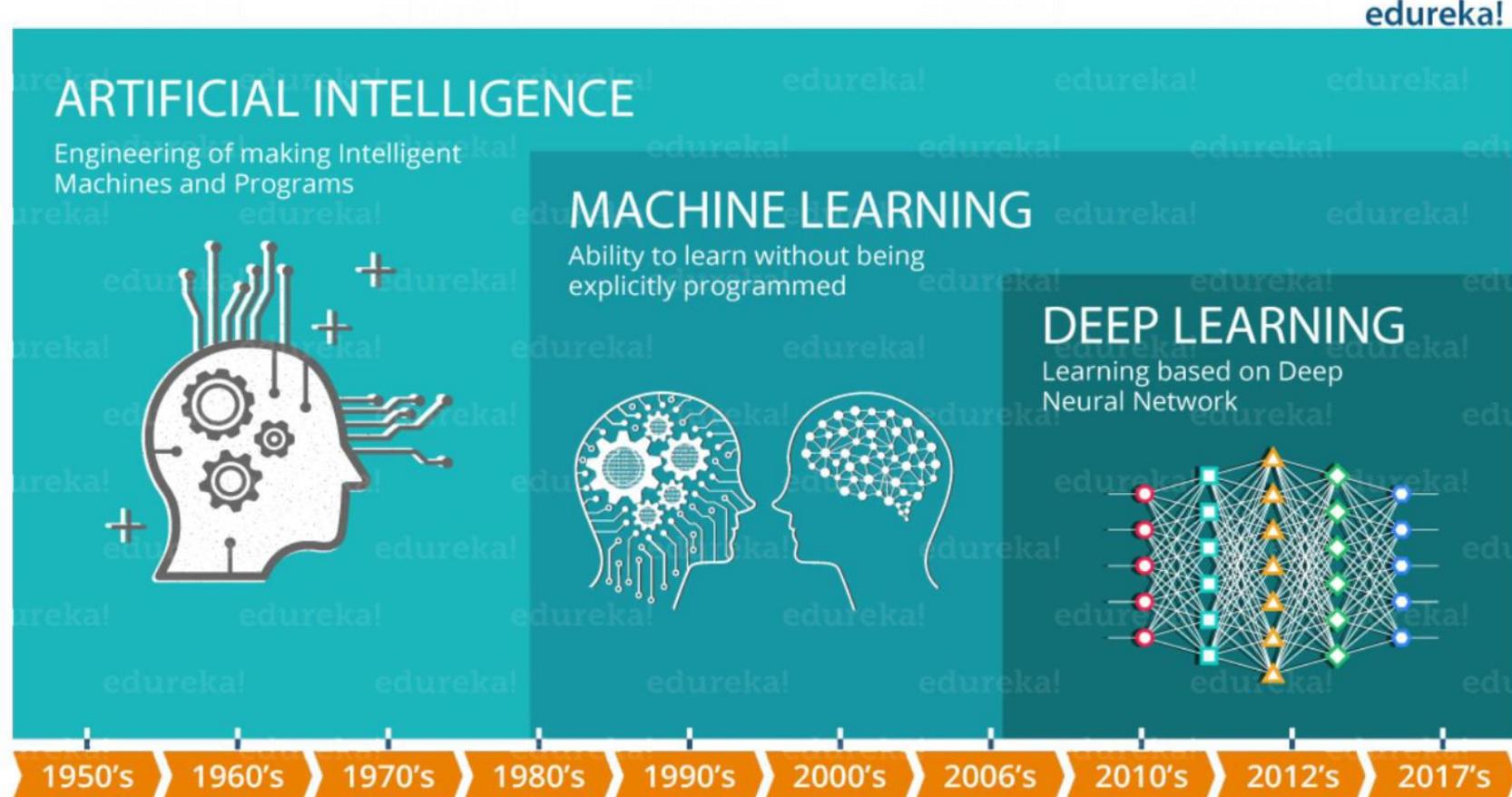
Vinicius Caridá, Ph.D.

- Executive Specialist, Artificial Intelligence and Data - Itaú
- MBA Professor – FIAP and ESPM



“

Redes Neurais



Motivação biológica

- O neurônio recebe impulsos (sinais) de outros neurônios por meio dos seus dendritos;
- O neurônio envia impulsos para outros neurônios por meio do seu axônio;
- O axônio termina num tipo de contato chamado sinapse, que conecta-o com o dendrito de outro neurônio.

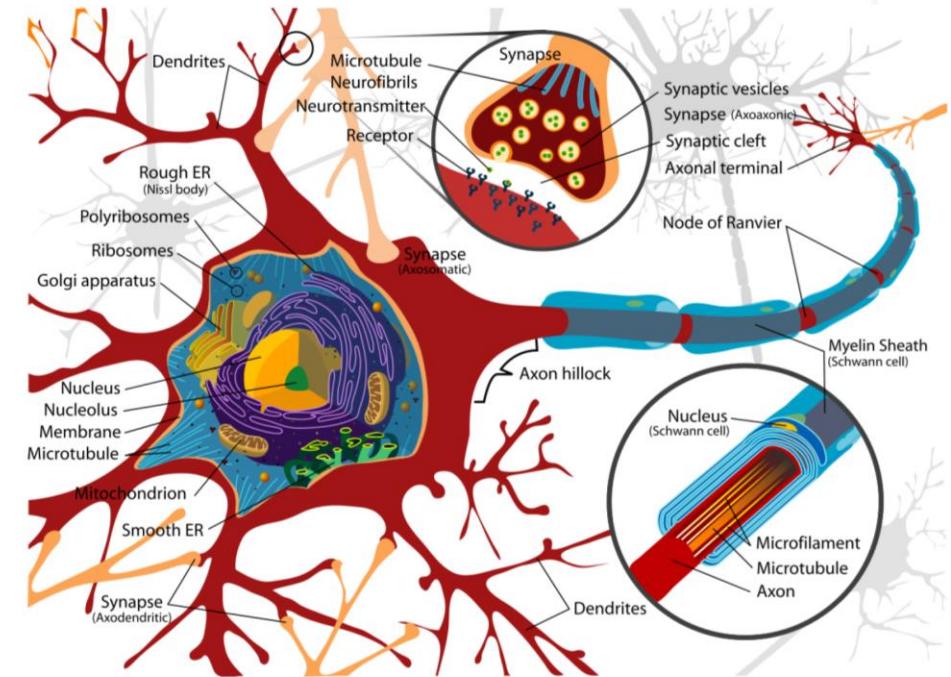
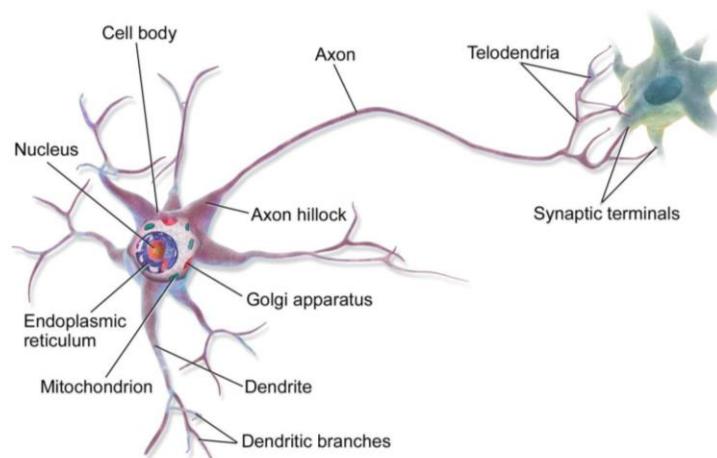


Figura 2 – Elementos constituintes de um neurônio

Extraído de: [http://upload.wikimedia.org/wikipedia/commons/thumb/a/a9/Complete_neuron_cell_diagram_en.svg/1280px-Complete_neuron_cell_diagram_en.svg.png]

Motivação biológica

- A sinapse libera substâncias químicas chamadas de neurotransmissores, em função do pulso elétrico disparado pelo axônio;
- O fluxo de neurotransmissores nas sinapses pode ter um efeito excitatório ou inbitório sobre o neurônio receptor



Destaque para os telodendros (ramificações do axônio)

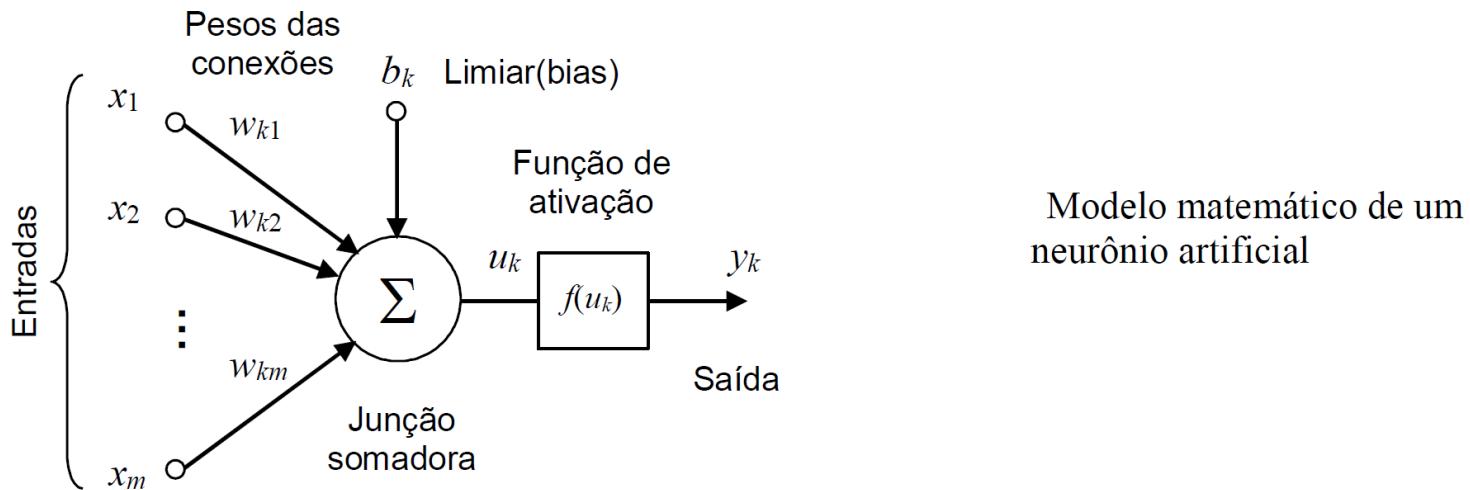
Fonte: https://en.wikipedia.org/wiki/Neural_circuit

Motivação biológica

- O aprendizado ocorre por sucessivas modificações nas sinapses que interconectam os neurônios, em função da maior ou menor liberação de neurotransmissores;
- À medida que novos eventos ocorrem, determinadas ligações entre neurônios são reforçadas, enquanto outras enfraquecidas;
- Este ajuste nas ligações entre os neurônios é uma das características das redes neurais artificiais.

Redes Neurais

- Modelo matemático: Simplificações da realidade com o propósito de representar aspectos relevantes de um sistema em estudo, sendo que detalhes de menor significância são descartados para viabilizar a modelagem.



- A saída do neurônio k pode ser descrita por: $y_k = f(u_k) = f\left(\sum_{j=1}^m w_{kj}x_j + b_k\right)$

Redes Neurais

O problema do OU-exclusivo em MLP

- Considere os pontos $(0,0), (0,1), (1,0)$ e $(1,1)$ no plano \mathbb{R}^2 , conforme apresentado na Figura 26. O objetivo é determinar uma rede com duas entradas $x_i \in \{0,1\}$ ($i=1,2$), e uma saída $y \in \{0,1\}$ de maneira que: $\begin{cases} (x_1, x_2) = (0,0) \text{ ou } (1,1) \Rightarrow y = 0 \\ (x_1, x_2) = (1,0) \text{ ou } (0,1) \Rightarrow y = 1 \end{cases}$

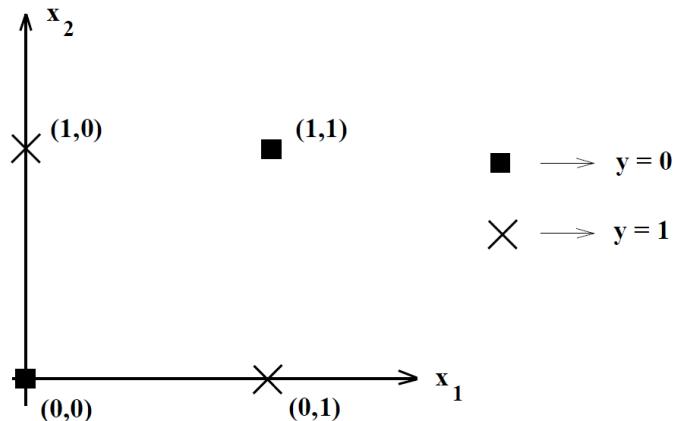


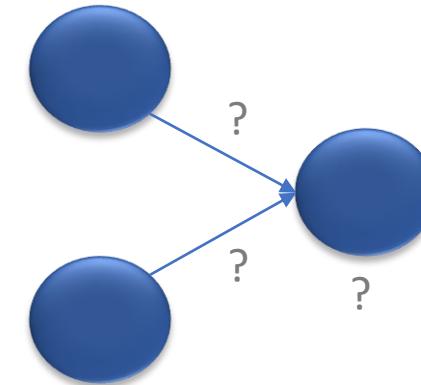
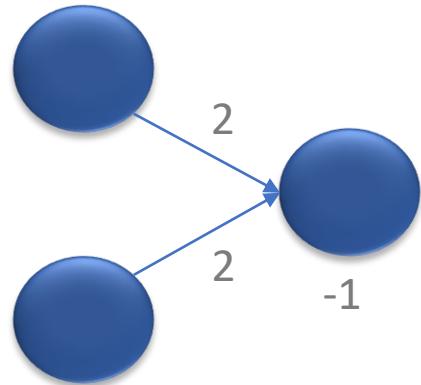
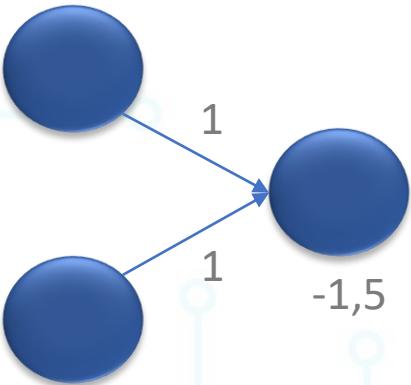
Figura 26 – O problema do OU-exclusivo

Redes Neurais

0	0	0
0	1	0
1	0	0
1	1	1

0	0	0
0	1	1
1	0	1
1	1	1

0	0	0
0	1	1
1	0	1
1	1	0



Função Degrau Padrão
 Se ≥ 0 então 1
 Se < 0 então 0

Redes Neurais

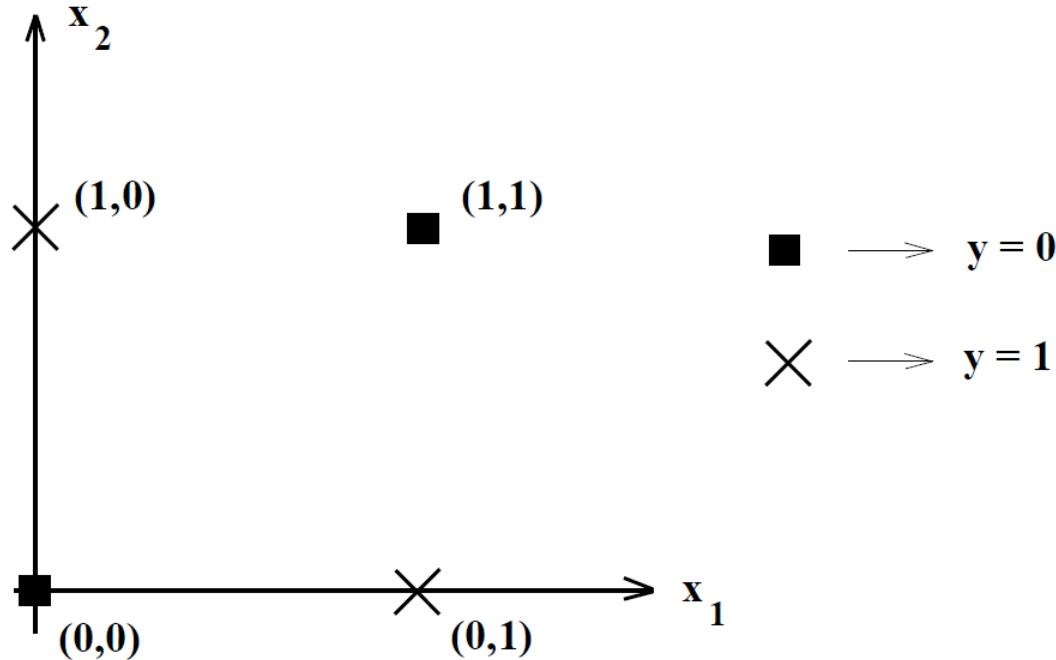
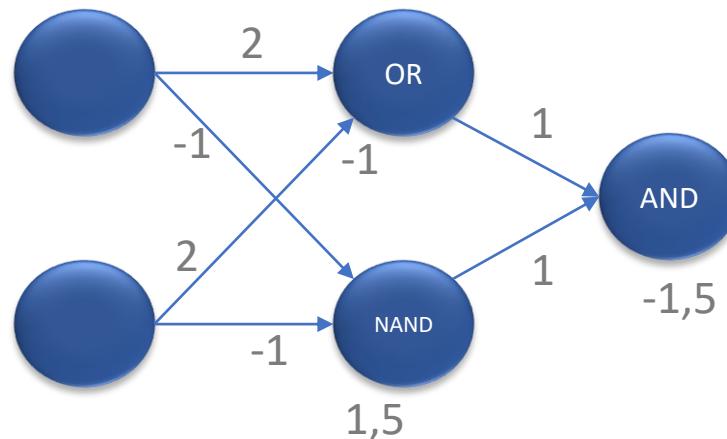


Figura 26 – O problema do OU-exclusivo

Redes Neurais

0	0	0
0	1	1
1	0	1
1	1	0



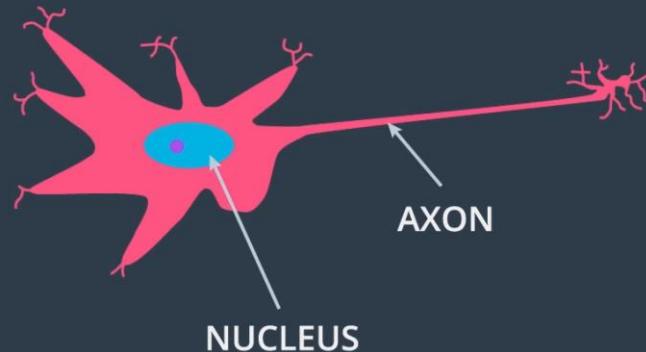
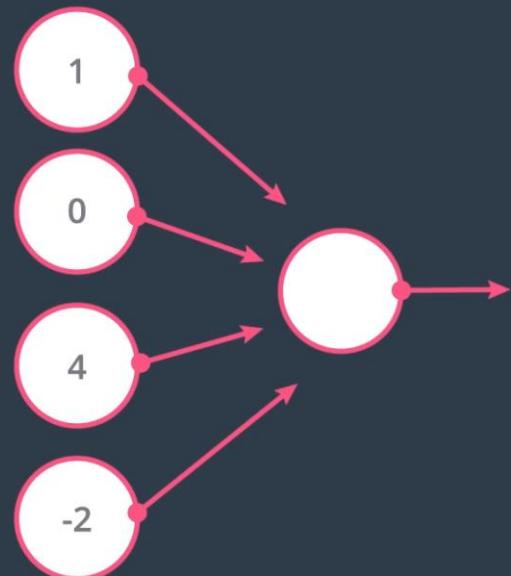
Função Degrau Padrão

Se ≥ 0 então 1

Se < 0 então 0

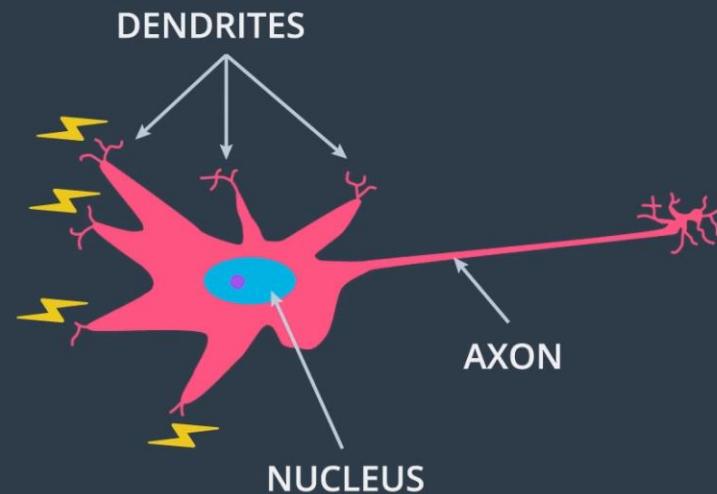
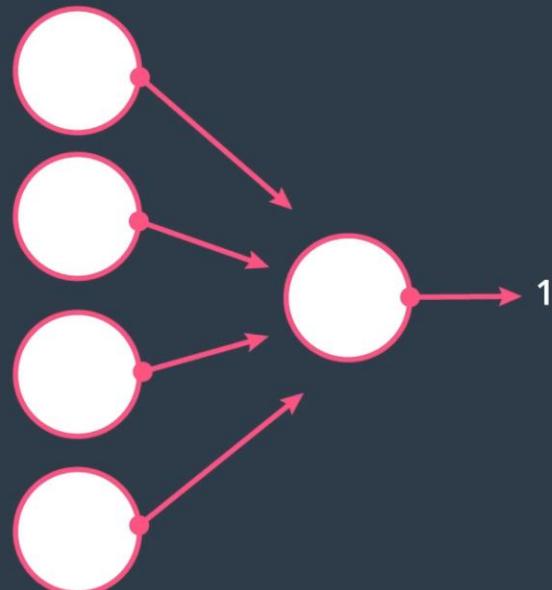
Redes Neurais

Perceptron



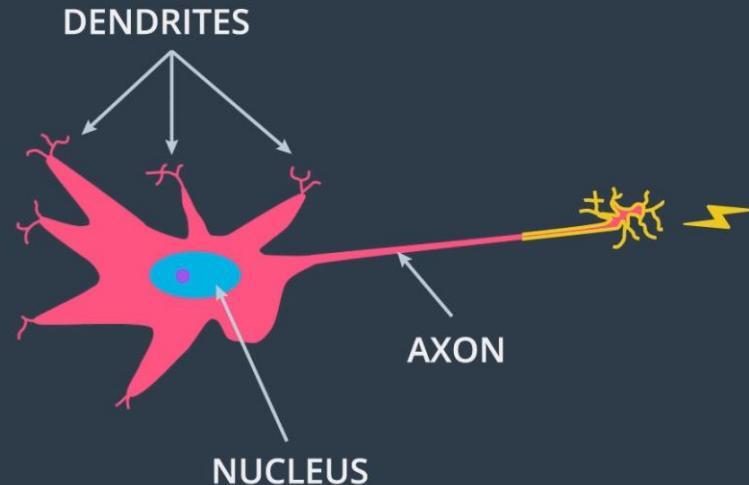
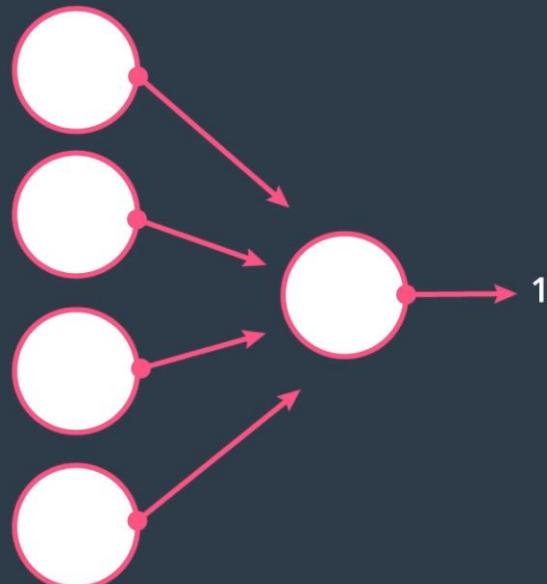
Redes Neurais

Perceptron



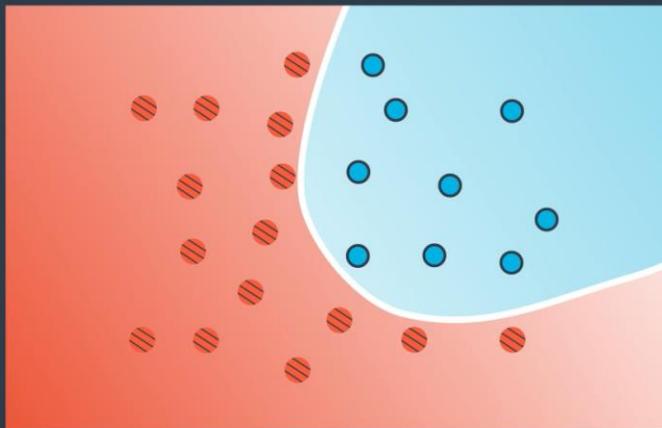
Redes Neurais

Perceptron



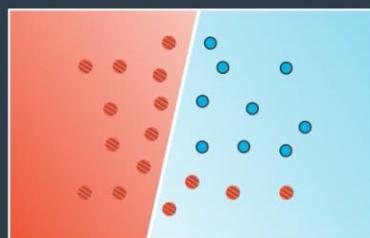
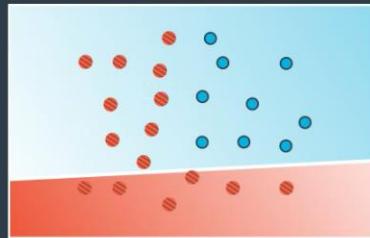
Redes Neurais - Arquitetura

Non-Linear Regions



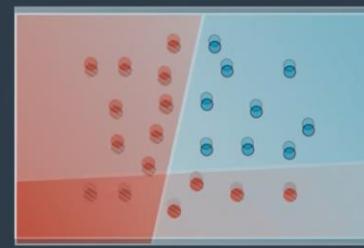
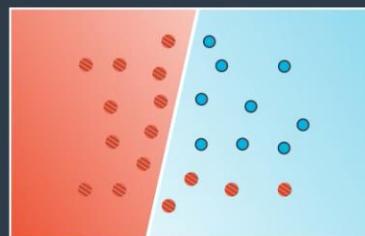
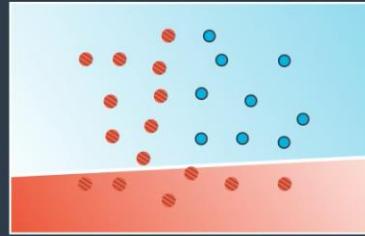
Redes Neurais - Arquitetura

Combining Regions



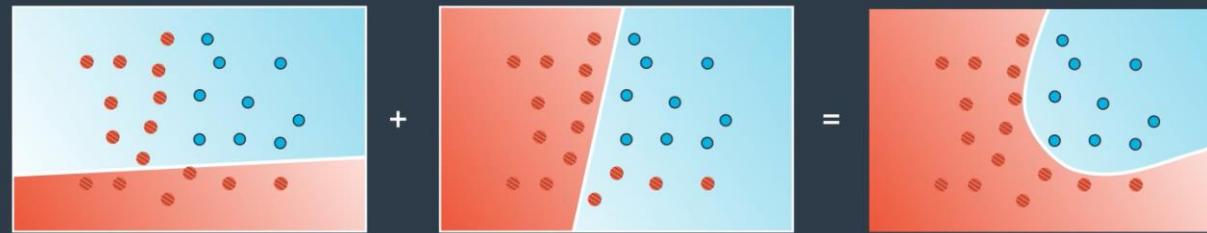
Redes Neurais - Arquitetura

Combining Regions



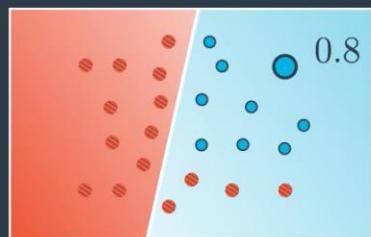
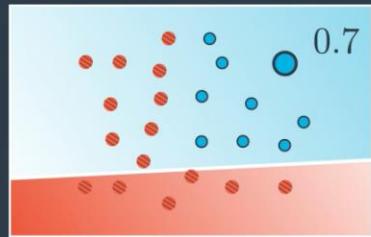
Redes Neurais - Arquitetura

Combining Regions



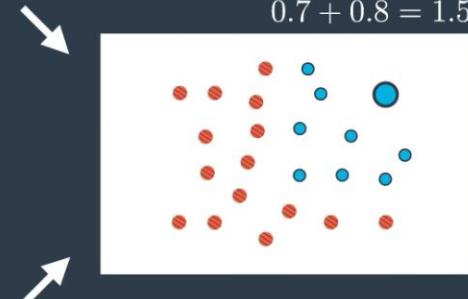
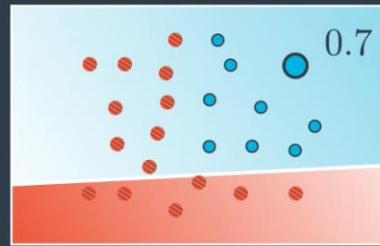
Redes Neurais - Arquitetura

Neural Network

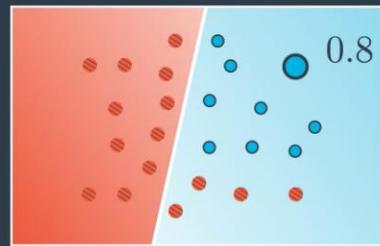


Redes Neurais - Arquitetura

Neural Network

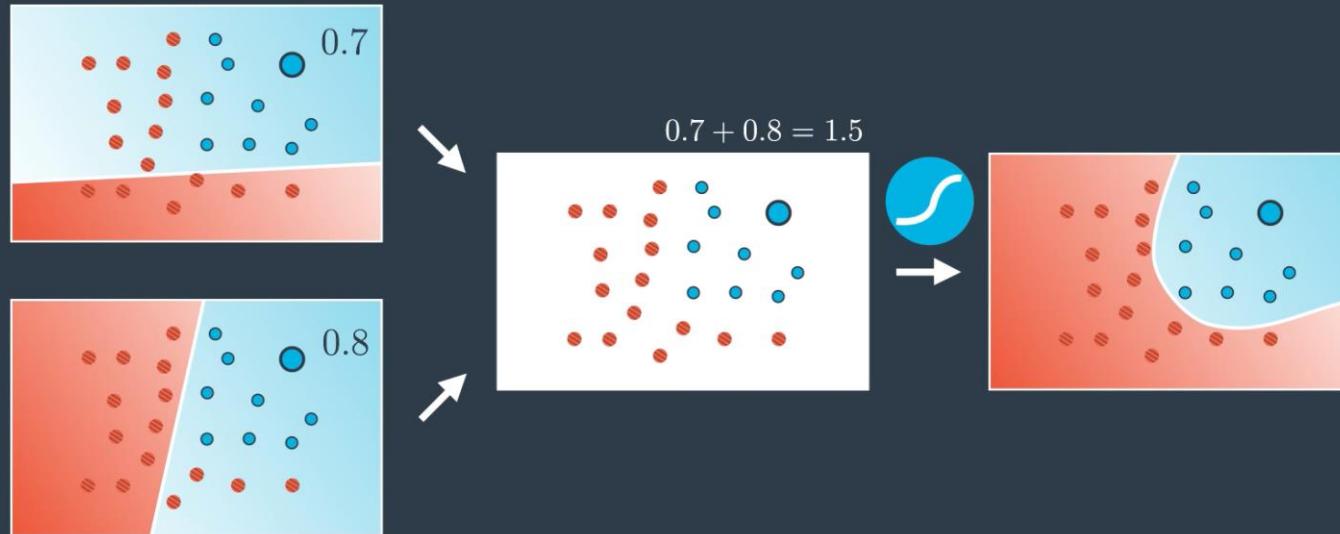


$$0.7 + 0.8 = 1.5$$

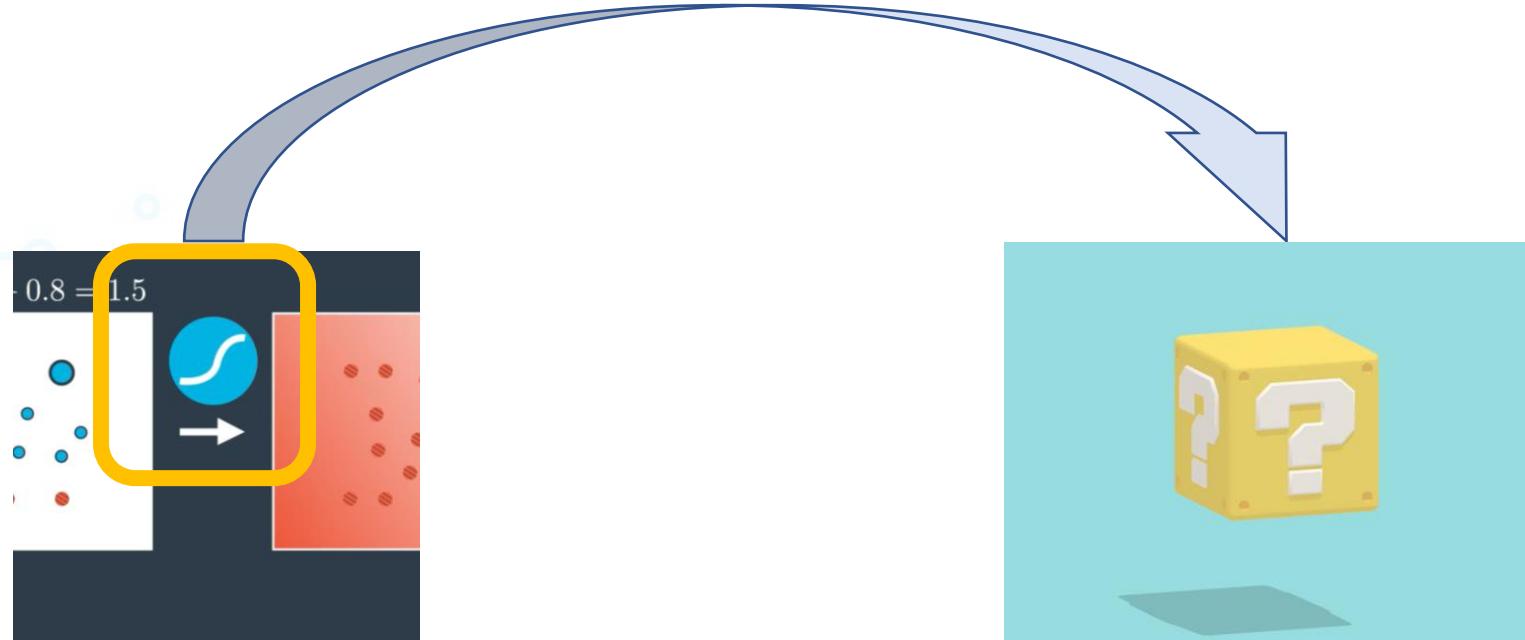


Redes Neurais - Arquitetura

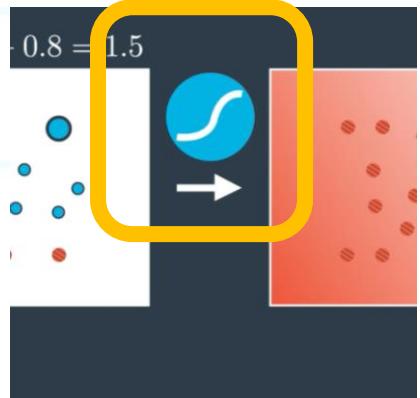
Neural Network



Redes Neurais - Arquitetura

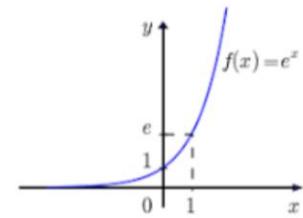


Redes Neurais - Arquitetura

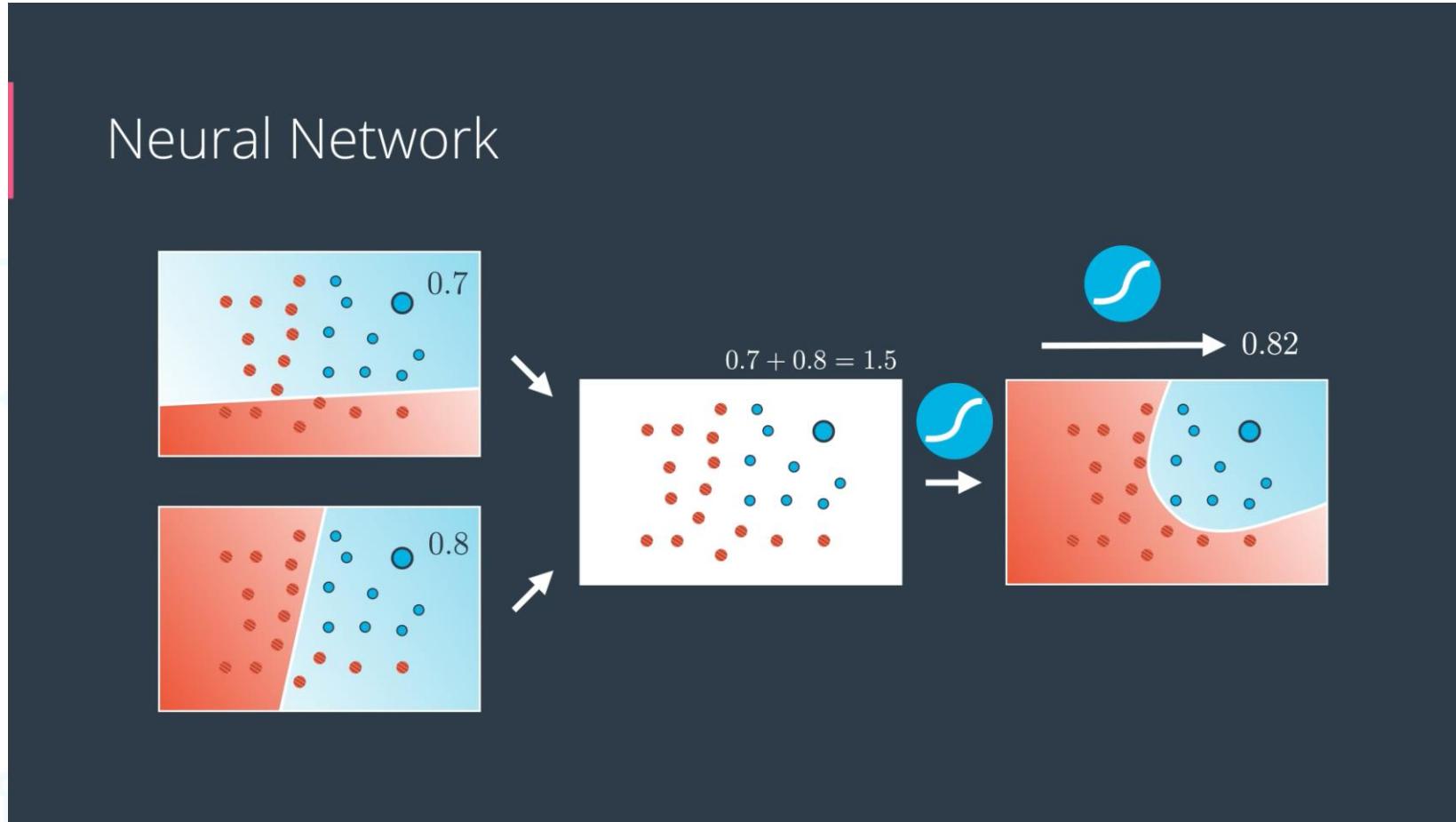


$$f(x) = \frac{1}{1 + e^{-x}} \text{ para todo } x \text{ real.}$$

A função **exponencial** natural, denotada e^x ou $\exp(x)$ é a função **exponencial** cuja base é o número de Euler (um número irracional que **vale** aproximadamente 2,718281828).

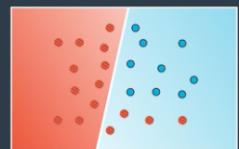
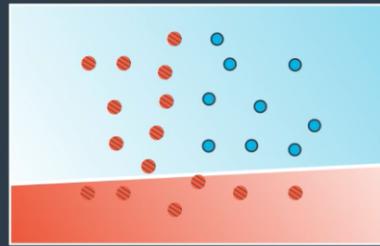


Redes Neurais - Arquitetura



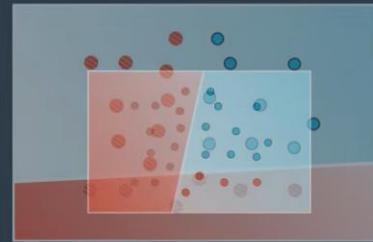
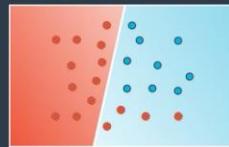
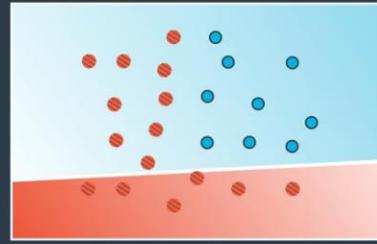
Redes Neurais - Arquitetura

Combining Regions



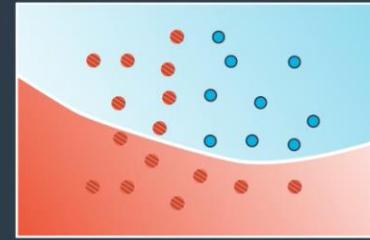
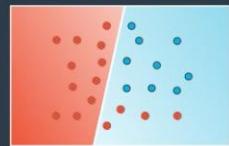
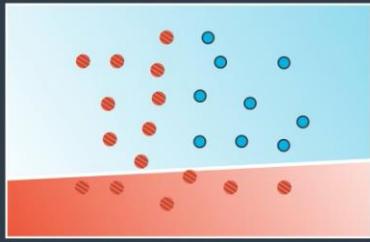
Redes Neurais - Arquitetura

Combining Regions



Redes Neurais - Arquitetura

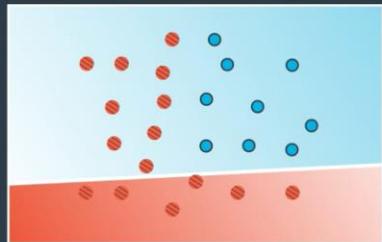
Combining Regions



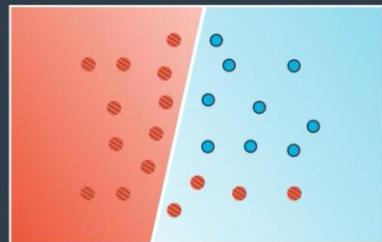
Redes Neurais - Arquitetura

Neural Network

7

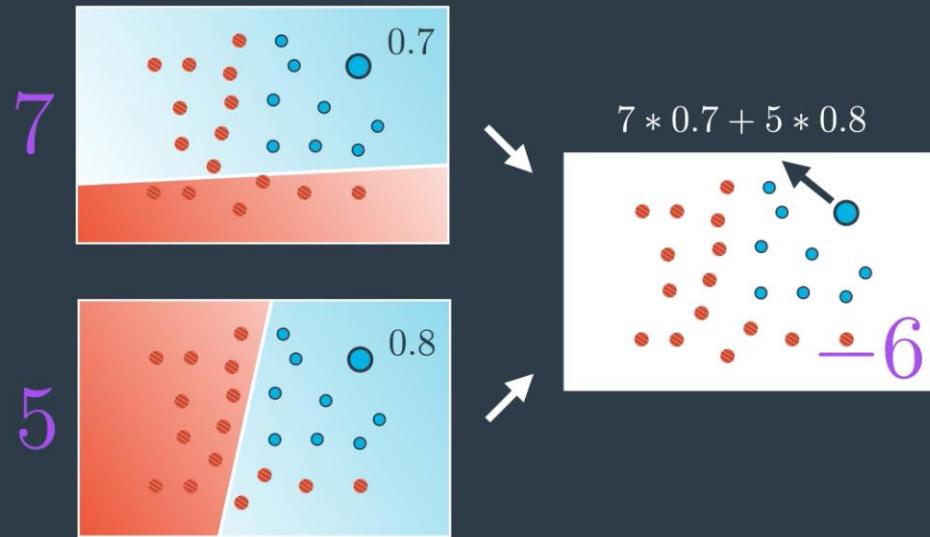


5



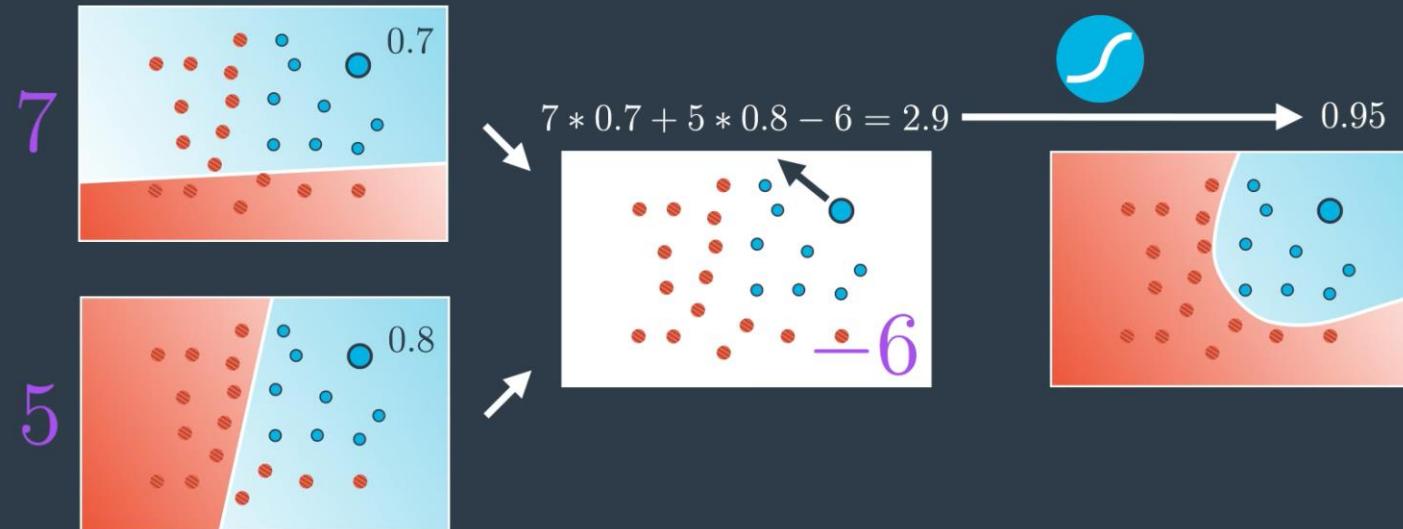
Redes Neurais - Arquitetura

Neural Network

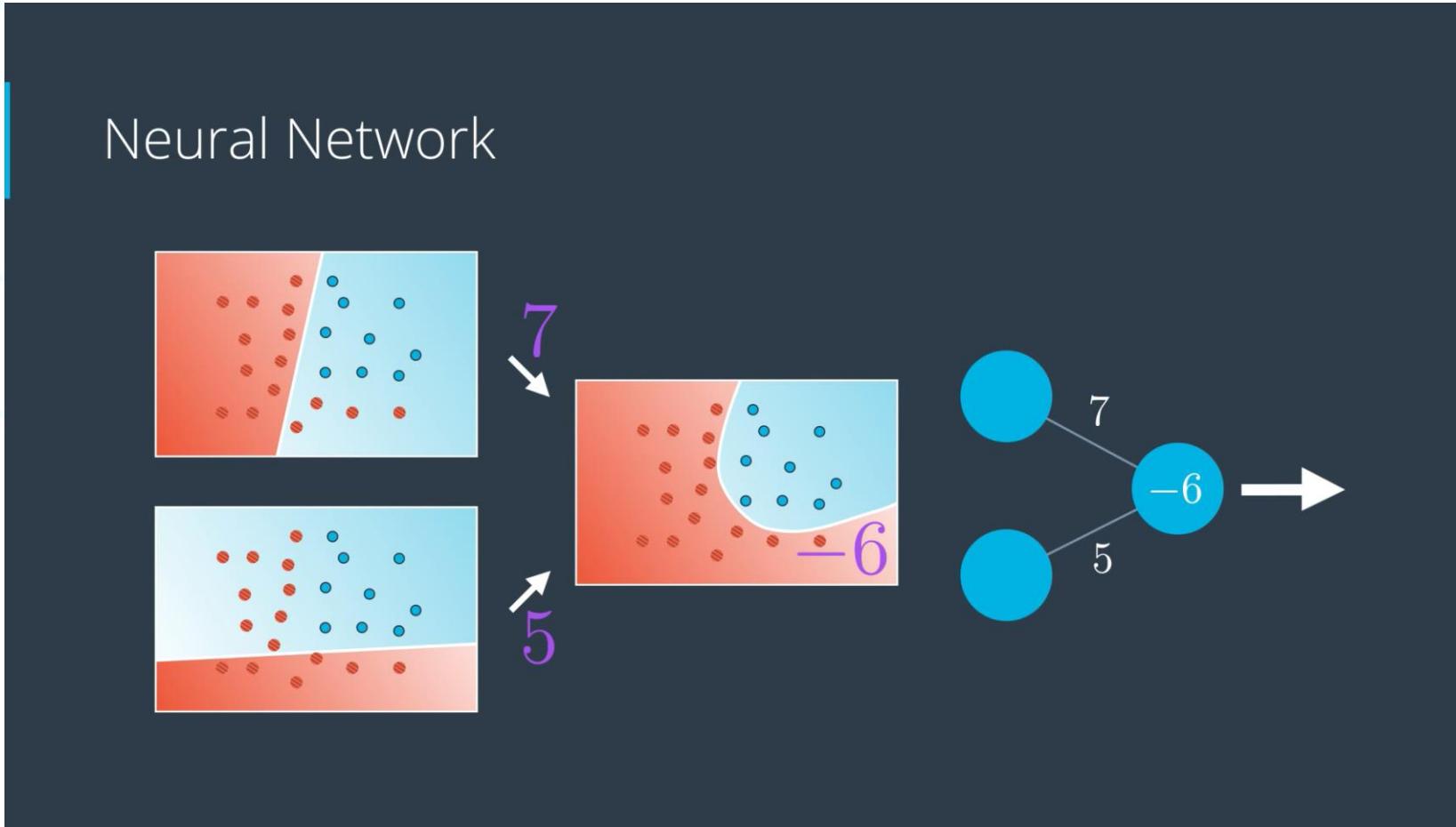


Redes Neurais - Arquitetura

Neural Network

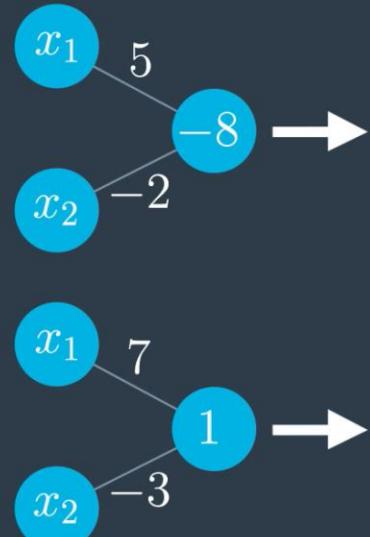
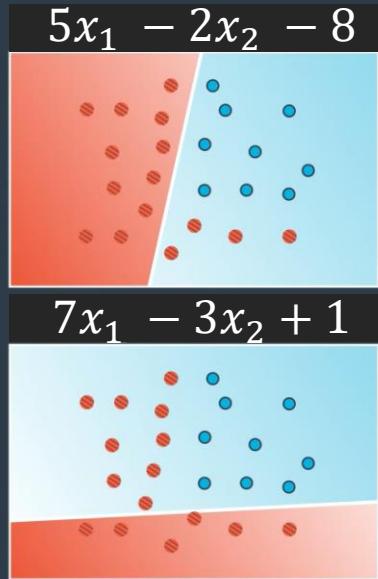


Redes Neurais - Arquitetura

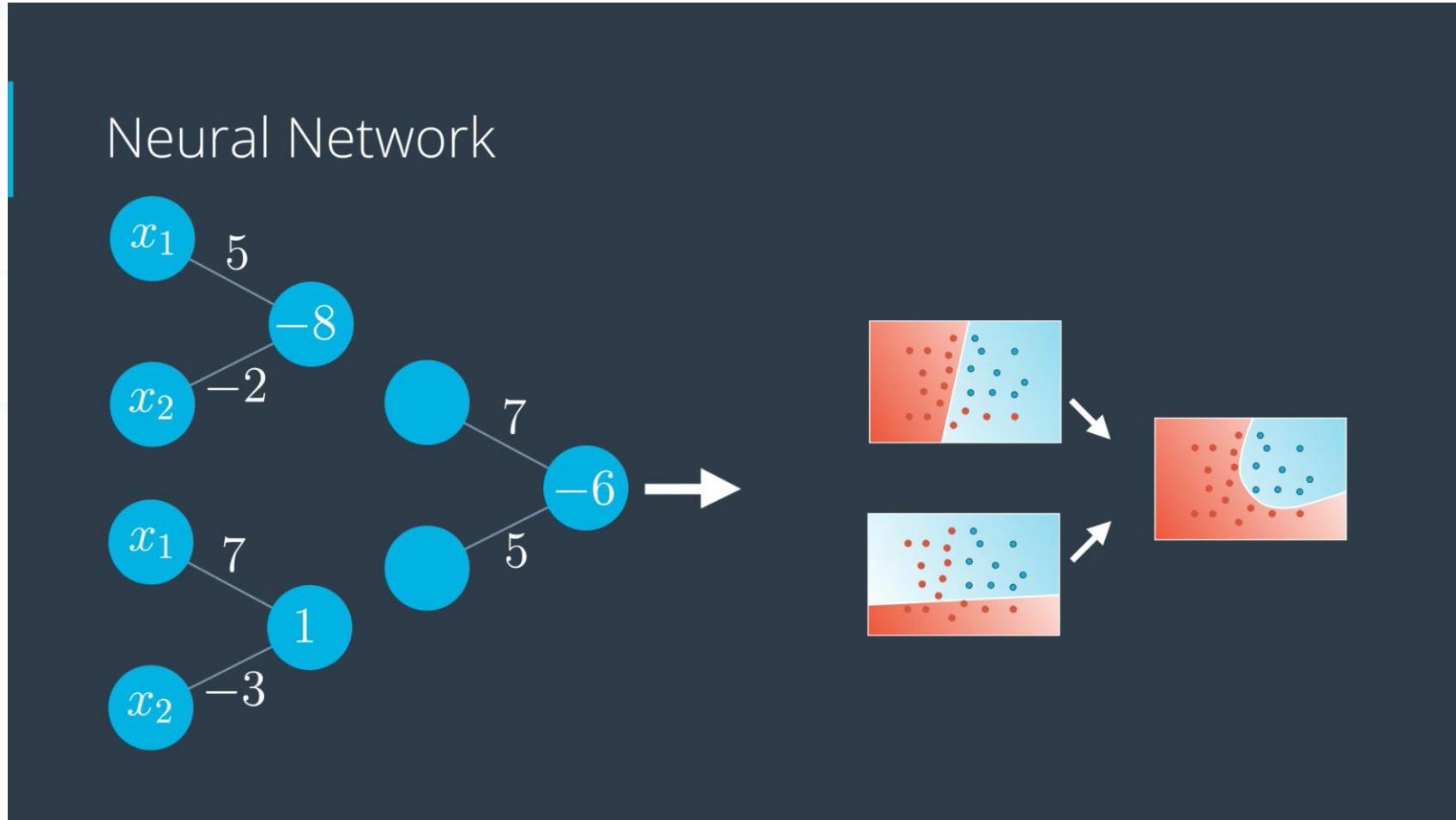


Redes Neurais - Arquitetura

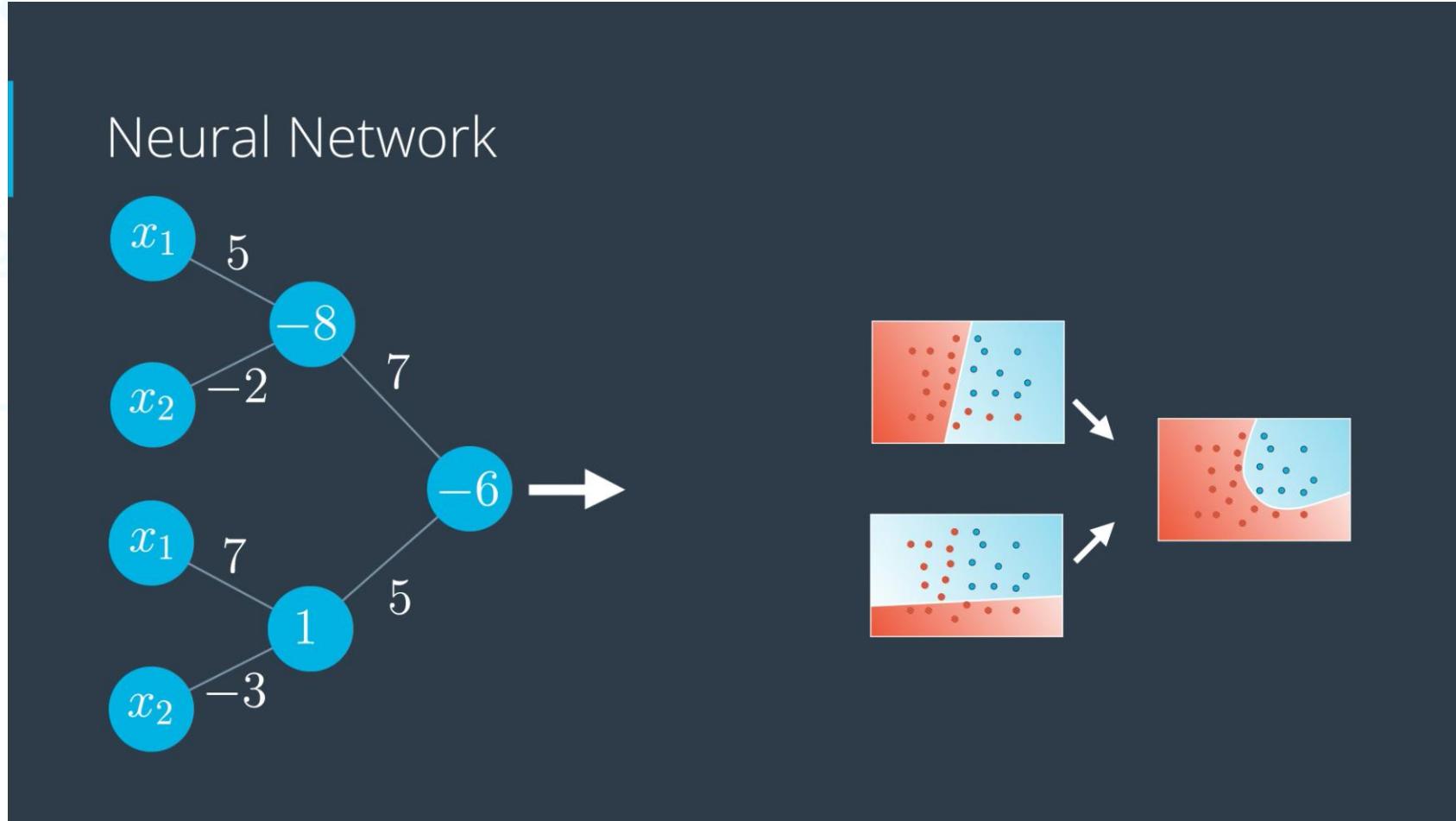
Neural Network



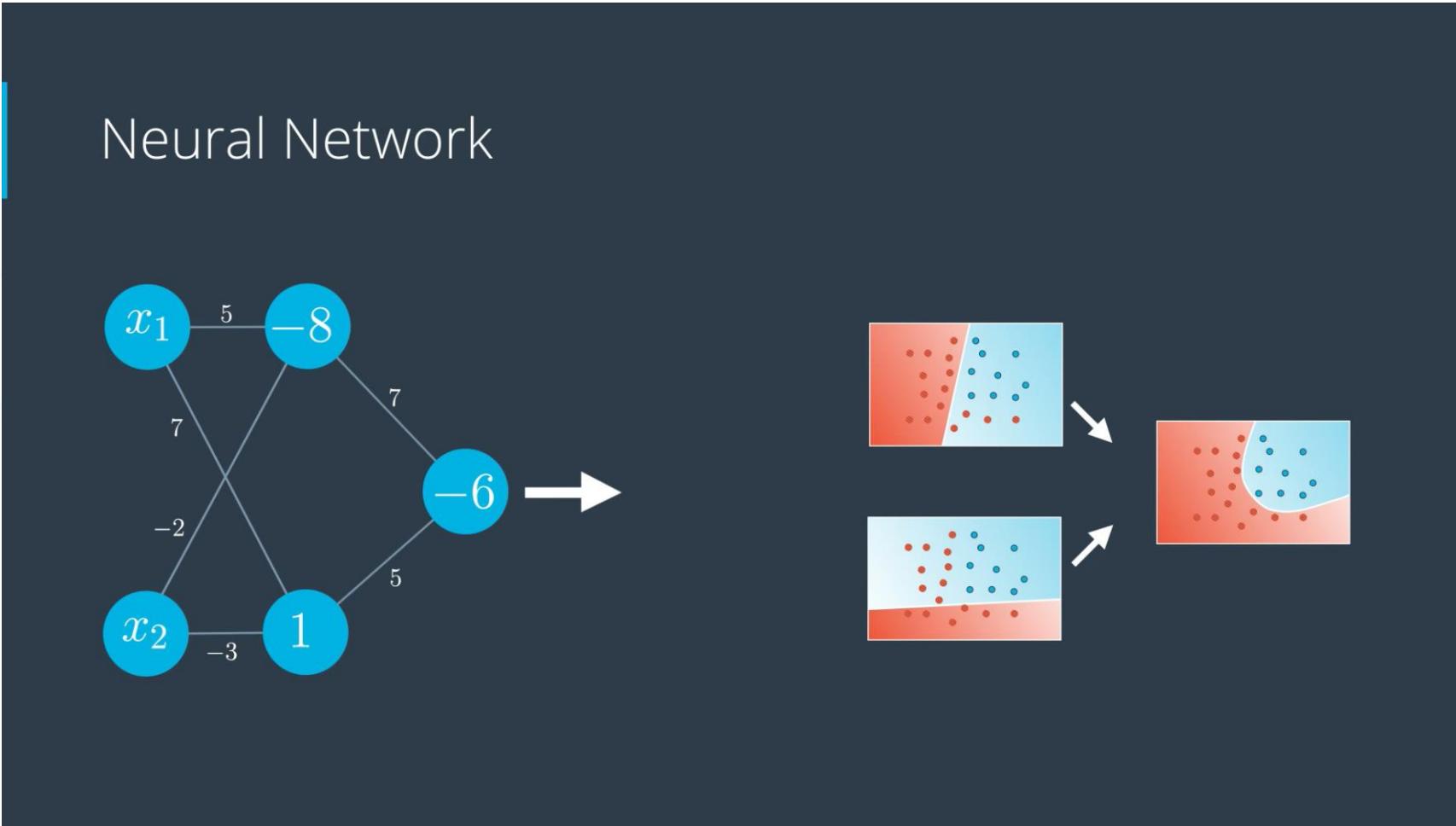
Redes Neurais - Arquitetura



Redes Neurais - Arquitetura

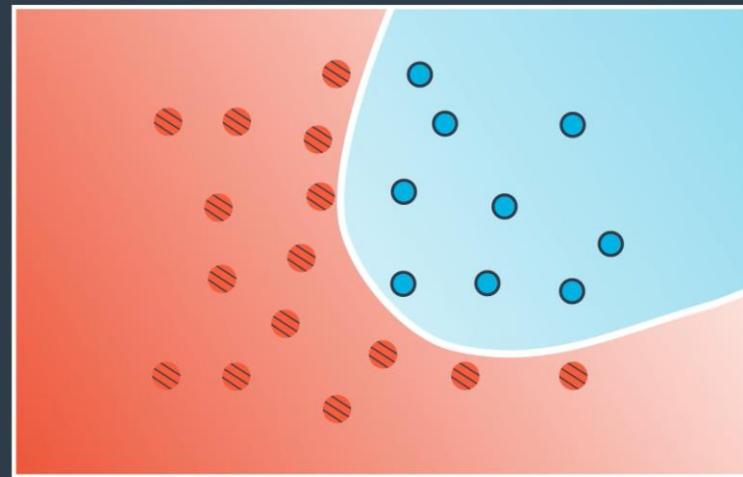
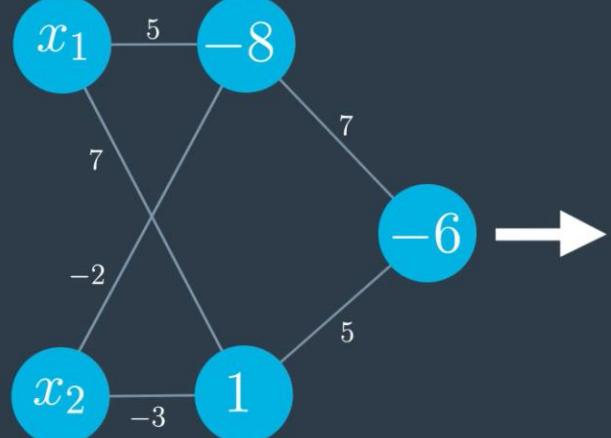


Redes Neurais - Arquitetura

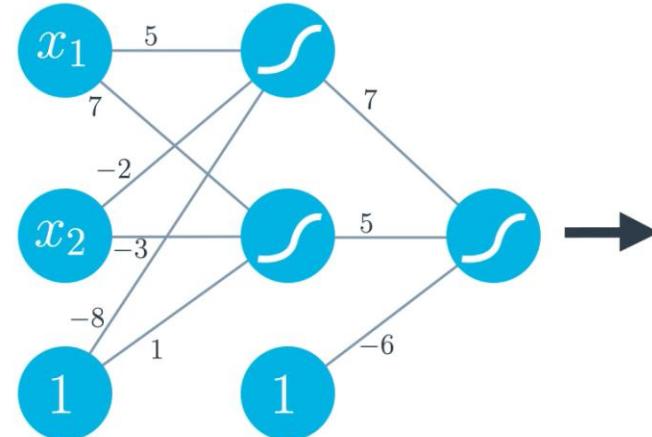
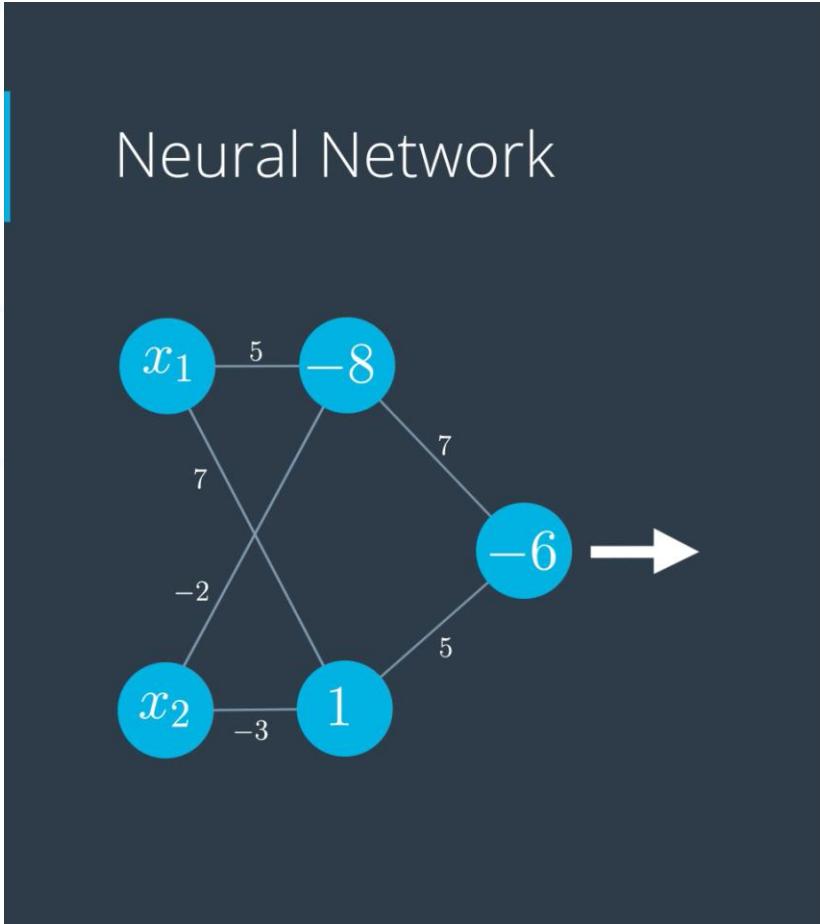


Redes Neurais - Arquitetura

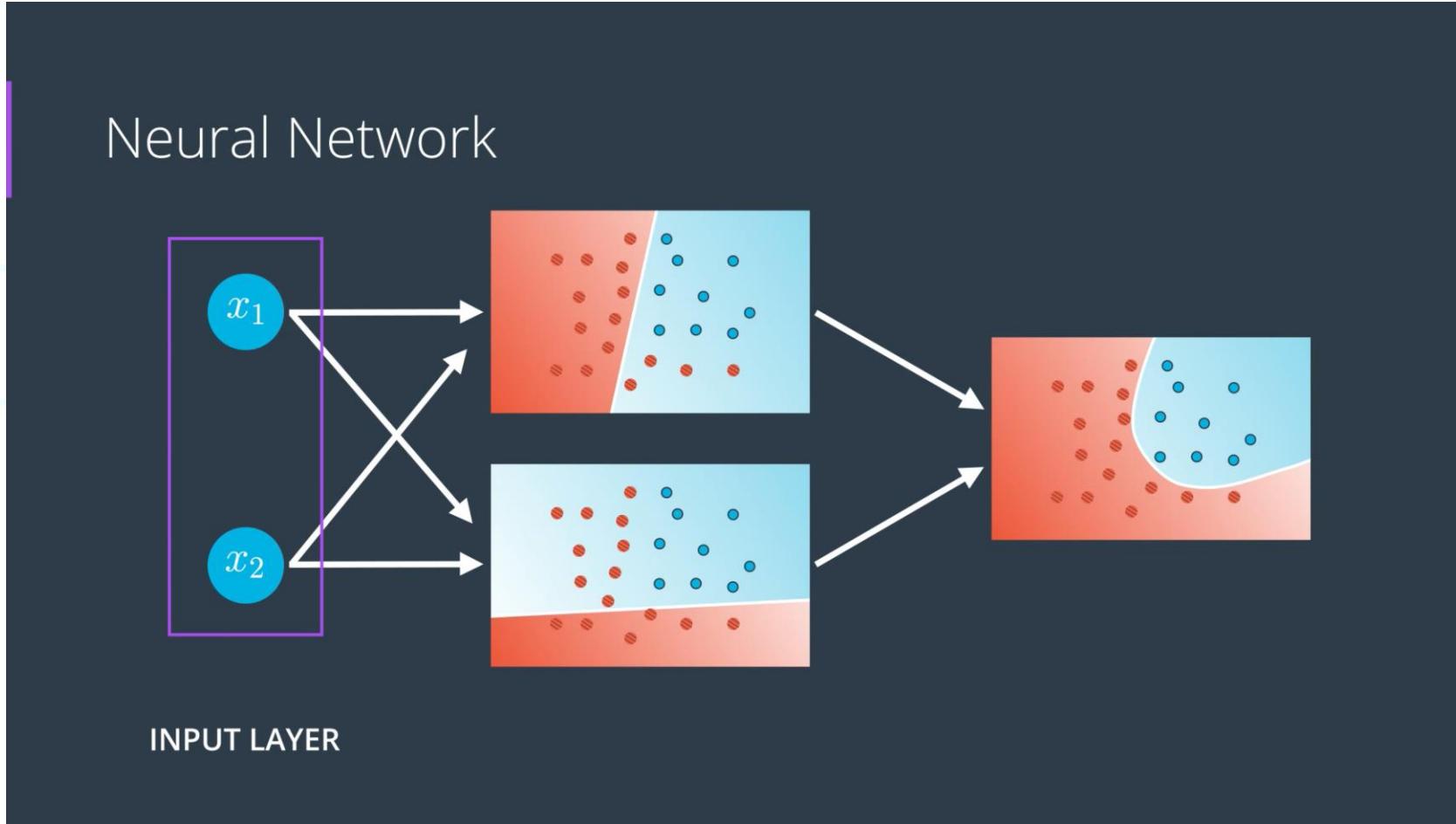
Neural Network



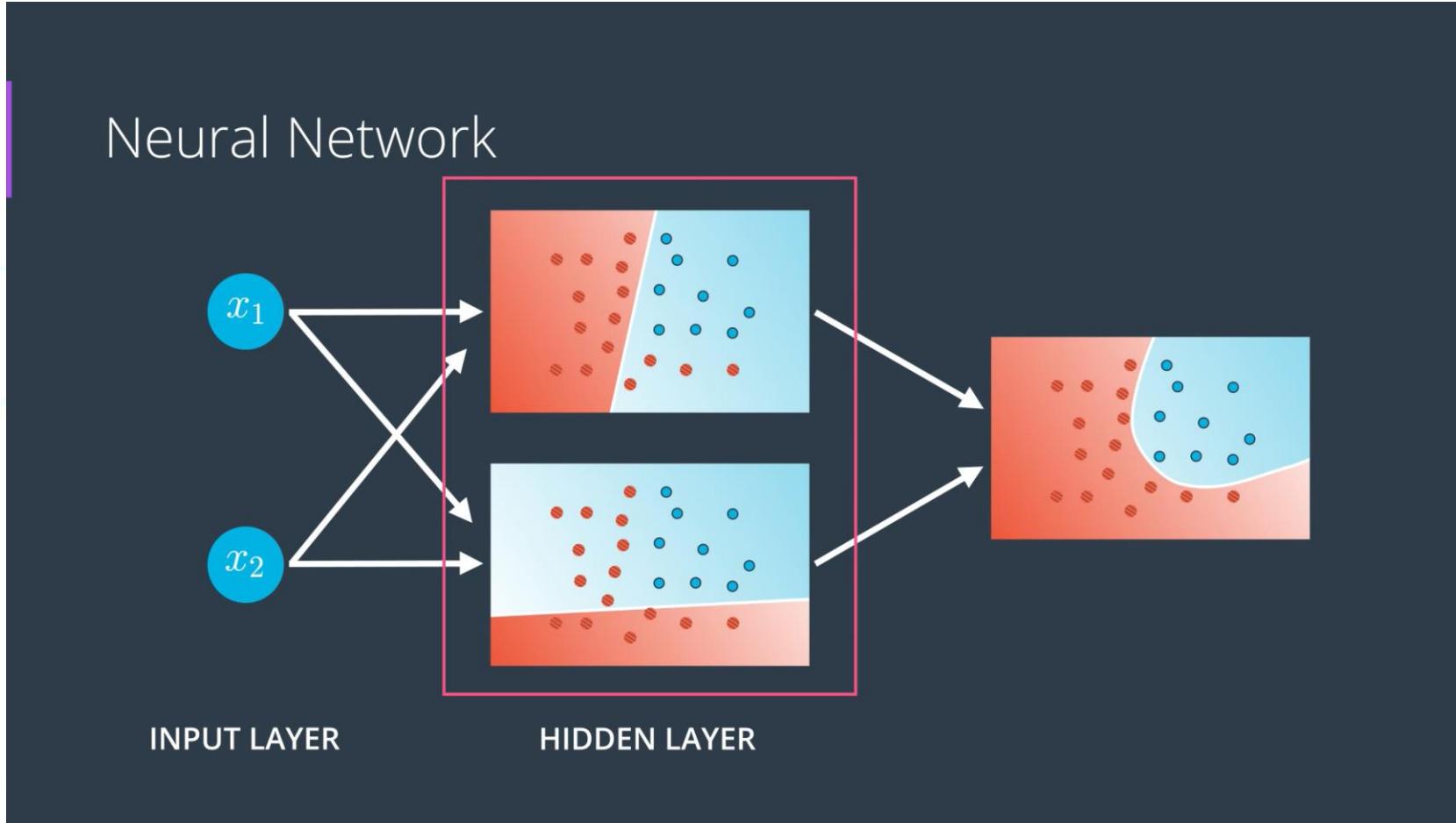
Redes Neurais - Arquitetura



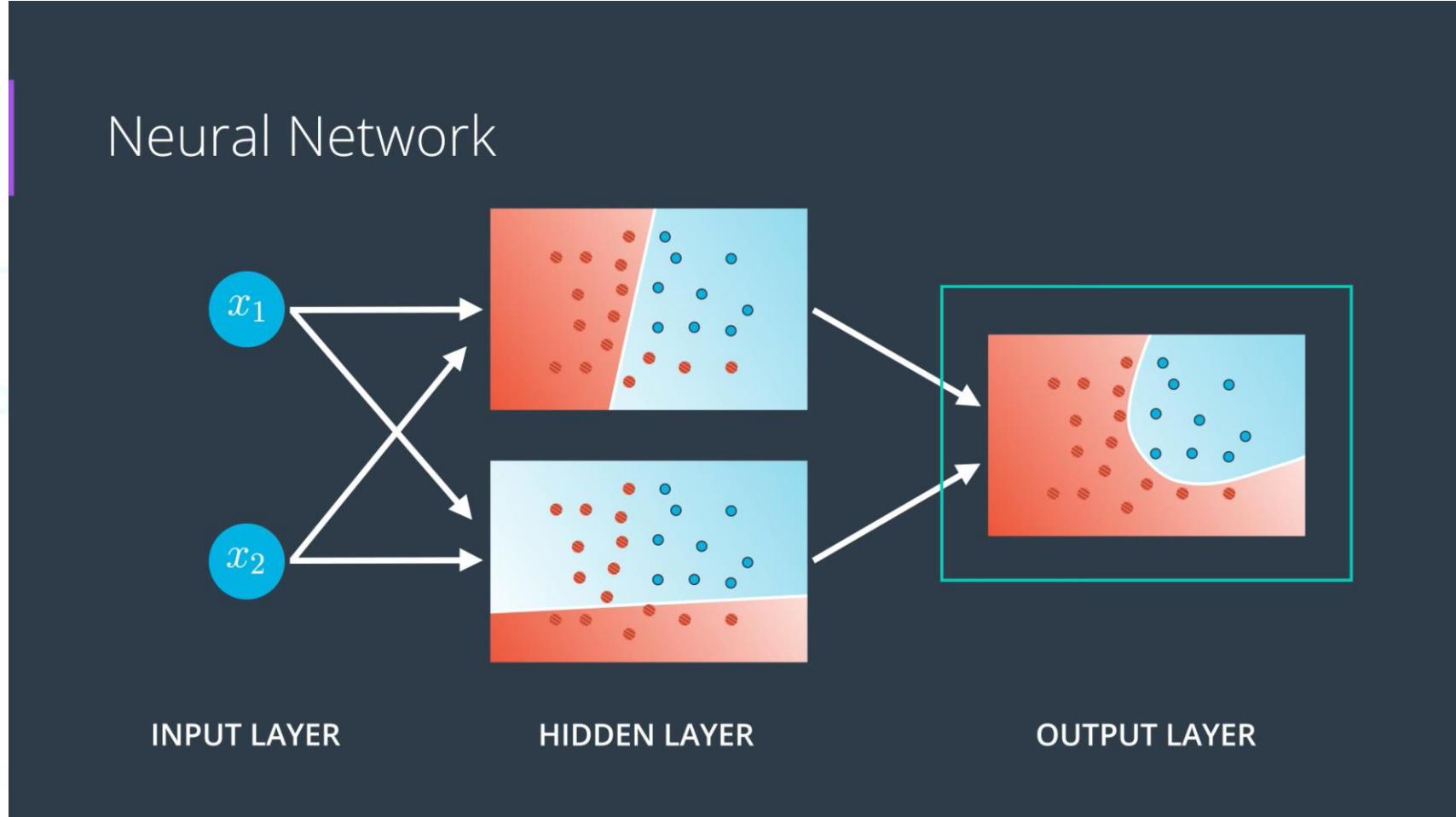
Redes Neurais - Multiple Layers



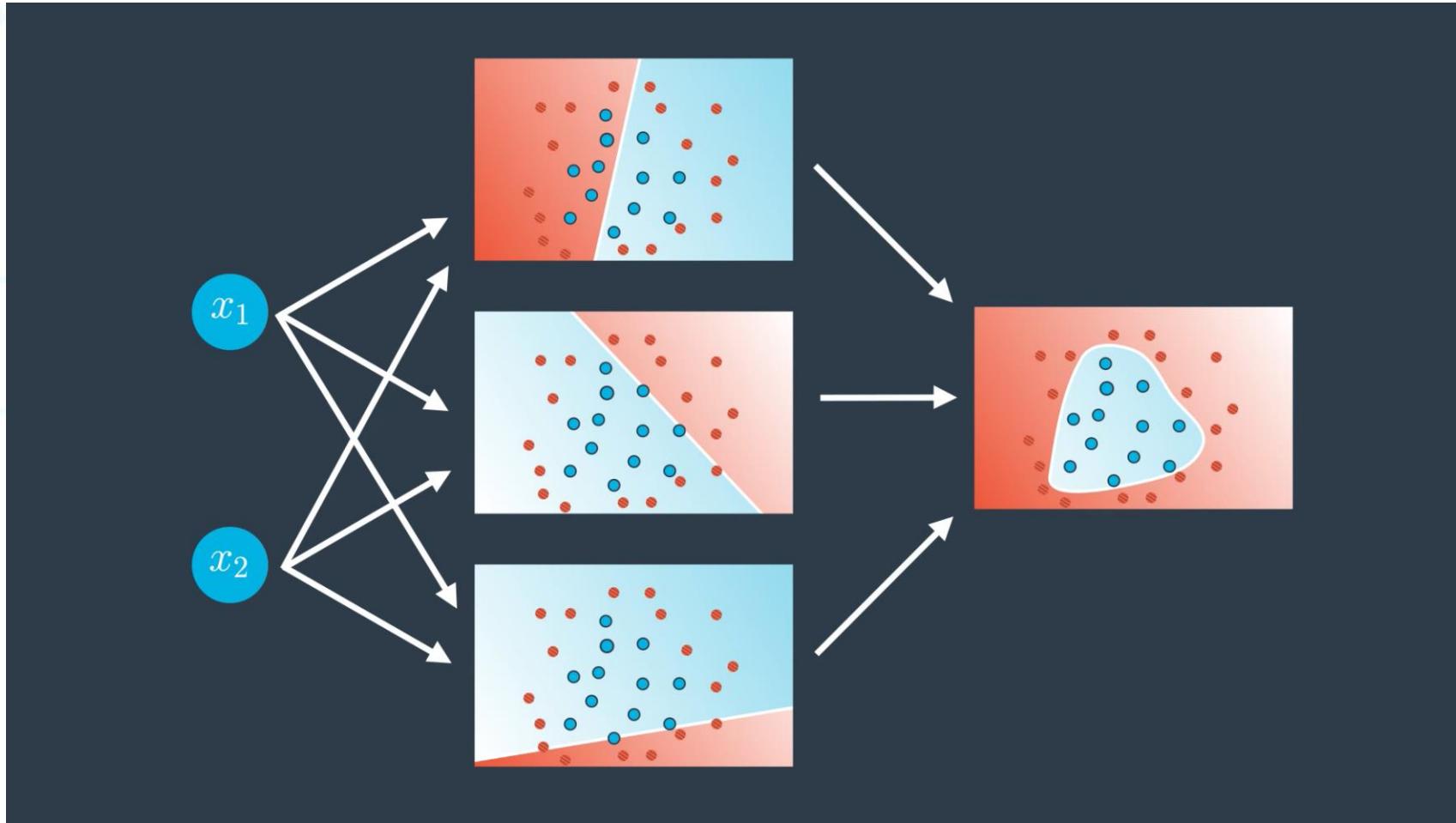
Redes Neurais - Multiple Layers



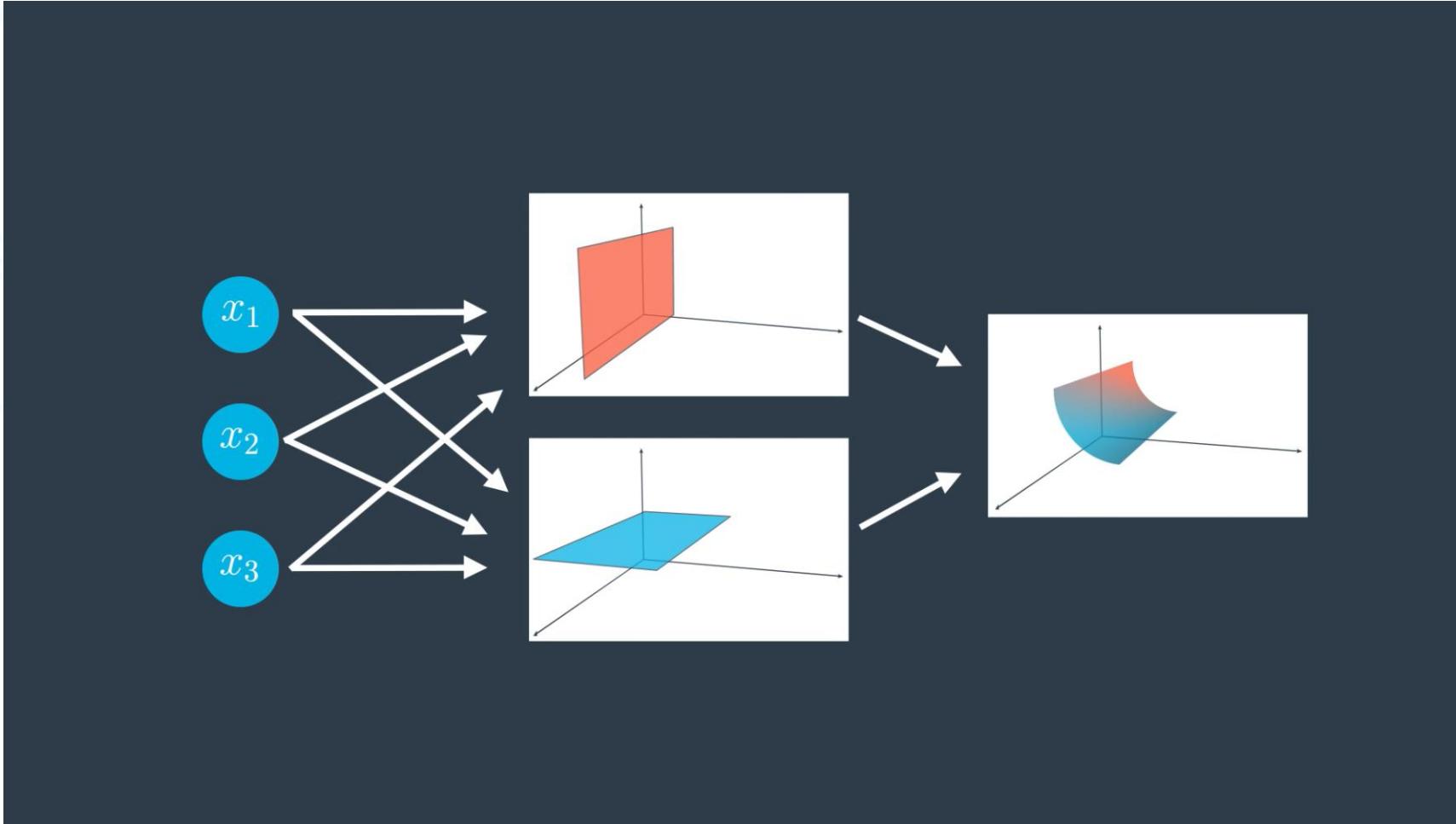
Redes Neurais - Multiple Layers



Redes Neurais - Multiple Layers

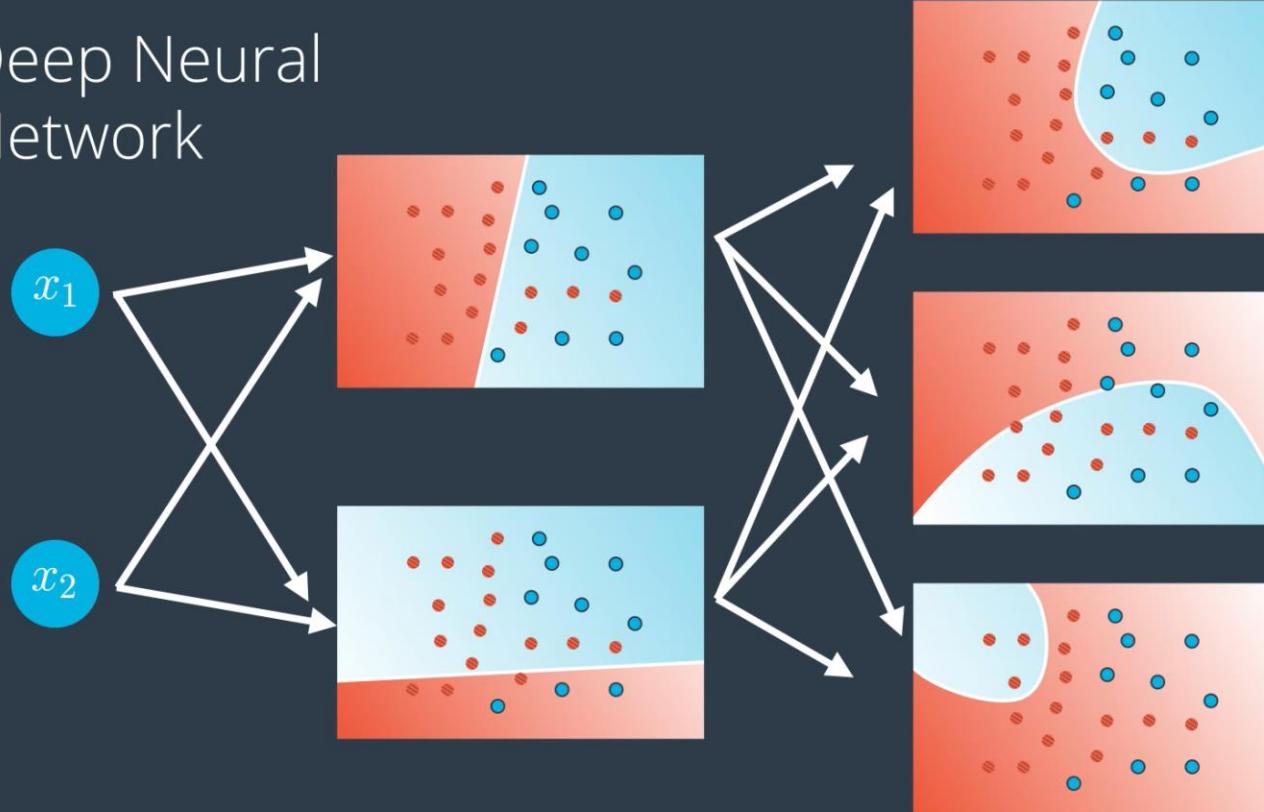


Redes Neurais - Multiple Layers

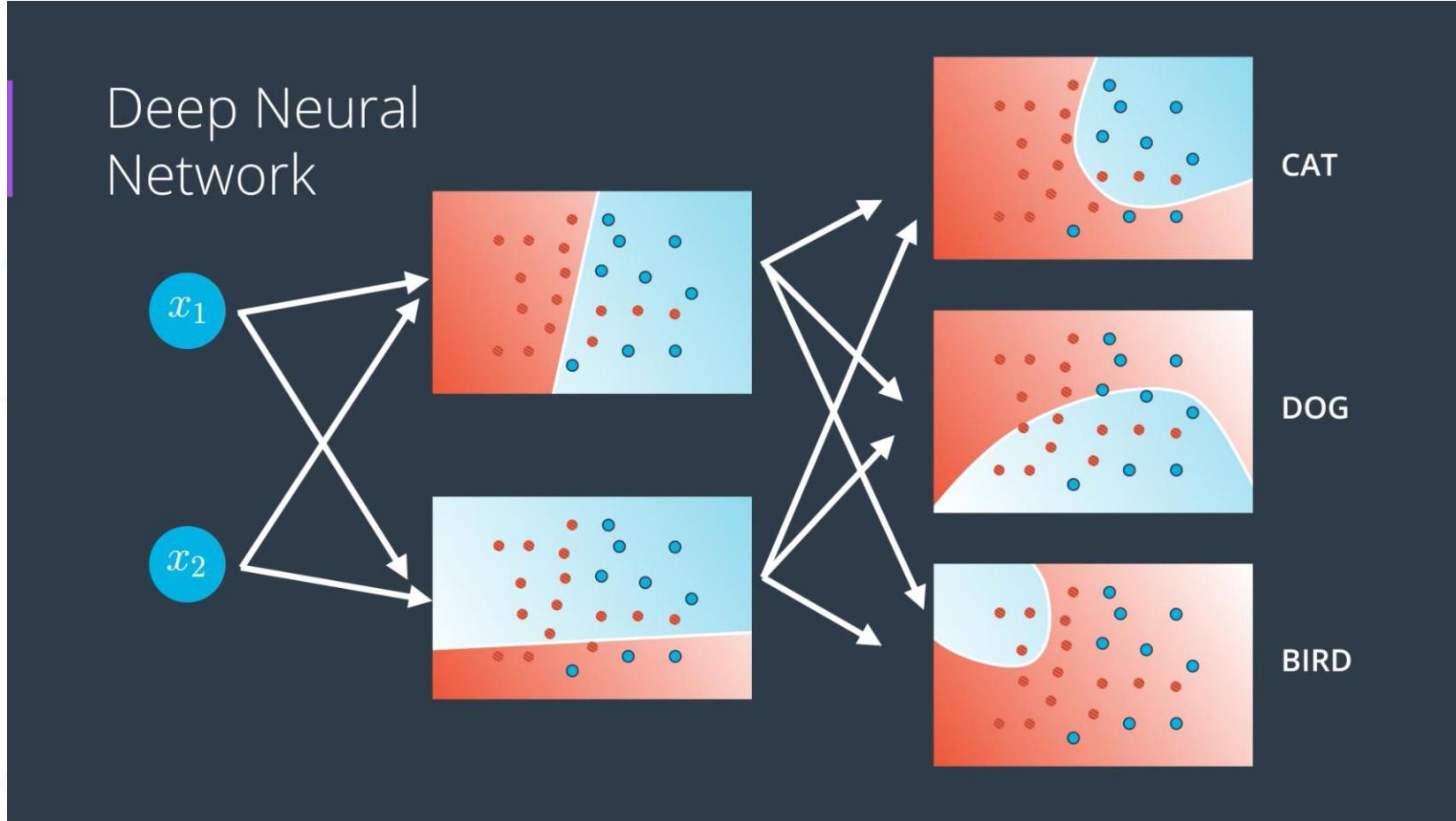


Redes Neurais - Multiple Layers

Deep Neural Network

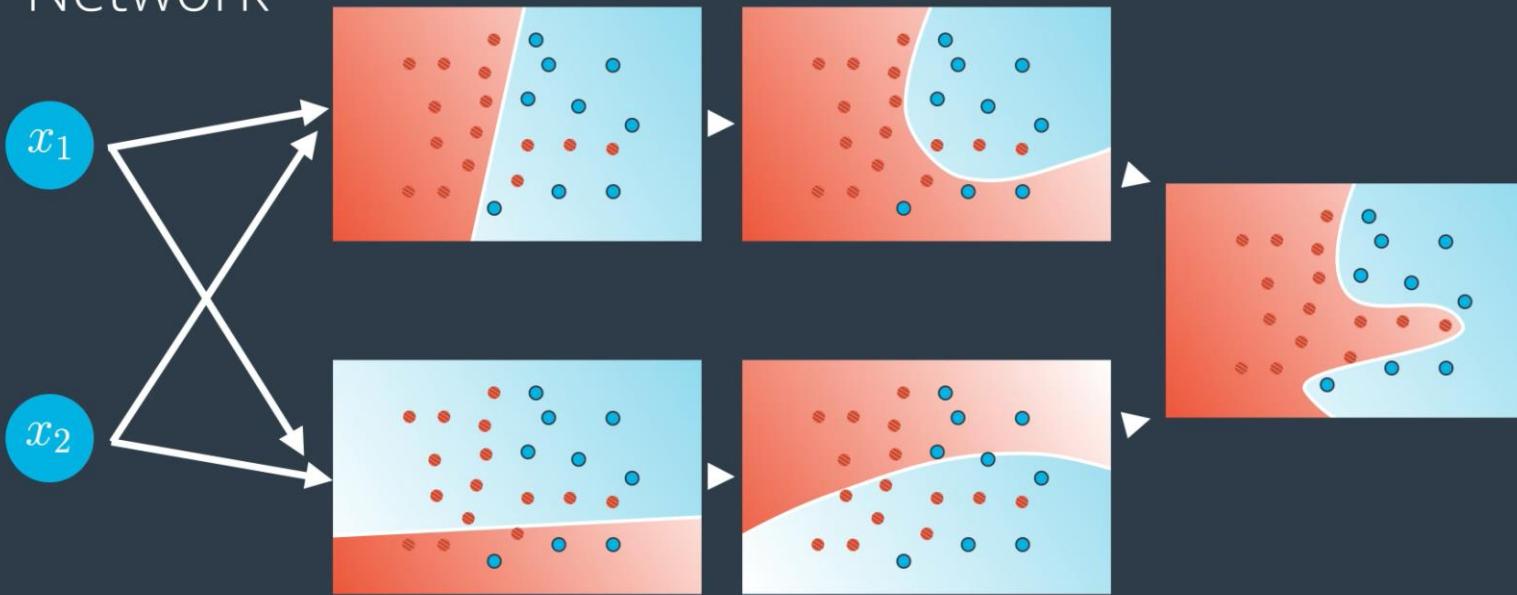


Redes Neurais - Multiple Layers



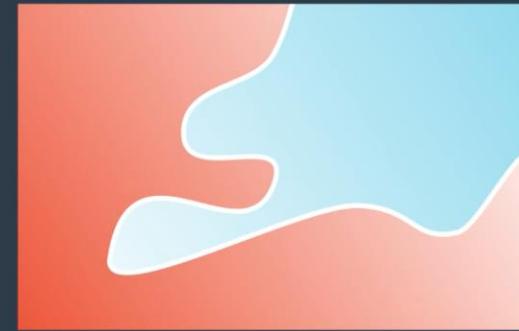
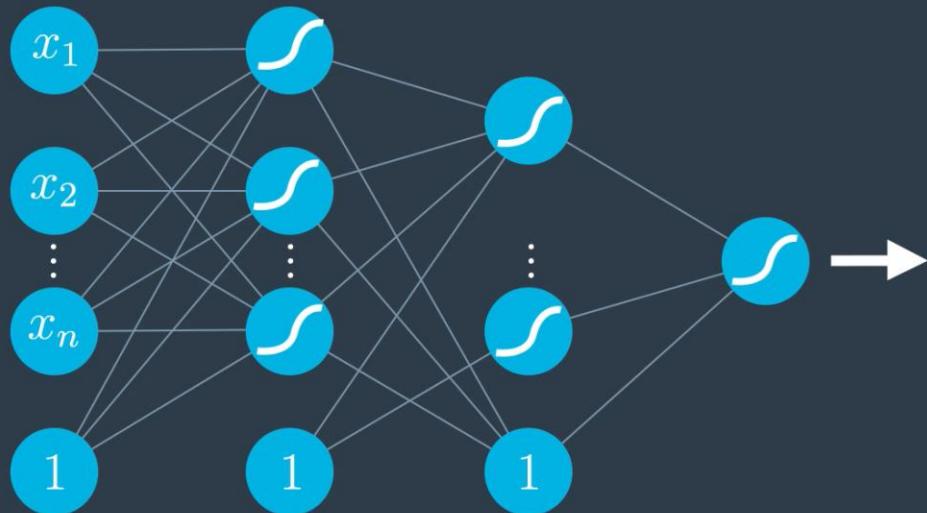
Redes Neurais - Multiple Layers

Deep Neural Network



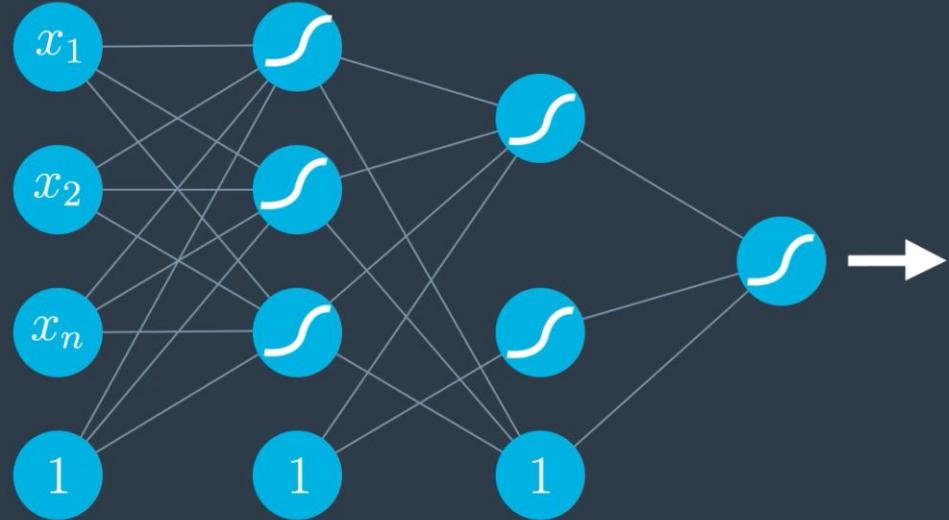
Redes Neurais - Multiple Layers

Neural Network



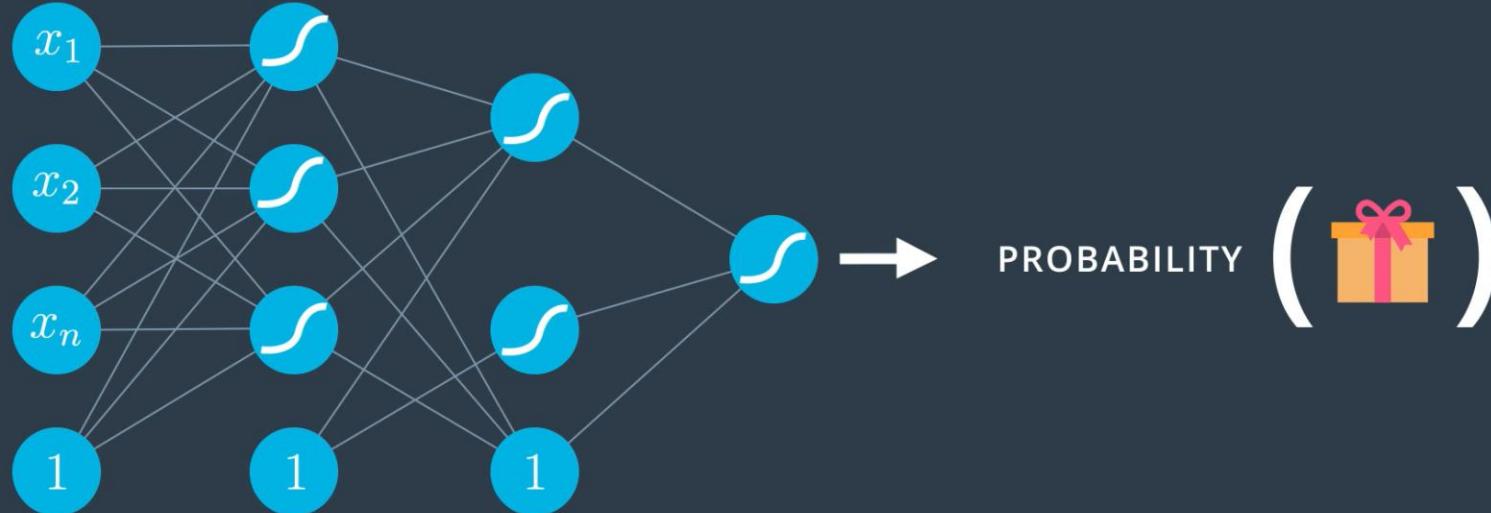
Redes Neurais - Multi-Class classification

Binary Classification



Redes Neurais - Multi-Class classification

Binary Classification



Redes Neurais - Multi-Class classification

Multi-Class
Classification

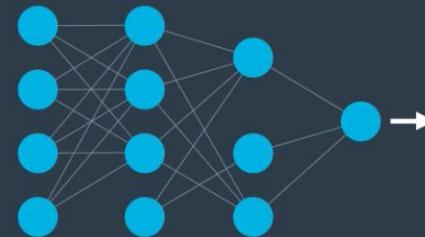
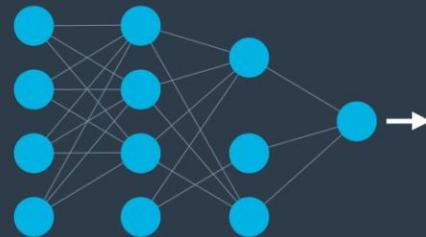
PROBABILITY 

PROBABILITY 

PROBABILITY 

Redes Neurais - Multi-Class classification

Multi-Class
Classification

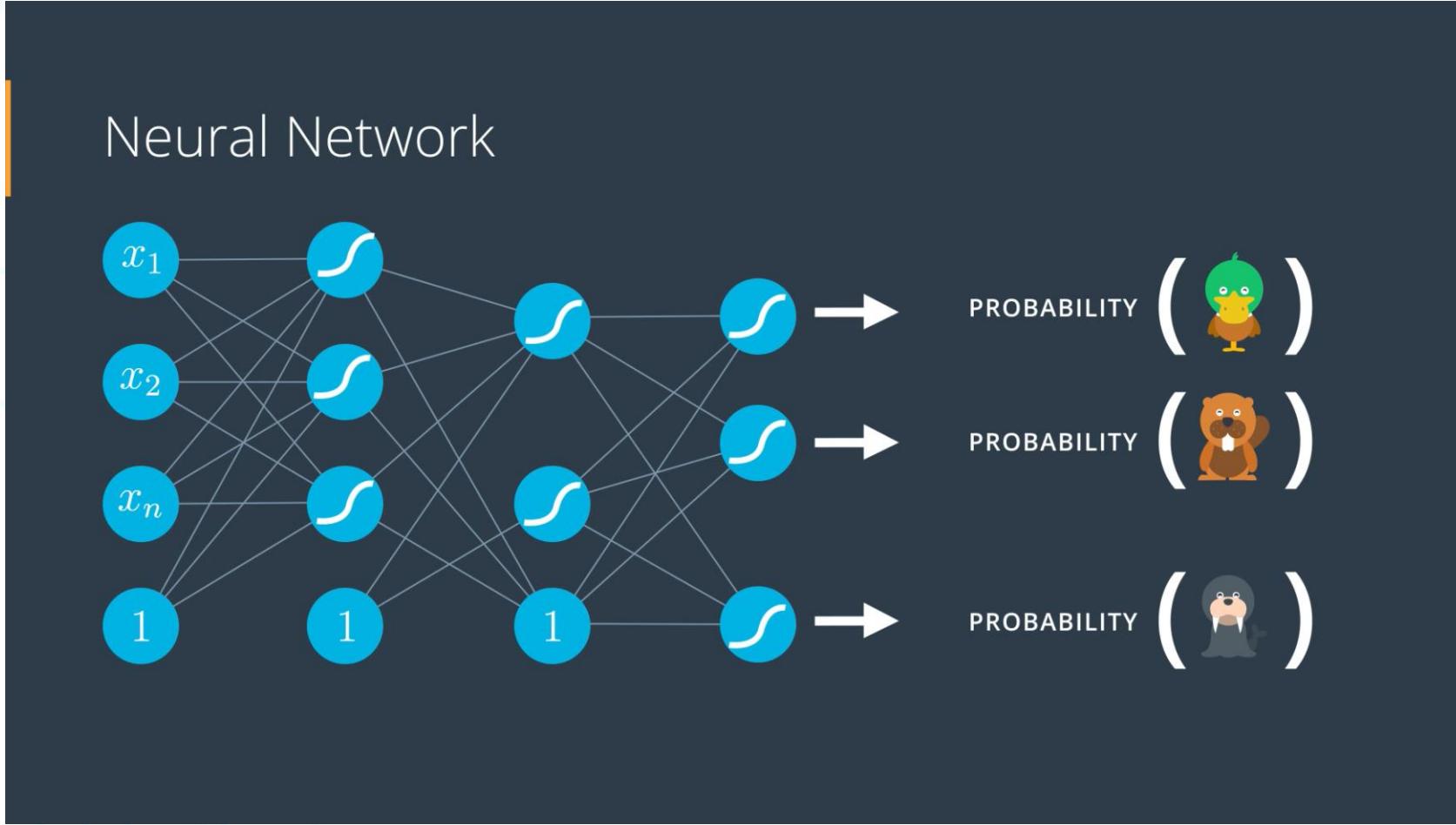


PROBABILITY 

PROBABILITY 

PROBABILITY 

Redes Neurais - Multi-Class classification



Redes Neurais - Feedforward

Perceptron

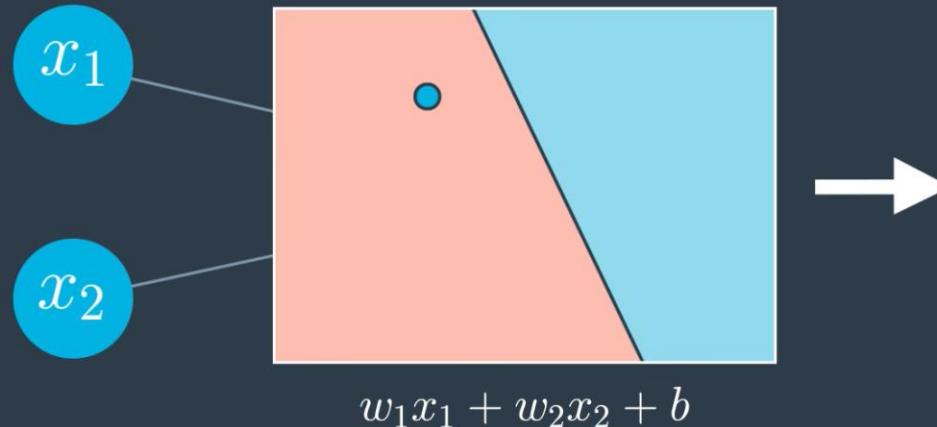


Redes Neurais - Feedforward

Perceptron

$$x = (x_1, x_2)$$

$$y = 1$$

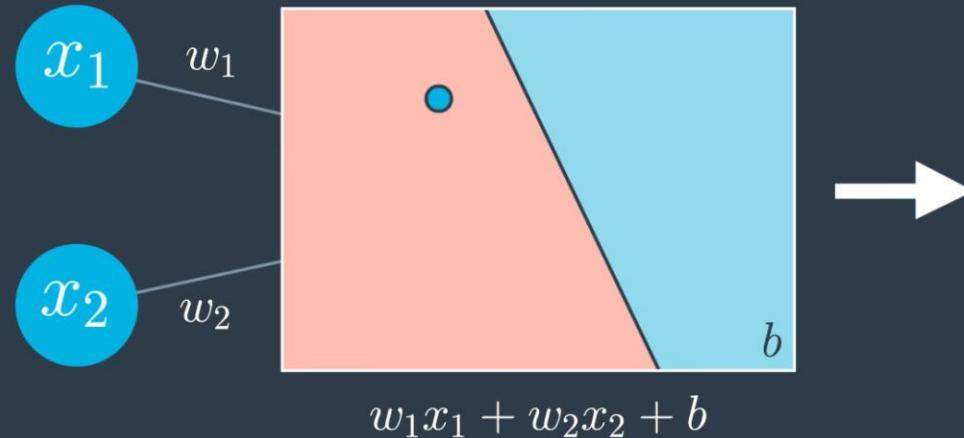


Redes Neurais - Feedforward

Perceptron

$$x = (x_1, x_2)$$

$$y = 1$$

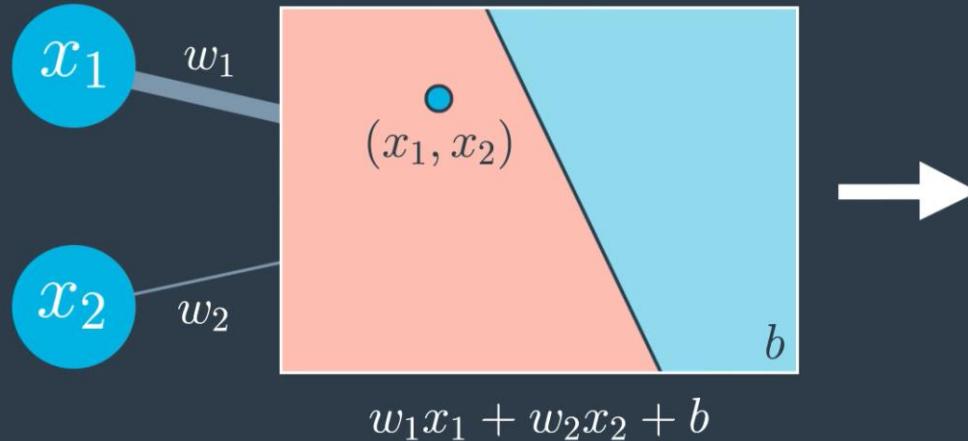


Redes Neurais - Feedforward

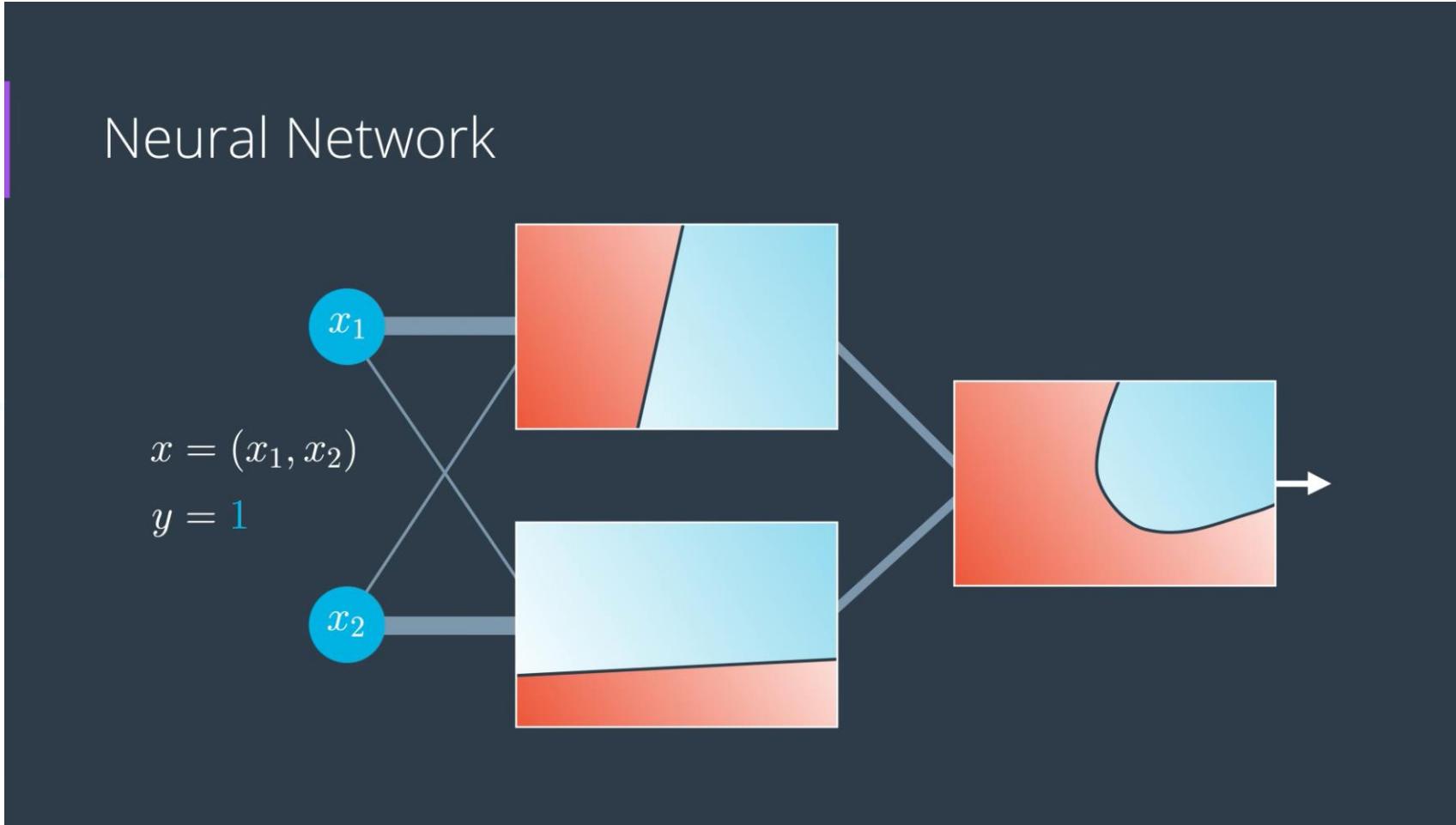
Perceptron

$$x = (x_1, x_2)$$

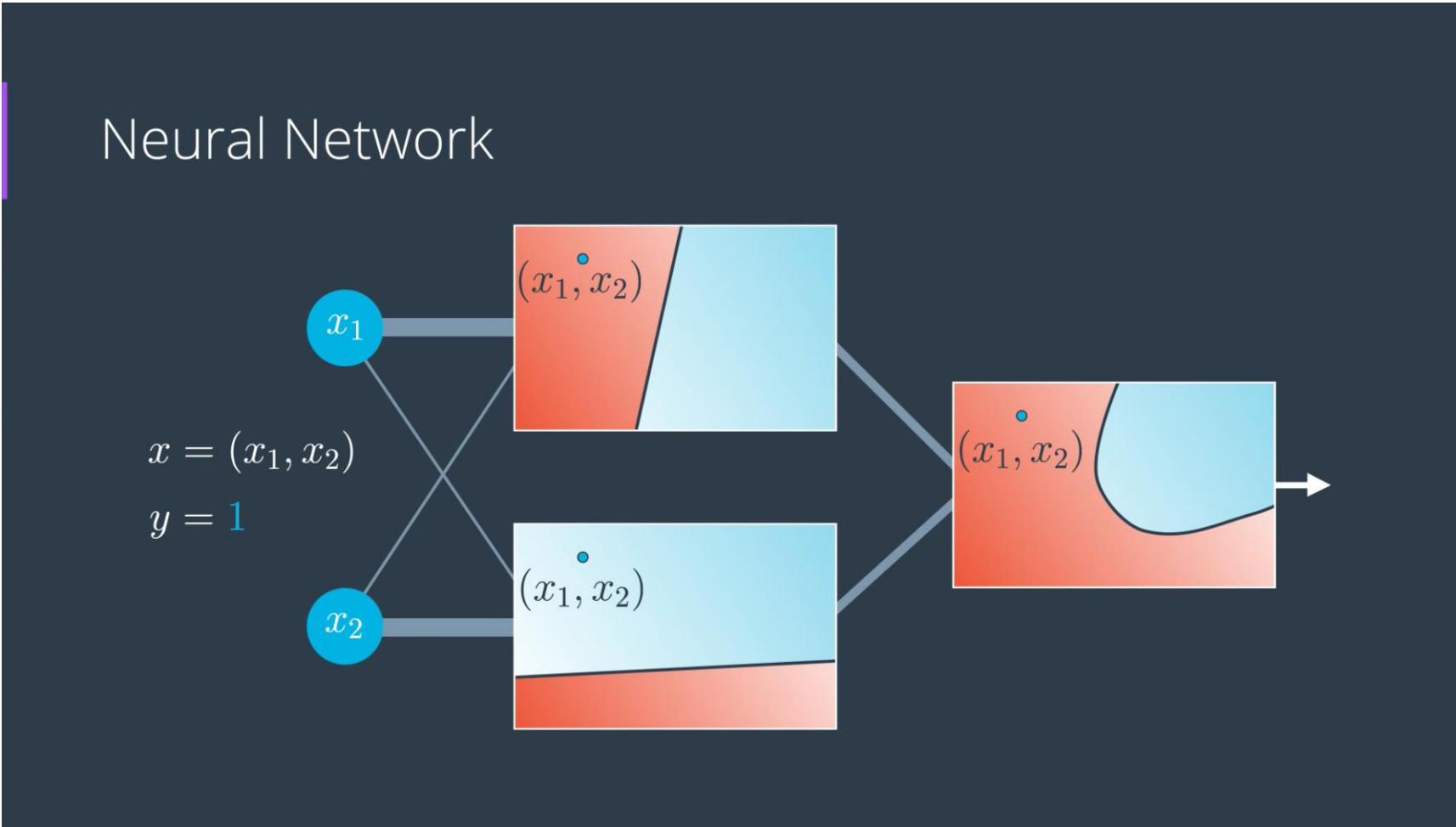
$$y = 1$$



Redes Neurais - Feedforward

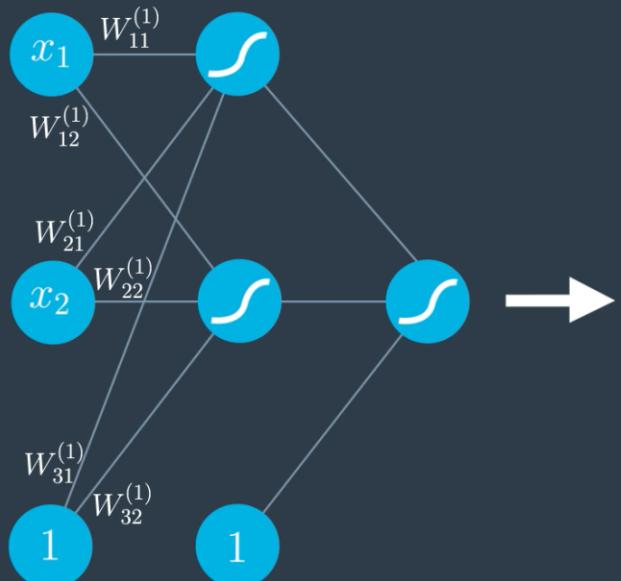


Redes Neurais - Feedforward

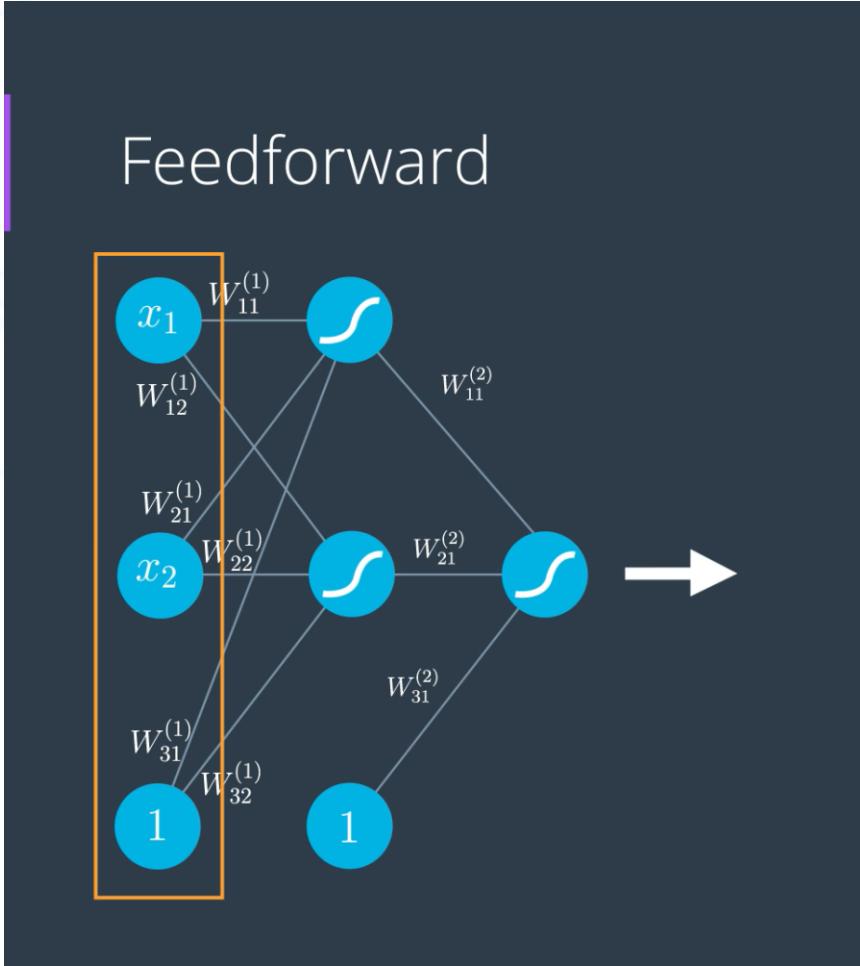


Redes Neurais - Feedforward

Feedforward



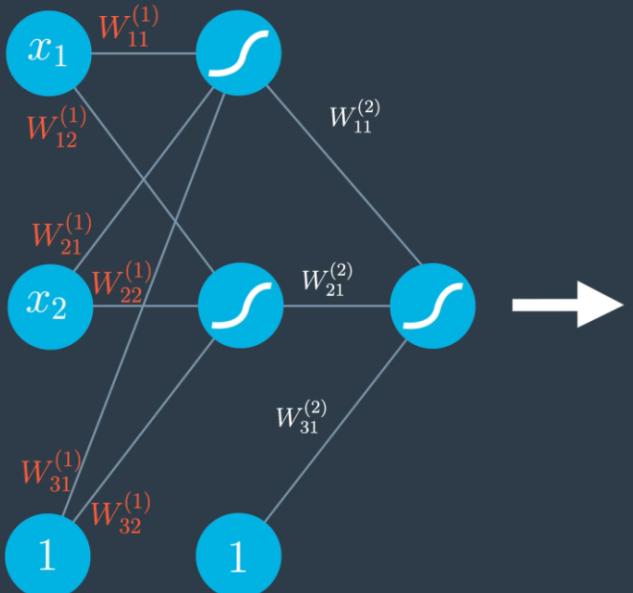
Redes Neurais - Feedforward



$$\begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

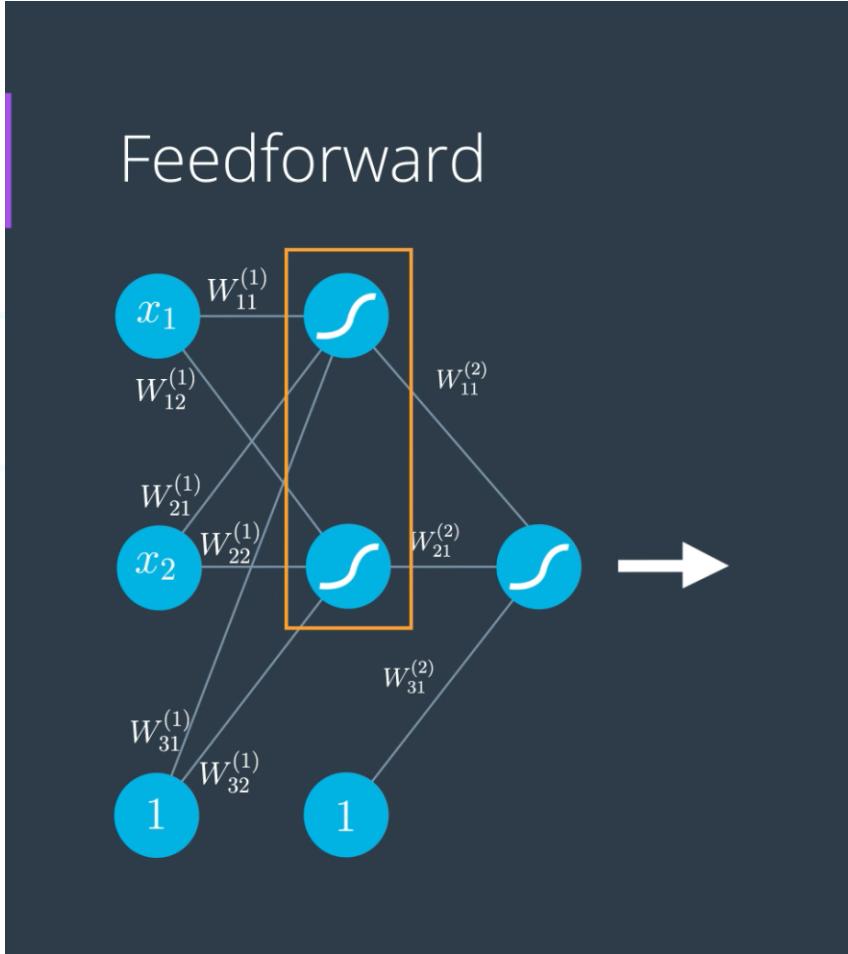
Redes Neurais - Feedforward

Feedforward



$$\begin{pmatrix} W_{11}^{(1)} & W_{12}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} \\ W_{31}^{(1)} & W_{32}^{(1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

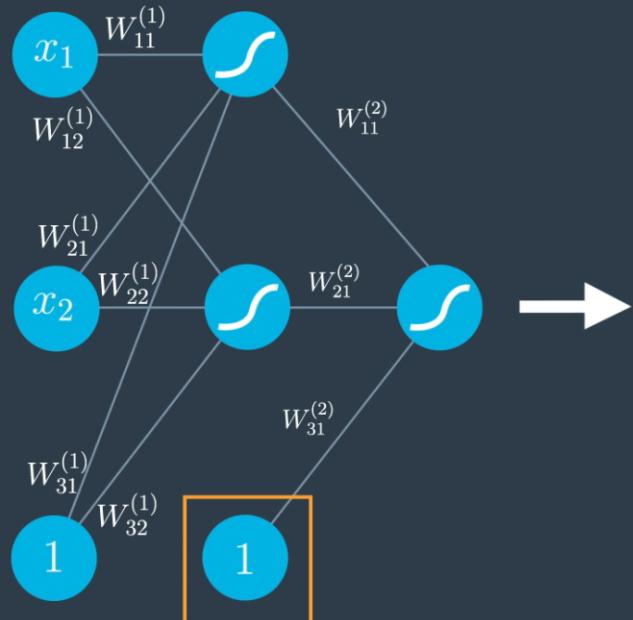
Redes Neurais - Feedforward



$$\sigma \begin{pmatrix} W_{11}^{(1)} & W_{12}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} \\ W_{31}^{(1)} & W_{32}^{(1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

Redes Neurais - Feedforward

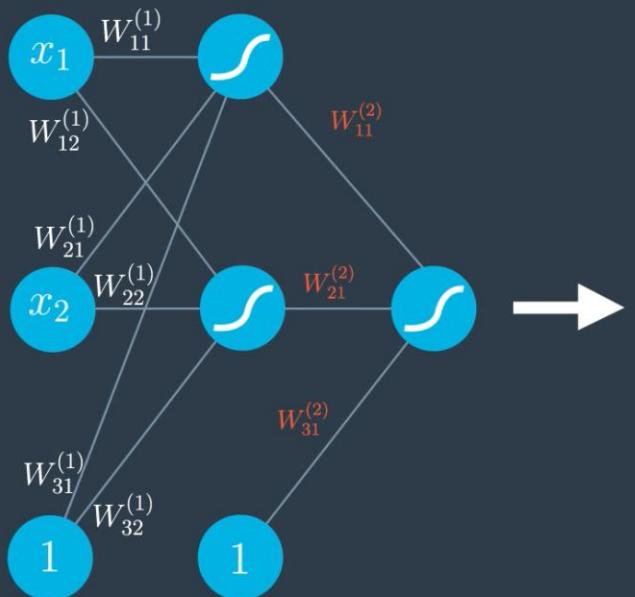
Feedforward



$$\sigma \begin{pmatrix} W_{11}^{(1)} & W_{12}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} \\ W_{31}^{(1)} & W_{32}^{(1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

Redes Neurais - Feedforward

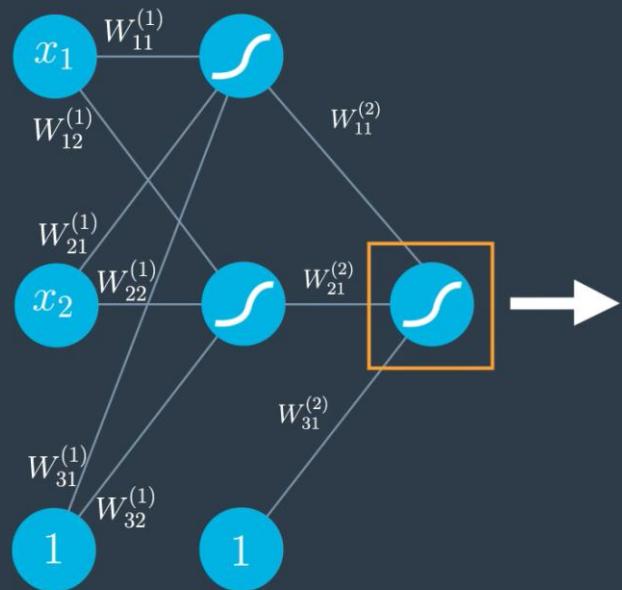
Feedforward



$$\begin{pmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{31}^{(2)} \end{pmatrix} \sigma \begin{pmatrix} W_{11}^{(1)} & W_{12}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} \\ W_{31}^{(1)} & W_{32}^{(1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

Redes Neurais - Feedforward

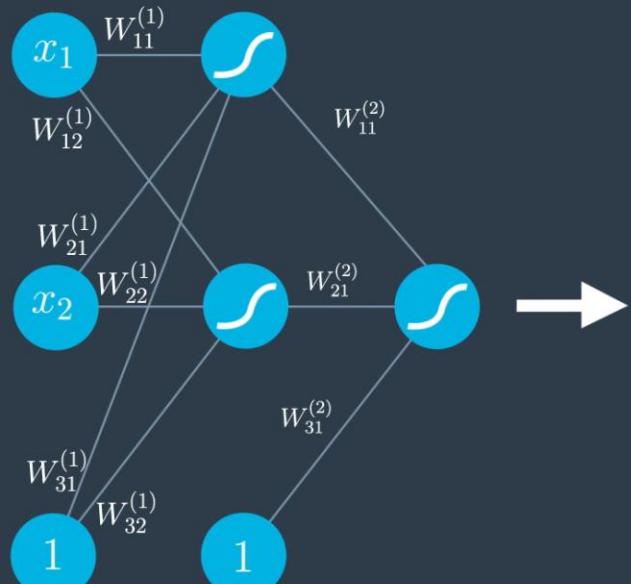
Feedforward



$$\sigma \begin{pmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{31}^{(2)} \end{pmatrix} \sigma \begin{pmatrix} W_{11}^{(1)} & W_{12}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} \\ W_{31}^{(1)} & W_{32}^{(1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

Redes Neurais - Feedforward

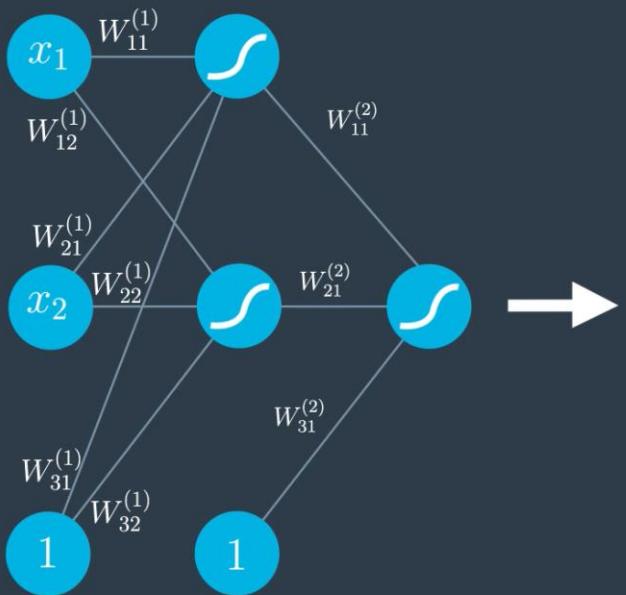
Feedforward



$$\hat{y} = \sigma \begin{pmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{31}^{(2)} \end{pmatrix} \sigma \begin{pmatrix} W_{11}^{(1)} & W_{12}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} \\ W_{31}^{(1)} & W_{32}^{(1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

Redes Neurais - Feedforward

Feedforward

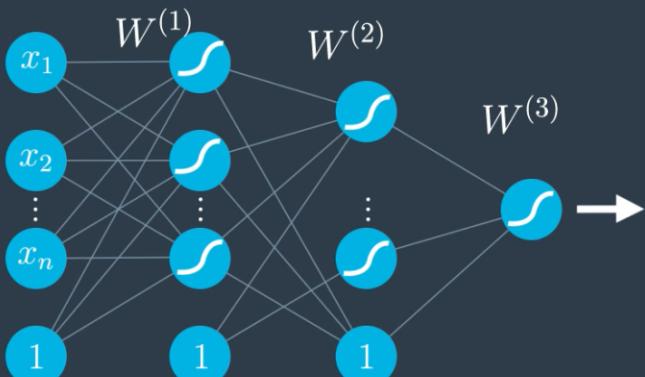


$$\hat{y} = \sigma \begin{pmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{31}^{(2)} \end{pmatrix} \sigma \begin{pmatrix} W_{11}^{(1)} & W_{12}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} \\ W_{31}^{(1)} & W_{32}^{(1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

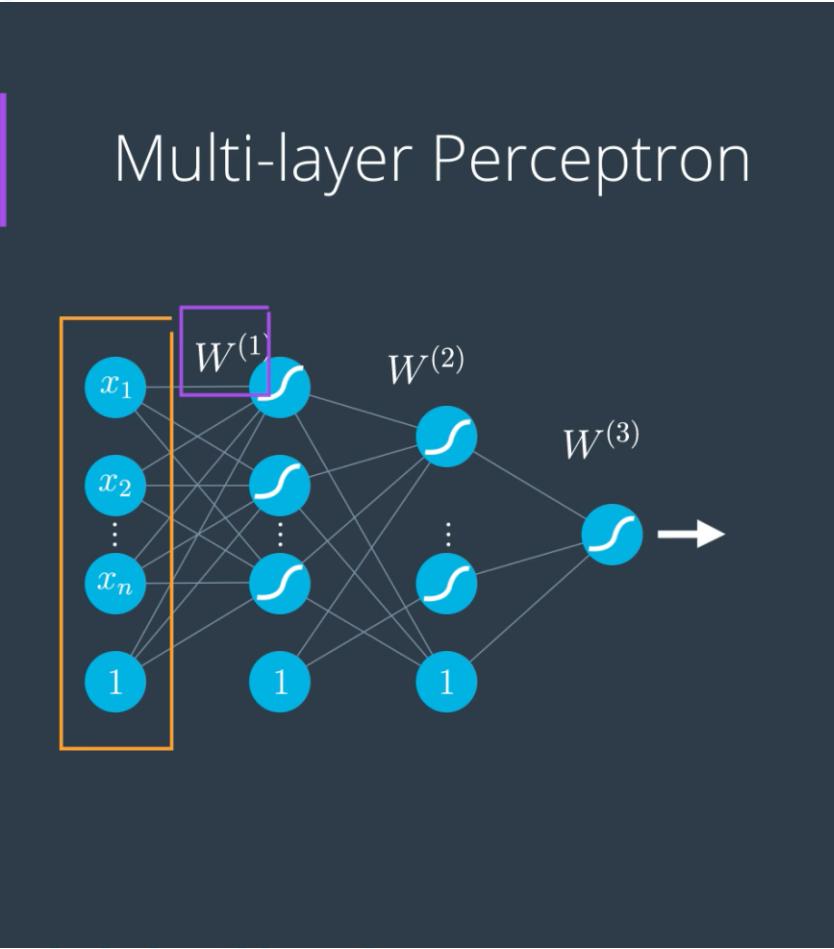
$$\hat{y} = \sigma \circ W^{(2)} \circ \sigma \circ W^{(1)}(x)$$

Redes Neurais - Feedforward

Multi-layer Perceptron



Redes Neurais - Feedforward



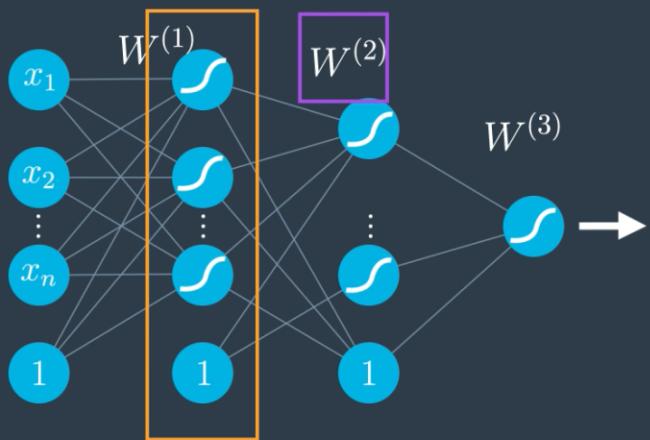
PREDICTION

$$\hat{y} =$$

$$\sigma \circ W^{(1)}(x)$$

Redes Neurais - Feedforward

Multi-layer Perceptron

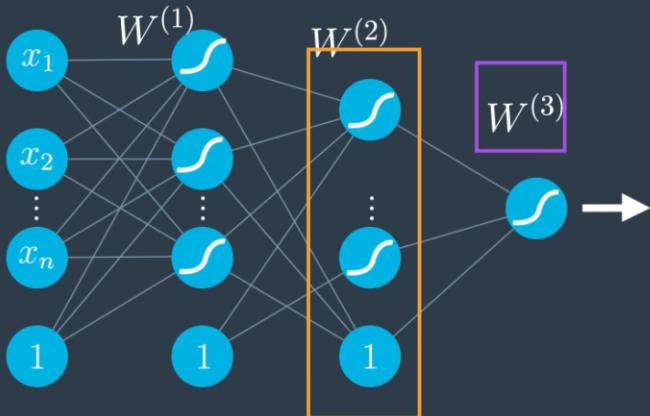


PREDICTION

$$\hat{y} = \sigma \circ W^{(2)} \circ \sigma \circ W^{(1)}(x)$$

Redes Neurais - Feedforward

Multi-layer Perceptron

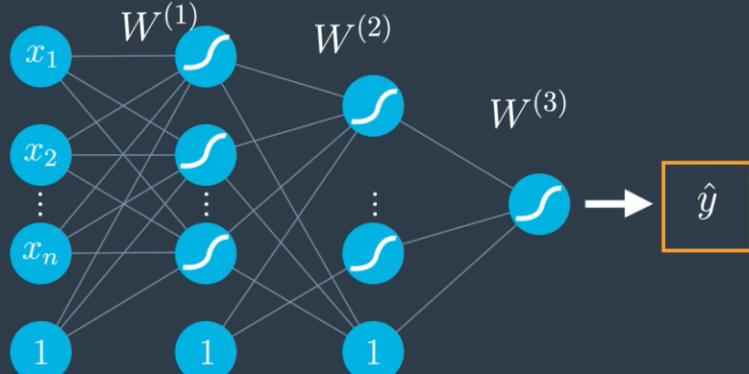


PREDICTION

$$\hat{y} = \sigma \circ W^{(3)} \circ \sigma \circ W^{(2)} \circ \sigma \circ W^{(1)}(x)$$

Redes Neurais - Feedforward

Multi-layer Perceptron

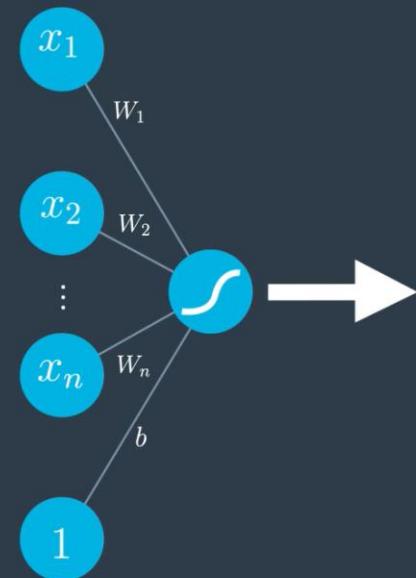


PREDICTION

$$\hat{y} = \sigma \circ W^{(3)} \circ \sigma \circ W^{(2)} \circ \sigma \circ W^{(1)}(x)$$

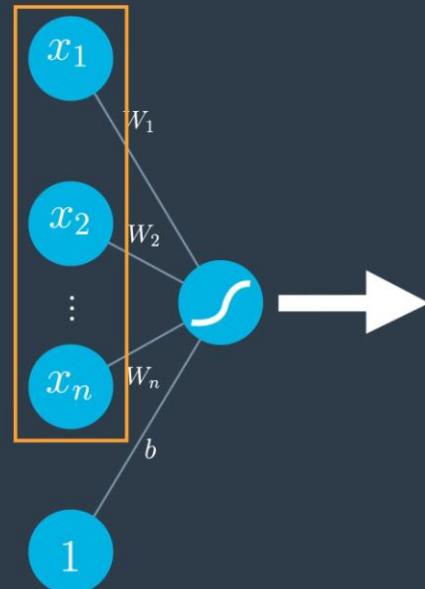
Redes Neurais - Error Function

Perceptron



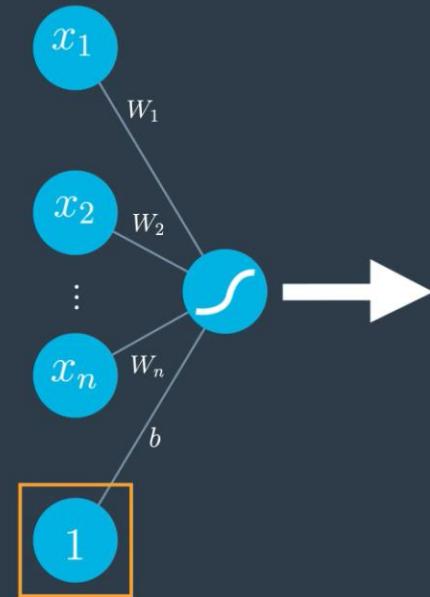
Redes Neurais - Error Function

Perceptron



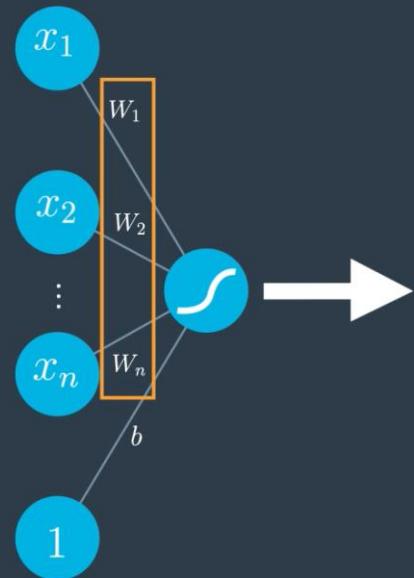
Redes Neurais - Error Function

Perceptron



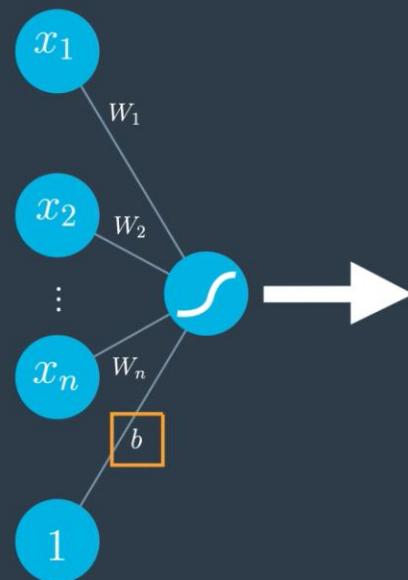
Redes Neurais - Error Function

Perceptron



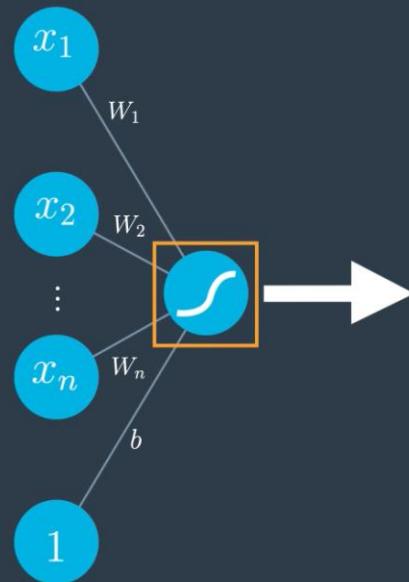
Redes Neurais - Error Function

Perceptron



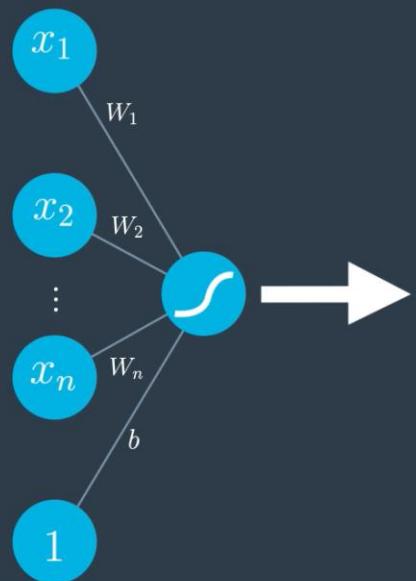
Redes Neurais - Error Function

Perceptron



Redes Neurais - Error Function

Perceptron

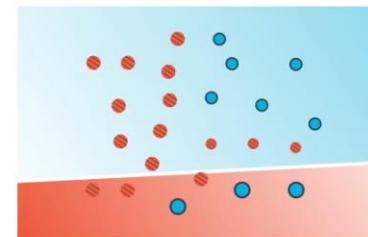


PREDICTION

$$\hat{y} = \sigma(Wx + b)$$

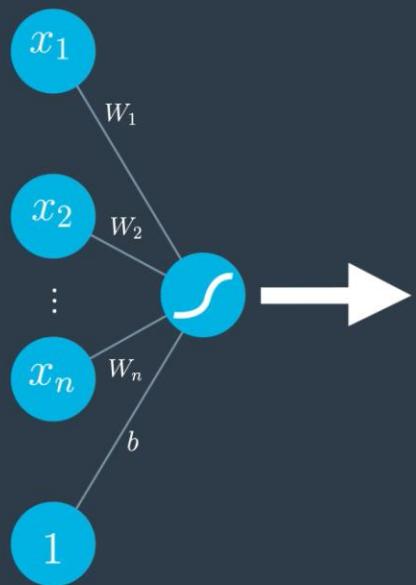
ERROR FUNCTION

$$E(W) = -\frac{1}{m} \sum_{i=1}^m y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i)$$



Redes Neurais - Error Function

Perceptron

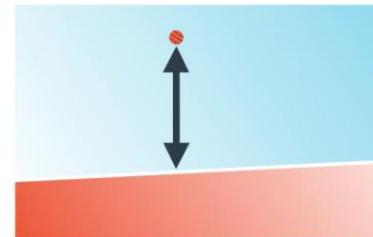


PREDICTION

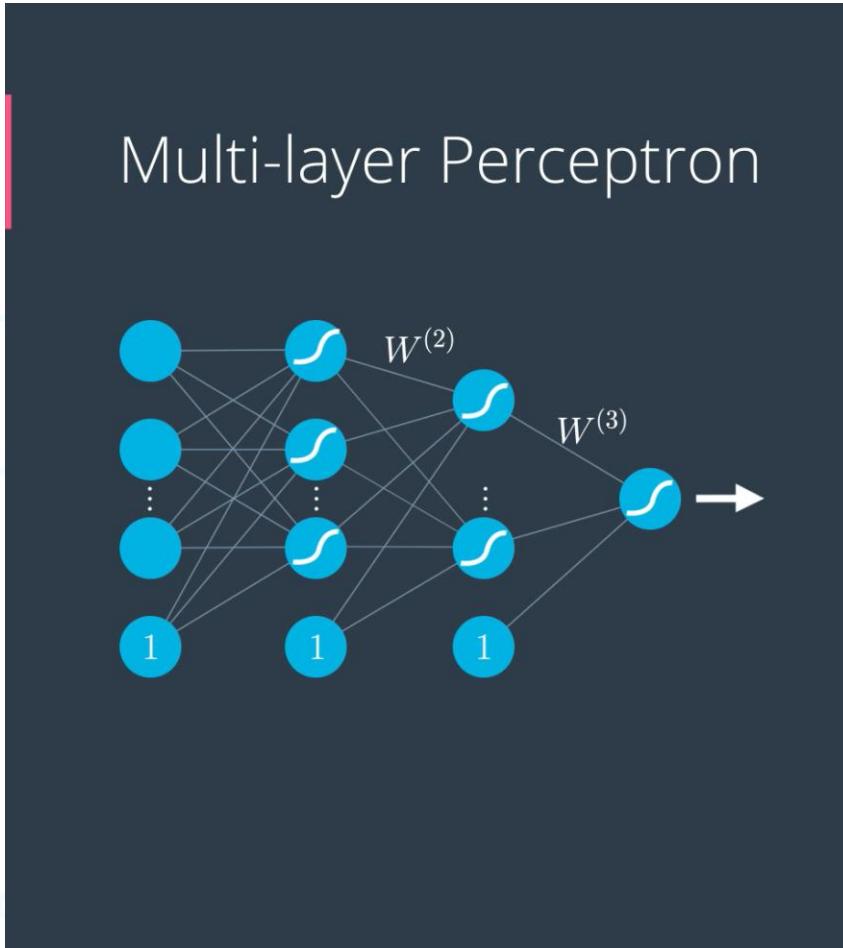
$$\hat{y} = \sigma(Wx + b)$$

ERROR FUNCTION

$$E(W) = -\frac{1}{m} \sum_{i=1}^m y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i)$$



Redes Neurais - Error Function

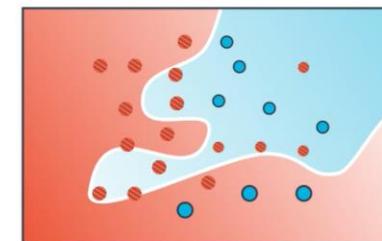


PREDICTION

$$\hat{y} = \sigma \circ W^{(3)} \circ \sigma \circ W^{(2)} \circ \sigma \circ W^{(1)}(x)$$

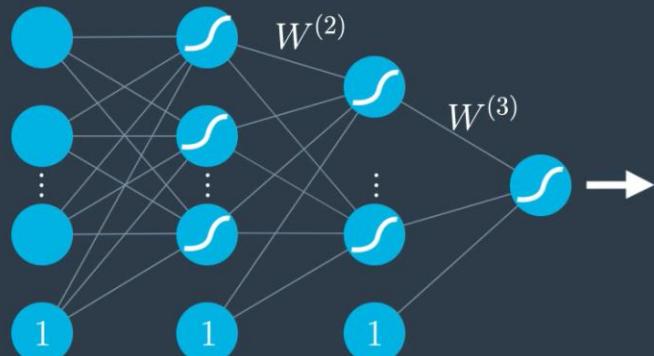
ERROR FUNCTION

$$E(W) = -\frac{1}{m} \sum_{i=1}^m y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i)$$



Redes Neurais - Error Function

Multi-layer Perceptron

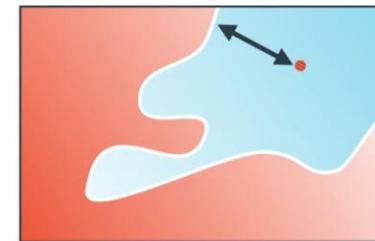


PREDICTION

$$\hat{y} = \sigma \circ W^{(3)} \circ \sigma \circ W^{(2)} \circ \sigma \circ W^{(1)}(x)$$

ERROR FUNCTION

$$E(W) = -\frac{1}{m} \sum_{i=1}^m y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i)$$

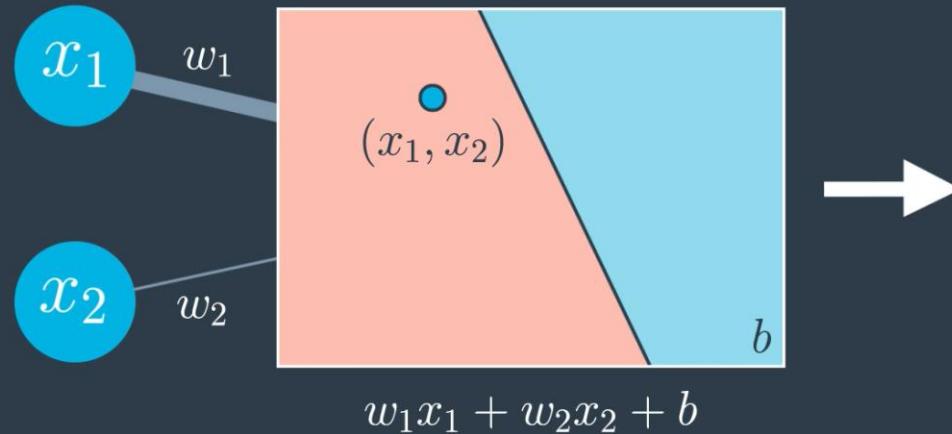


Redes Neurais - Backpropagation

FeedForward

$$x = (x_1, x_2)$$

$$y = 1$$

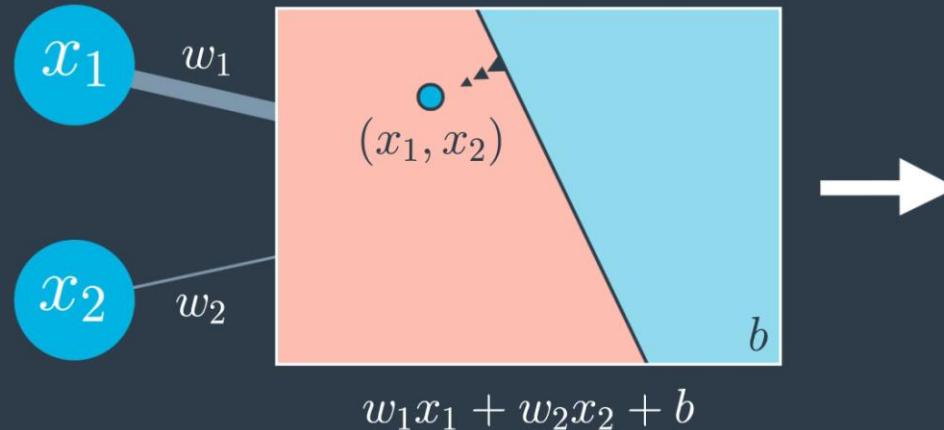


Redes Neurais - Backpropagation

Backpropagation

$$x = (x_1, x_2)$$

$$y = 1$$

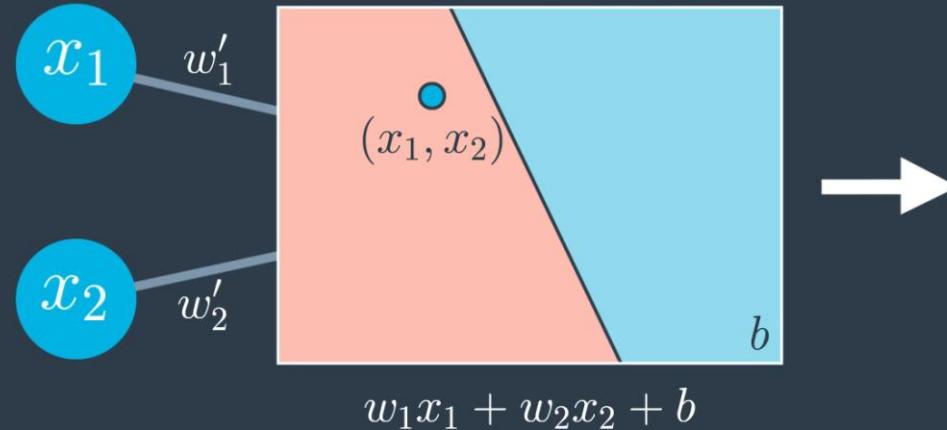


Redes Neurais - Backpropagation

Backpropagation

$$x = (x_1, x_2)$$

$$y = 1$$



Redes Neurais - Backpropagation

Gradient Descent

$$E(W) = -\frac{1}{m} \sum_{i=1}^m y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i)$$



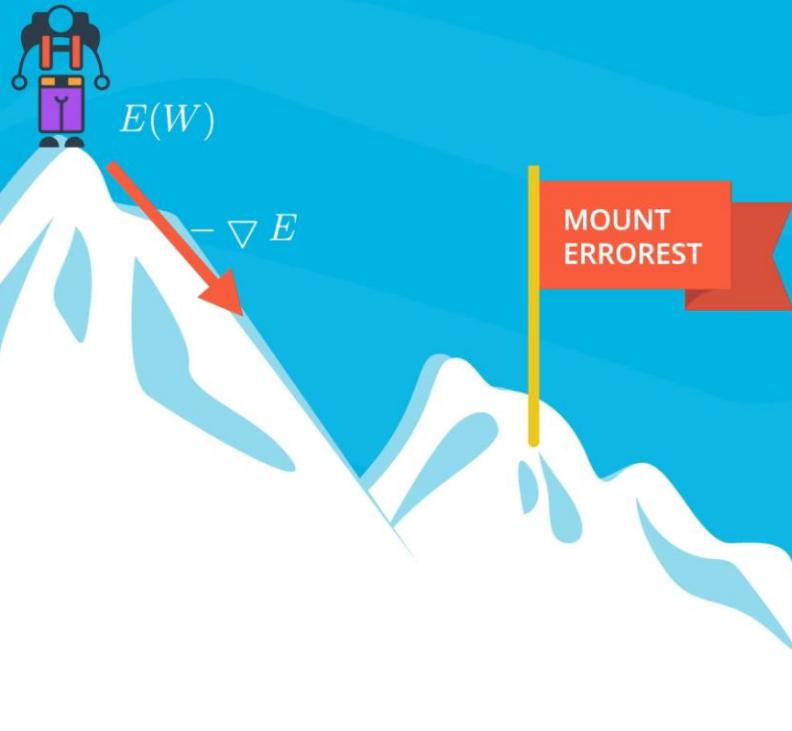
MOUNT
ERROREST



Redes Neurais - Backpropagation

Gradient Descent

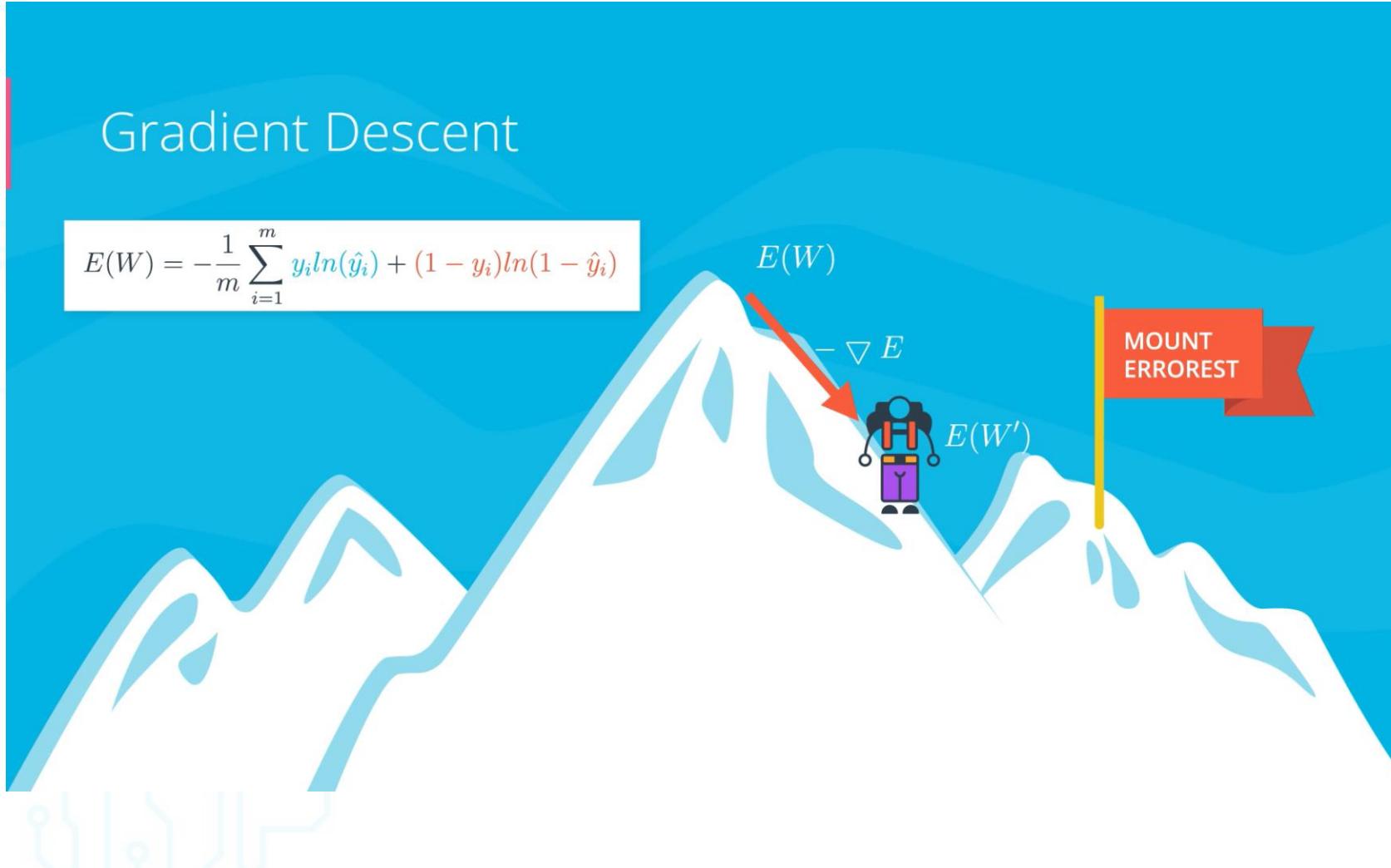
$$E(W) = -\frac{1}{m} \sum_{i=1}^m y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i)$$



Redes Neurais - Backpropagation

Gradient Descent

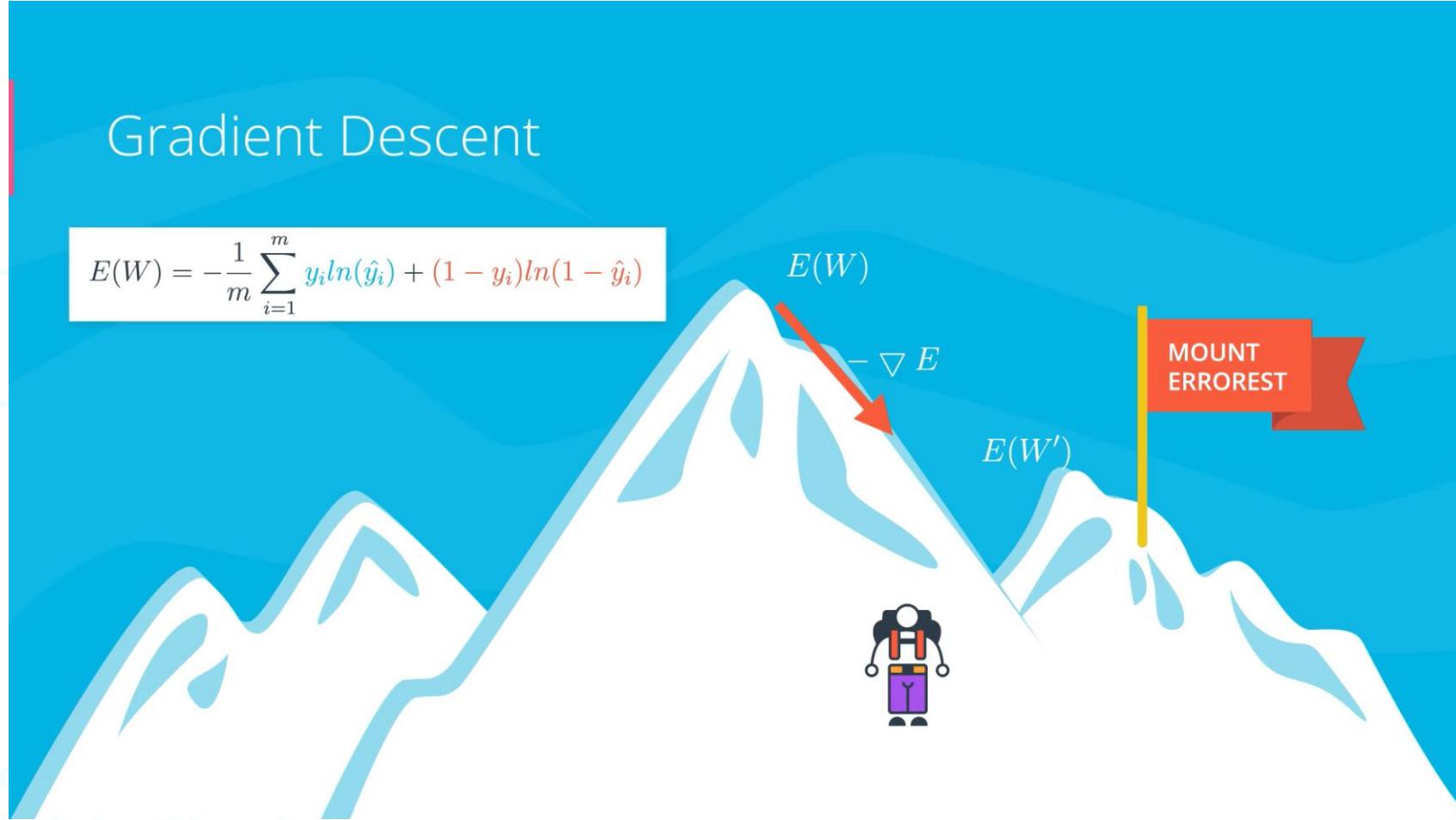
$$E(W) = -\frac{1}{m} \sum_{i=1}^m y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i)$$



Redes Neurais - Backpropagation

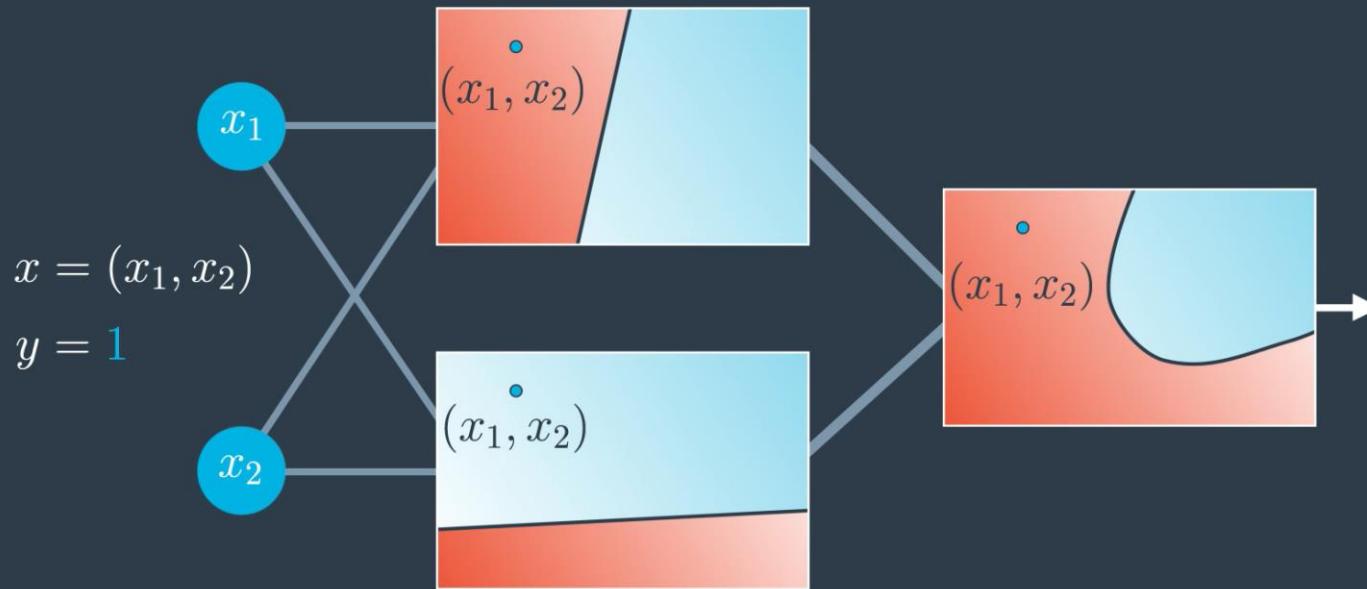
Gradient Descent

$$E(W) = -\frac{1}{m} \sum_{i=1}^m y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i)$$



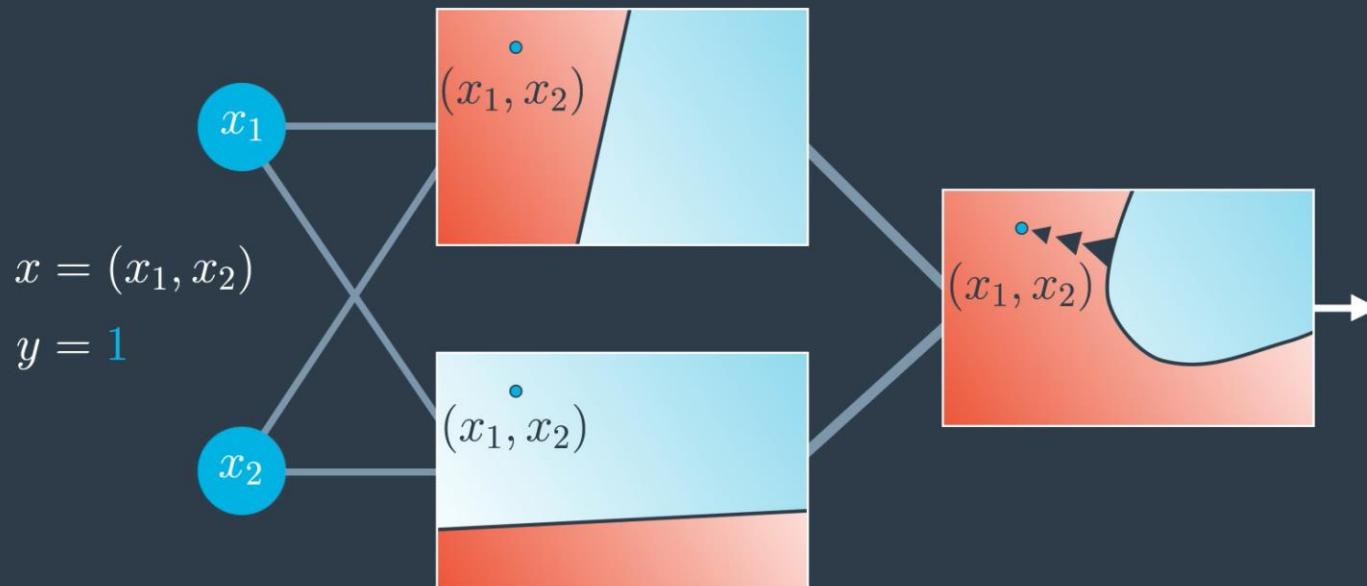
Redes Neurais - Backpropagation

FeedForward



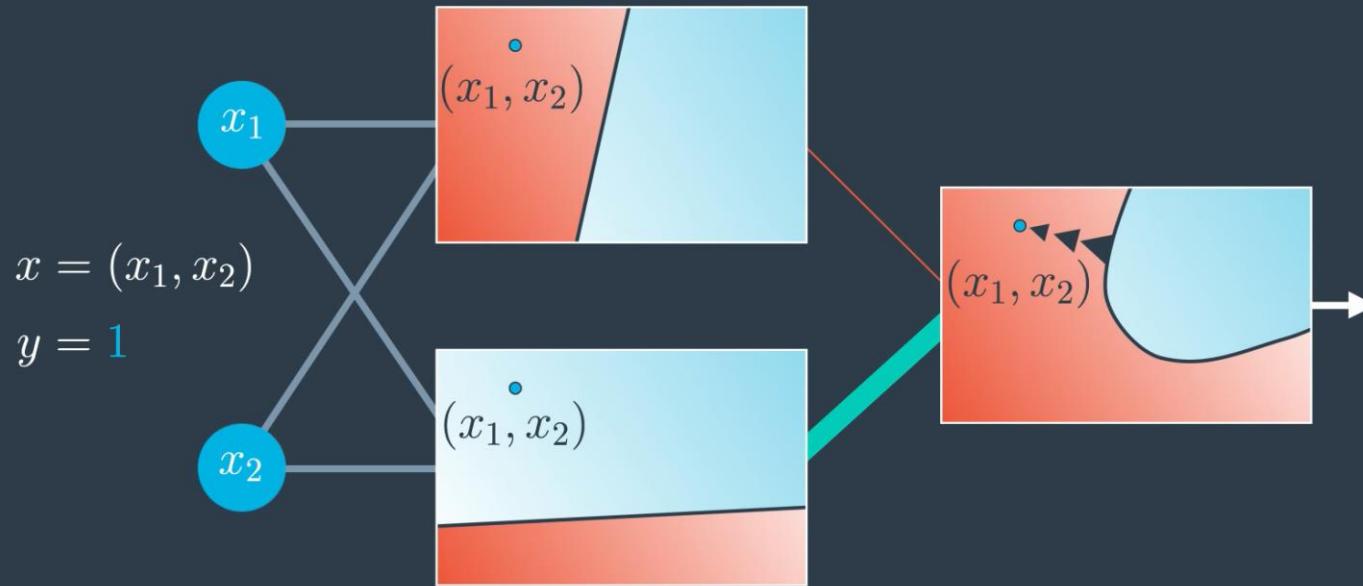
Redes Neurais - Backpropagation

Backpropagation



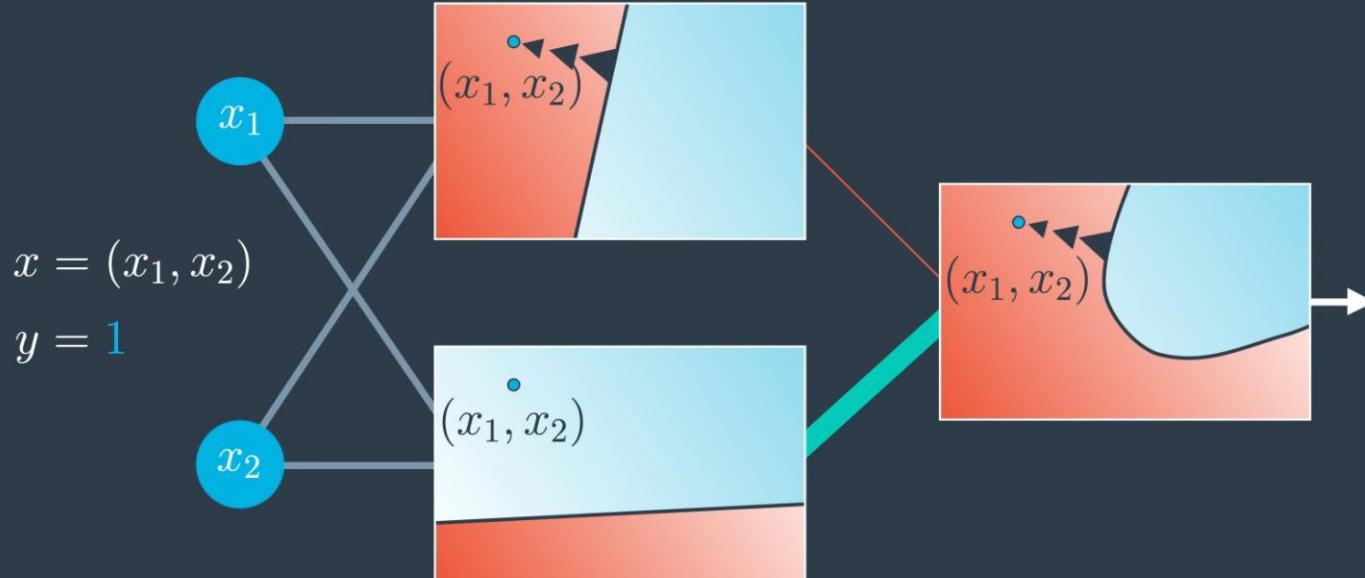
Redes Neurais - Backpropagation

Backpropagation



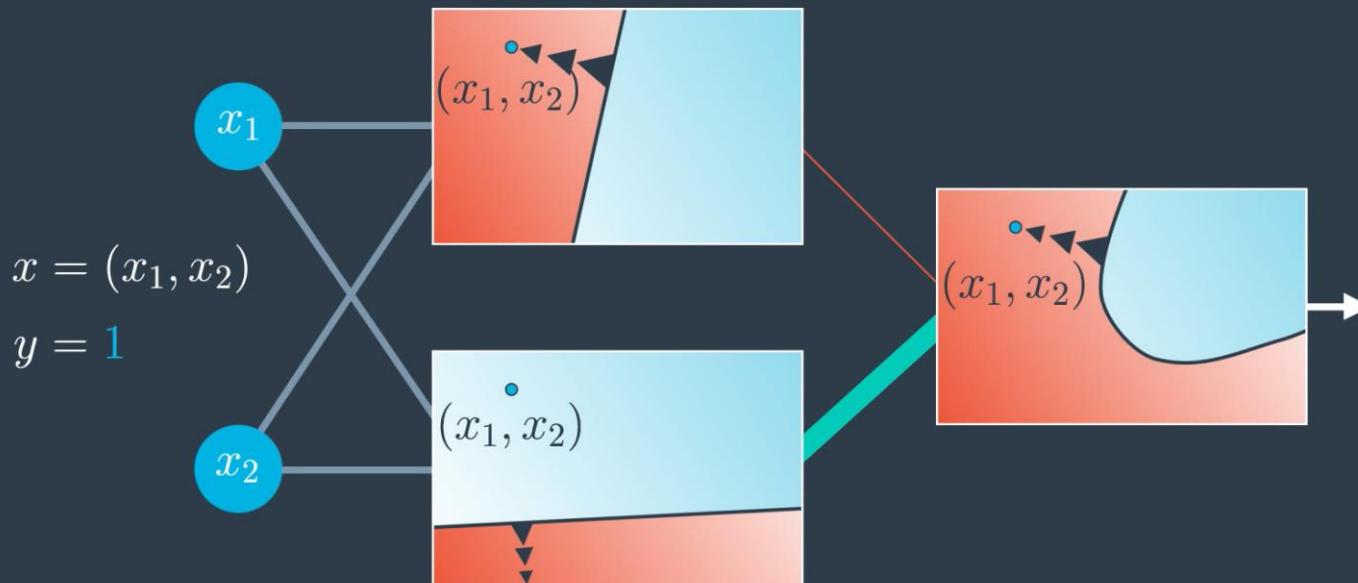
Redes Neurais - Backpropagation

Backpropagation



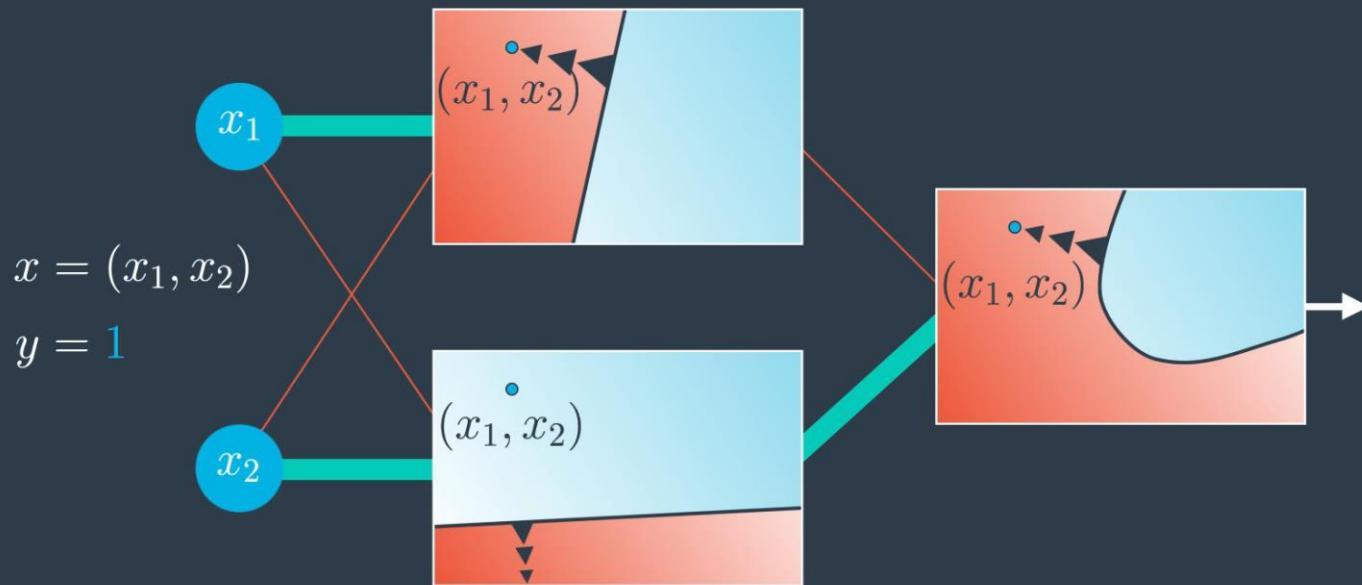
Redes Neurais - Backpropagation

Backpropagation



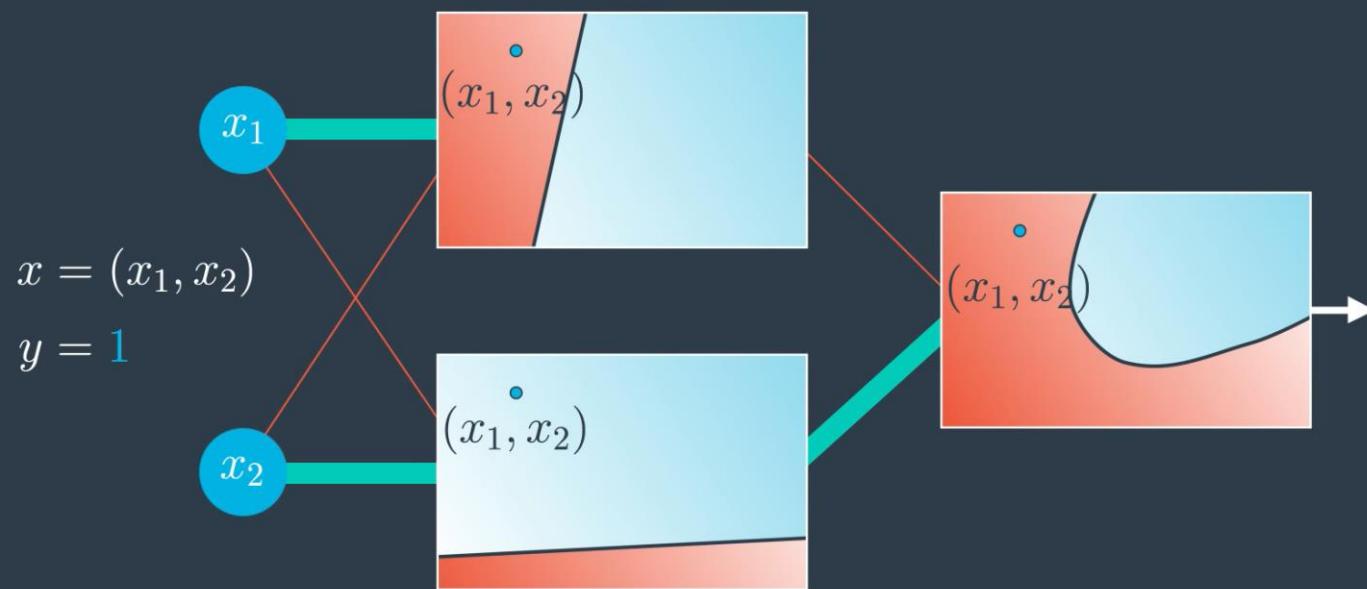
Redes Neurais - Backpropagation

Backpropagation

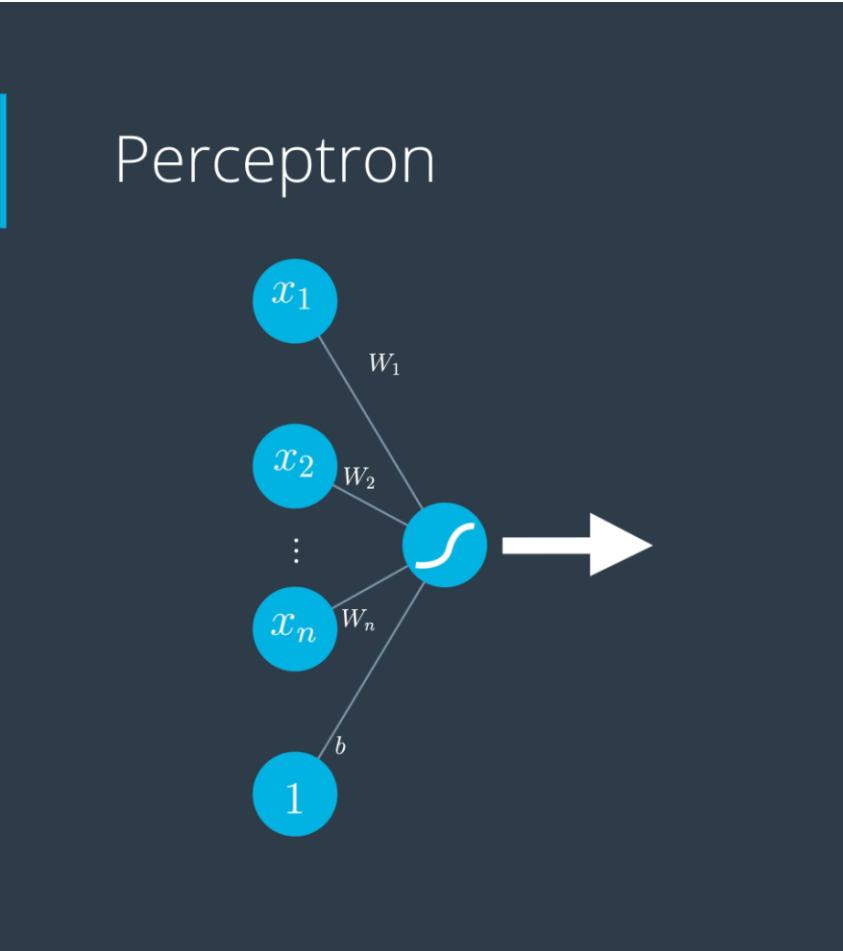


Redes Neurais - Backpropagation

Backpropagation



Redes Neurais - Backpropagation



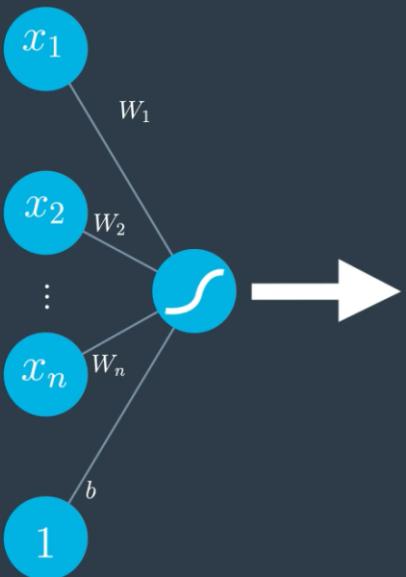
PREDICTION

ERROR FUNCTION

GRADIENT OF THE ERROR FUNCTION

Redes Neurais - Backpropagation

Perceptron



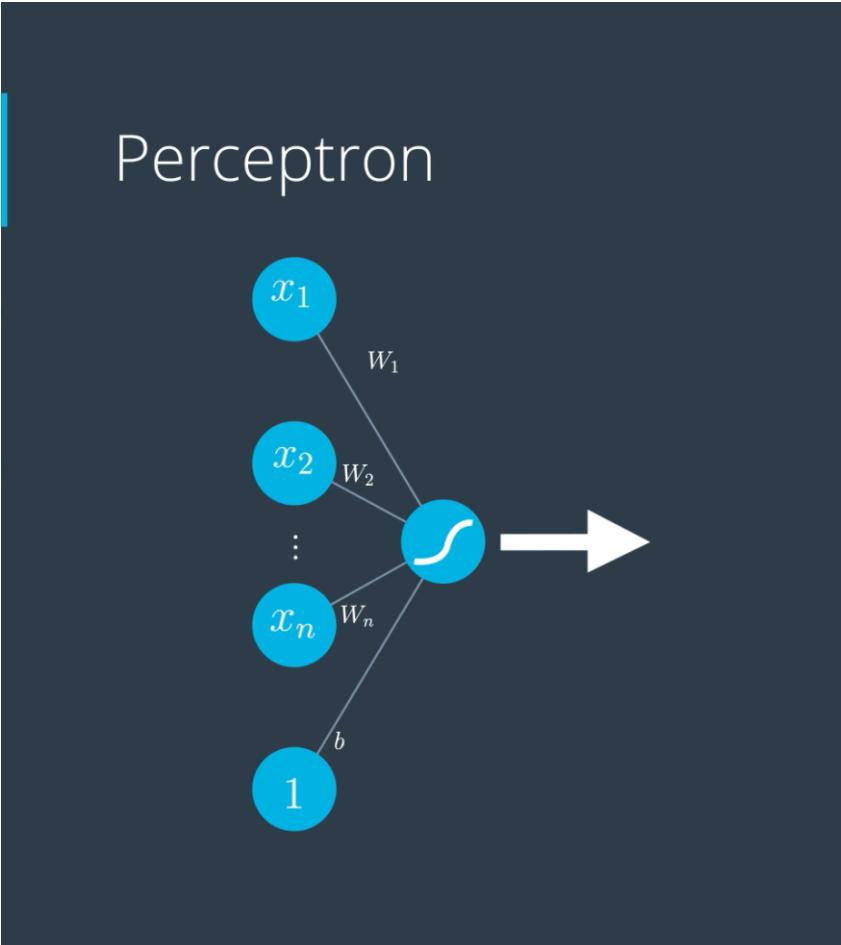
PREDICTION

$$\hat{y} = \sigma(Wx + b)$$

ERROR FUNCTION

GRADIENT OF THE ERROR FUNCTION

Redes Neurais - Backpropagation



PREDICTION

$$\hat{y} = \sigma(Wx + b)$$

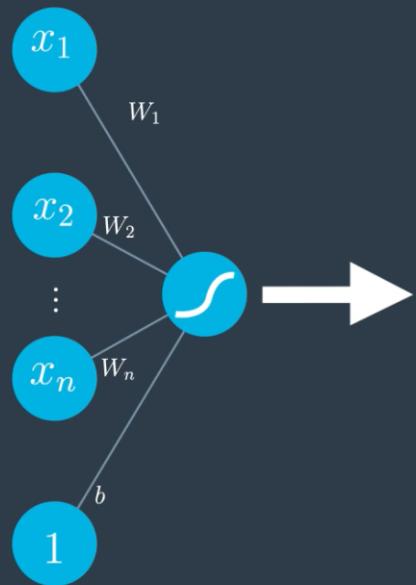
ERROR FUNCTION

$$E(W) = -\frac{1}{m} \sum_{i=1}^m y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i)$$

GRADIENT OF THE ERROR FUNCTION

Redes Neurais - Backpropagation

Perceptron



PREDICTION

$$\hat{y} = \sigma(Wx + b)$$

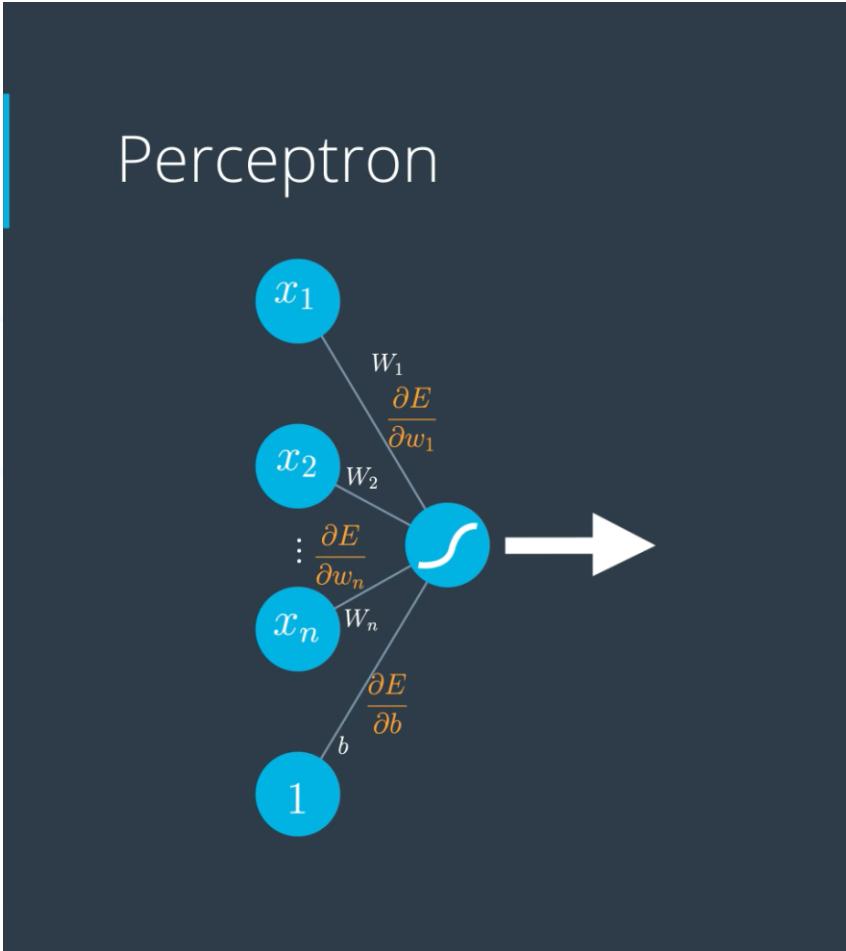
ERROR FUNCTION

$$E(W) = -\frac{1}{m} \sum_{i=1}^m y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i)$$

GRADIENT OF THE ERROR FUNCTION

$$\nabla E = \left(\frac{\partial E}{\partial w_1}, \dots, \frac{\partial E}{\partial w_n}, \frac{\partial E}{\partial b} \right)$$

Redes Neurais - Backpropagation



PREDICTION

$$\hat{y} = \sigma(Wx + b)$$

ERROR FUNCTION

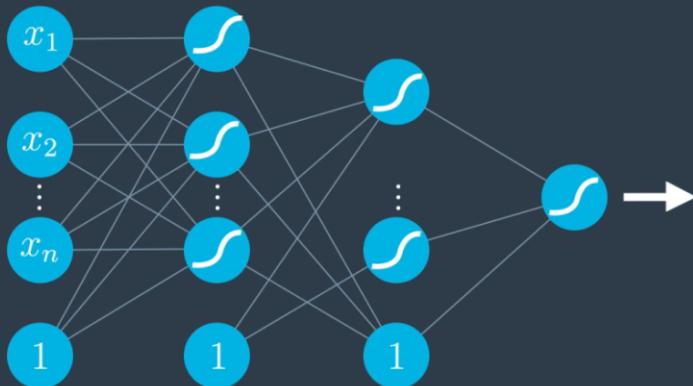
$$E(W) = -\frac{1}{m} \sum_{i=1}^m y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i)$$

GRADIENT OF THE ERROR FUNCTION

$$\nabla E = \left(\frac{\partial E}{\partial w_1}, \dots, \frac{\partial E}{\partial w_n}, \frac{\partial E}{\partial b} \right)$$

Redes Neurais - Backpropagation

Multi-layer Perceptron



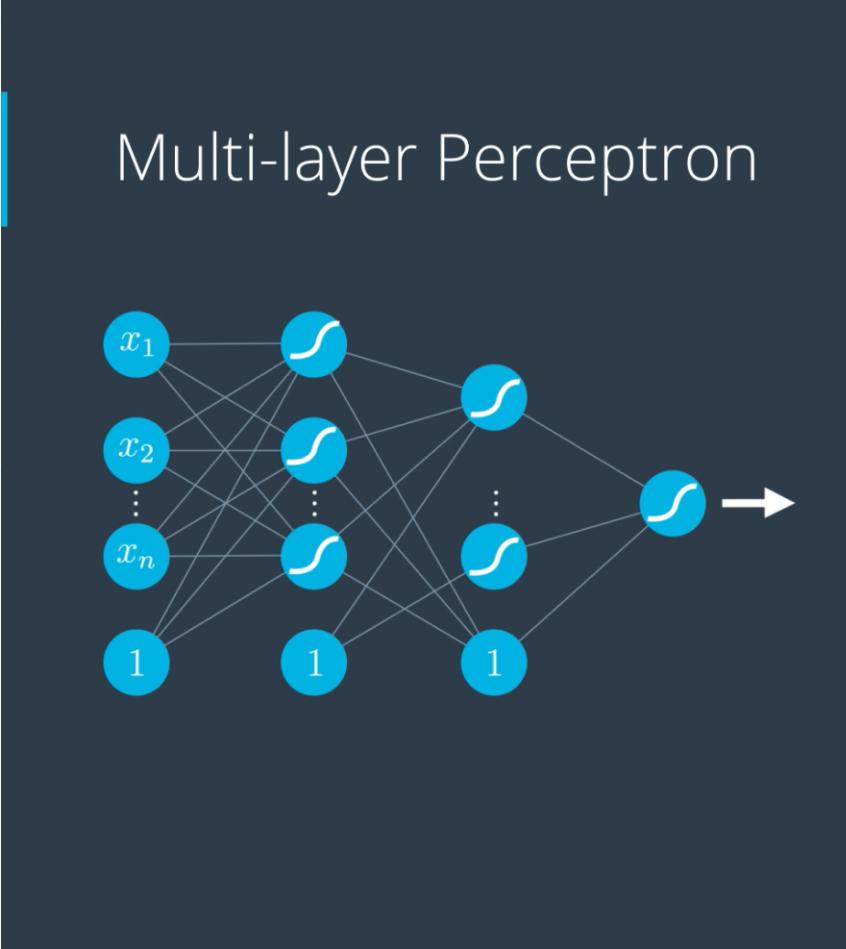
PREDICTION

ERROR FUNCTION

$$E(W) = -\frac{1}{m} \sum_{i=1}^m y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i)$$

GRADIENT OF THE ERROR FUNCTION

Redes Neurais - Backpropagation



PREDICTION

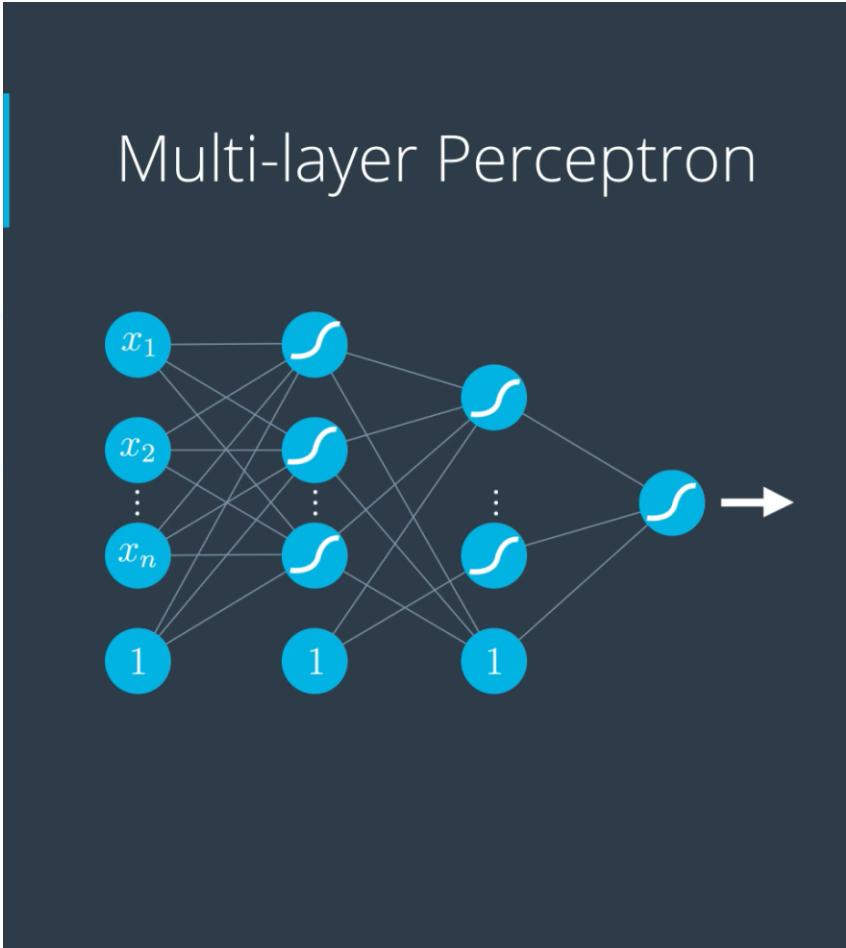
$$\hat{y} = \sigma W^{(3)} \circ \sigma W^{(2)} \circ \sigma \circ W^{(1)}(x)$$

ERROR FUNCTION

$$E(W) = -\frac{1}{m} \sum_{i=1}^m y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i)$$

GRADIENT OF THE ERROR FUNCTION

Redes Neurais - Backpropagation



PREDICTION

$$\hat{y} = \sigma W^{(3)} \circ \sigma W^{(2)} \circ \sigma \circ W^{(1)}(x)$$

ERROR FUNCTION

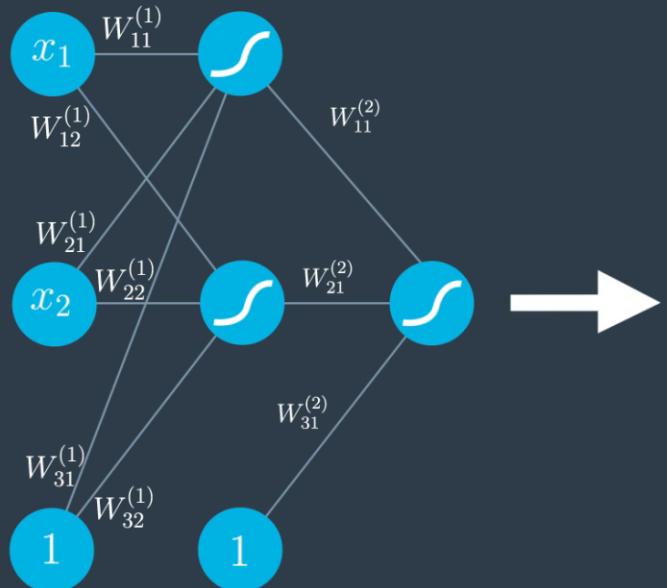
$$E(W) = -\frac{1}{m} \sum_{i=1}^m y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i)$$

GRADIENT OF THE ERROR FUNCTION

$$\nabla E = \left(\dots, \frac{\partial E}{\partial w_j^{(i)}}, \dots \right)$$

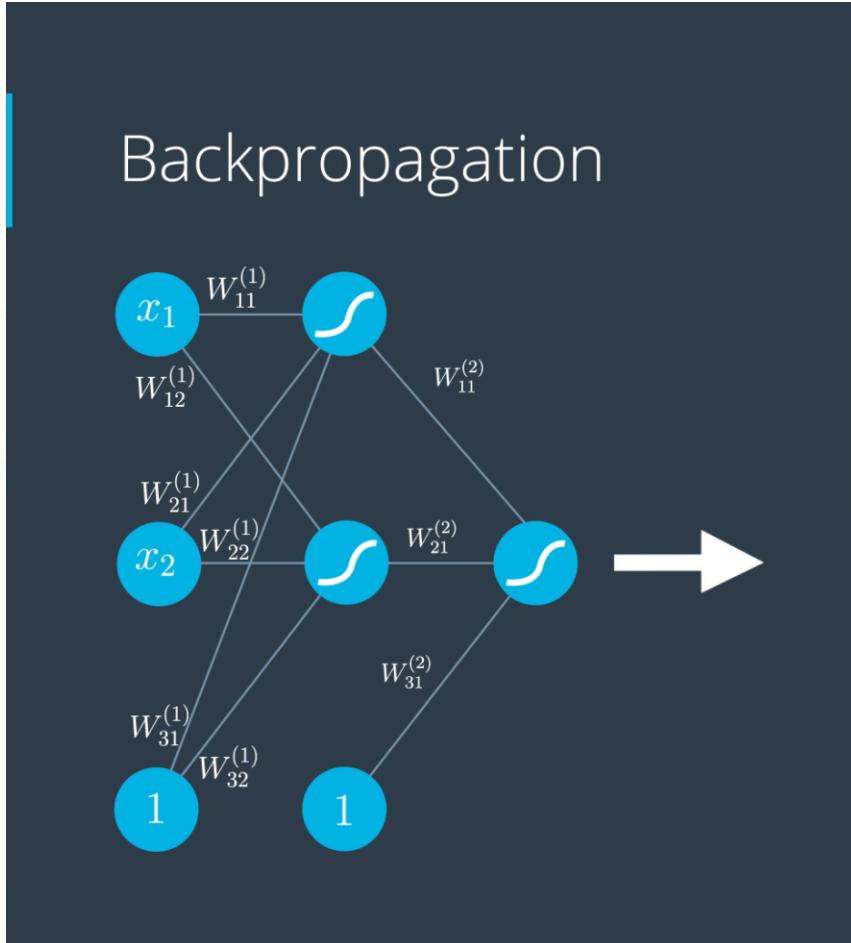
Redes Neurais - Backpropagation

Backpropagation



$$\hat{y} = \sigma W^{(2)} \circ \sigma \circ W^{(1)}(x)$$

Redes Neurais - Backpropagation



$$\hat{y} = \sigma W^{(2)} \circ \sigma \circ W^{(1)}(x)$$

$$W^{(1)} = \begin{pmatrix} W_{11}^{(1)} & W_{12}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} \\ W_{31}^{(1)} & W_{32}^{(1)} \end{pmatrix} \quad W^{(2)} = \begin{pmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{31}^{(2)} \end{pmatrix}$$

$$\nabla E = \begin{pmatrix} \frac{\partial E}{\partial W_{11}^{(1)}} & \frac{\partial E}{\partial W_{12}^{(1)}} & \frac{\partial E}{\partial W_{11}^{(2)}} \\ \frac{\partial E}{\partial W_{21}^{(1)}} & \frac{\partial E}{\partial W_{22}^{(1)}} & \frac{\partial E}{\partial W_{21}^{(2)}} \\ \frac{\partial E}{\partial W_{31}^{(1)}} & \frac{\partial E}{\partial W_{32}^{(1)}} & \frac{\partial E}{\partial W_{31}^{(2)}} \end{pmatrix}$$

$$W'_{ij}^{(k)} \leftarrow W_{ij}^{(k)} - \alpha \frac{\partial E}{\partial W_{ij}^{(k)}}$$



Thank you!



@vinicius caridá



@vfcarida



vfcarida@gmail.com



<https://linktr.ee/vfcarida>