VINI CARIDÁ

Human-Centered Data & AI

Vinicius Caridá, Ph.D.

Data Science Manager, Itaú Unibanco

MBA Professor, FIAP

GDE – Machine Learning

@vinicius caridá @vfcarida @vinicius caridá @vfcarida @vinicius caridá @vfcarida

VINI CARIDÁ
Human-Centered Data & AI

VINI CARIDÁ
Human-Centered Data & AI

Redes Neurais

Recapitulando...

Regressão Linear

	Peso	Altura
 Pessoa 1	80 kg	163
 Pessoa 2	85 kg	168
 Pessoa 3	90 kg	175
 Pessoa 4	95 kg	188
 Pessoa 5	88 kg	175,4

$$\hat{y} = \beta_0 + \beta_1 X_1$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

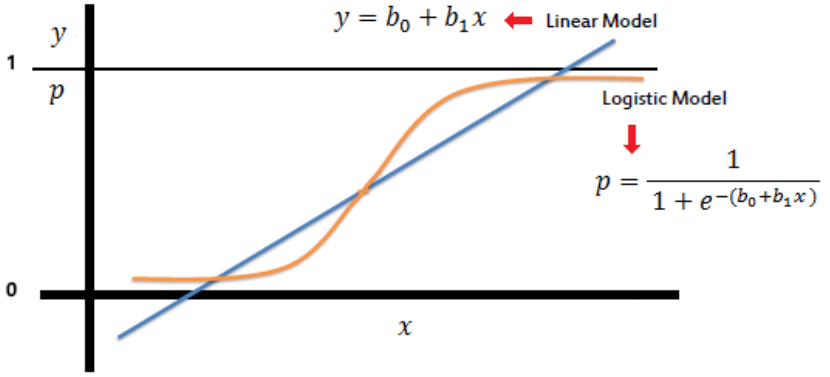
$\beta_0 = 17$

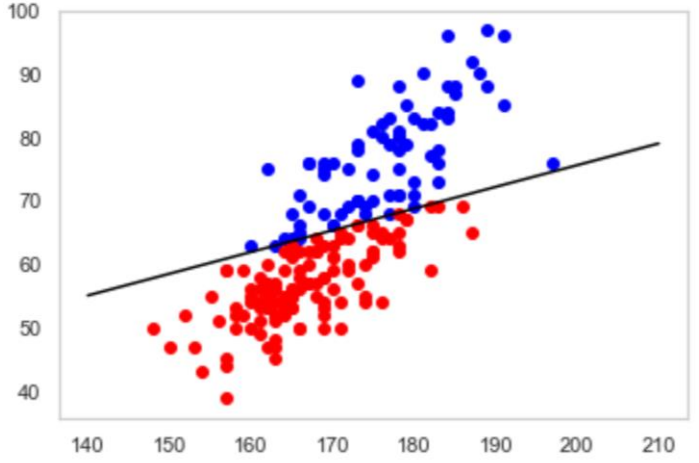
$\beta_1 = 1{,}8$

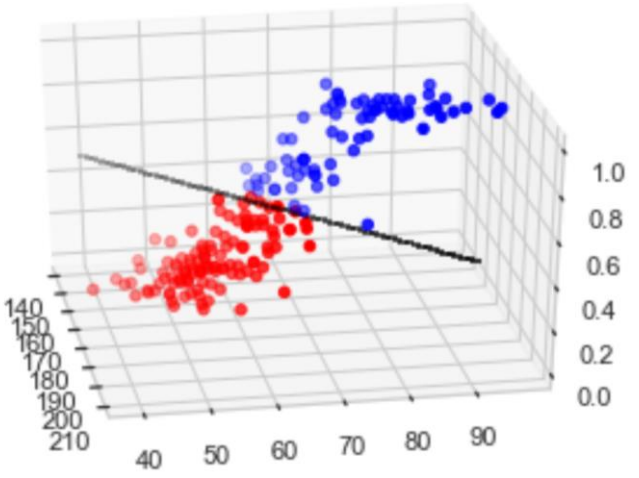
MSE = 6

$$\hat{y} = 17 + 1{,}8\, X_1$$

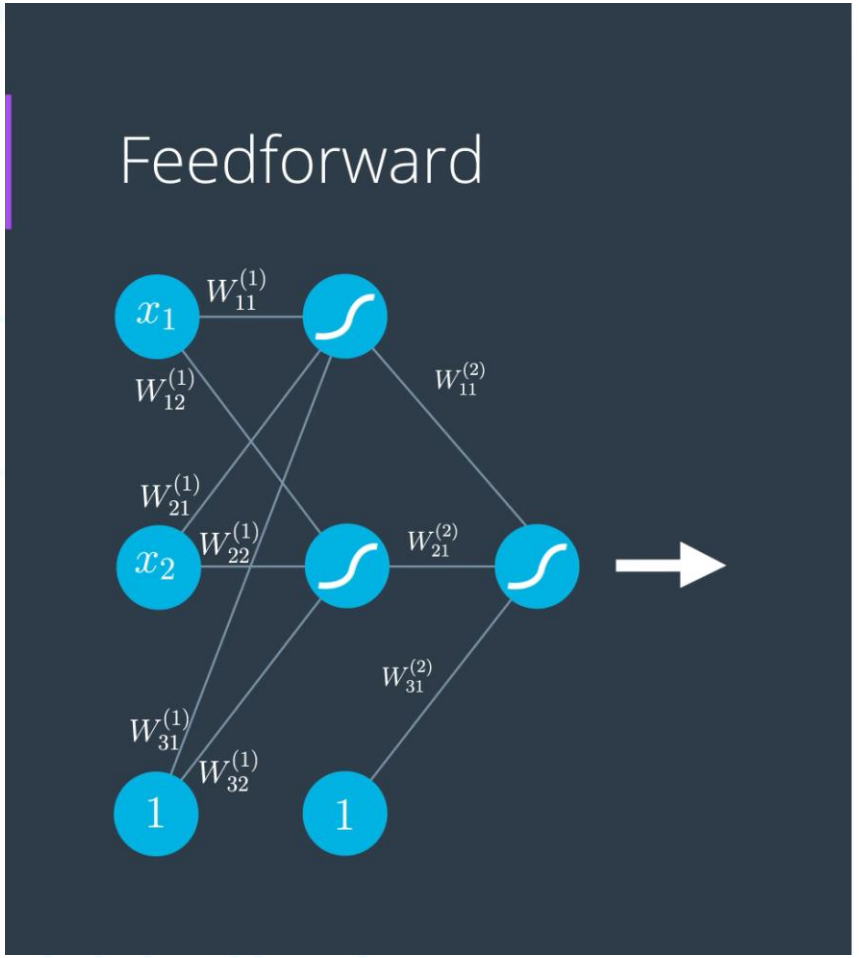
Redes Neurais – Feedforward







Redes Neurais – Feedforward



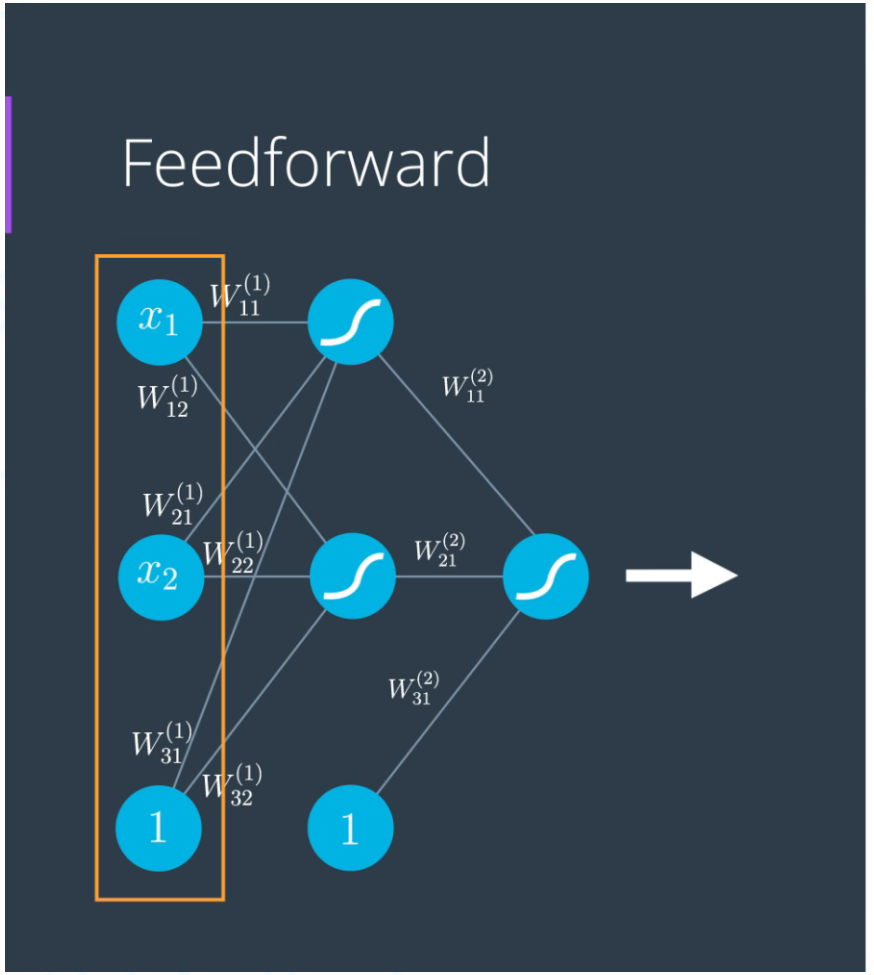
$$\hat{y} = \sigma \begin{pmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{31}^{(2)} \end{pmatrix} \sigma \begin{pmatrix} W_{11}^{(1)} & W_{12}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} \\ W_{31}^{(1)} & W_{32}^{(1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

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

VINI CARIDÁ
Human-Centered Data & AI

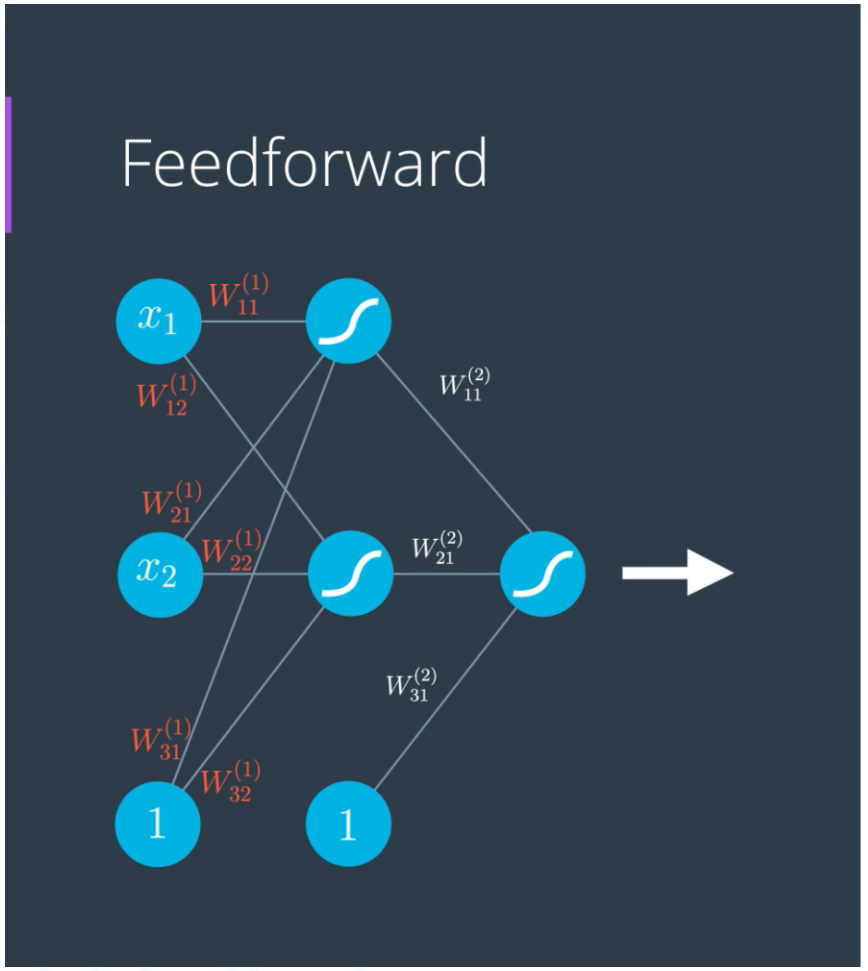
Redes Neurais

Redes Neurais – Feedforward



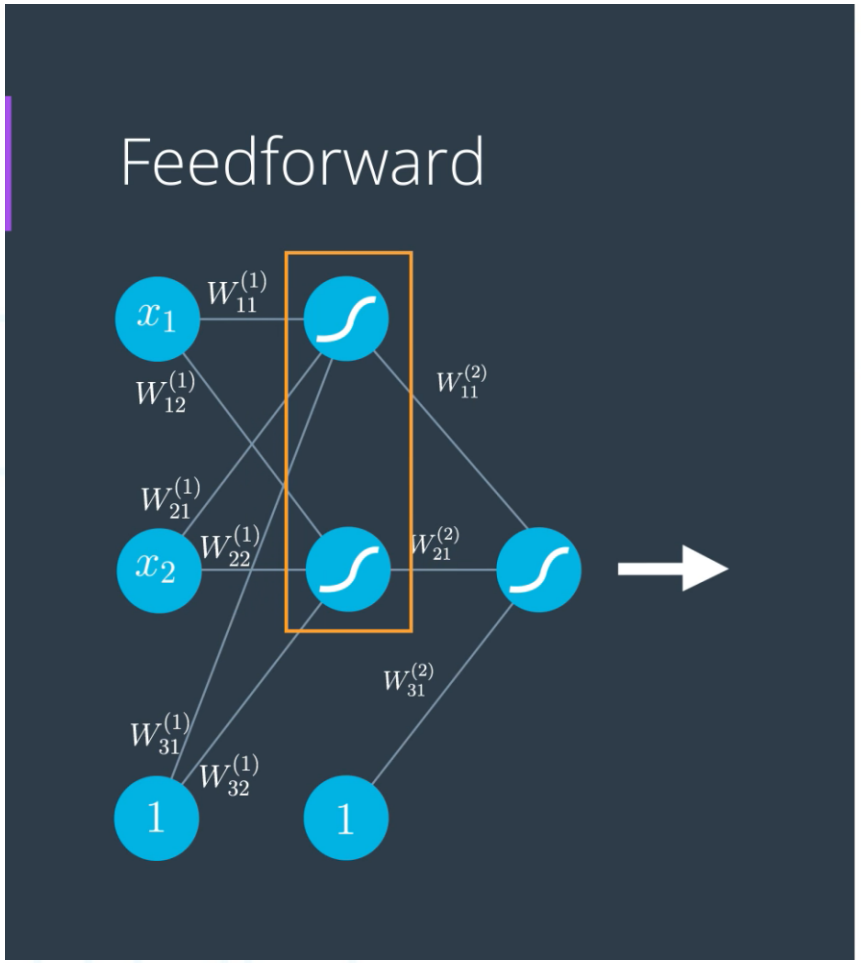
$$\begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

Redes Neurais – Feedforward



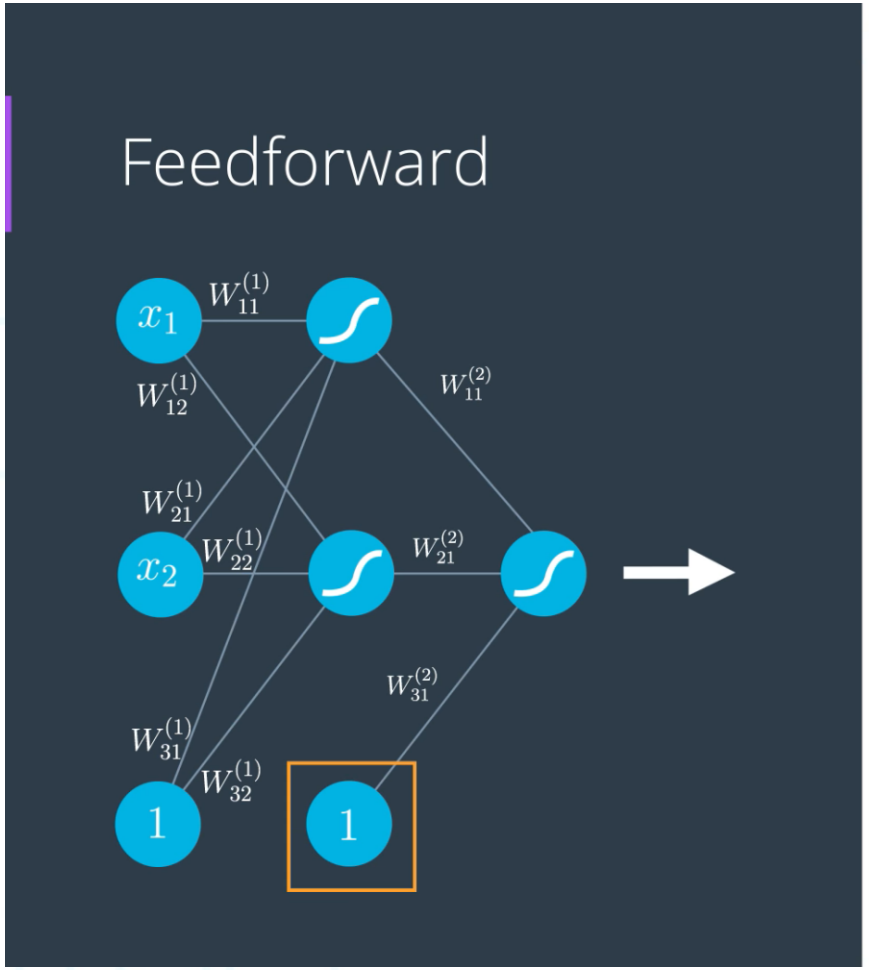
$$\begin{pmatrix} W_{11}^{(1)} & W_{12}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} \\ W_{31}^{(1)} & W_{32}^{(1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

Redes Neurais – Feedforward



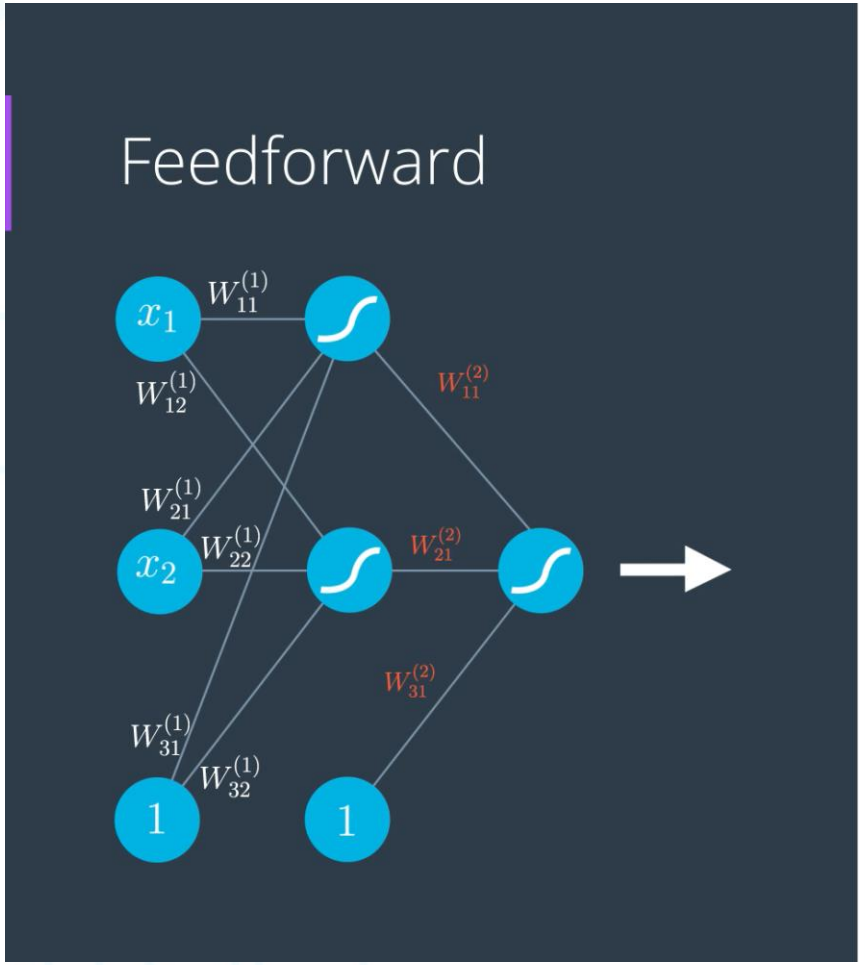
$$\sigma \begin{pmatrix} W_{11}^{(1)} & W_{12}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} \\ W_{31}^{(1)} & W_{32}^{(1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

Redes Neurais – Feedforward



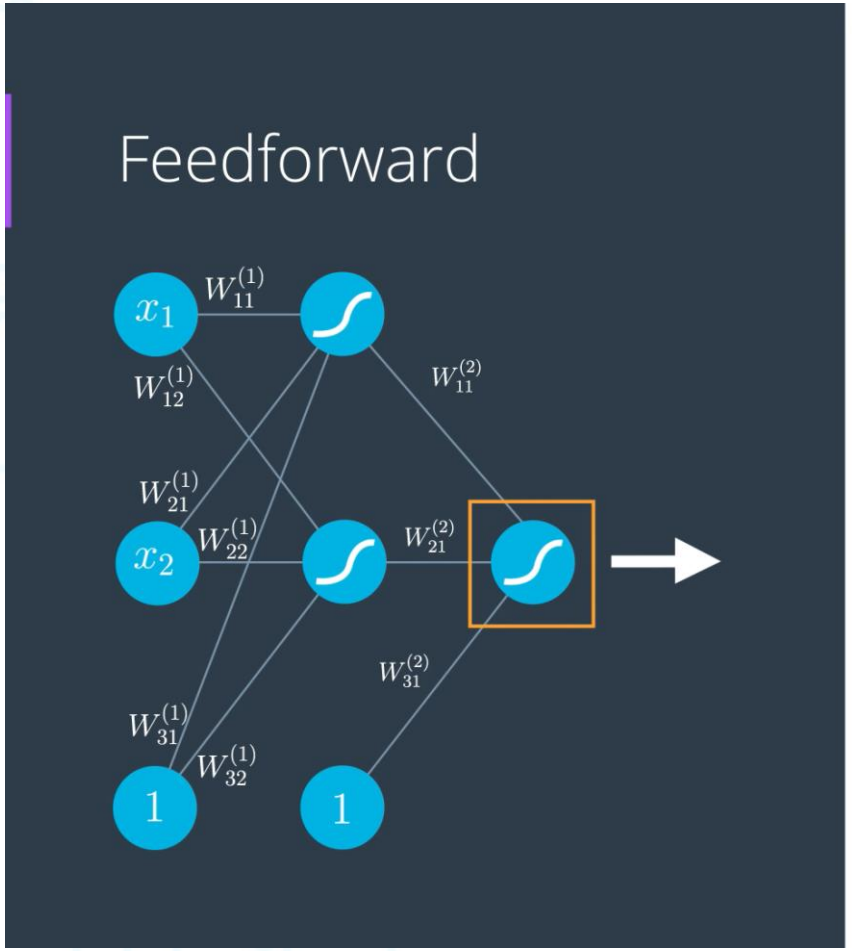
$$\sigma \begin{pmatrix} W_{11}^{(1)} & W_{12}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} \\ W_{31}^{(1)} & W_{32}^{(1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

Redes Neurais – Feedforward



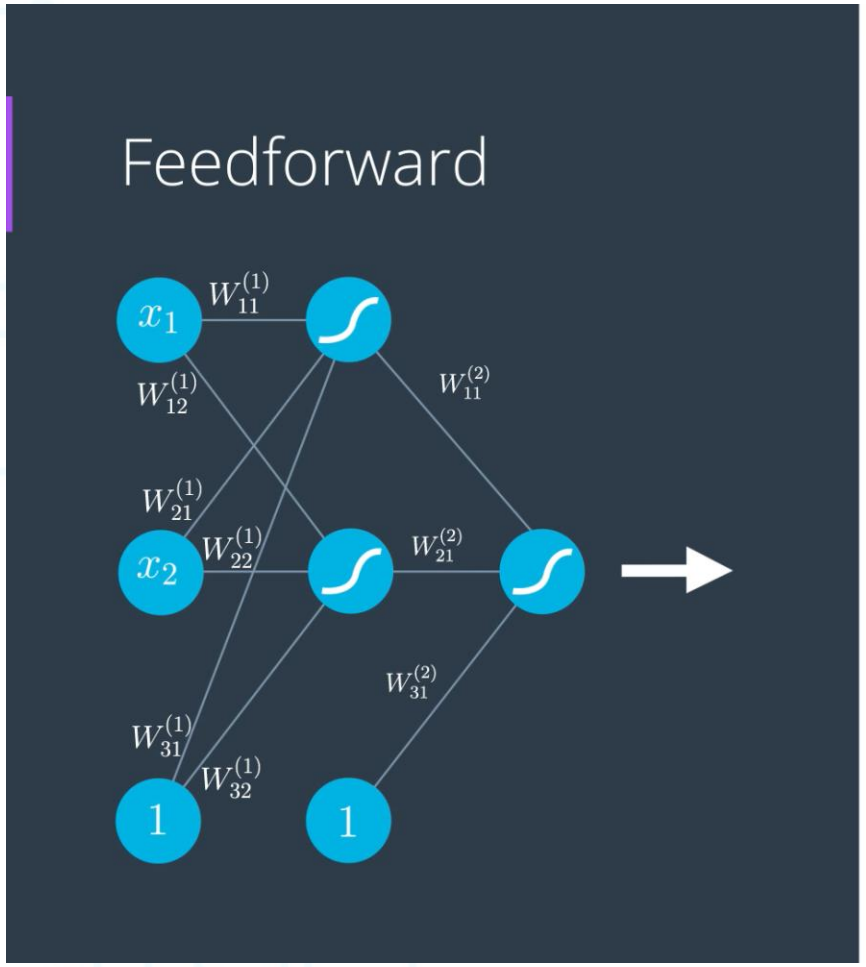
$$\begin{pmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{31}^{(2)} \end{pmatrix} \sigma \begin{pmatrix} W_{11}^{(1)} & W_{12}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} \\ W_{31}^{(1)} & W_{32}^{(1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

Redes Neurais – Feedforward



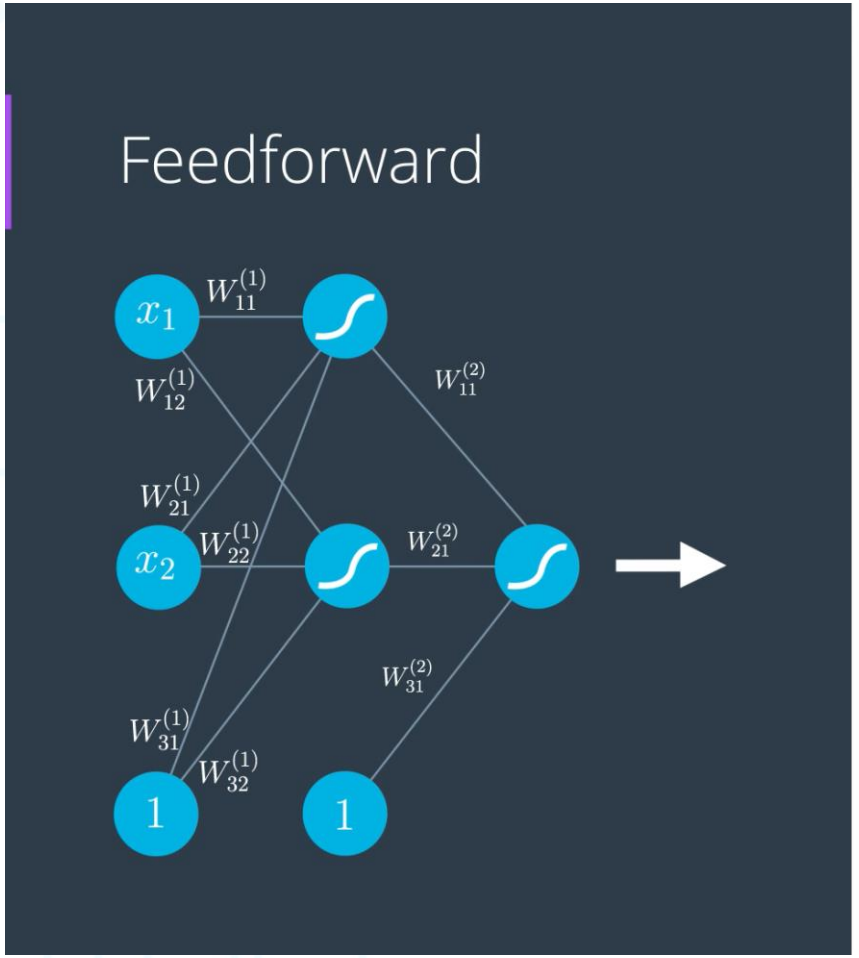
$$\sigma \begin{pmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{31}^{(2)} \end{pmatrix} \sigma \begin{pmatrix} W_{11}^{(1)} & W_{12}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} \\ W_{31}^{(1)} & W_{32}^{(1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

Redes Neurais – Feedforward



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

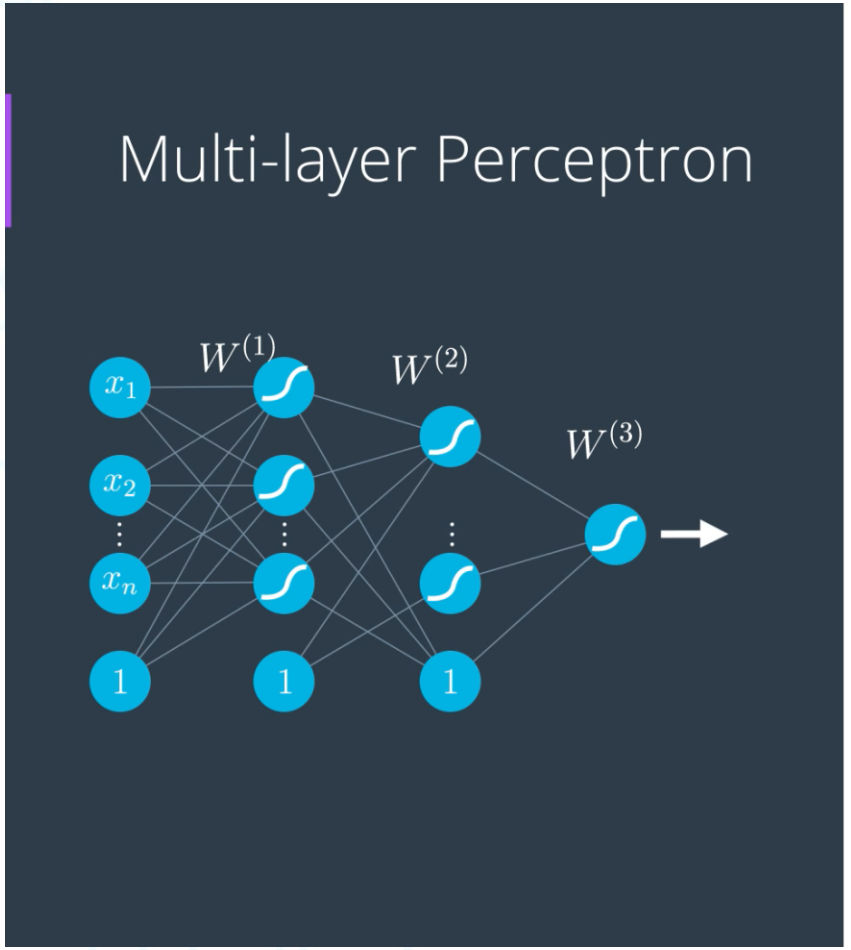
Redes Neurais – Feedforward



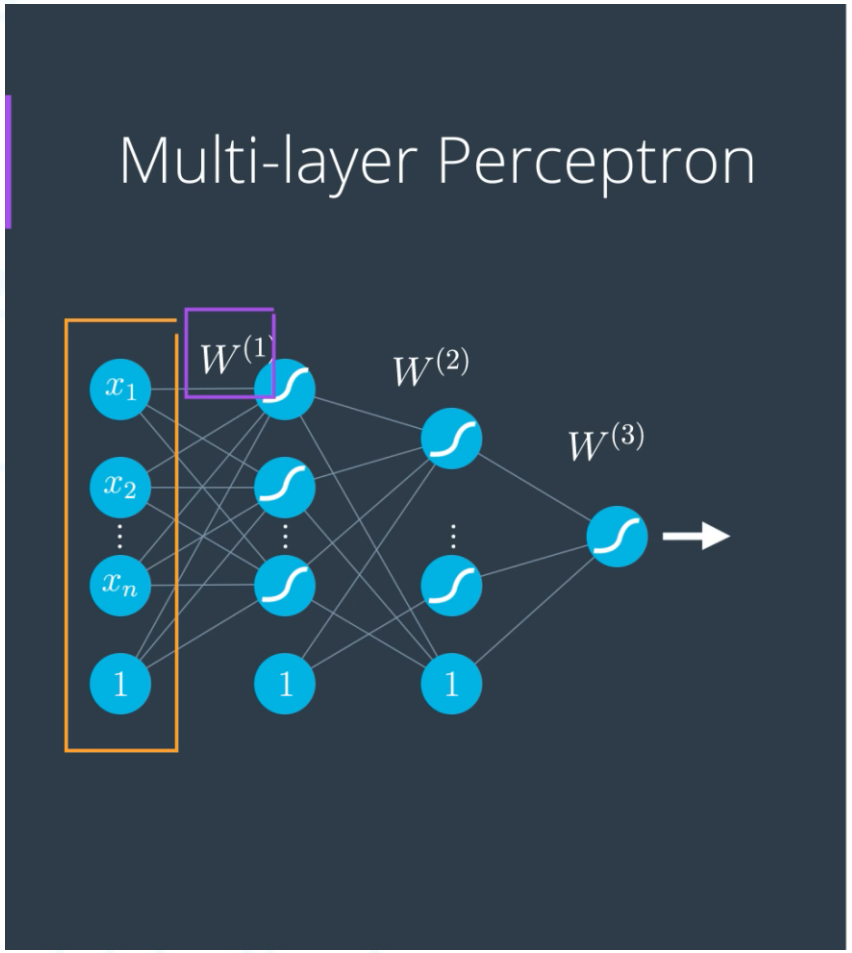
$$\hat{y} = \sigma \begin{pmatrix} W_{11}^{(2)} \\ W_{21}^{(2)} \\ W_{31}^{(2)} \end{pmatrix} \sigma \begin{pmatrix} W_{11}^{(1)} & W_{12}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} \\ W_{31}^{(1)} & W_{32}^{(1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

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

Redes Neurais – Feedforward



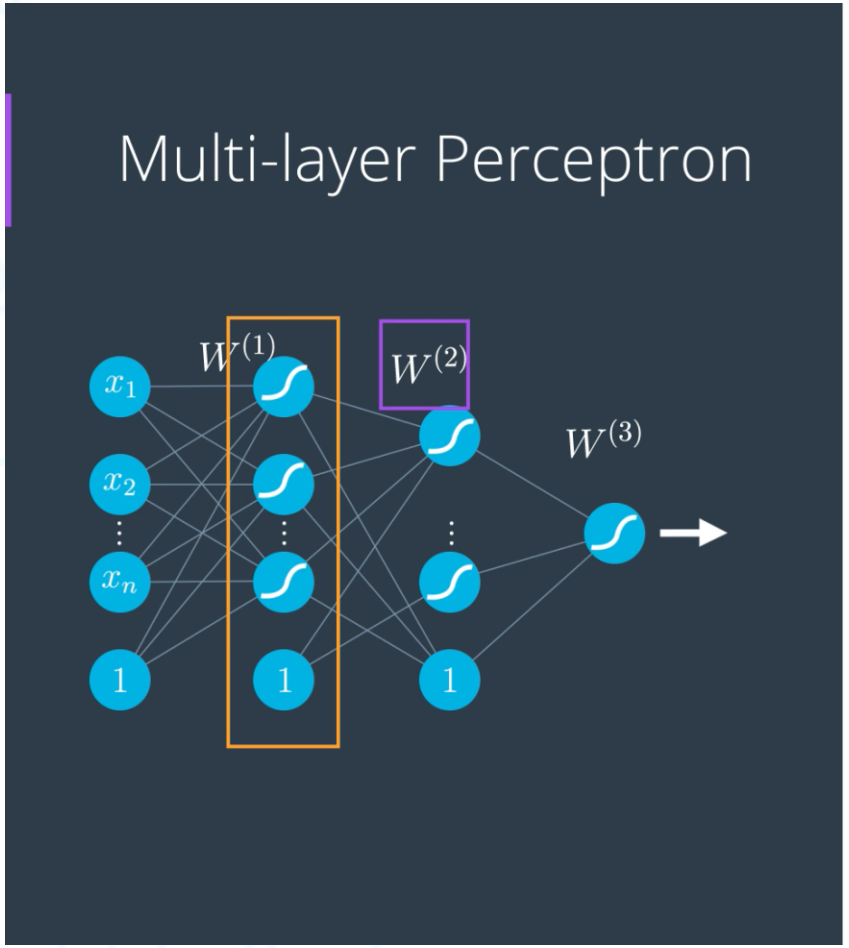
Redes Neurais – Feedforward



PREDICTION

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

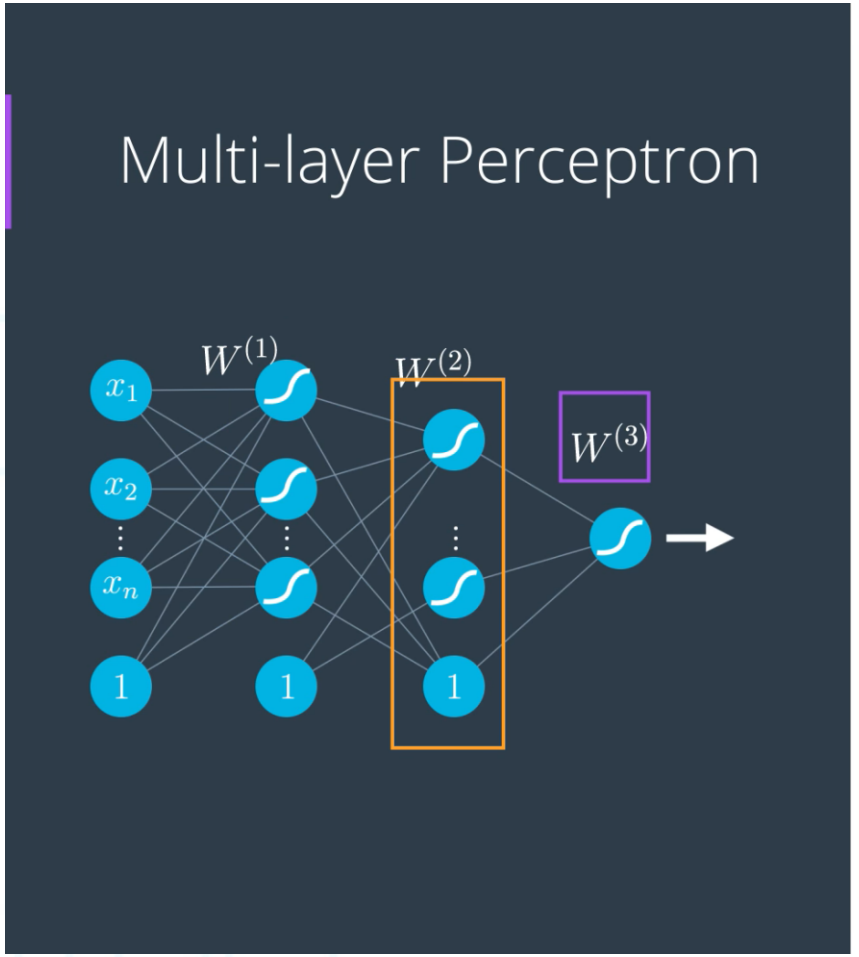
Redes Neurais – Feedforward



PREDICTION

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

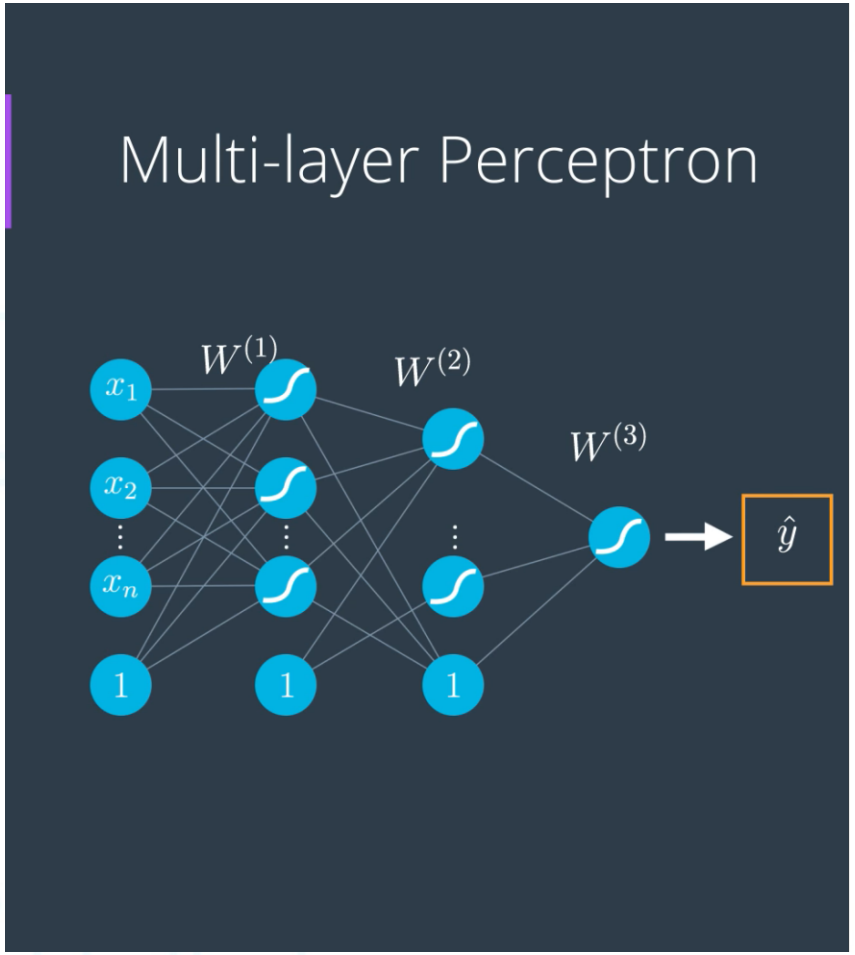
Redes Neurais – Feedforward



PREDICTION

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

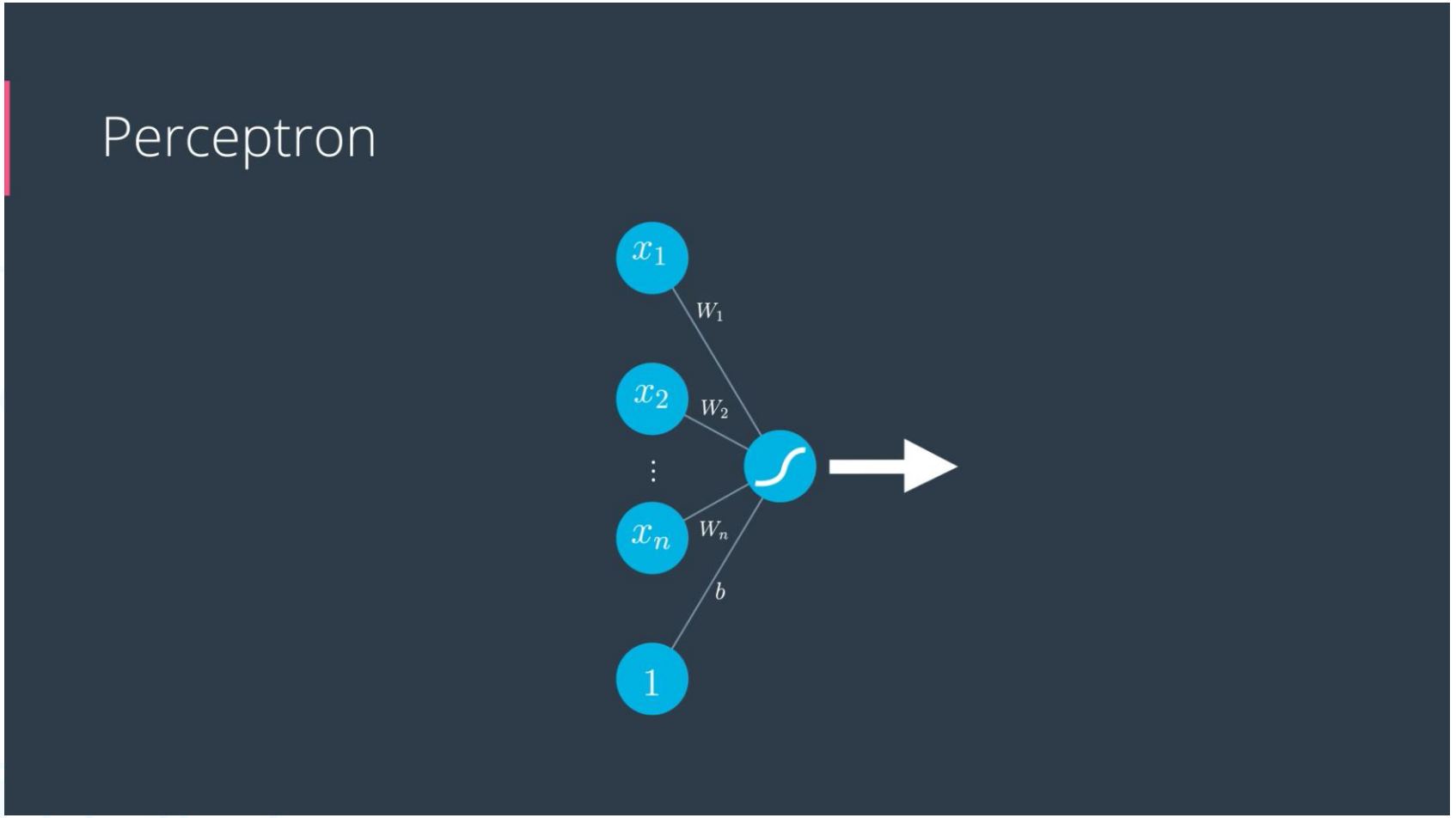
Redes Neurais – Feedforward



PREDICTION

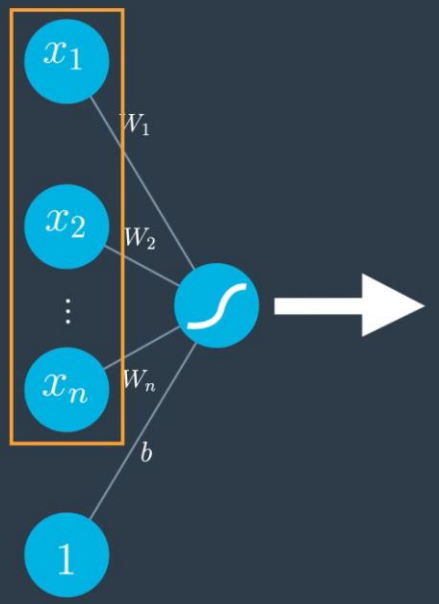
$$\hat{y} = \sigma \circ W^{(3)} \circ \sigma \circ W^{(2)} \circ \sigma \circ W^{(1)}(x)$$

Redes Neurais – Error Function



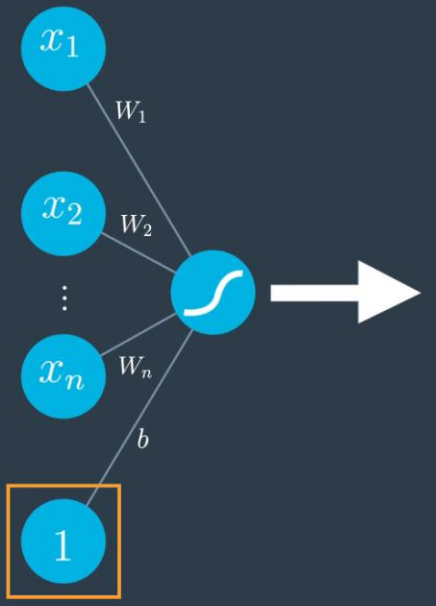
Redes Neurais – Error Function

Perceptron



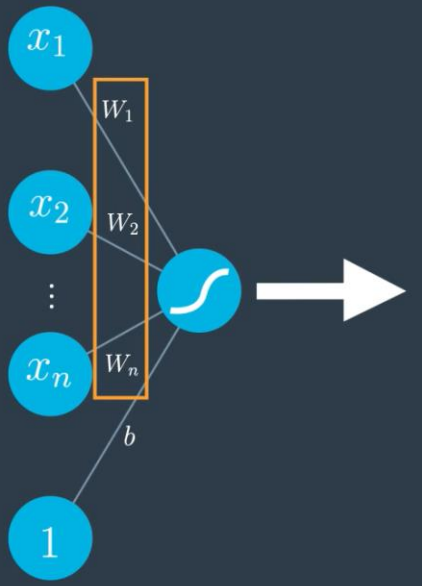
Redes Neurais – Error Function

Perceptron



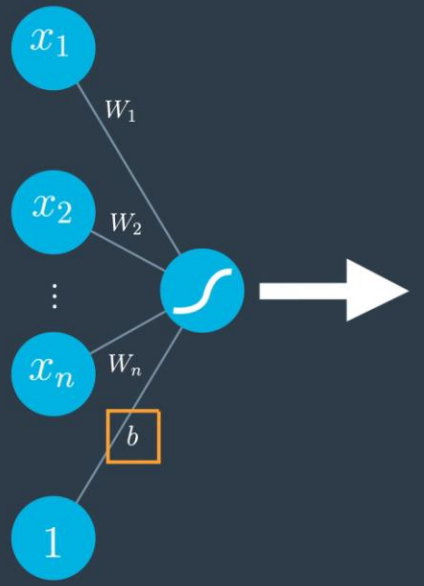
Redes Neurais – Error Function

Perceptron



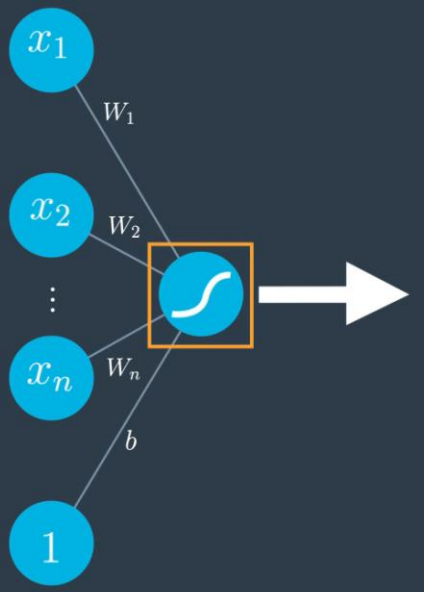
Redes Neurais – Error Function

Perceptron

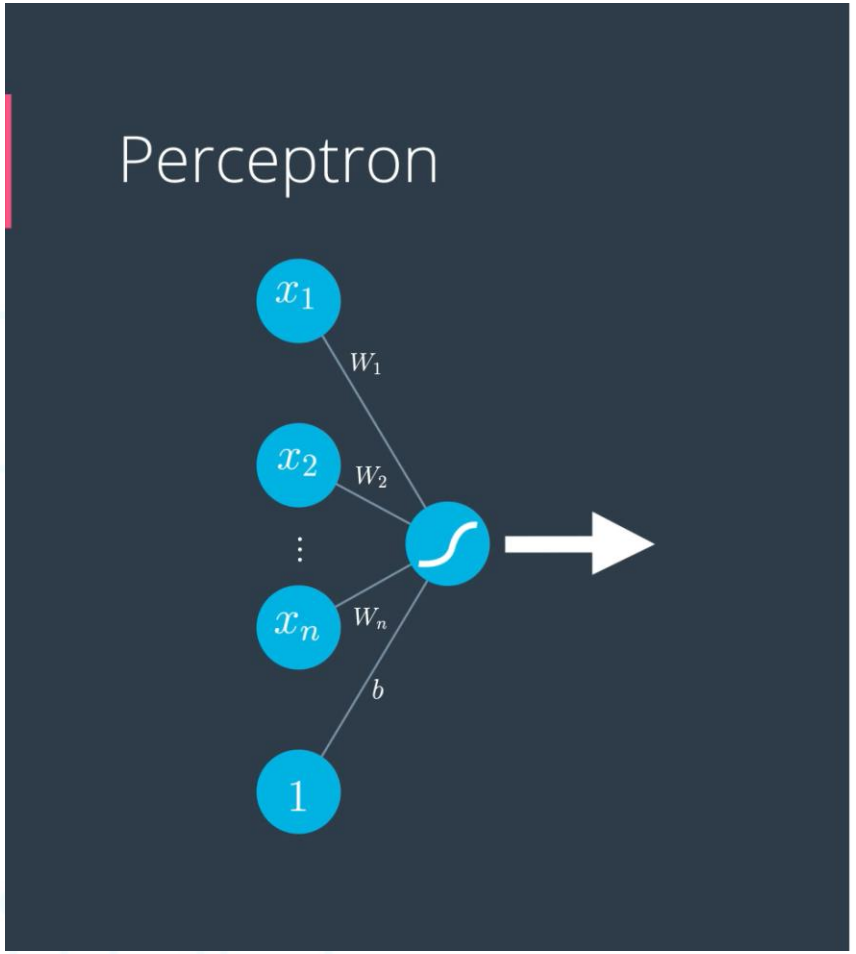


Redes Neurais – Error Function

Perceptron



Redes Neurais – Error Function

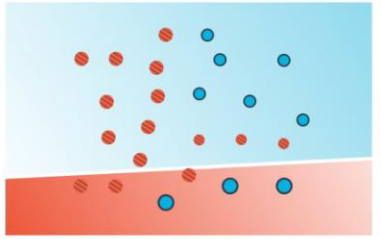


PREDICTION

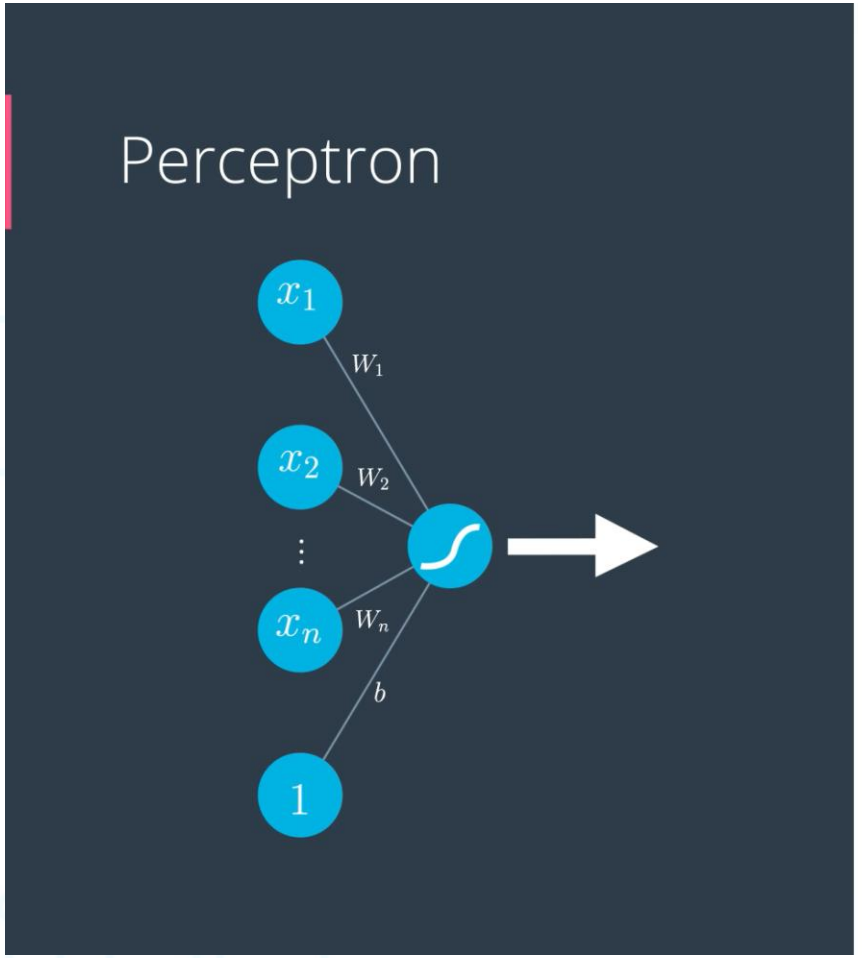
$$\hat{y} = \sigma(Wx + b)$$

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

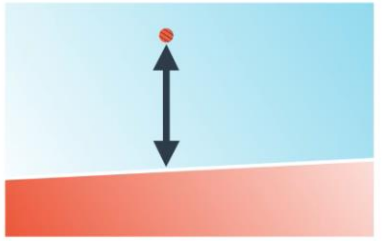


PREDICTION

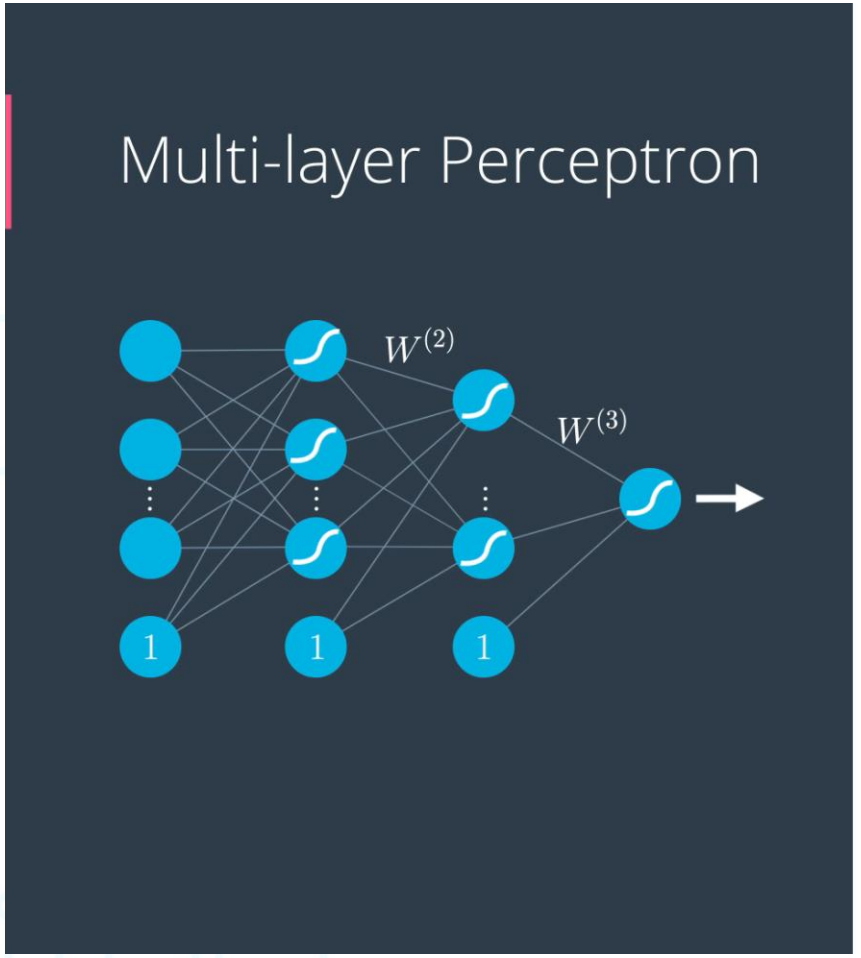
$$\hat{y} = \sigma(Wx + b)$$

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

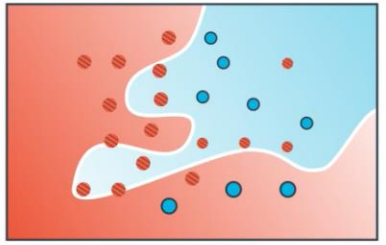


PREDICTION

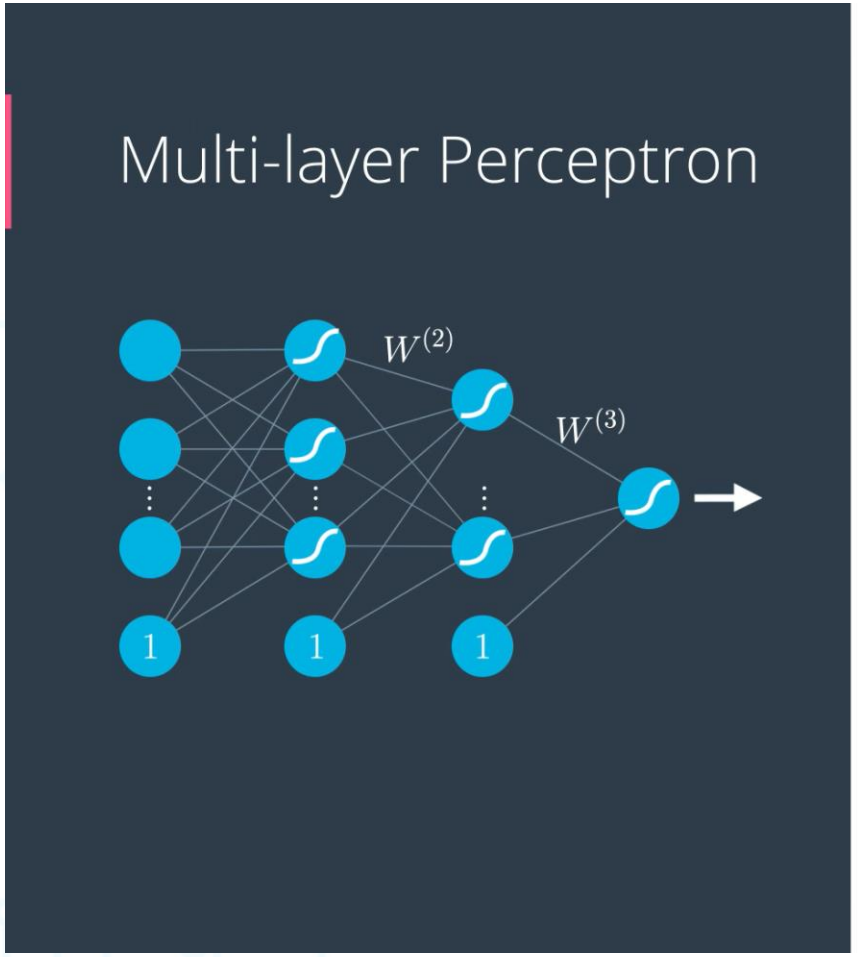
$$\hat{y} = \sigma \circ W^{(3)} \circ \sigma \circ W^{(2)} \circ \sigma \circ W^{(1)}(x)$$

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

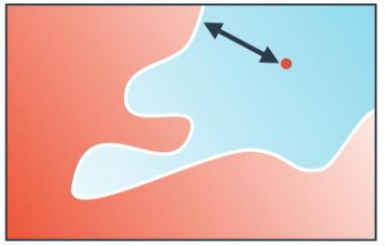


PREDICTION

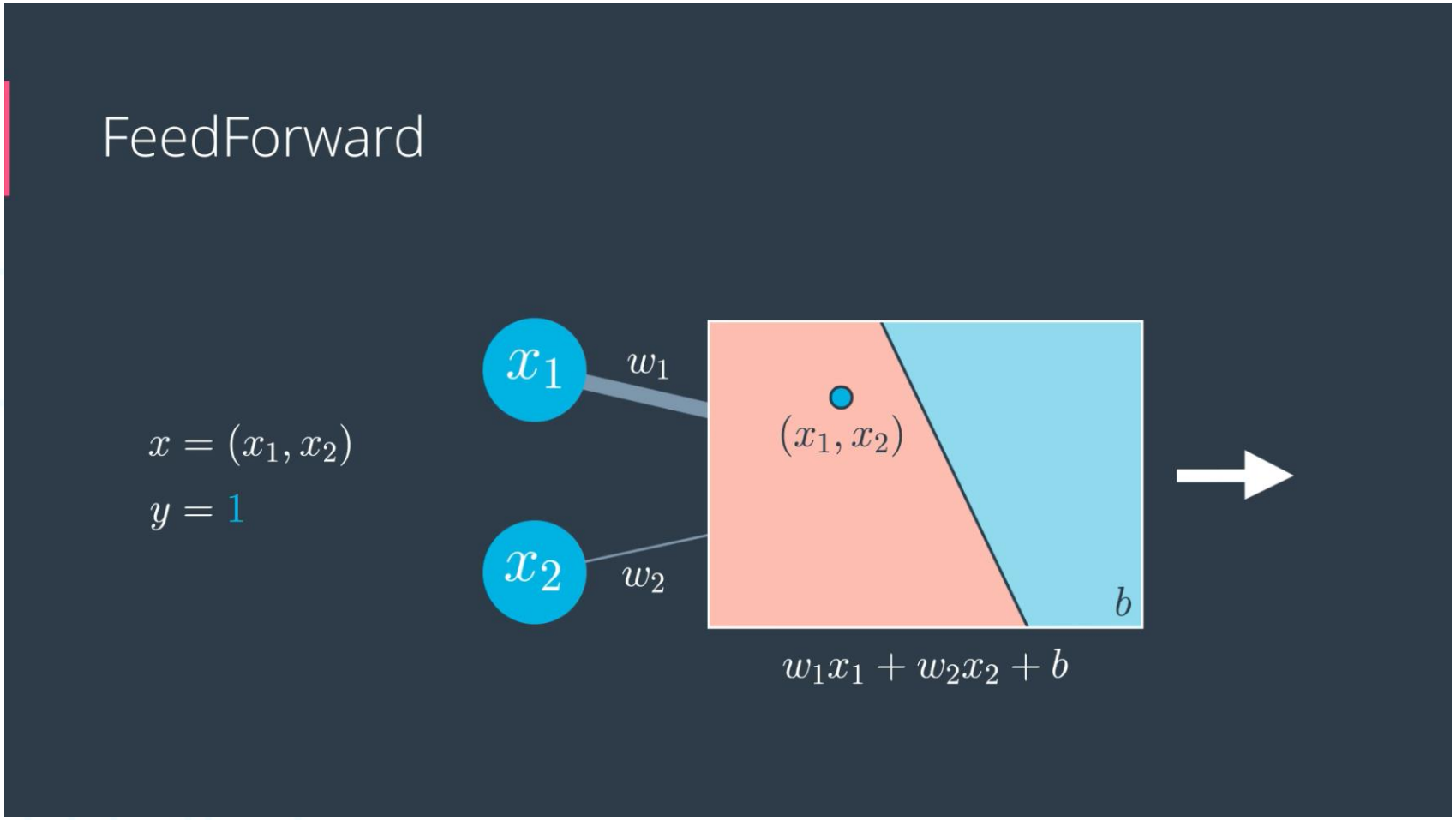
$$\hat{y} = \sigma \circ W^{(3)} \circ \sigma \circ W^{(2)} \circ \sigma \circ W^{(1)}(x)$$

ERROR FUNCTION

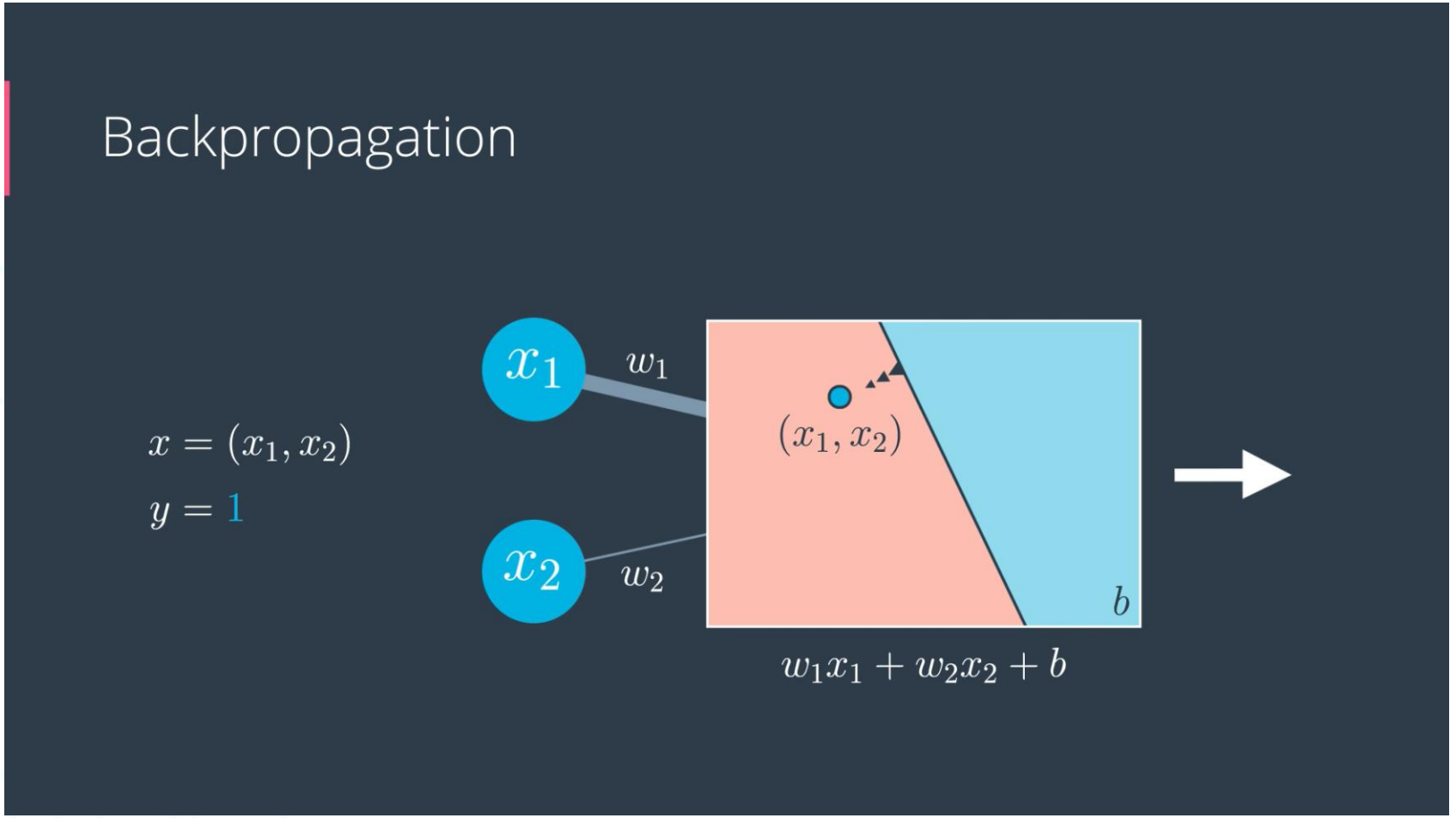
$$E(W) = -\frac{1}{m}\sum_{i=1}^{m} y_i ln(\hat{y}_i) + (1 - y_i) ln(1 - \hat{y}_i)$$



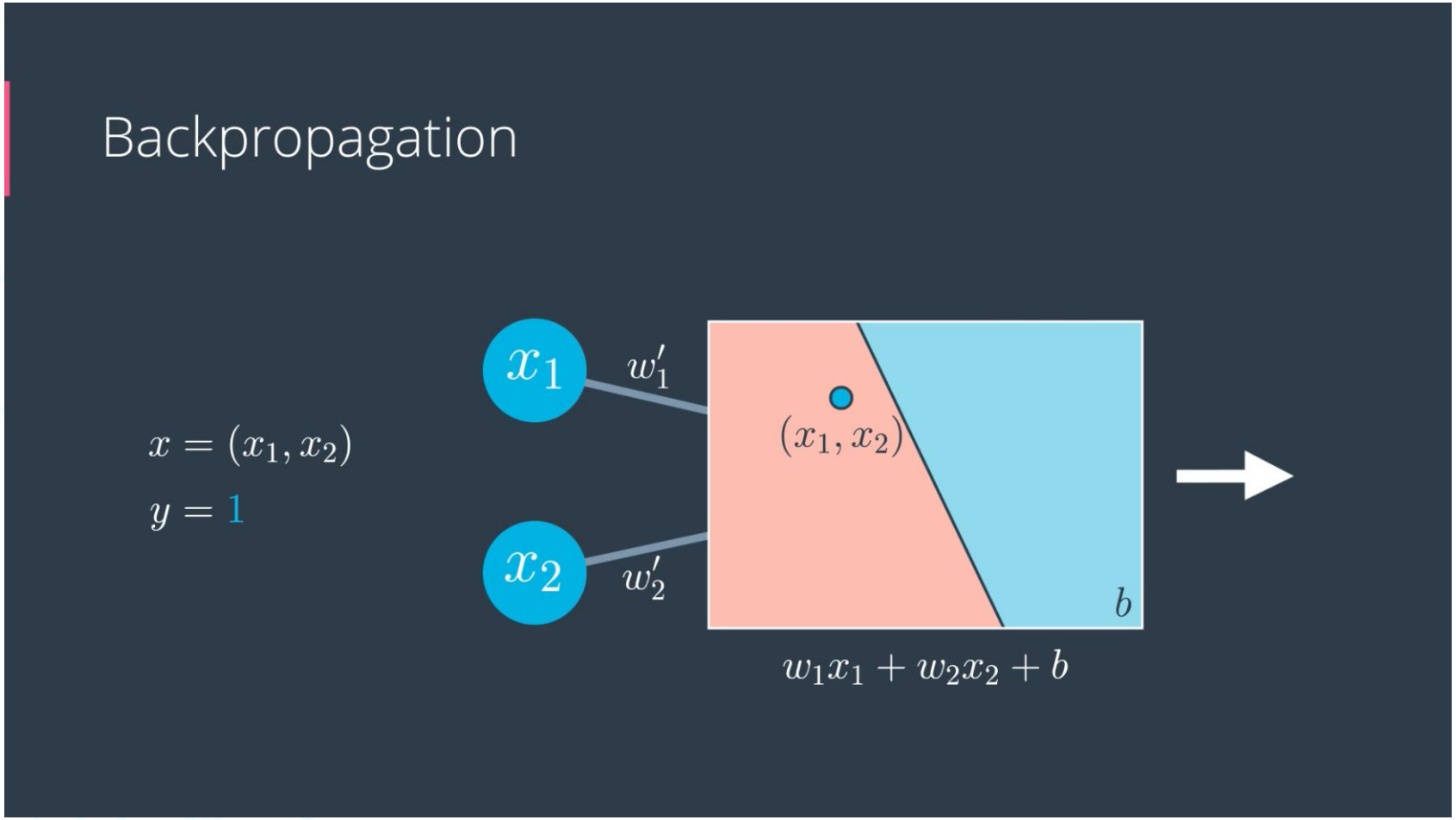
Redes Neurais – Backpropagation



Redes Neurais – Backpropagation

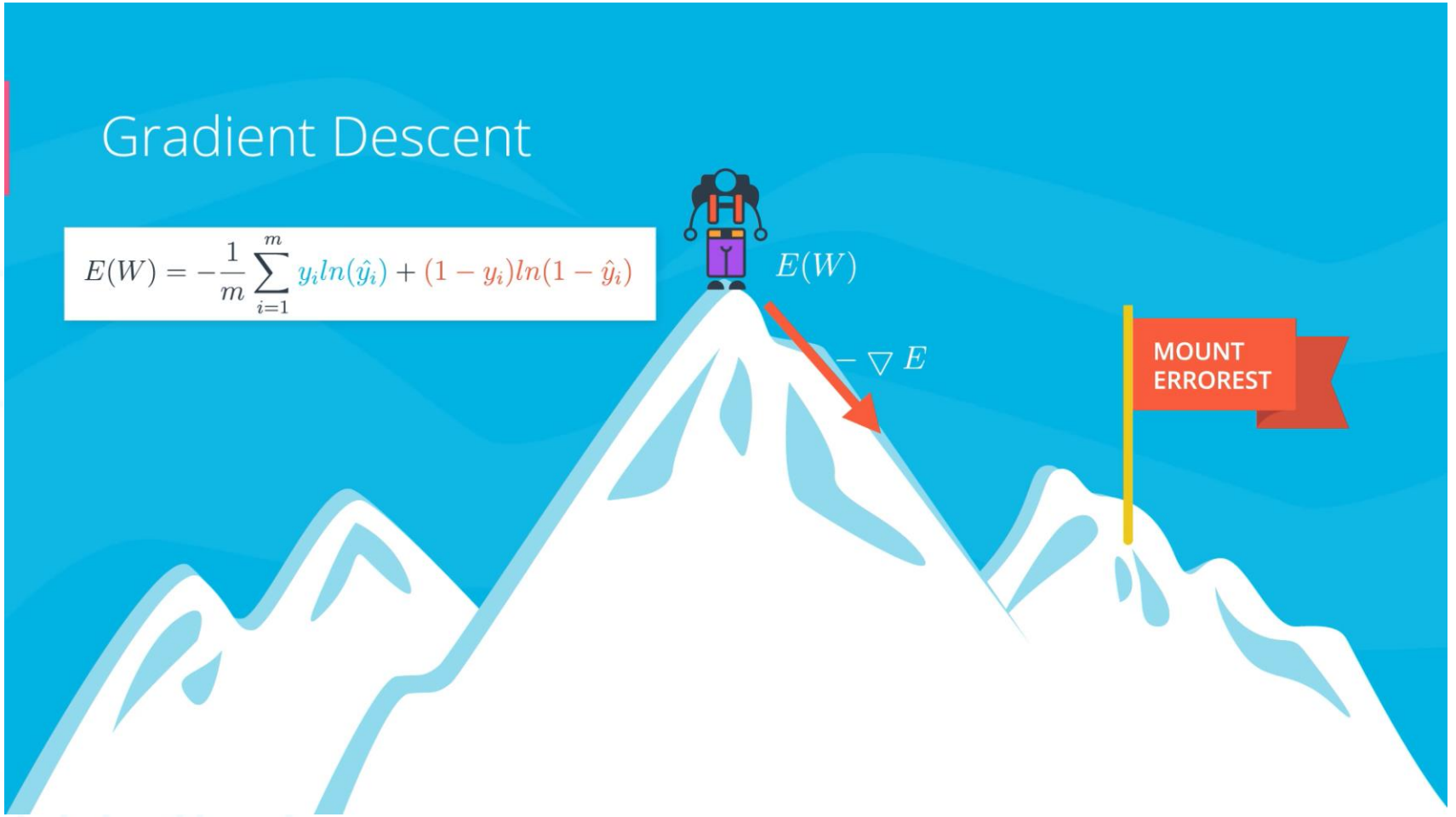


Redes Neurais – Backpropagation

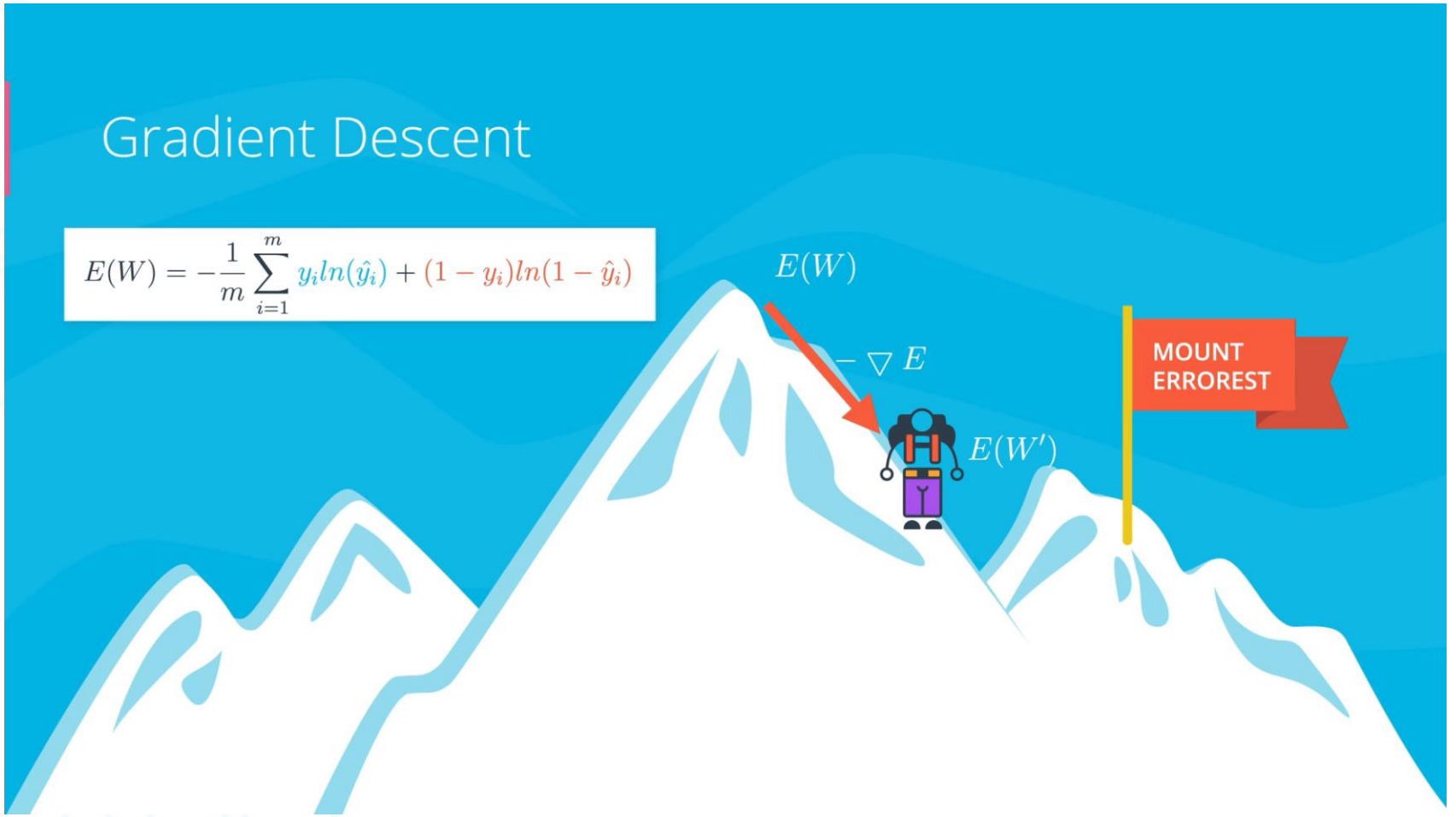


Redes Neurais – Backpropagation

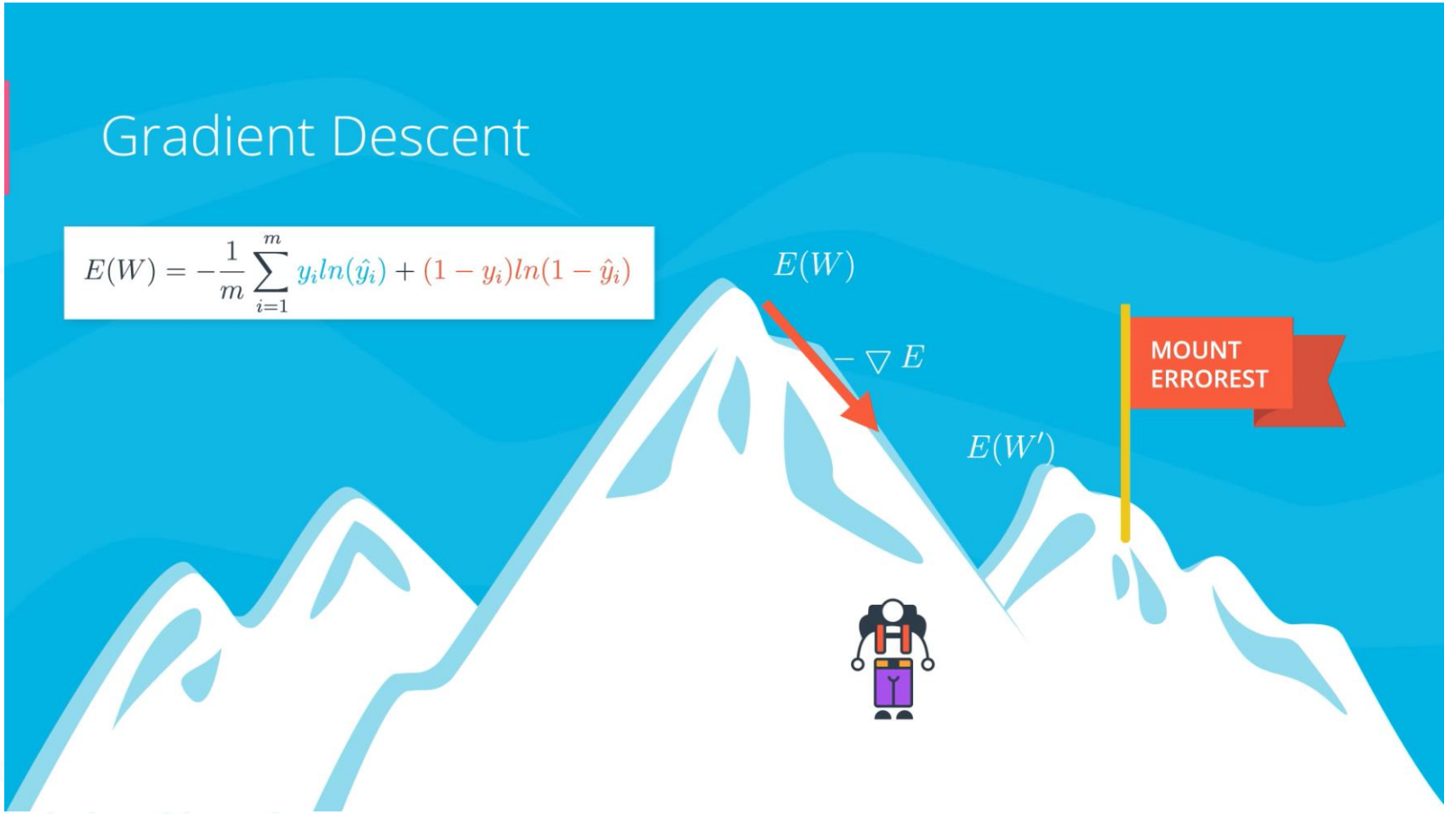
Redes Neurais – Backpropagation



Redes Neurais – Backpropagation

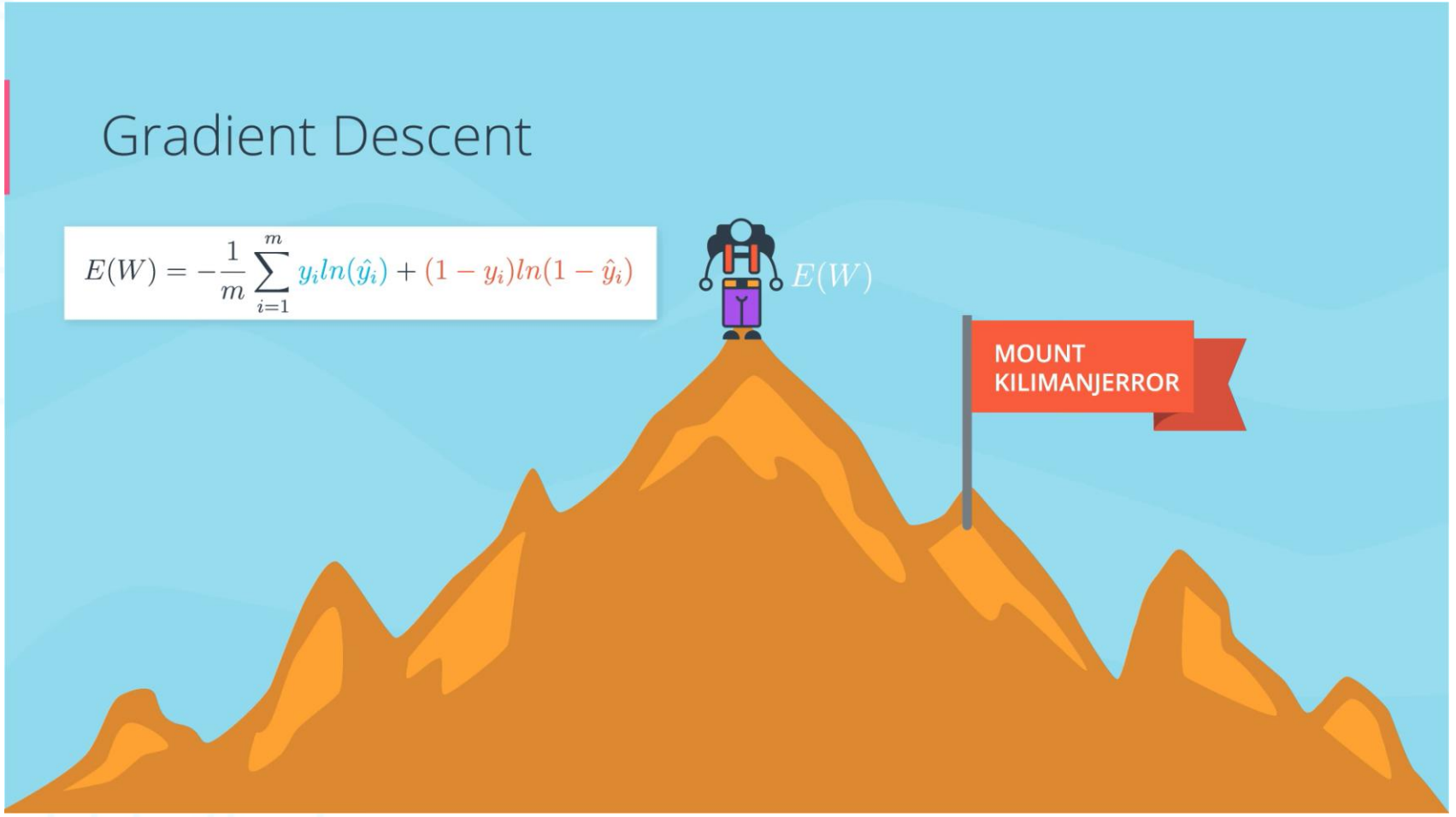


Redes Neurais – Backpropagation

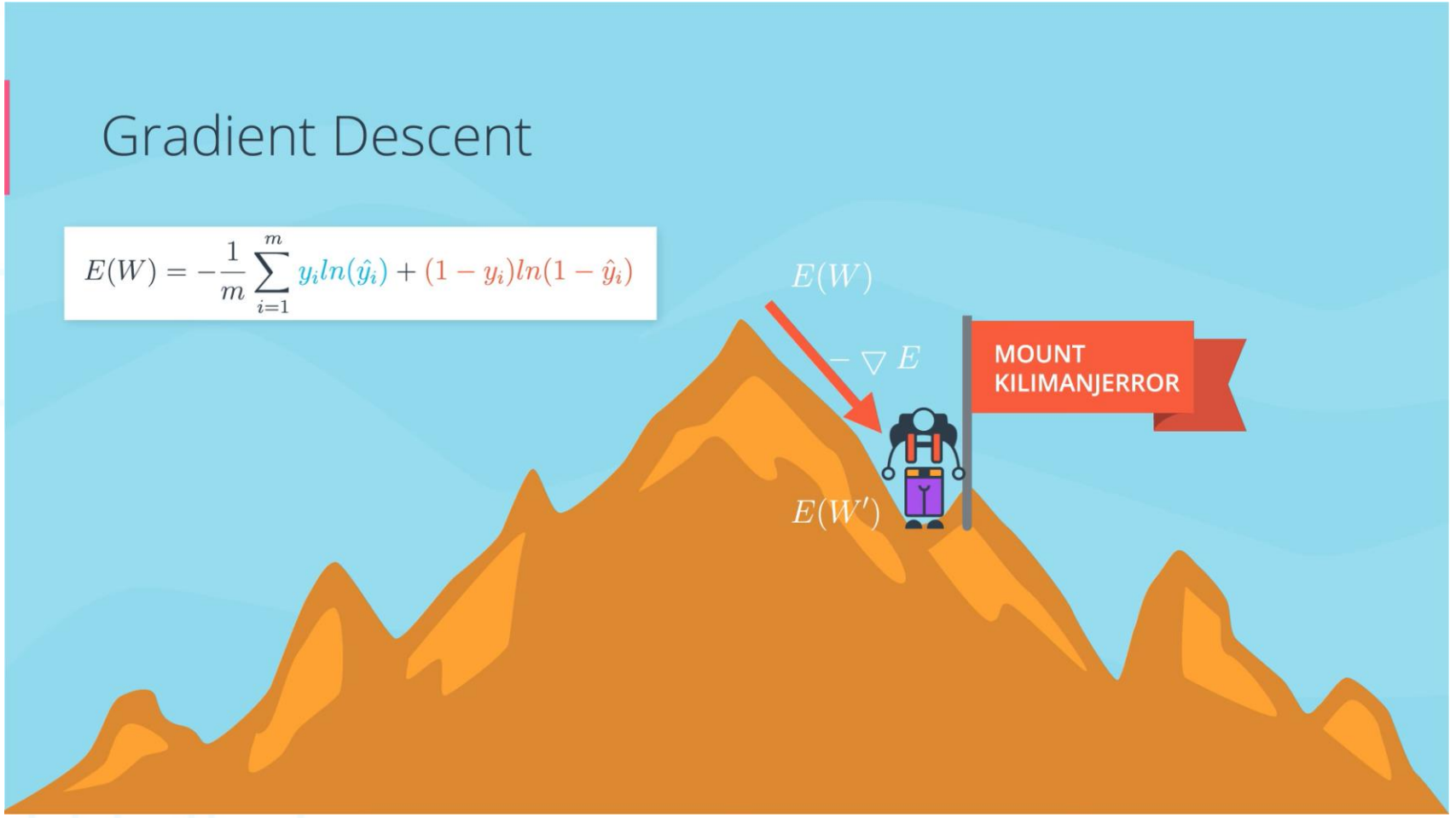


What about Multi-Layer Perceptrons?

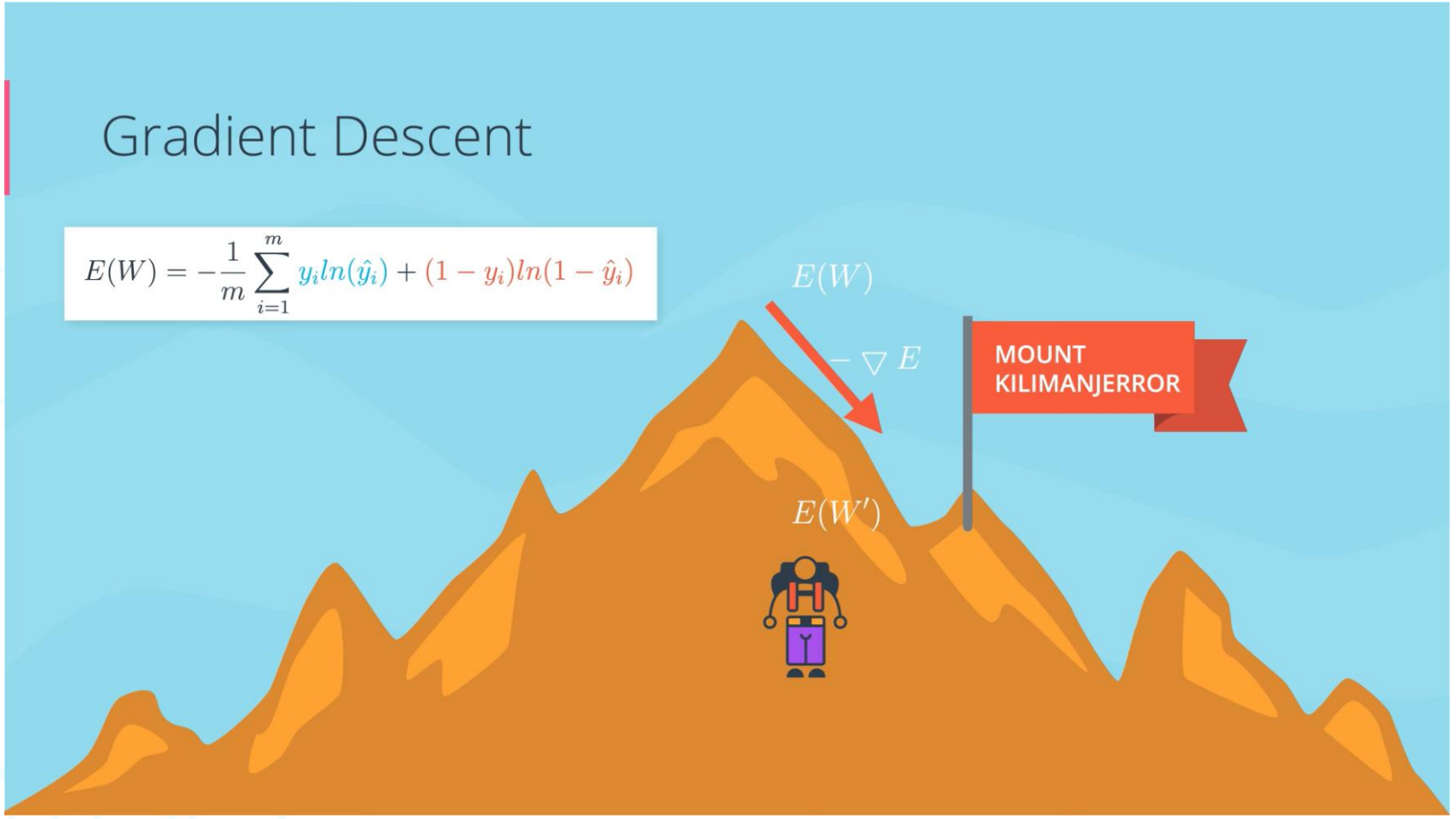
Redes Neurais – Backpropagation



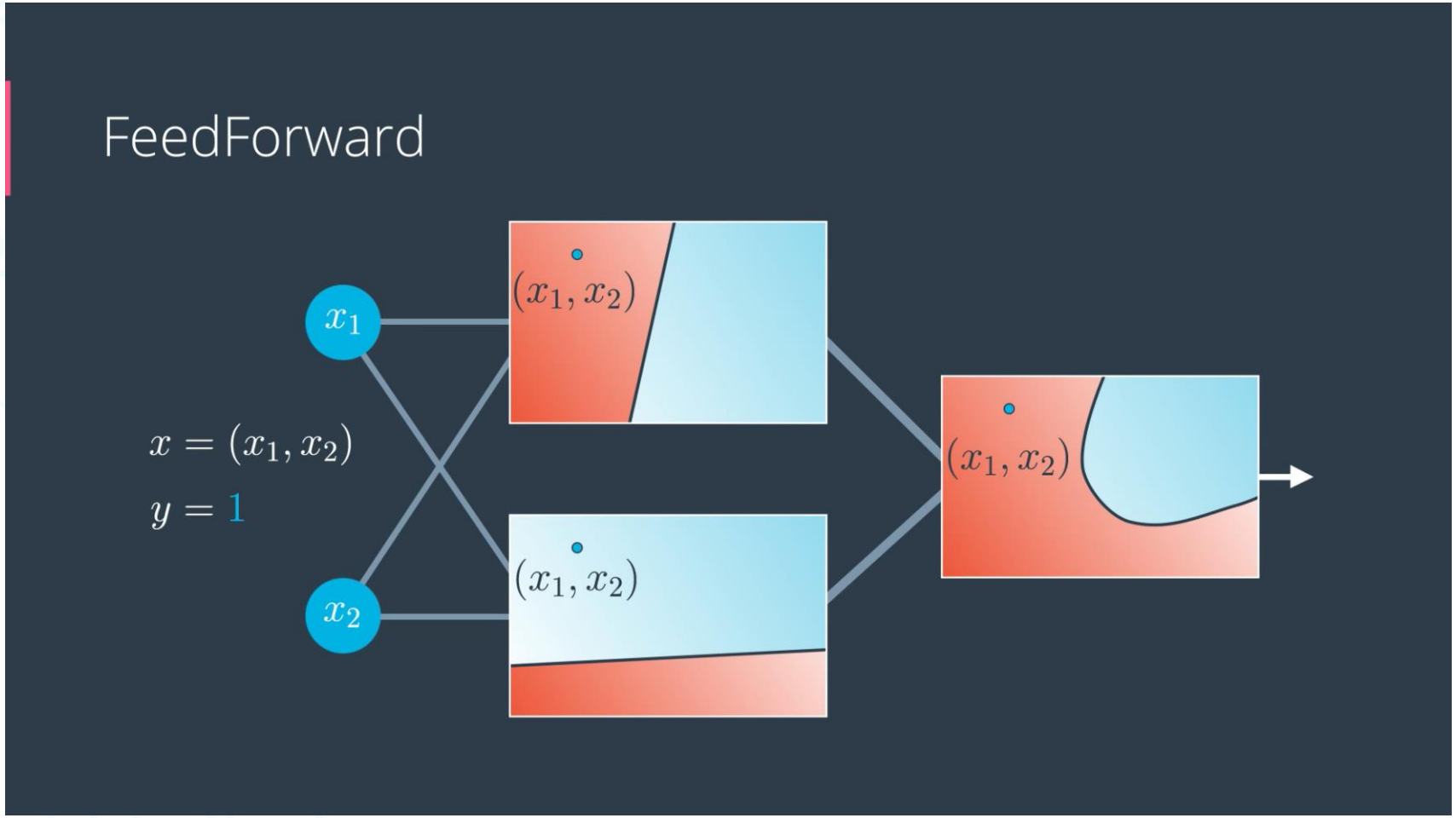
Redes Neurais – Backpropagation



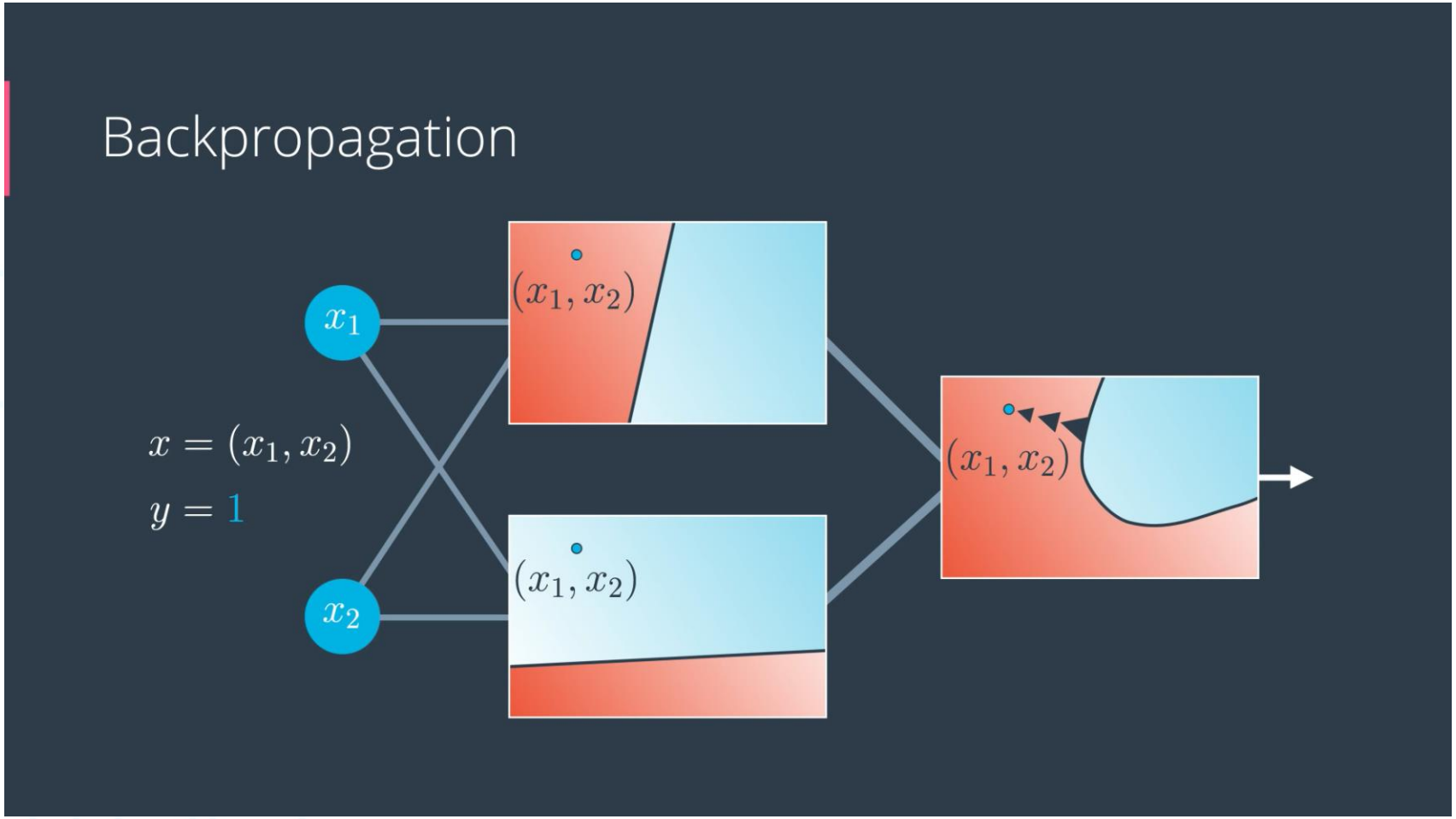
Redes Neurais – Backpropagation



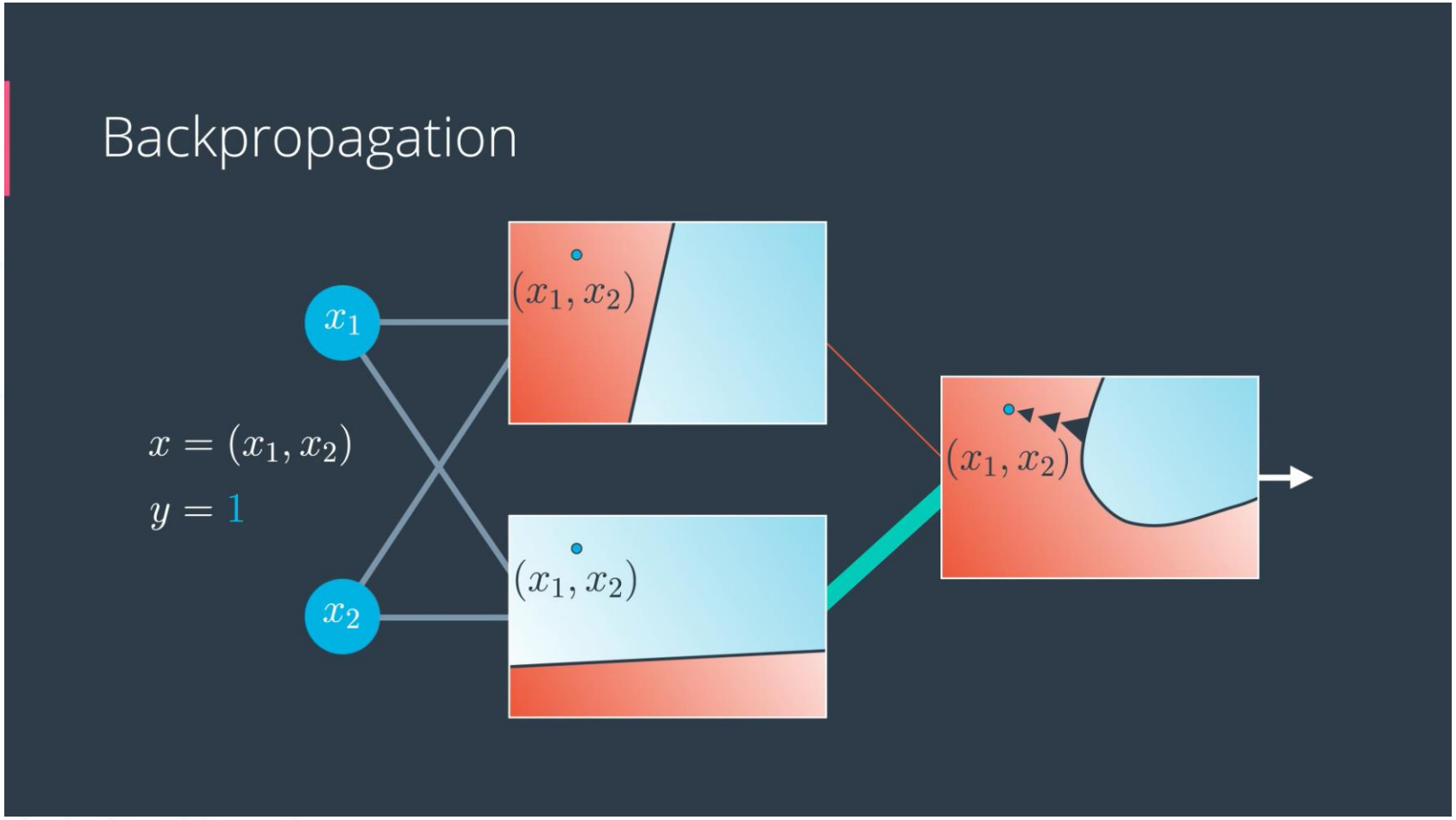
Redes Neurais – Backpropagation



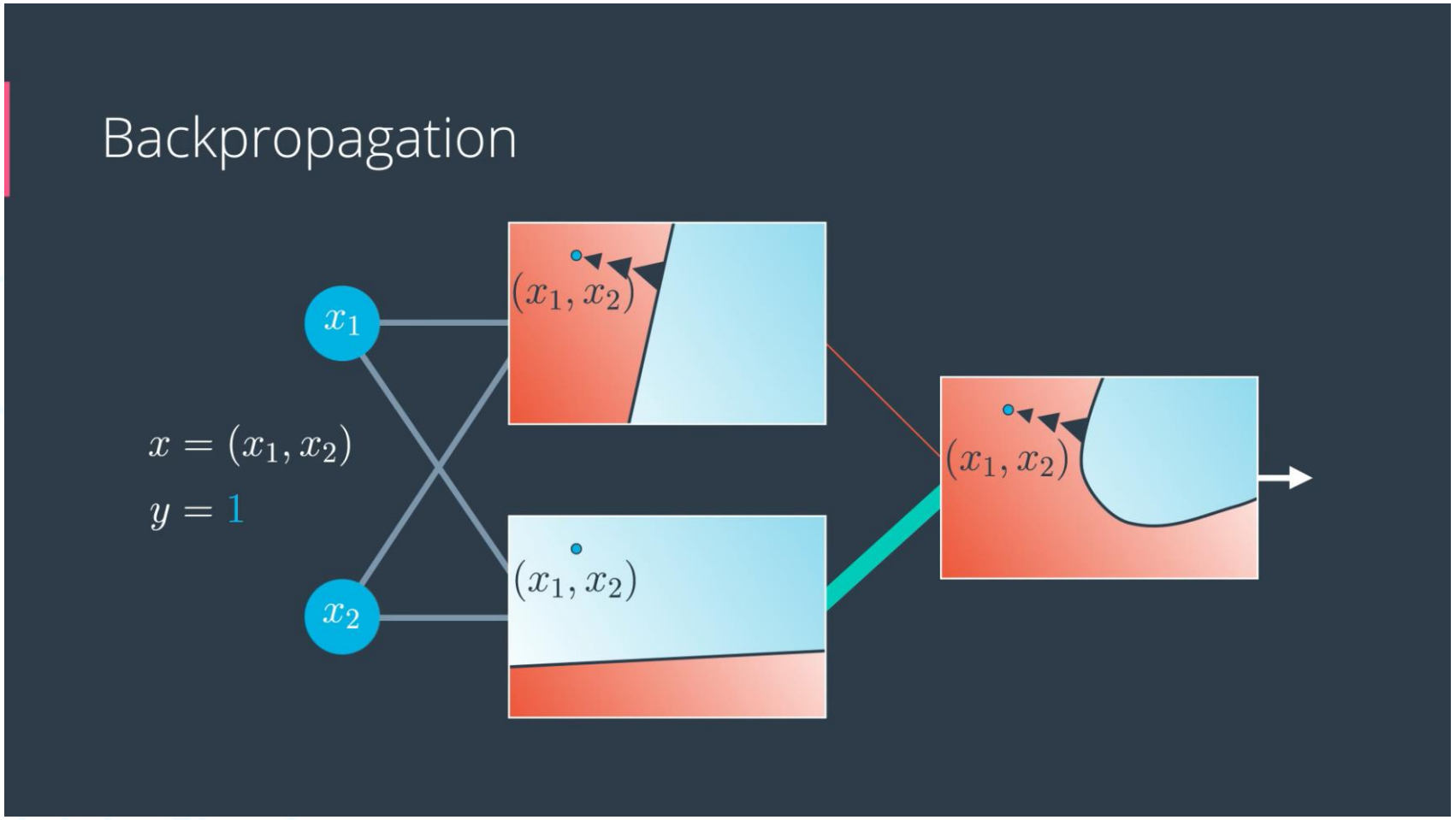
Redes Neurais – Backpropagation



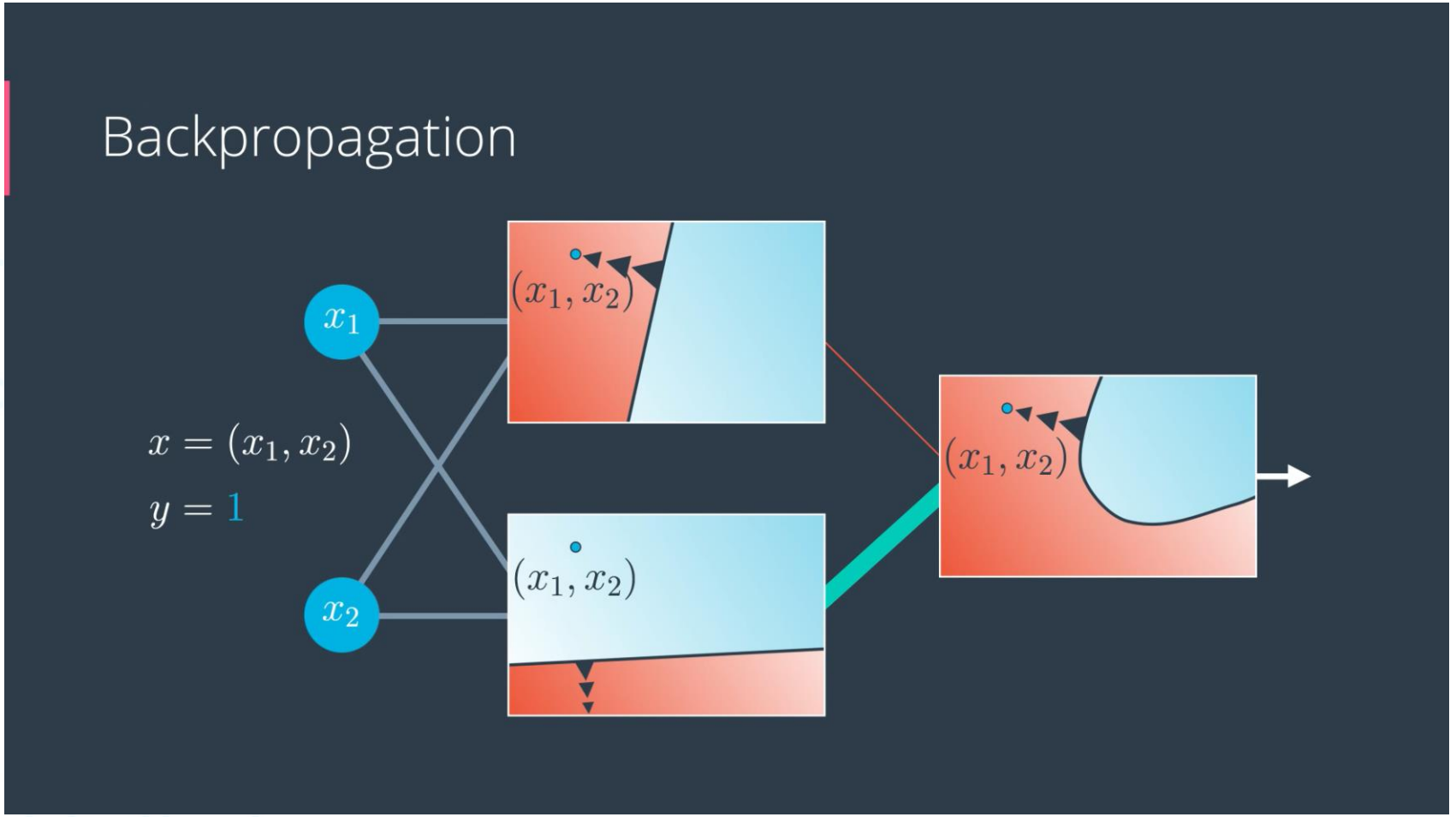
Redes Neurais – Backpropagation



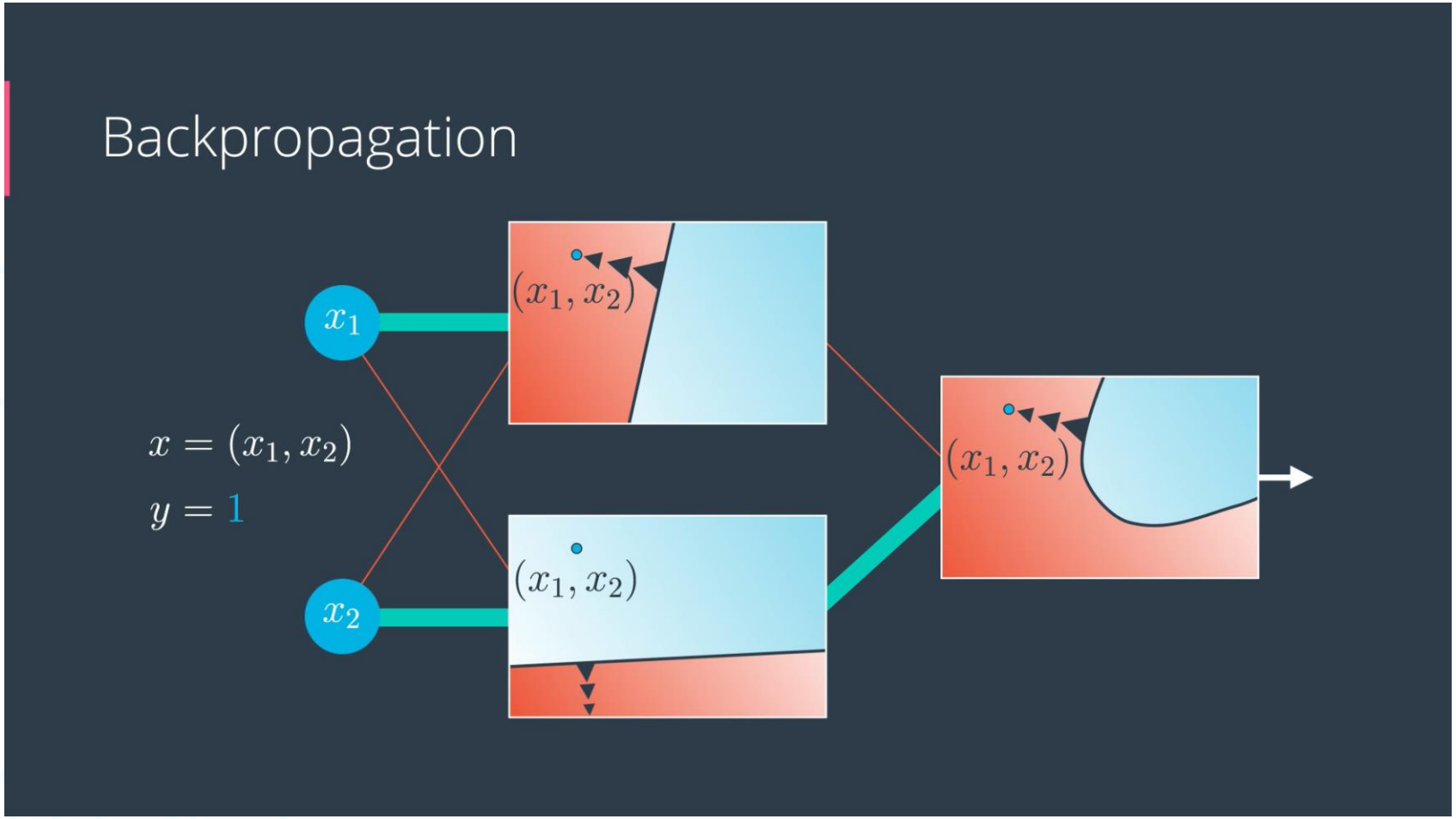
Redes Neurais – Backpropagation



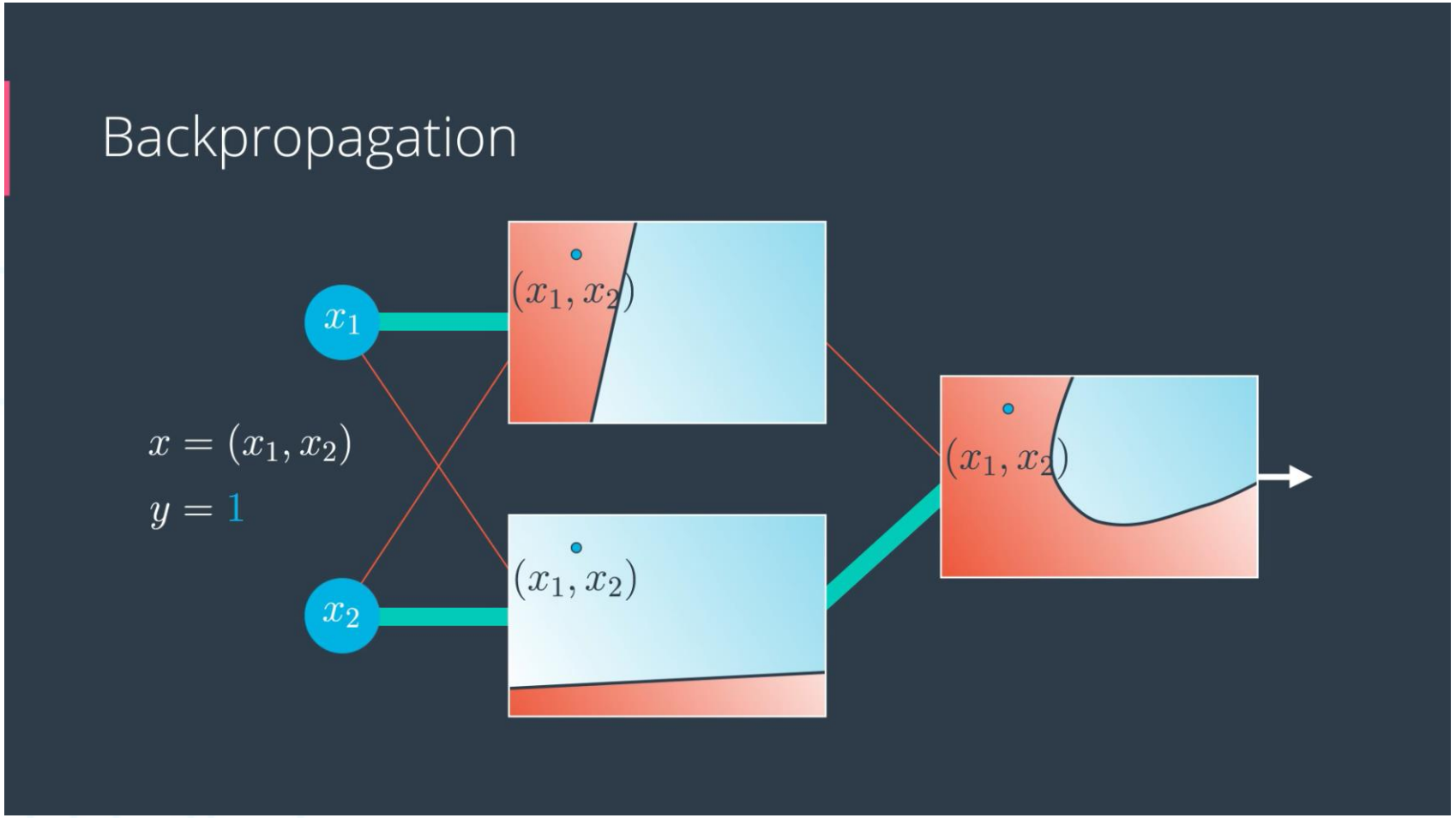
Redes Neurais – Backpropagation



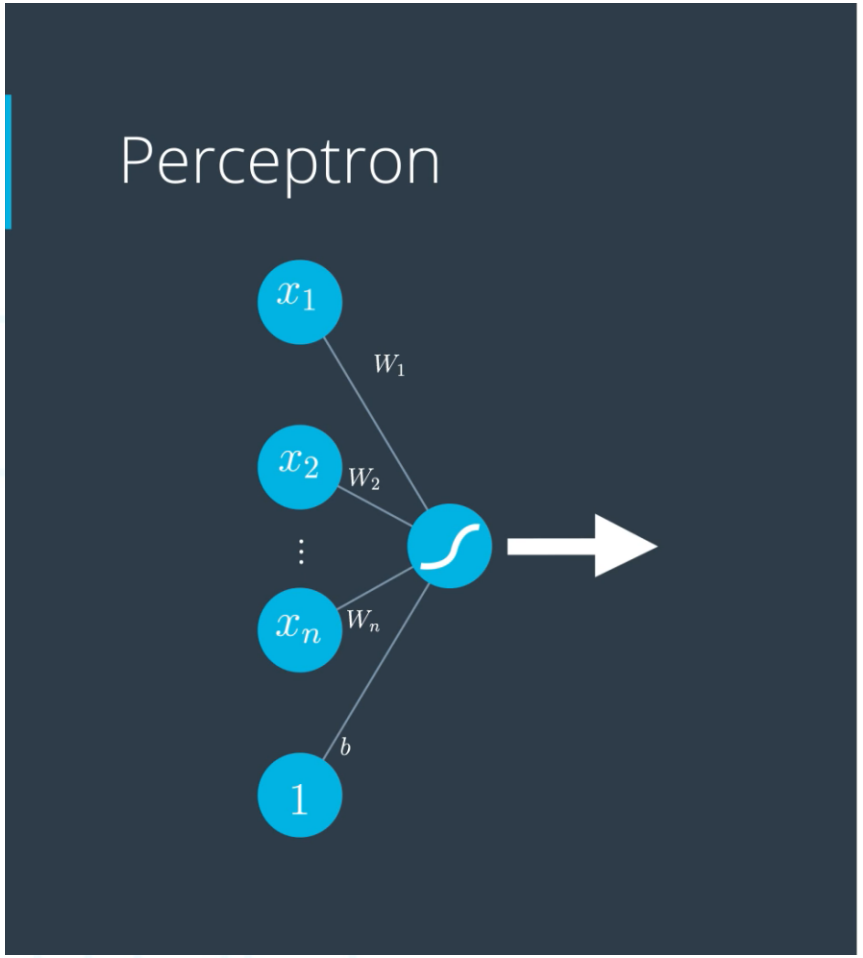
Redes Neurais – Backpropagation



Redes Neurais – Backpropagation



Redes Neurais – Backpropagation

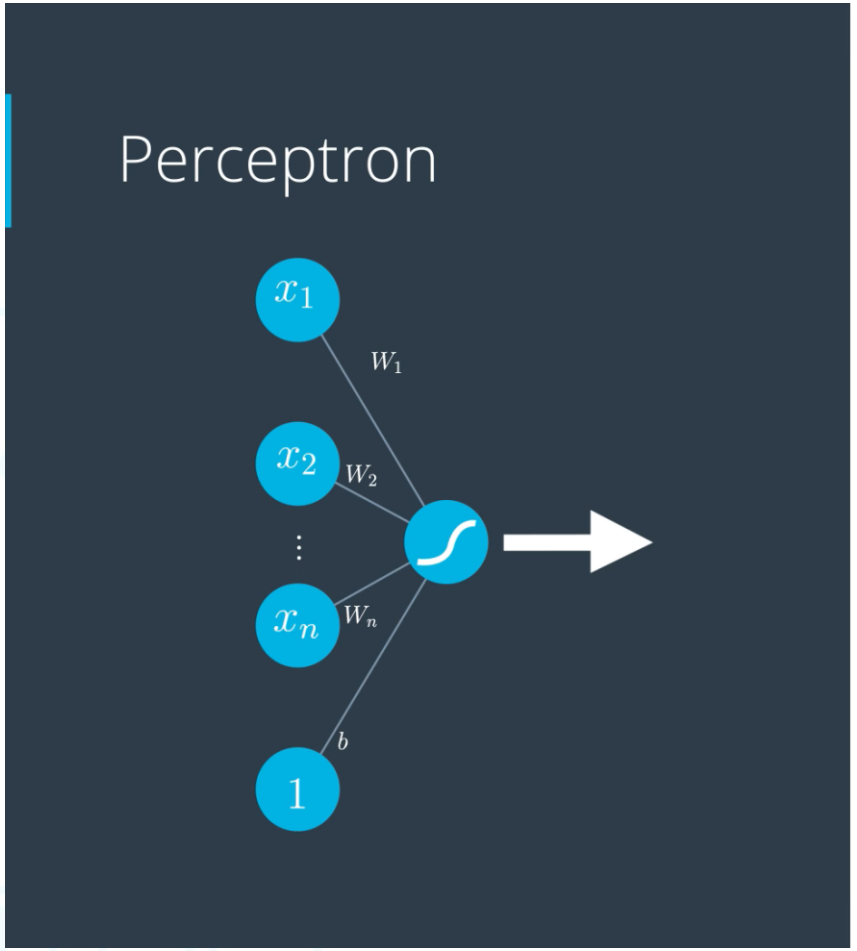


PREDICTION

ERROR FUNCTION

GRADIENT OF THE ERROR FUNCTION

Redes Neurais – Backpropagation



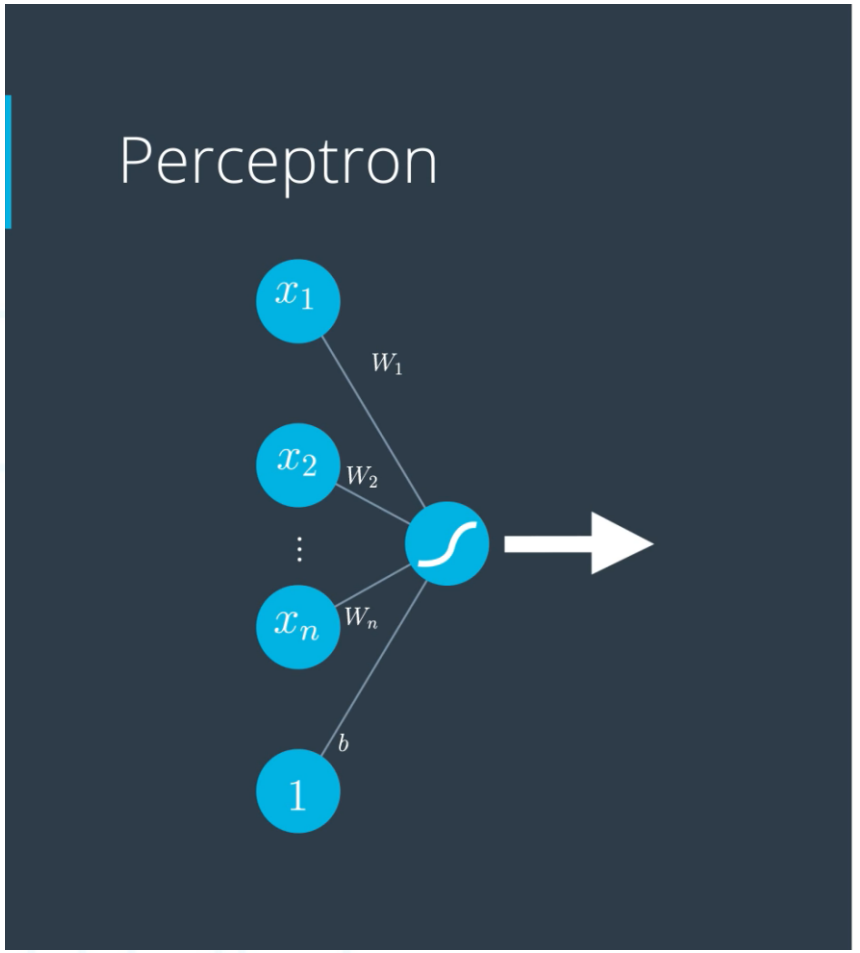
PREDICTION

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

ERROR FUNCTION

GRADIENT OF THE ERROR FUNCTION

Redes Neurais – Backpropagation



PREDICTION

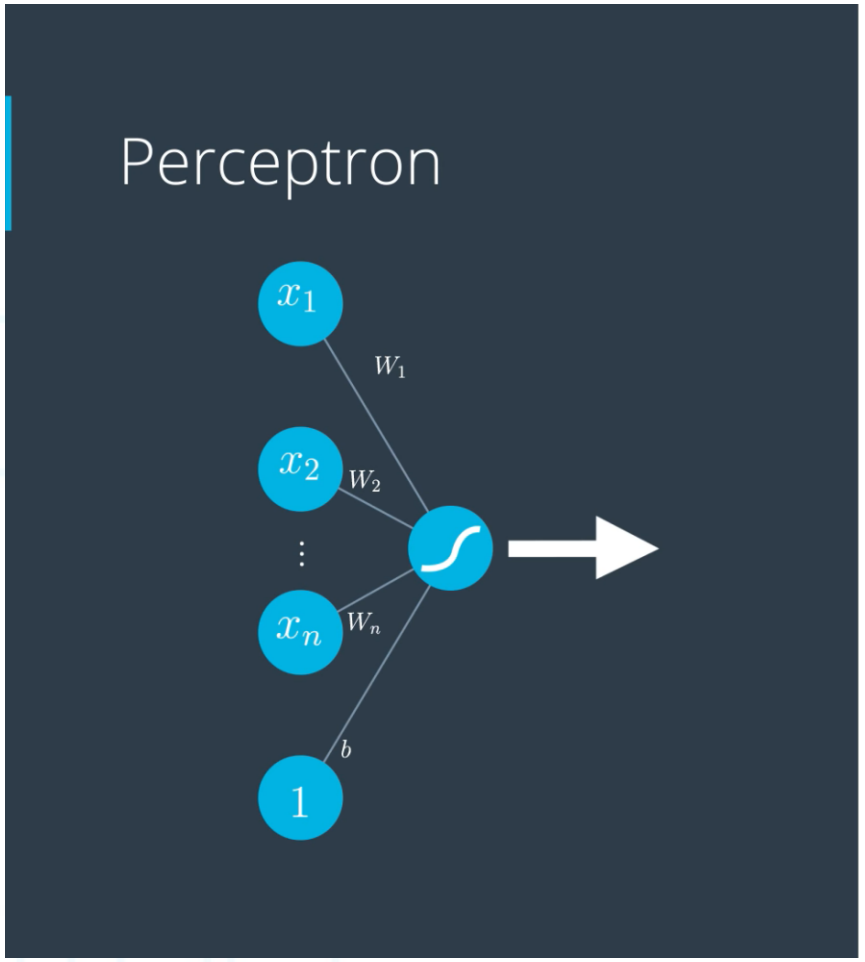
$$\hat{y} = \sigma(Wx + b)$$

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



PREDICTION

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

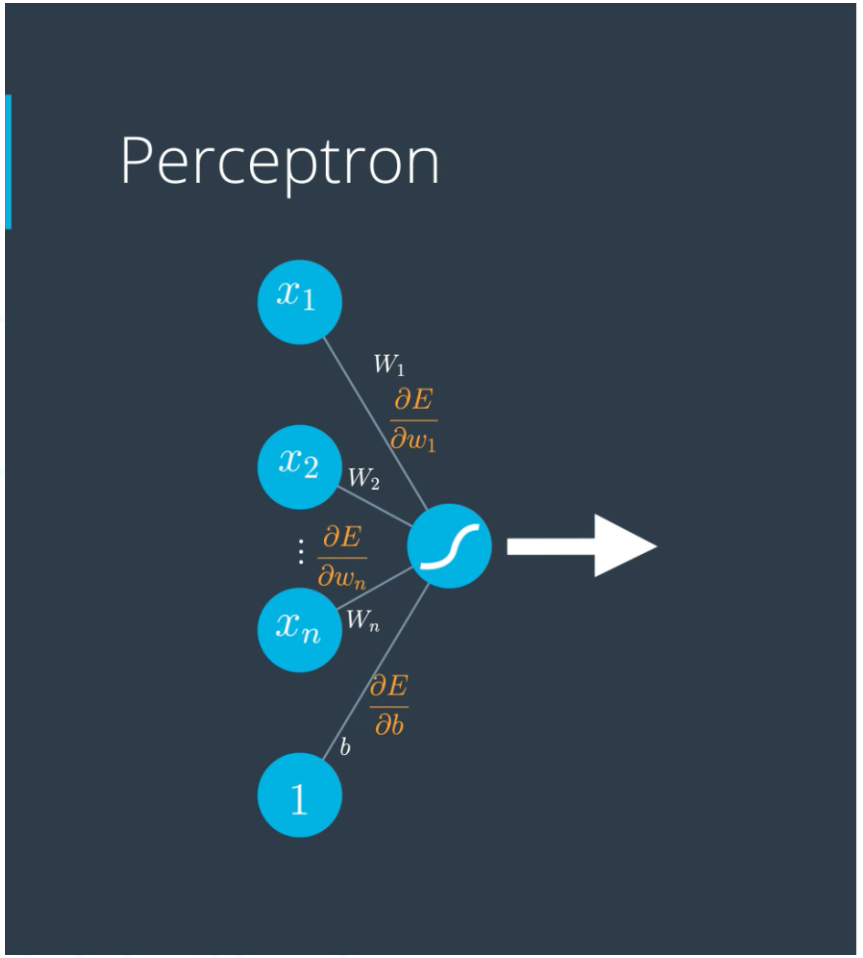
ERROR FUNCTION

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

GRADIENT OF THE ERROR FUNCTION

$$\nabla E = (\frac{\partial E}{\partial w_1}, ..., \frac{\partial E}{\partial w_n}, \frac{\partial E}{\partial b})$$

Redes Neurais – Backpropagation



PREDICTION

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

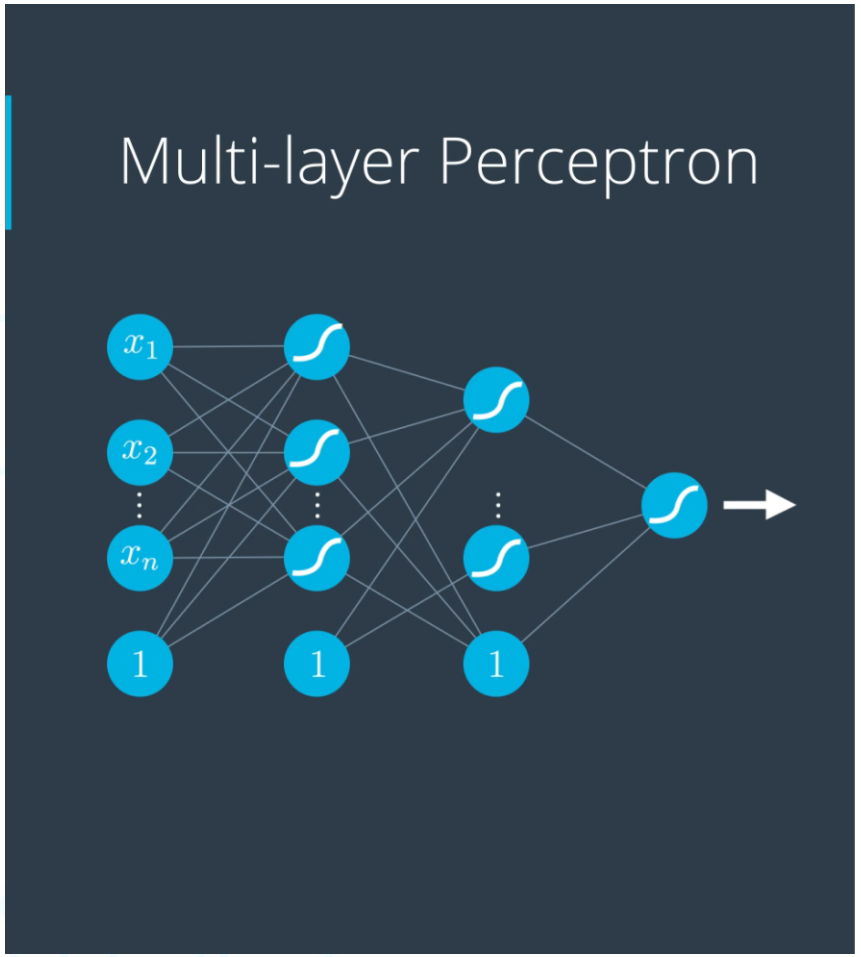
ERROR FUNCTION

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

GRADIENT OF THE ERROR FUNCTION

$$\nabla E = (\frac{\partial E}{\partial w_1}, ..., \frac{\partial E}{\partial w_n}, \frac{\partial E}{\partial b})$$

Redes Neurais – Backpropagation



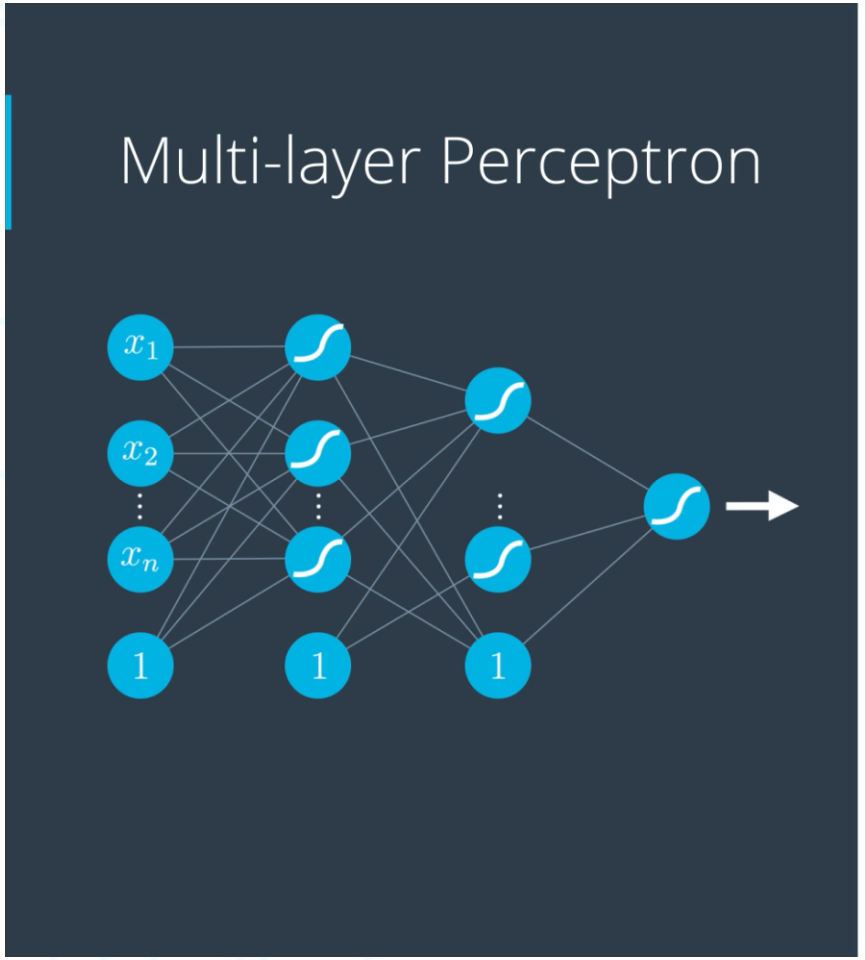
PREDICTION

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

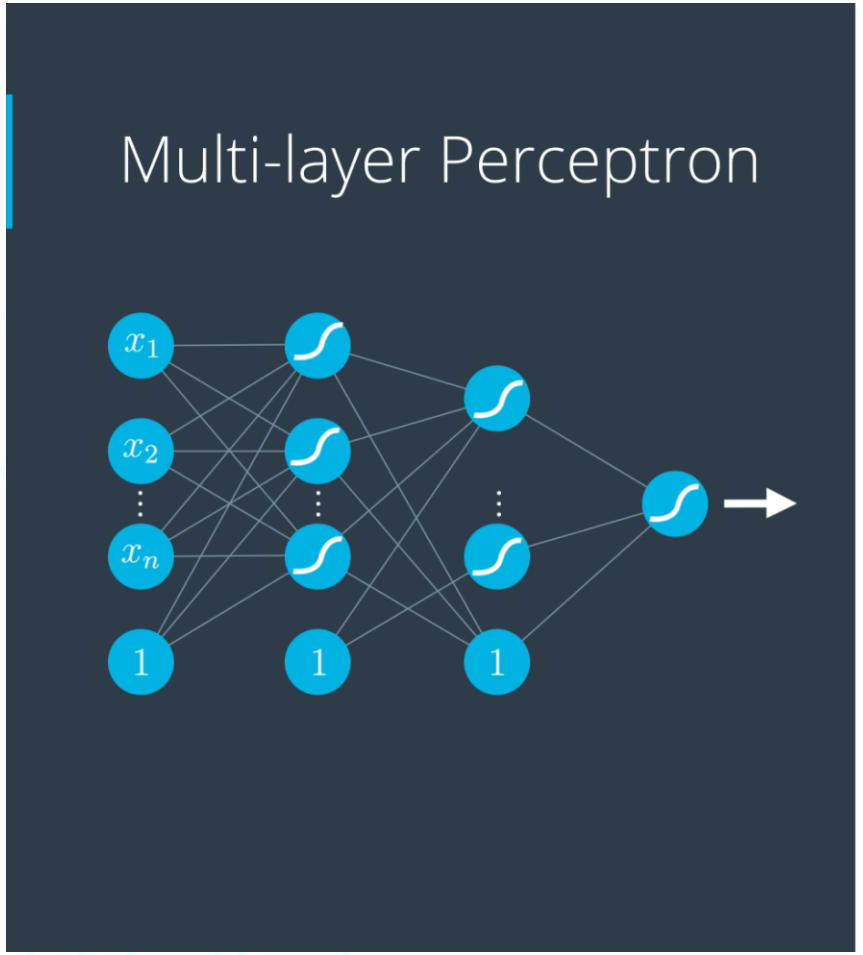
$$\hat{y} = \sigma W^{(3)} \circ \sigma W^{(2)} \circ \sigma \circ W^{(1)}(x)$$

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

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

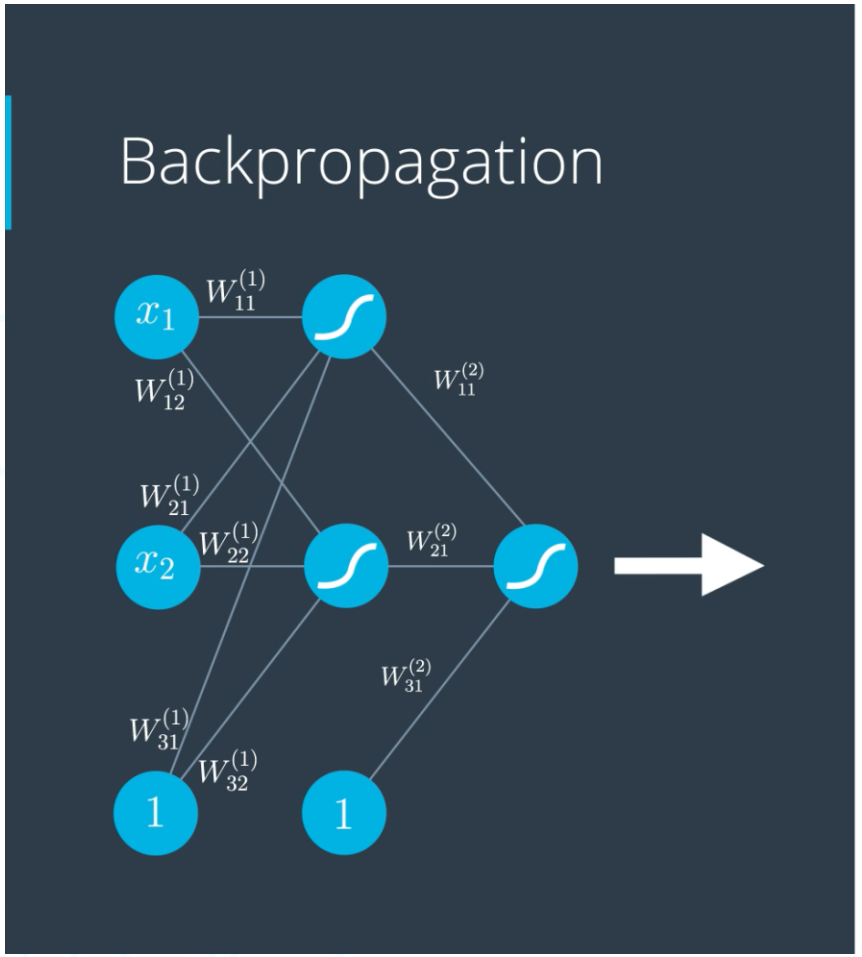
ERROR FUNCTION

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

GRADIENT OF THE ERROR FUNCTION

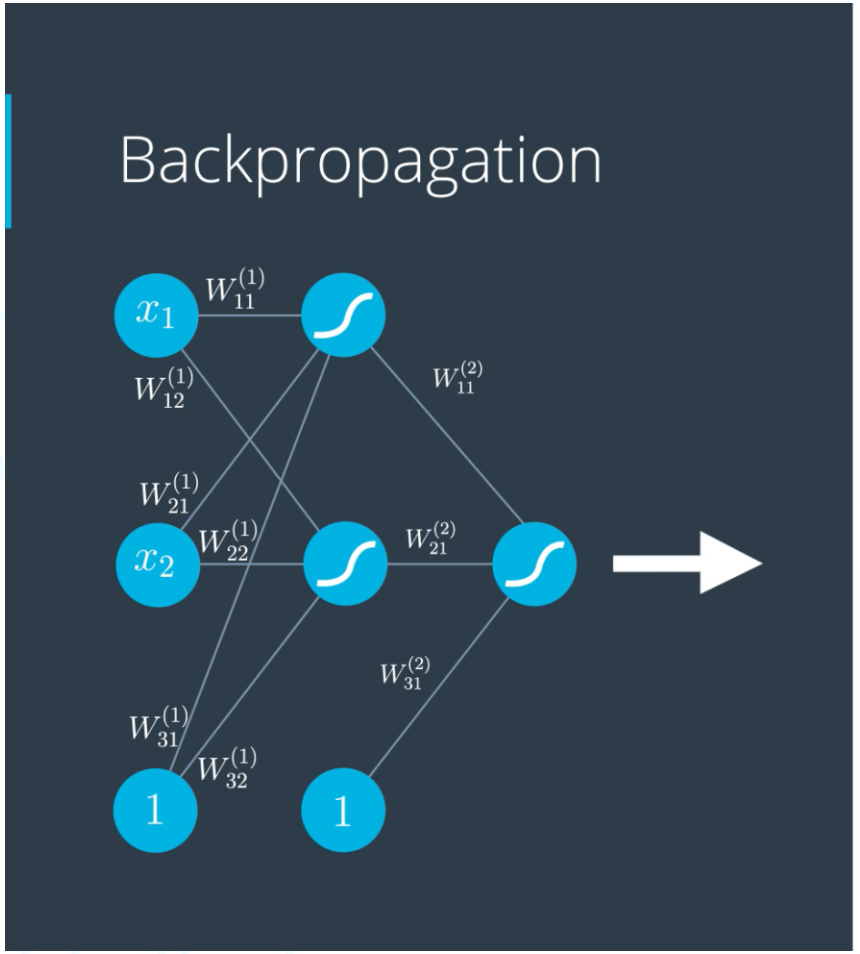
$$\nabla E = (..., \frac{\partial E}{\partial w_j^{(i)}}, ...)$$

Redes Neurais – Backpropagation



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

Redes Neurais – Backpropagation



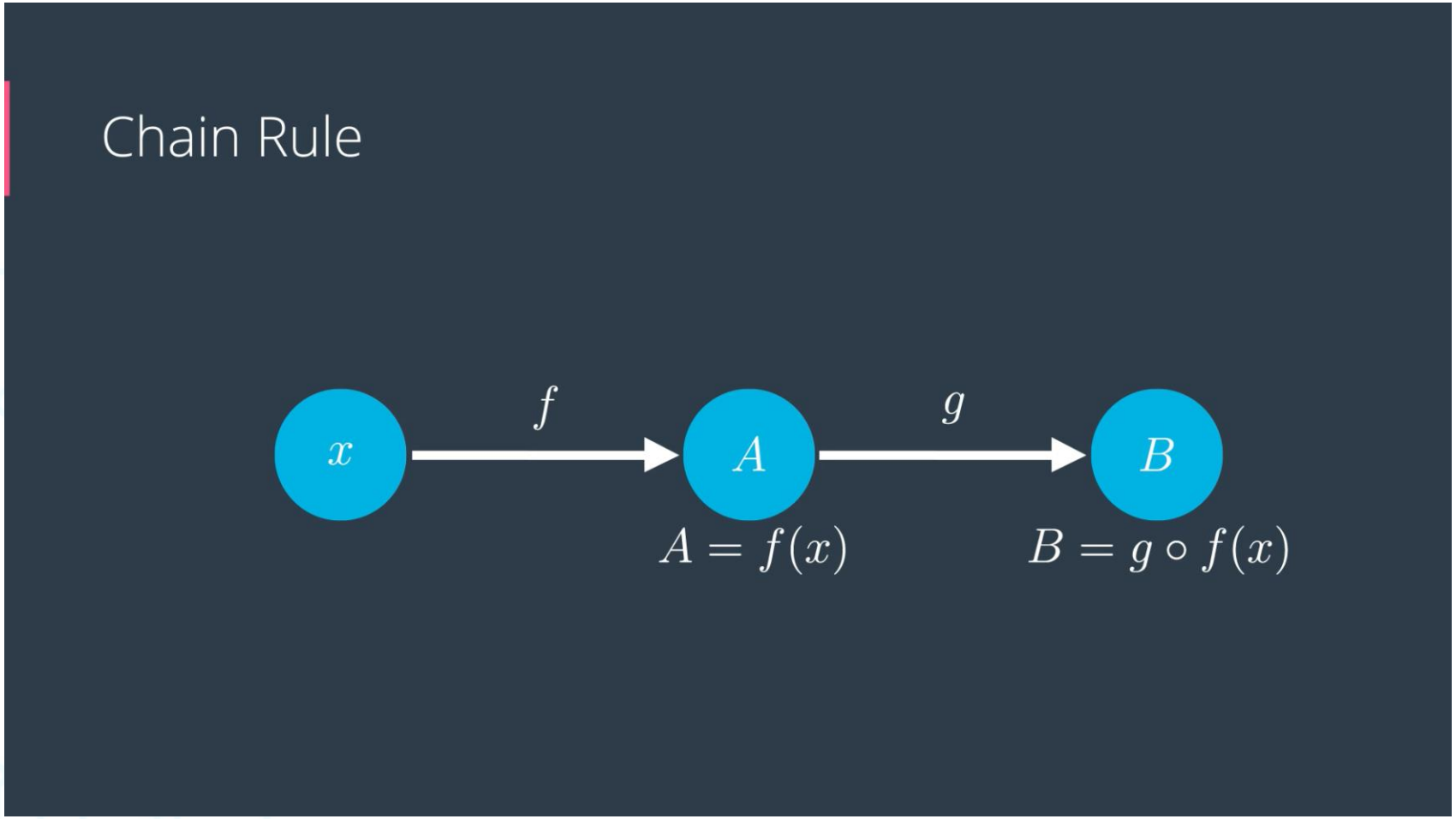
$$\hat{y} = \sigma W^{(2)} \circ \sigma \circ W^{(1)}(x)$$

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

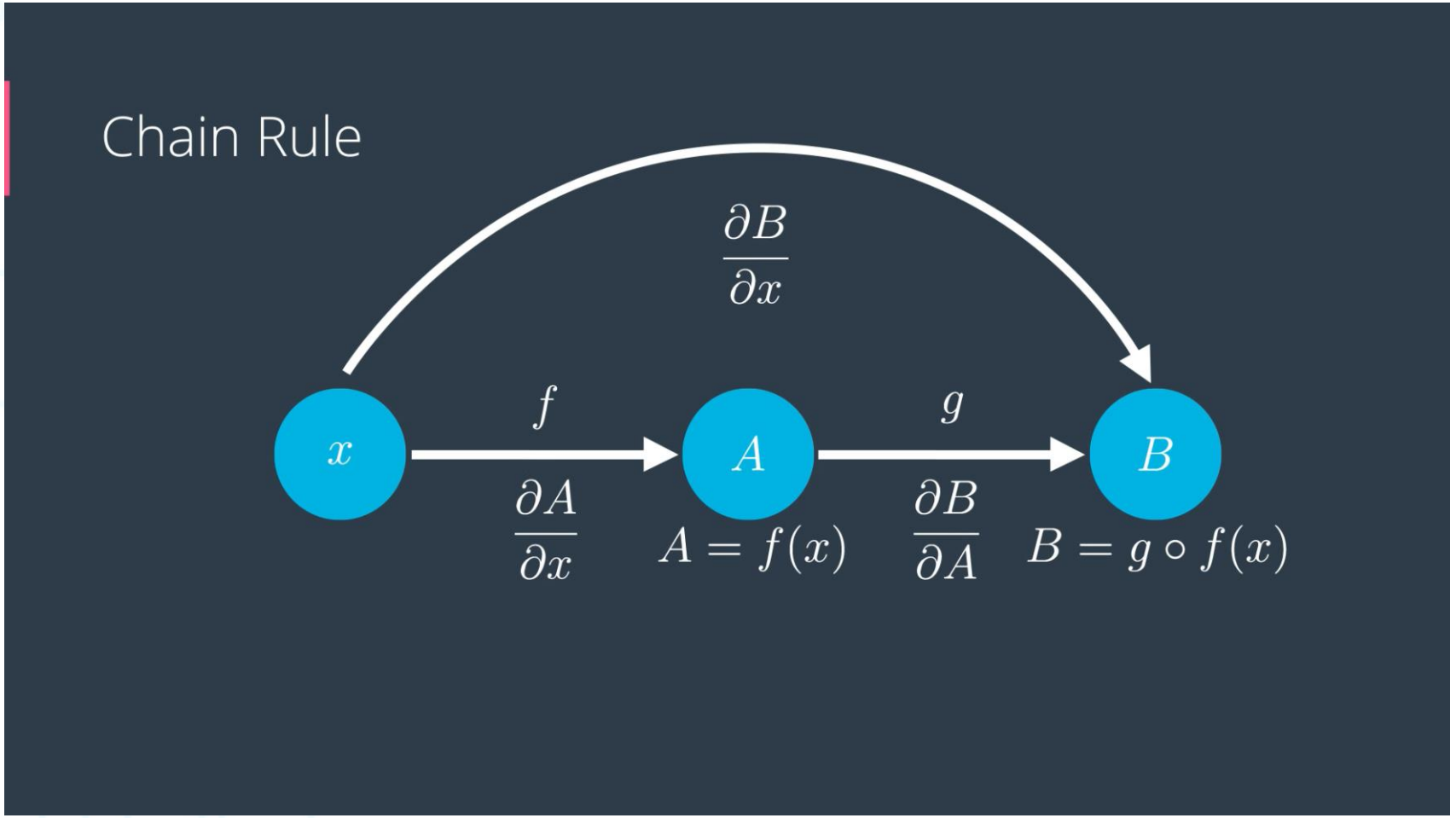
$$\nabla E = \begin{pmatrix} \frac{\partial E}{\partial W_{11}^{(1)}} & \frac{\partial E}{\partial W_{12}^{(1)}} & \frac{\partial E}{\partial W_{11}^{(2)}} \\ \frac{\partial E}{\partial W_{21}^{(1)}} & \frac{\partial E}{\partial W_{22}^{(1)}} & \frac{\partial E}{\partial W_{21}^{(2)}} \\ \frac{\partial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

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

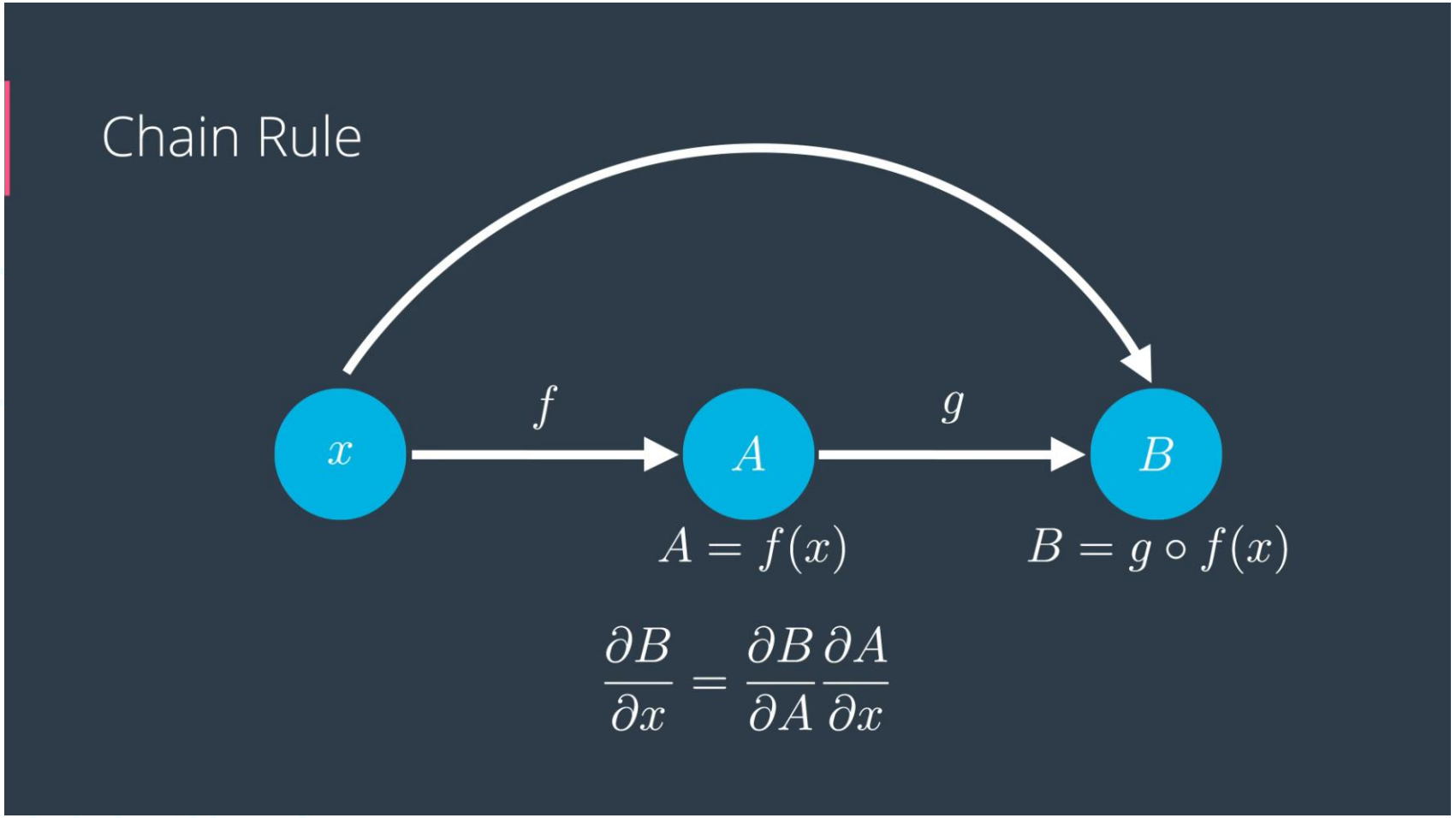
Redes Neurais – Backpropagation



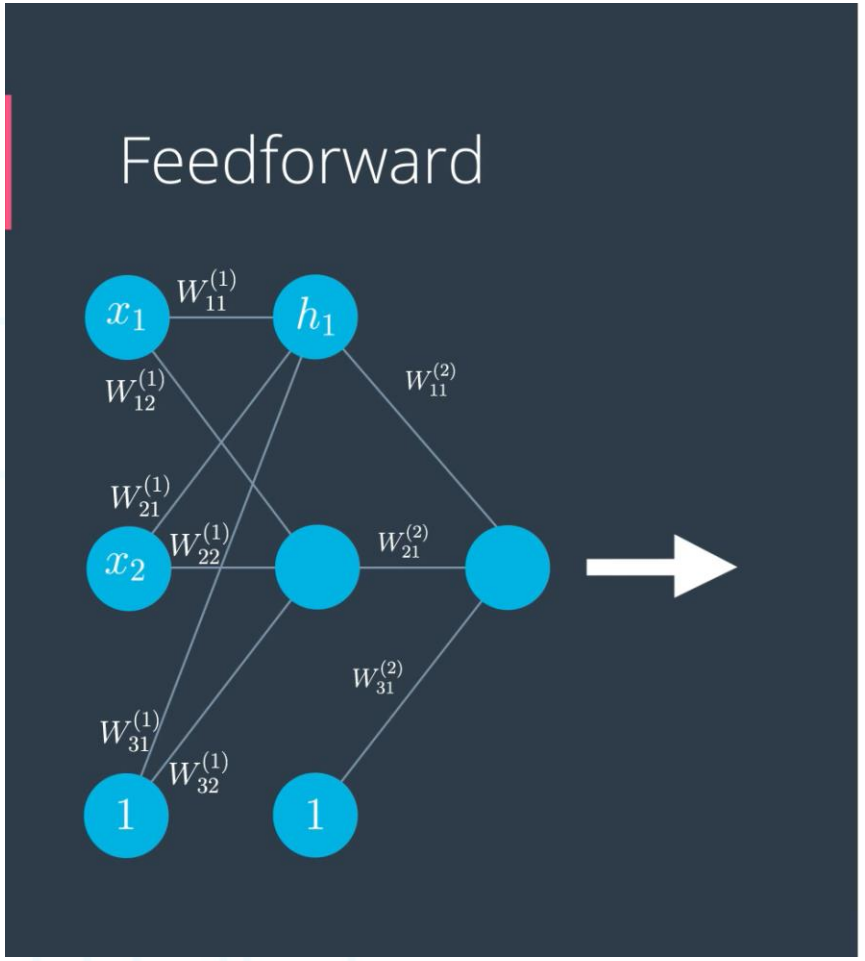
Redes Neurais – Backpropagation



Redes Neurais – Backpropagation

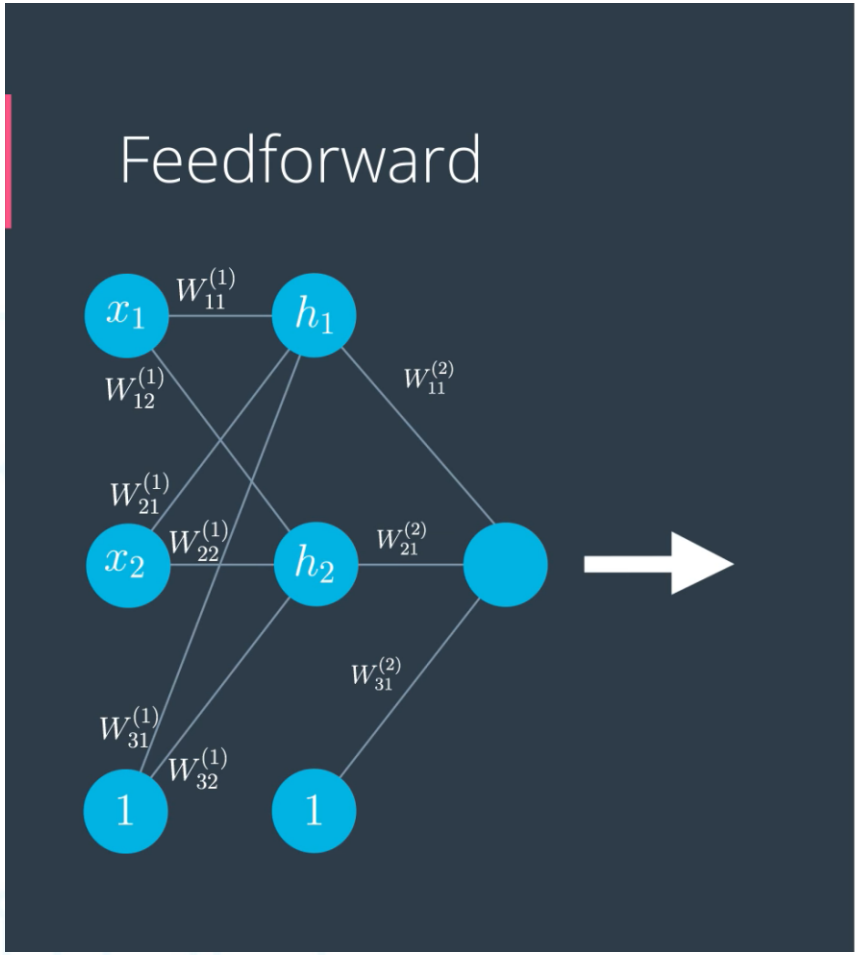


Redes Neurais – Backpropagation



$$h_1 = W_{11}^{(1)} x_1 + W_{21}^{(1)} x_2 + W_{31}^{(1)}$$

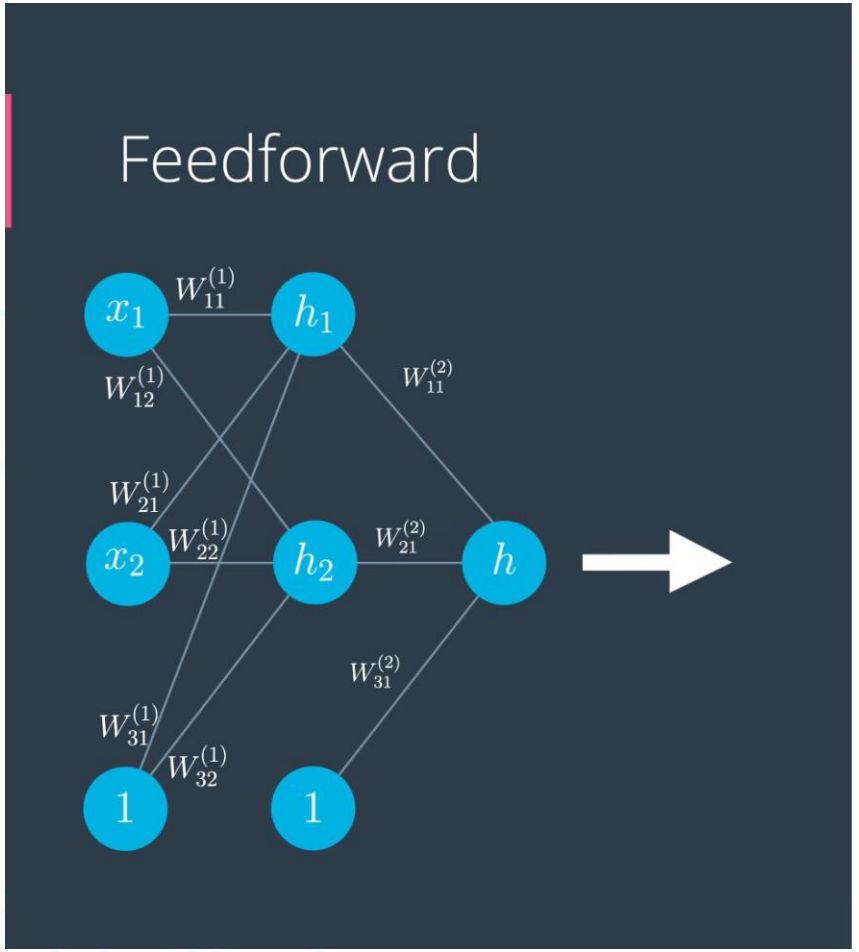
Redes Neurais – Backpropagation



$$h_1 = W_{11}^{(1)} x_1 + W_{21}^{(1)} x_2 + W_{31}^{(1)}$$

$$h_2 = W_{12}^{(1)} x_1 + W_{22}^{(1)} x_2 + W_{32}^{(1)}$$

Redes Neurais – Backpropagation

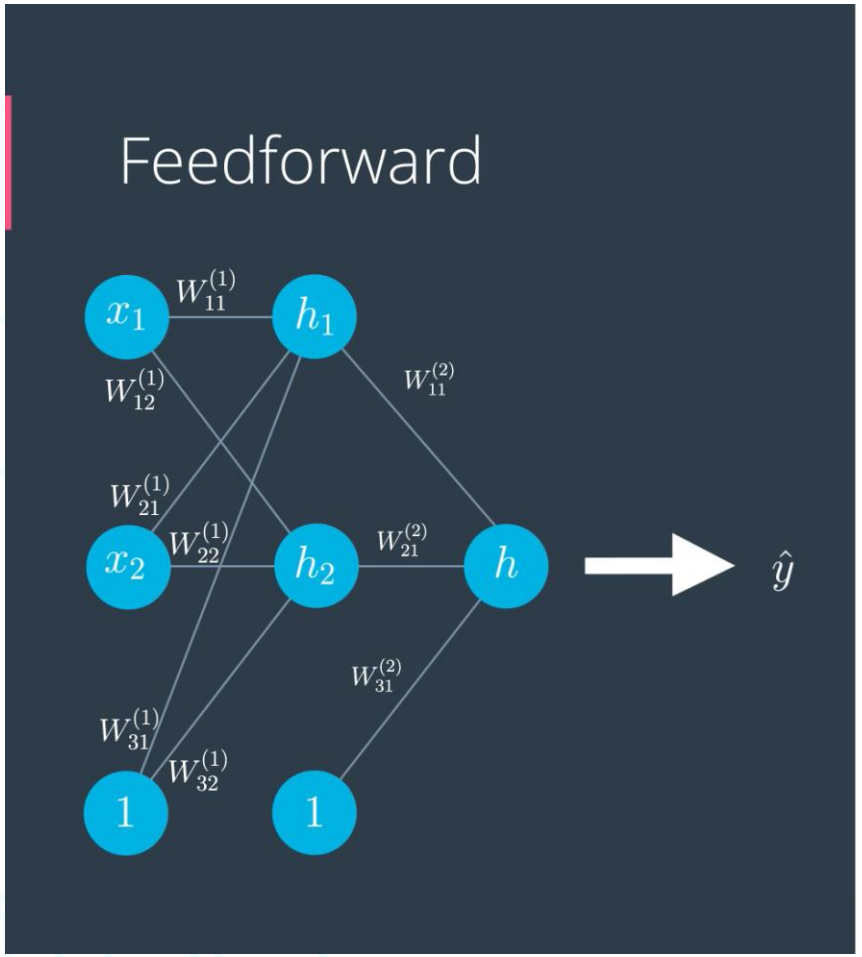


$$h_1 = W_{11}^{(1)} x_1 + W_{21}^{(1)} x_2 + W_{31}^{(1)}$$

$$h_2 = W_{12}^{(1)} x_1 + W_{22}^{(1)} x_2 + W_{32}^{(1)}$$

$$h = W_{11}^{(2)} \sigma(h_1) + W_{21}^{(2)} \sigma(h_2) + W_{31}^{(2)}$$

Redes Neurais – Backpropagation



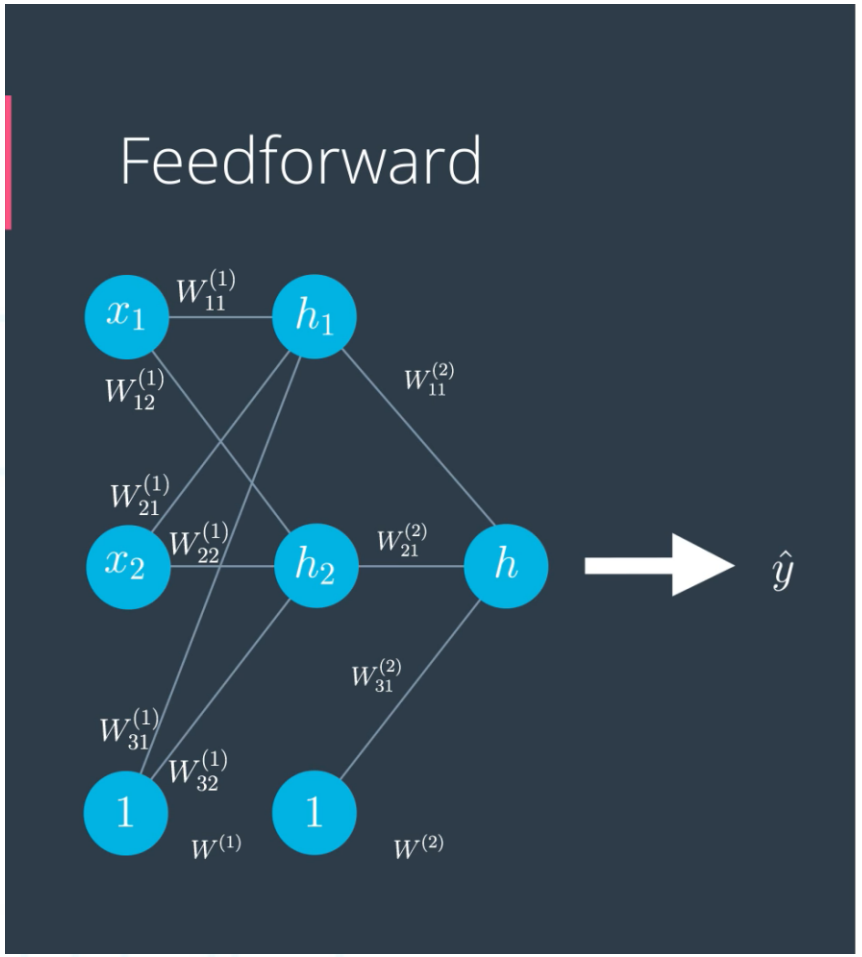
$$h_1 = W_{11}^{(1)} x_1 + W_{21}^{(1)} x_2 + W_{31}^{(1)}$$

$$h_2 = W_{12}^{(1)} x_1 + W_{22}^{(1)} x_2 + W_{32}^{(1)}$$

$$h = W_{11}^{(2)} \sigma(h_1) + W_{21}^{(2)} \sigma(h_2) + W_{31}^{(2)}$$

$$\hat{y} = \sigma(h)$$

Redes Neurais – Backpropagation



$$h_1 = W_{11}^{(1)} x_1 + W_{21}^{(1)} x_2 + W_{31}^{(1)}$$

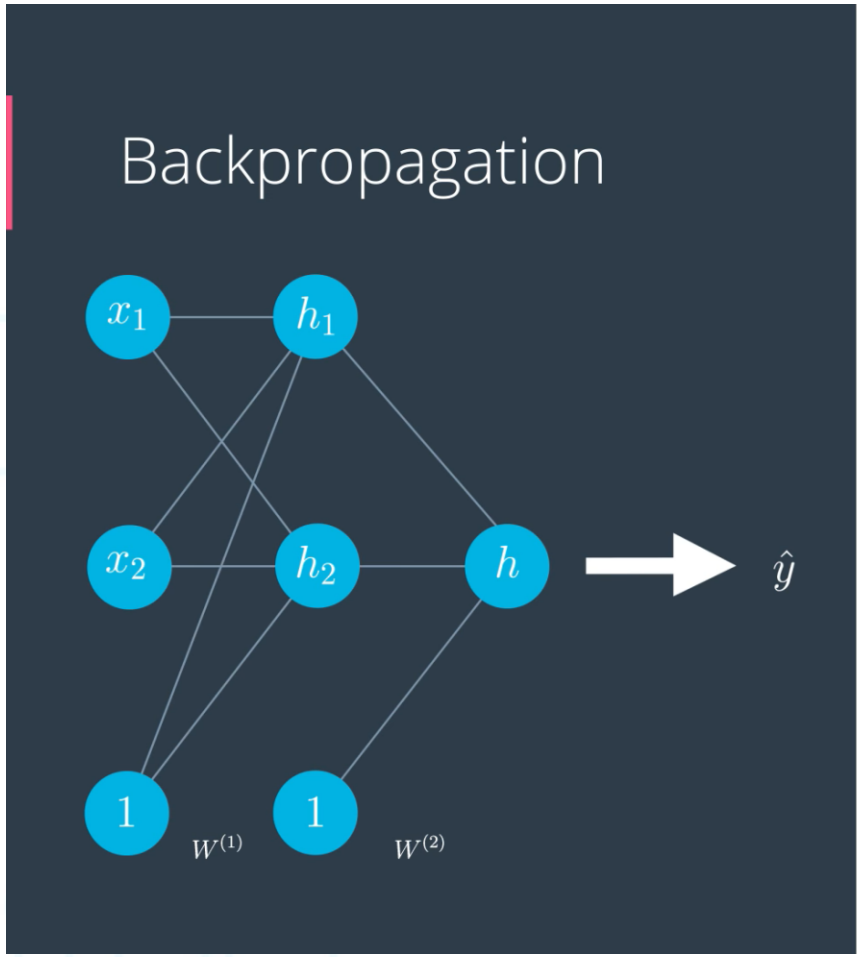
$$h_2 = W_{12}^{(1)} x_1 + W_{22}^{(1)} x_2 + W_{32}^{(1)}$$

$$h = W_{11}^{(2)} \sigma(h_1) + W_{21}^{(2)} \sigma(h_2) + W_{31}^{(2)}$$

$$\hat{y} = \sigma(h)$$

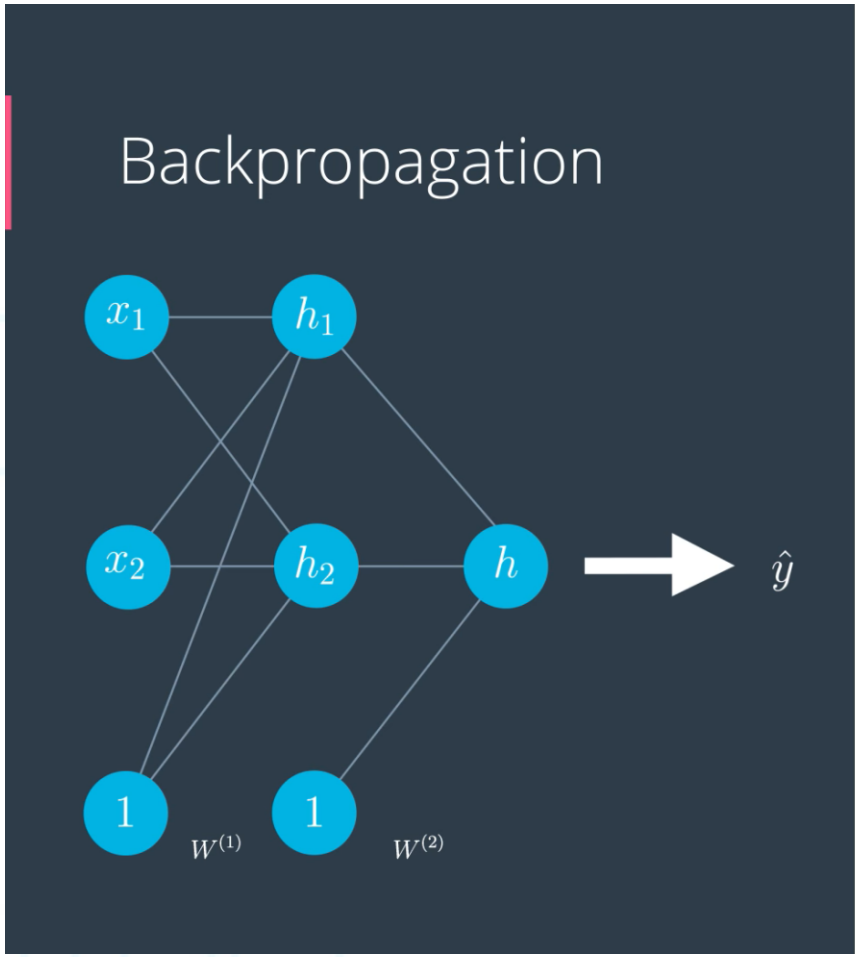
$$\hat{y} = \sigma \circ W^{(2)} \circ \sigma \circ W^{(1)}(x)$$

Redes Neurais – Backpropagation



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

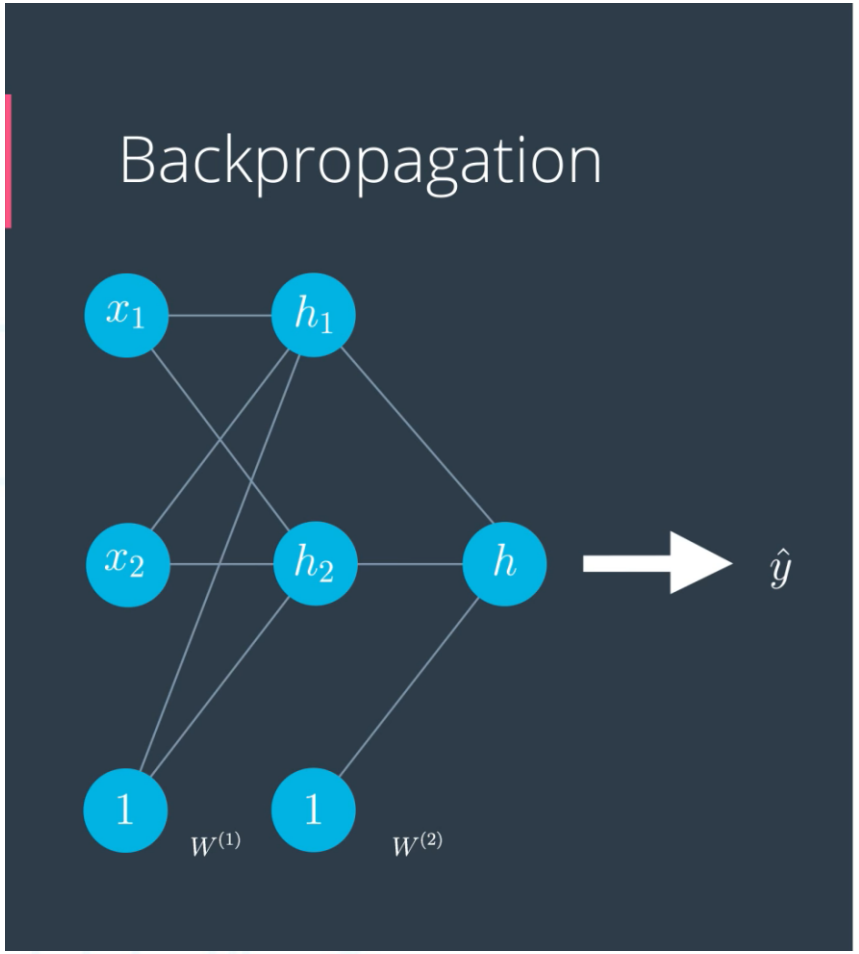
Redes Neurais – Backpropagation



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

$$E(W) = E(W_{11}^{(1)}, W_{12}^{(1)}, ..., W_{31}^{(2)})$$

Redes Neurais – Backpropagation

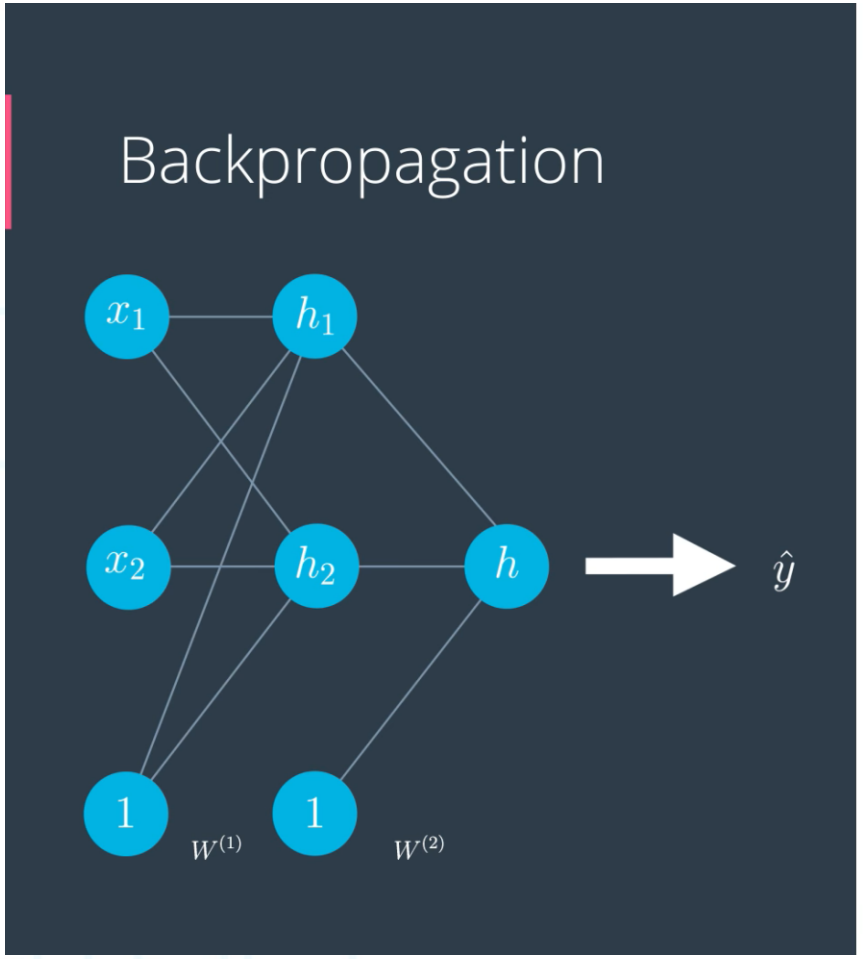


$$E(W) = -\frac{1}{m}\sum_{i=1}^{m} y_i ln(\hat{y}_i) + (1 - y_i) ln(1 - \hat{y}_i)$$

$$E(W) = E(W_{11}^{(1)}, W_{12}^{(1)}, ..., W_{31}^{(2)})$$

$$\nabla E = (\frac{\partial E}{\partial W_{11}^{(1)}}, ..., \frac{\partial E}{\partial W_{31}^{(2)}})$$

Redes Neurais – Backpropagation



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

$$E(W) = E(W_{11}^{(1)}, W_{12}^{(1)}, ..., W_{31}^{(2)})$$

$$\nabla E = (\frac{\partial E}{\partial W_{11}^{(1)}}, ..., \frac{\partial E}{\partial W_{31}^{(2)}})$$

$$\frac{\partial E}{\partial W_{11}^{(1)}} = \frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial h} \frac{\partial h}{\partial h_1} \frac{\partial h_1}{\partial W_{11}^{(1)}}$$

Redes Neurais

	Peso	Altura
 Pessoa 1	80 kg	163
 Pessoa 2	85 kg	168
 Pessoa 3	90 kg	175
 Pessoa 4	95 kg	188

$$\hat{y} = \beta_0 + \beta_1 X_1$$

$$\hat{y} = 3 + 0{,}7\ x\ 80$$

$$\hat{y} = 60$$

$$\hat{y} = 5 + 1.3\ x\ 85$$

$$\hat{y} = 115{,}5$$

$$\hat{y} = 10 + 1.6\ x\ 90$$

$$\hat{y} = 154$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$MSE = \sum(163 - 60)^2$$

$$MSE = 10.609$$

$$MSE = \sum(168 - 115{,}5)^2$$

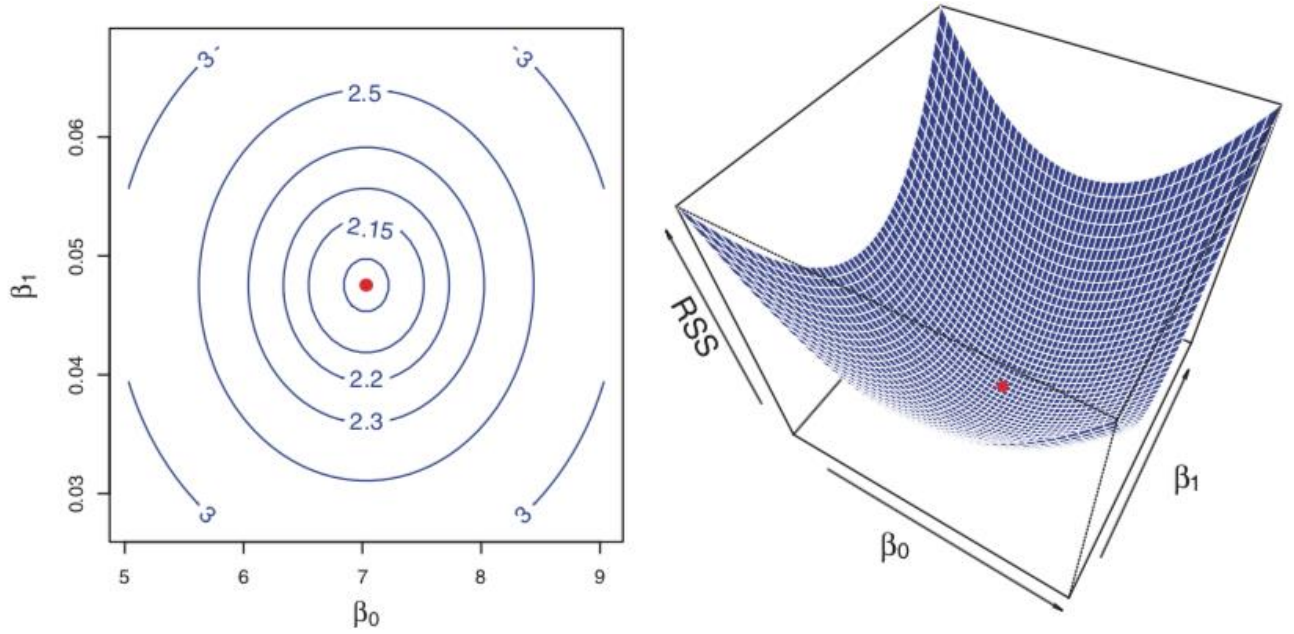
$$MSE = 2.756{,}25$$

$$MSE = \sum(175 - 154)^2$$

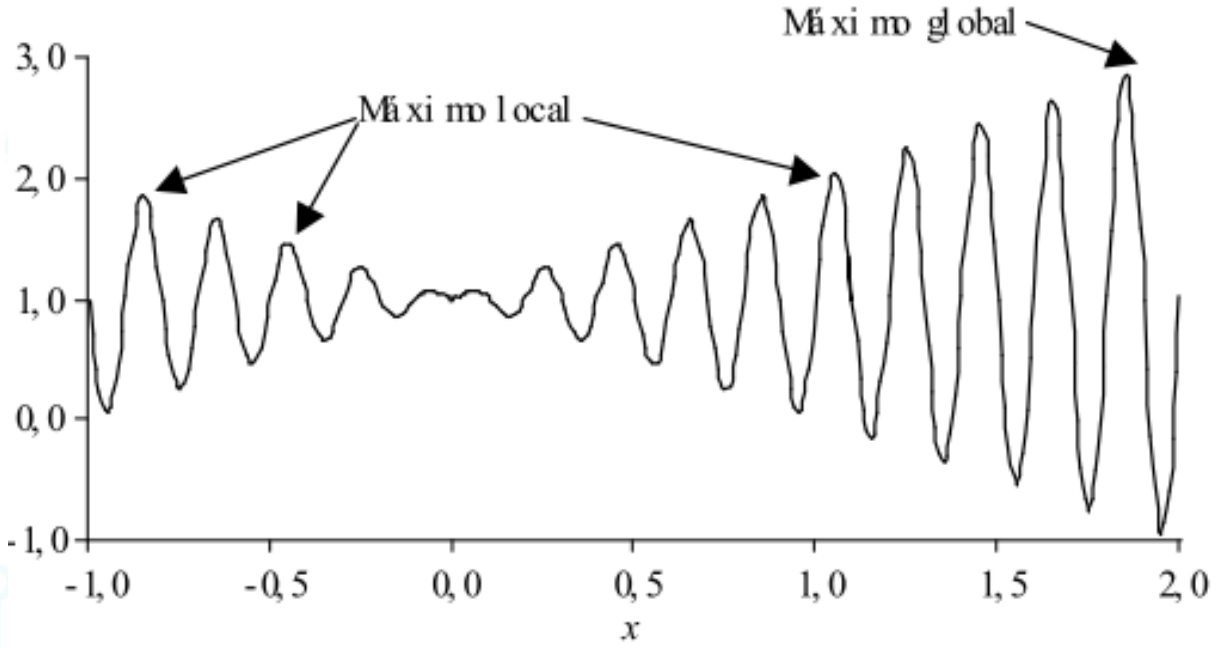
$$MSE = 441$$

Regressão Linear

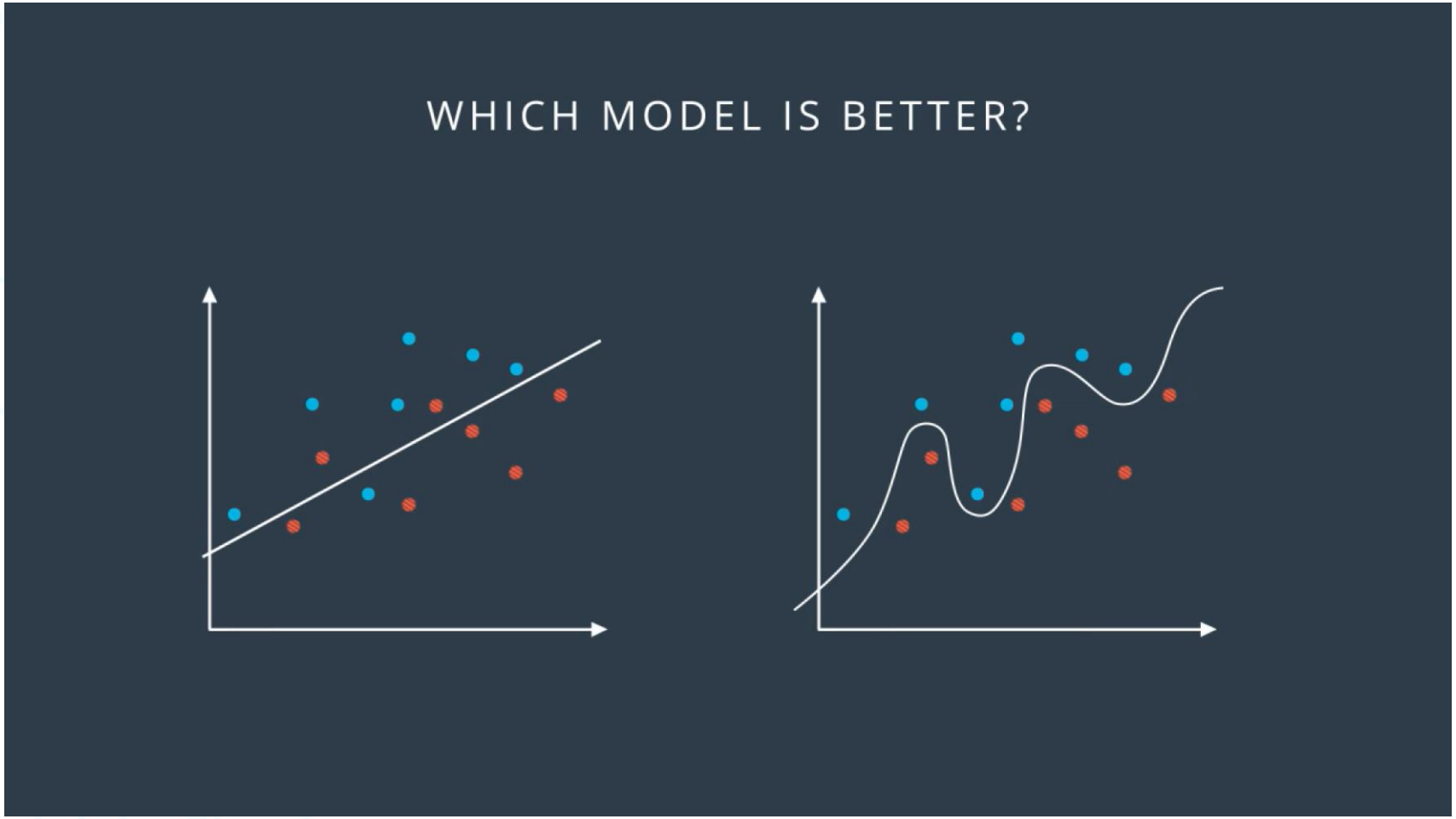
- Como a função RMSE é convexa, é possível encontrar o valor mínimo por meio de algoritmos de otimização



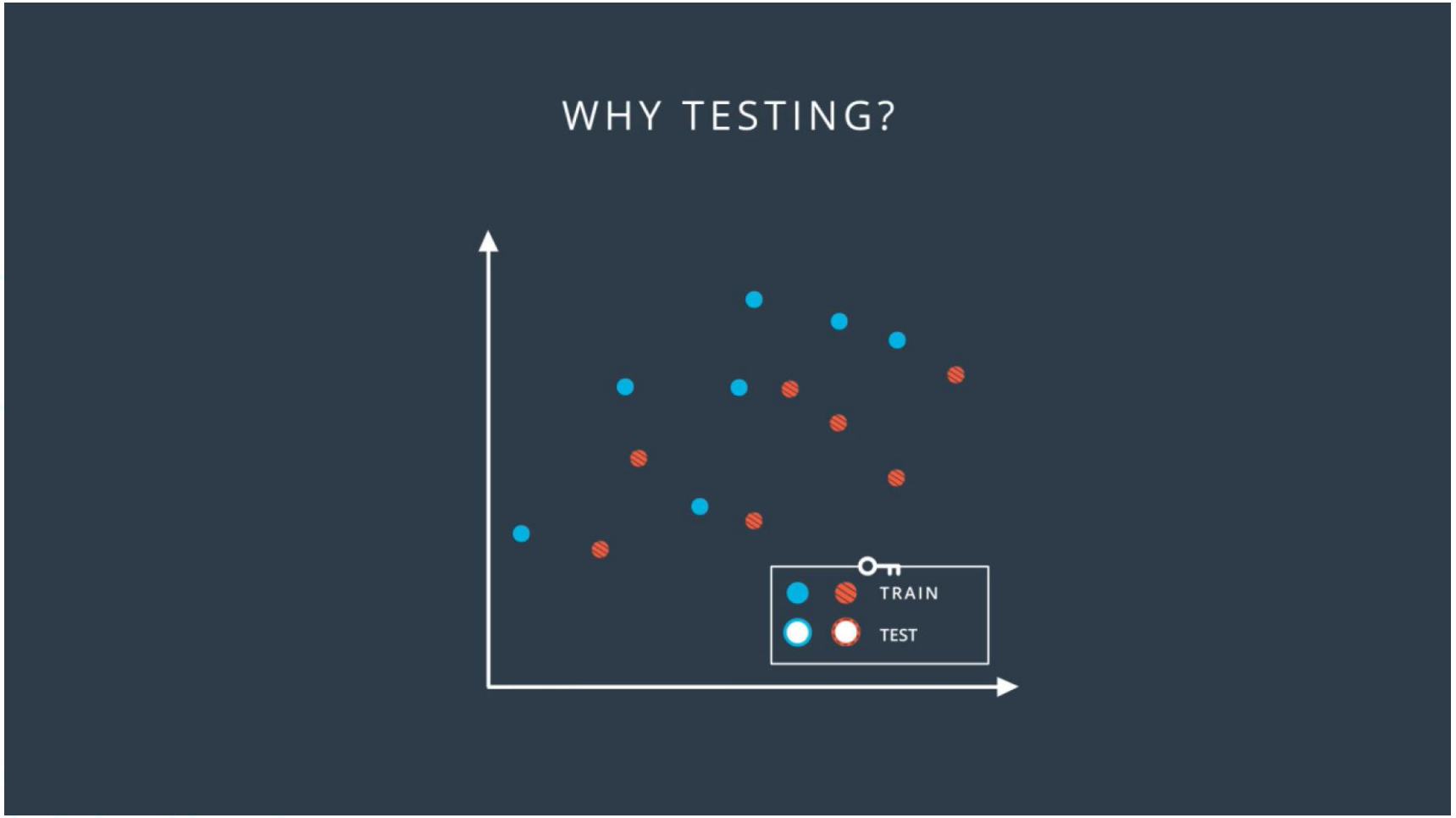
Redes Neurais



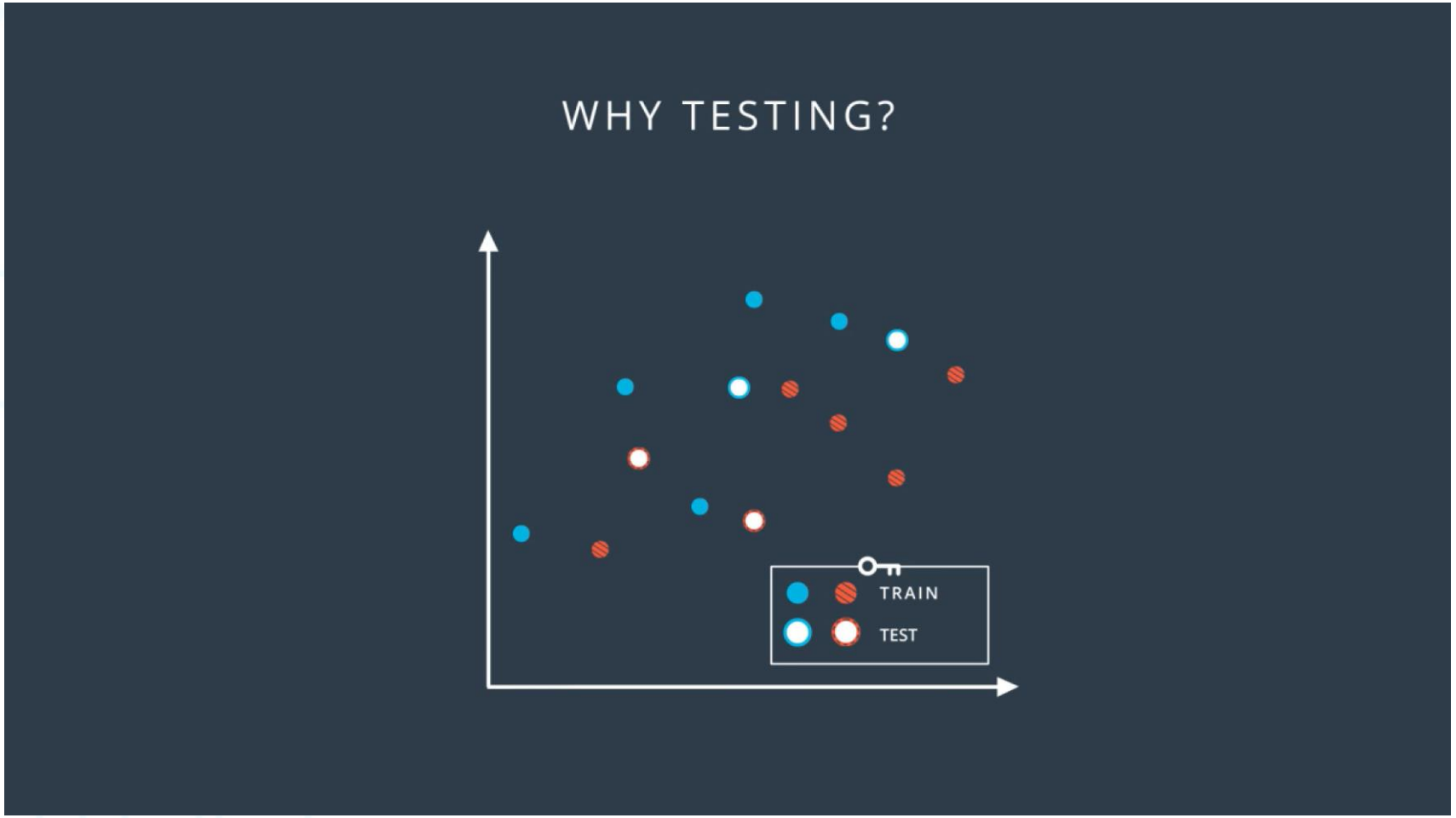
Redes Neurais – Testing



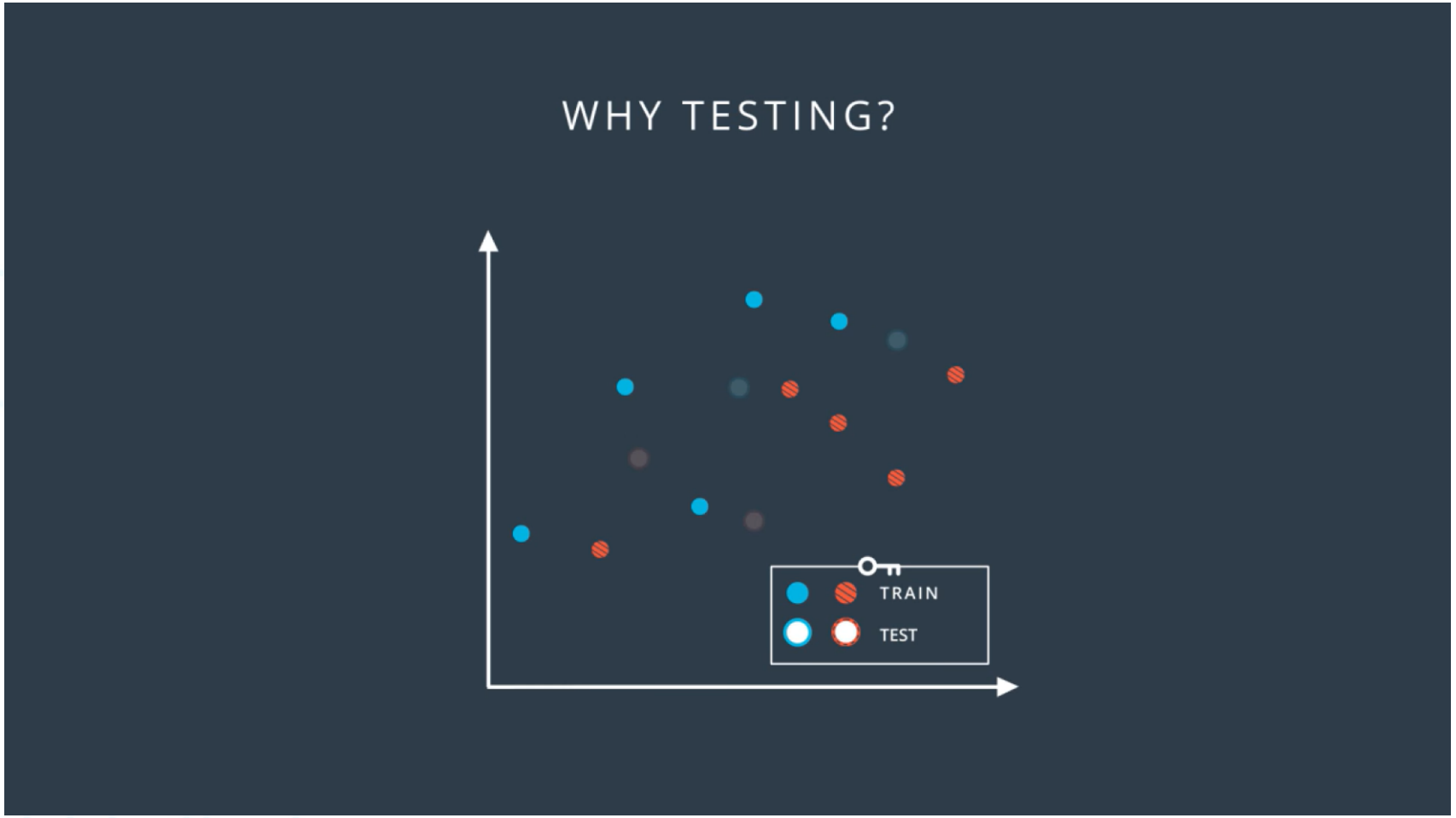
Redes Neurais – Testing



Redes Neurais – Testing

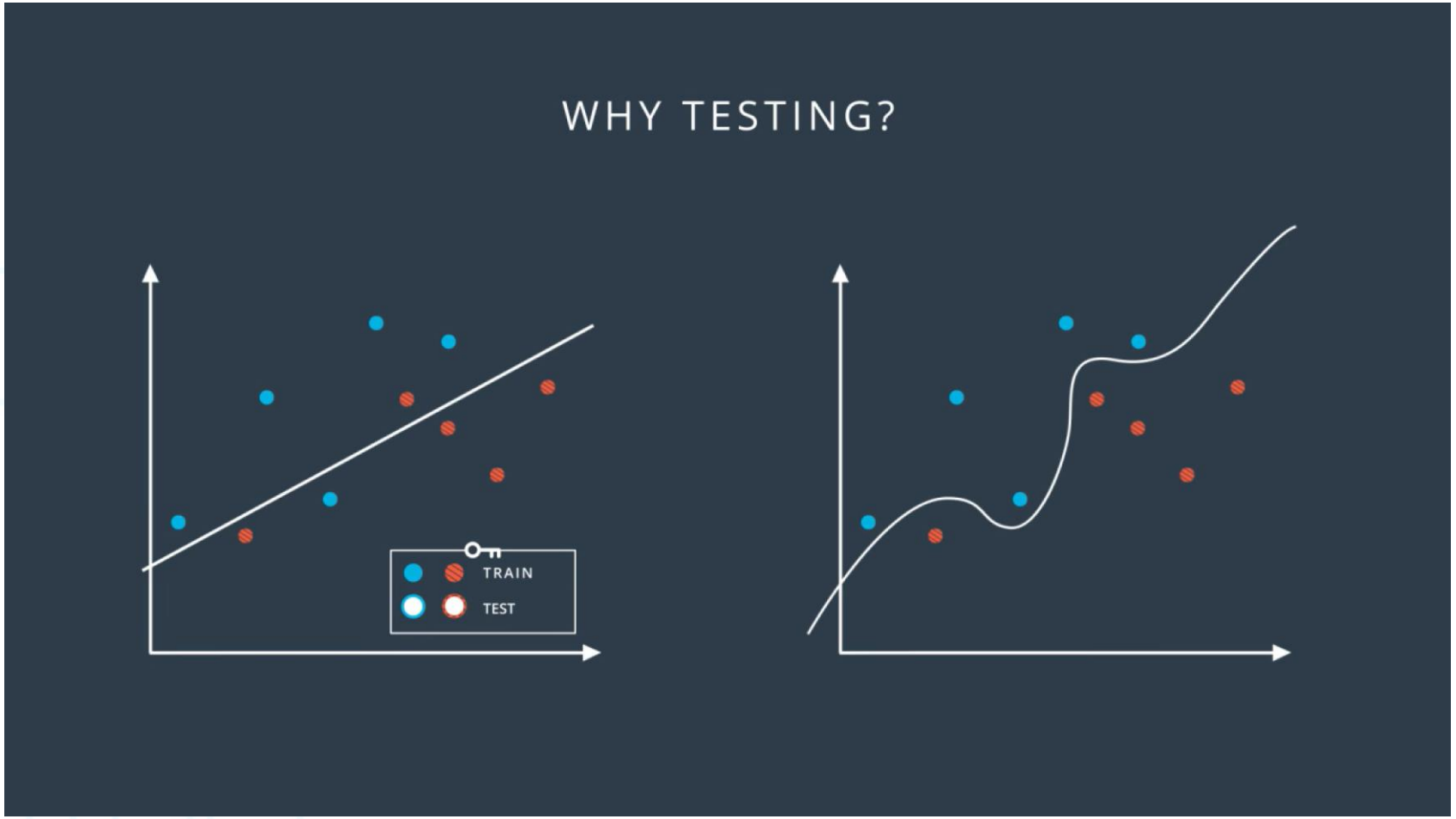


Redes Neurais – Testing

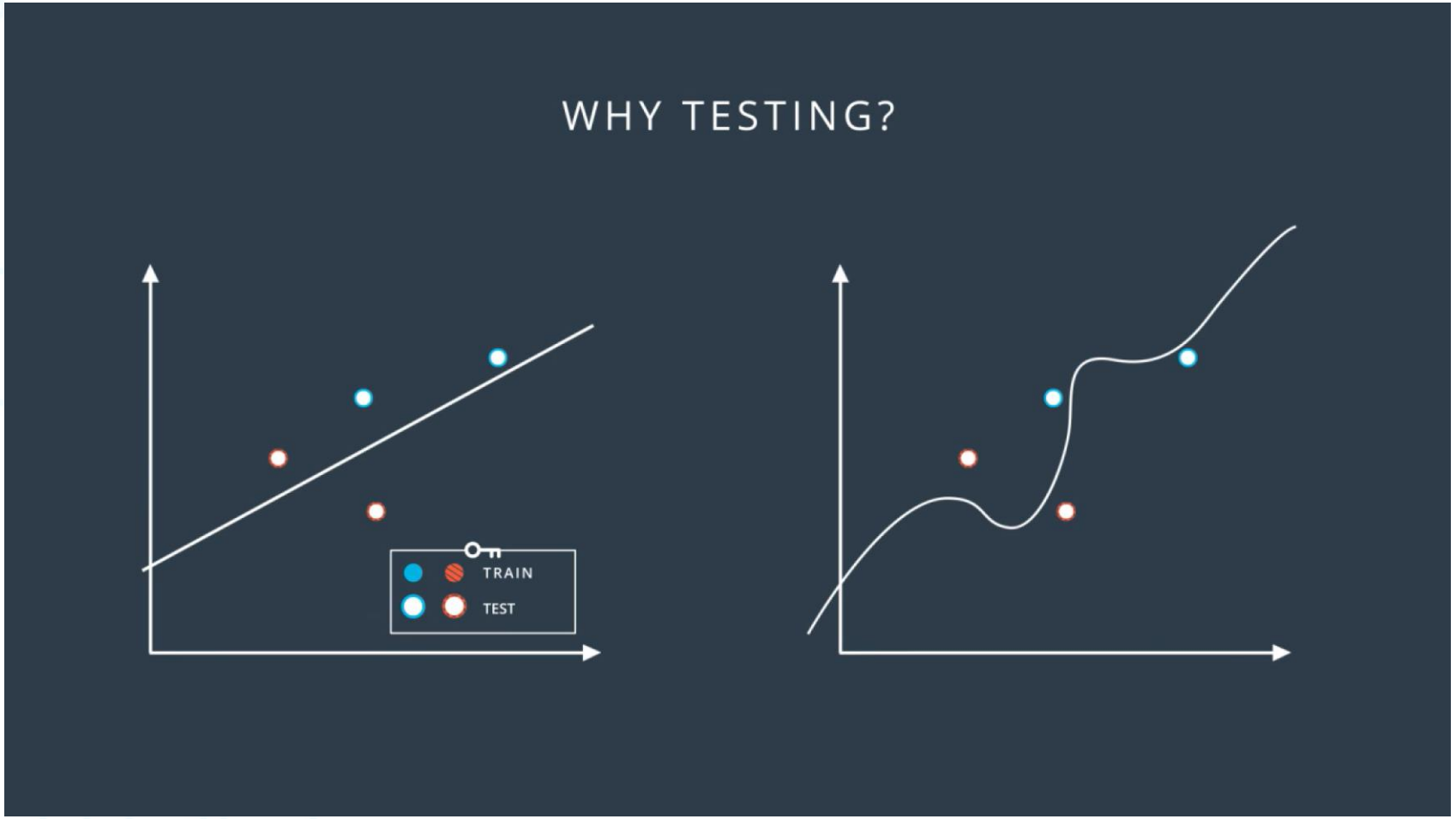


Redes Neurais – Testing

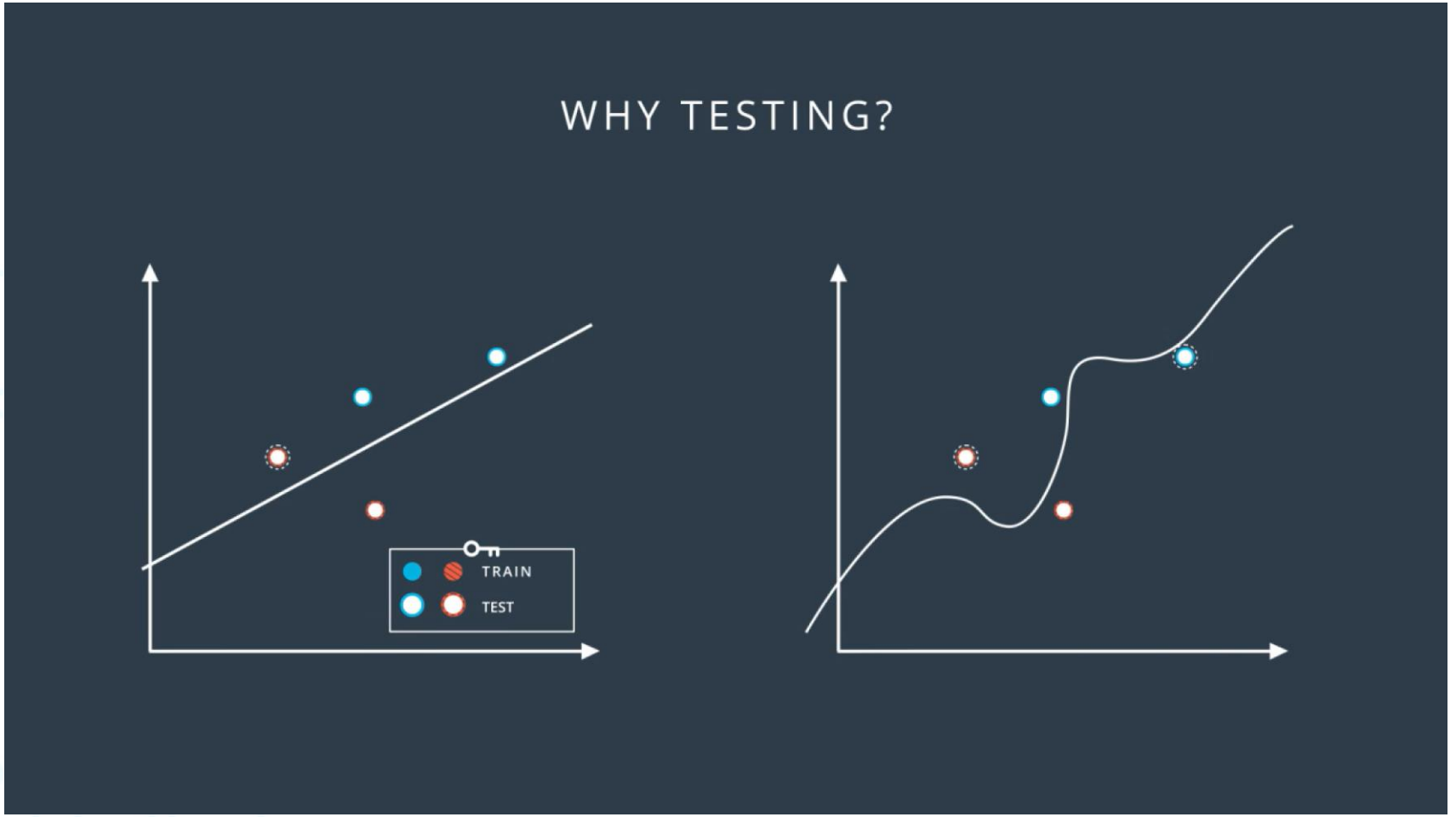
Redes Neurais – Testing



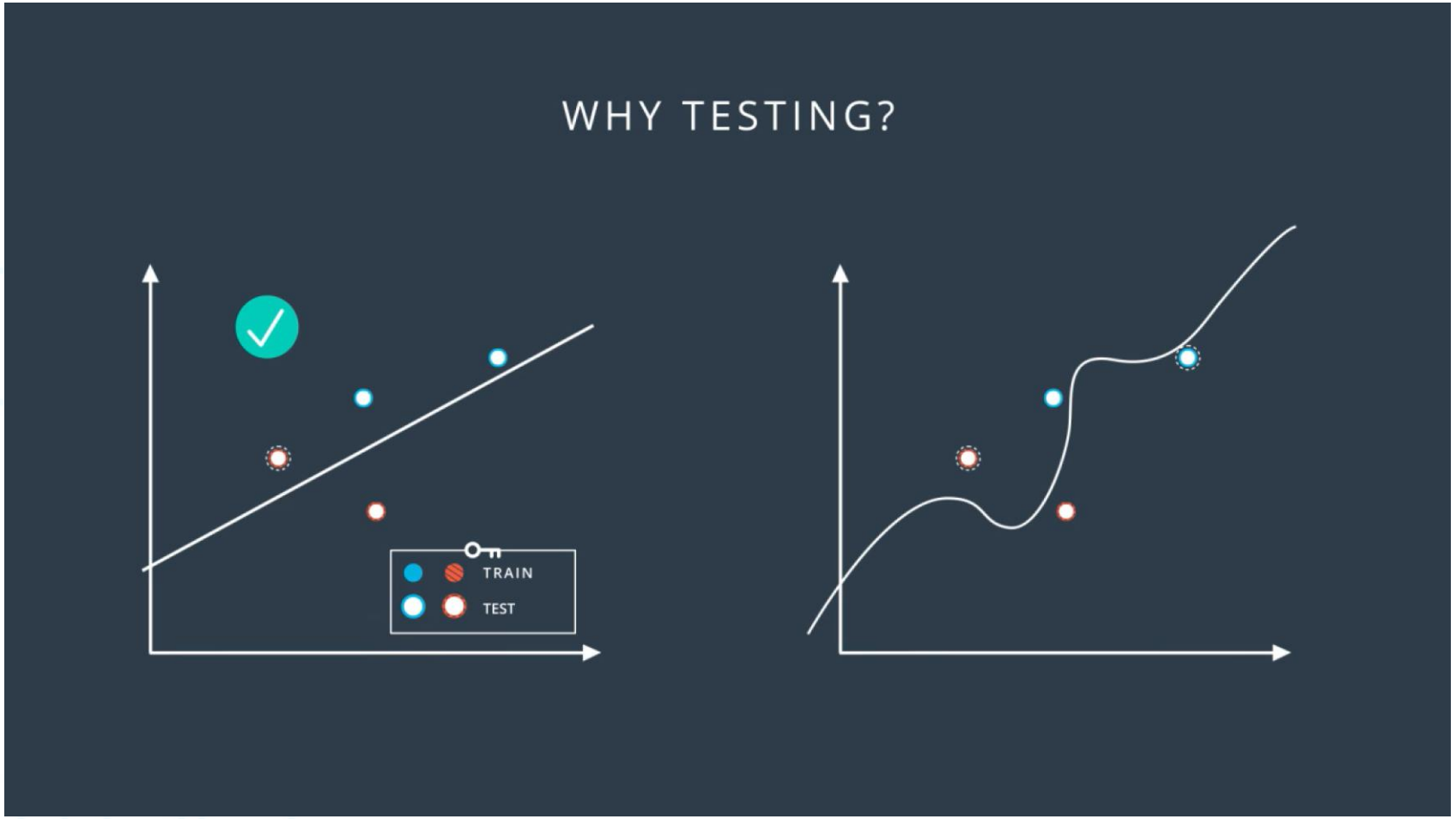
Redes Neurais – Testing



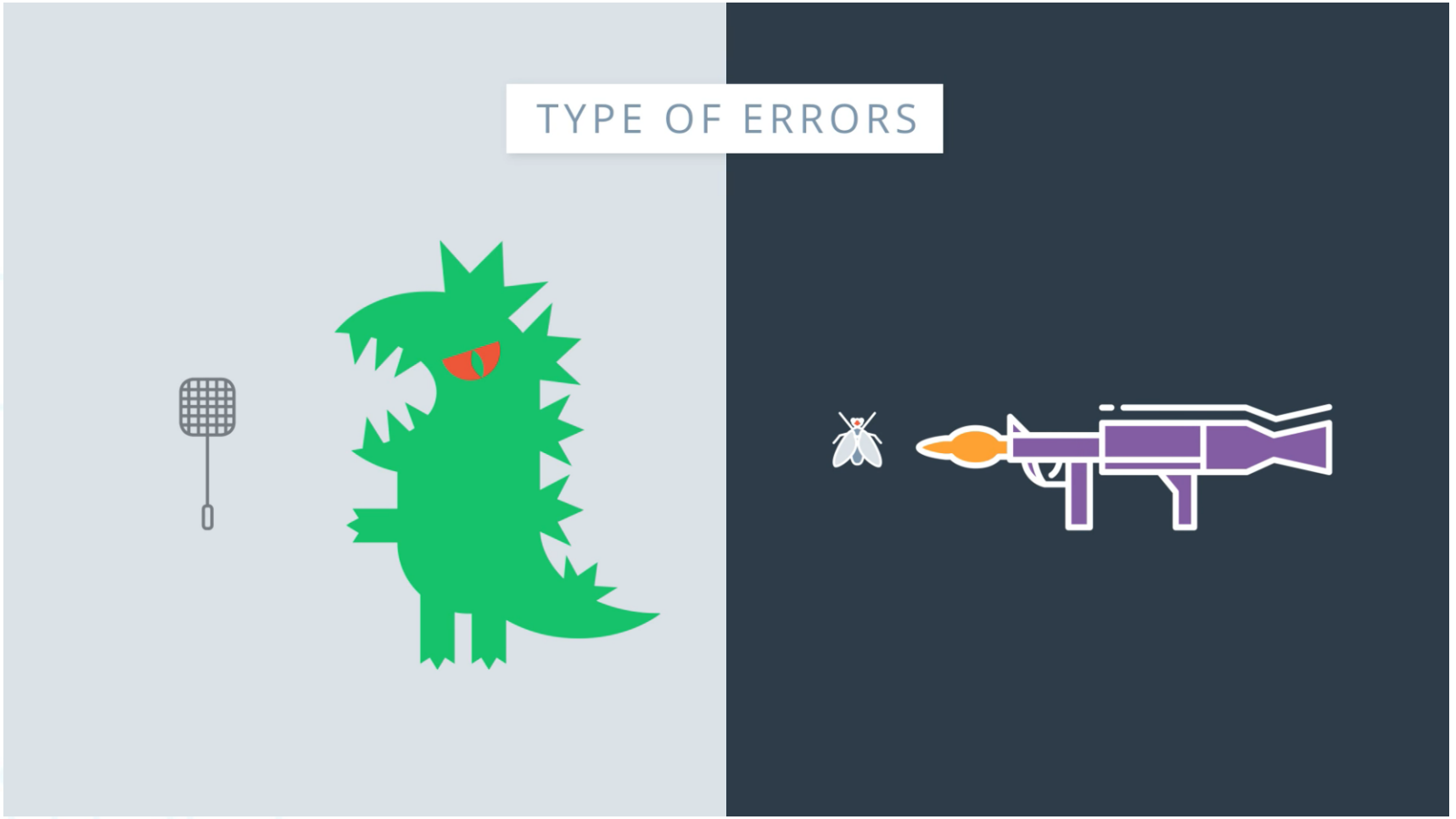
Redes Neurais – Testing



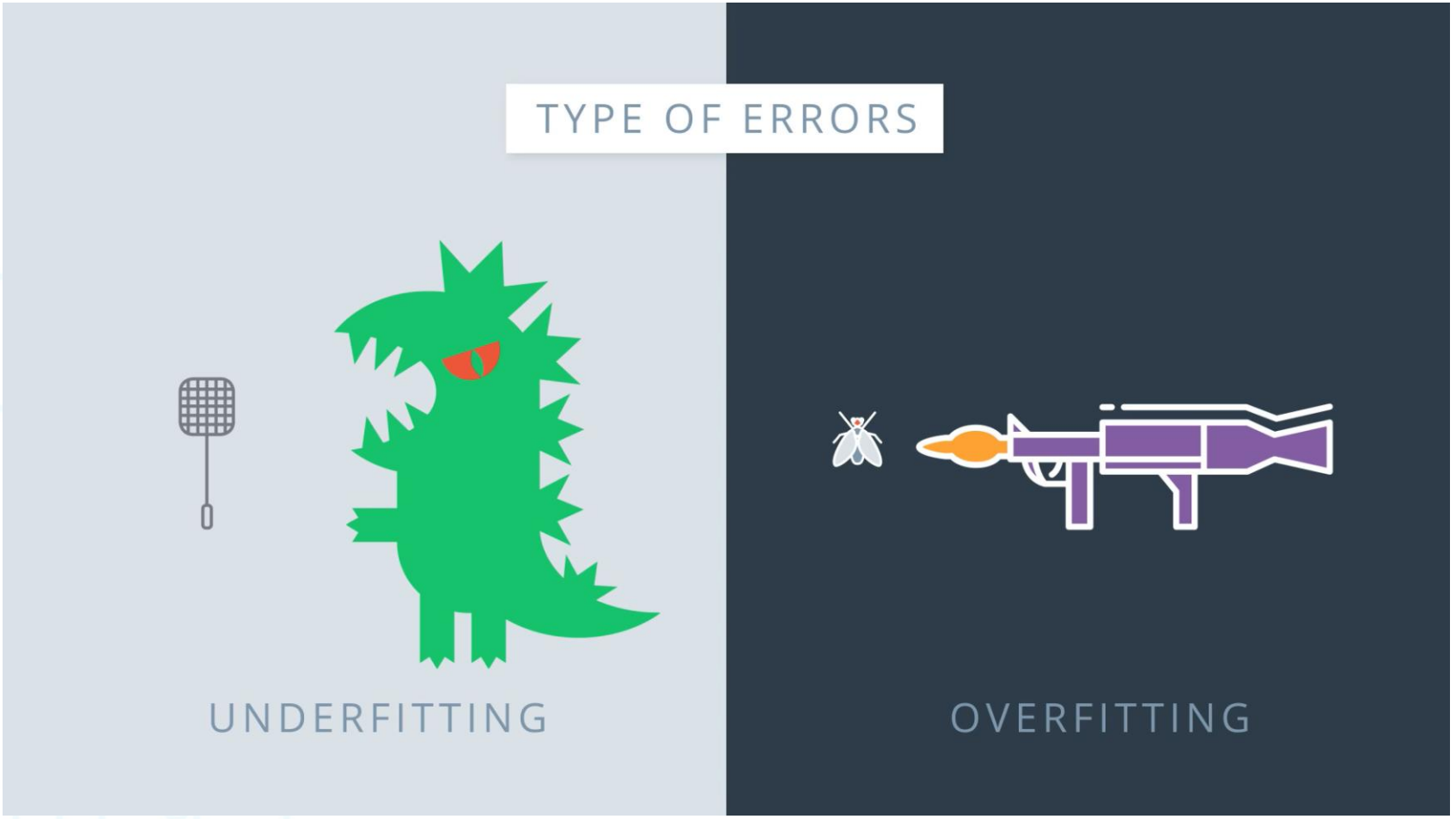
Redes Neurais – Testing



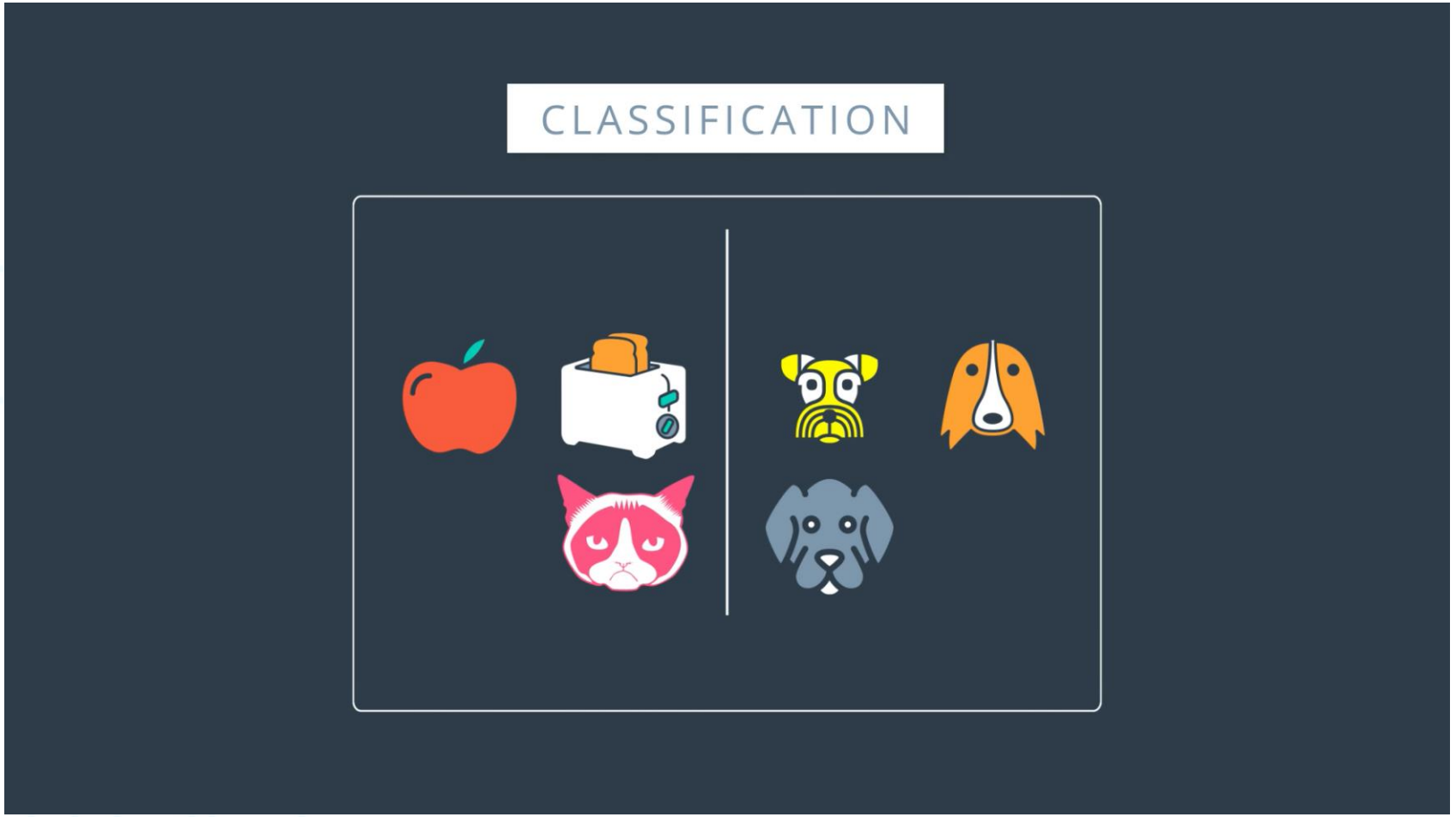
Redes Neurais – Overfitting and Underfitting



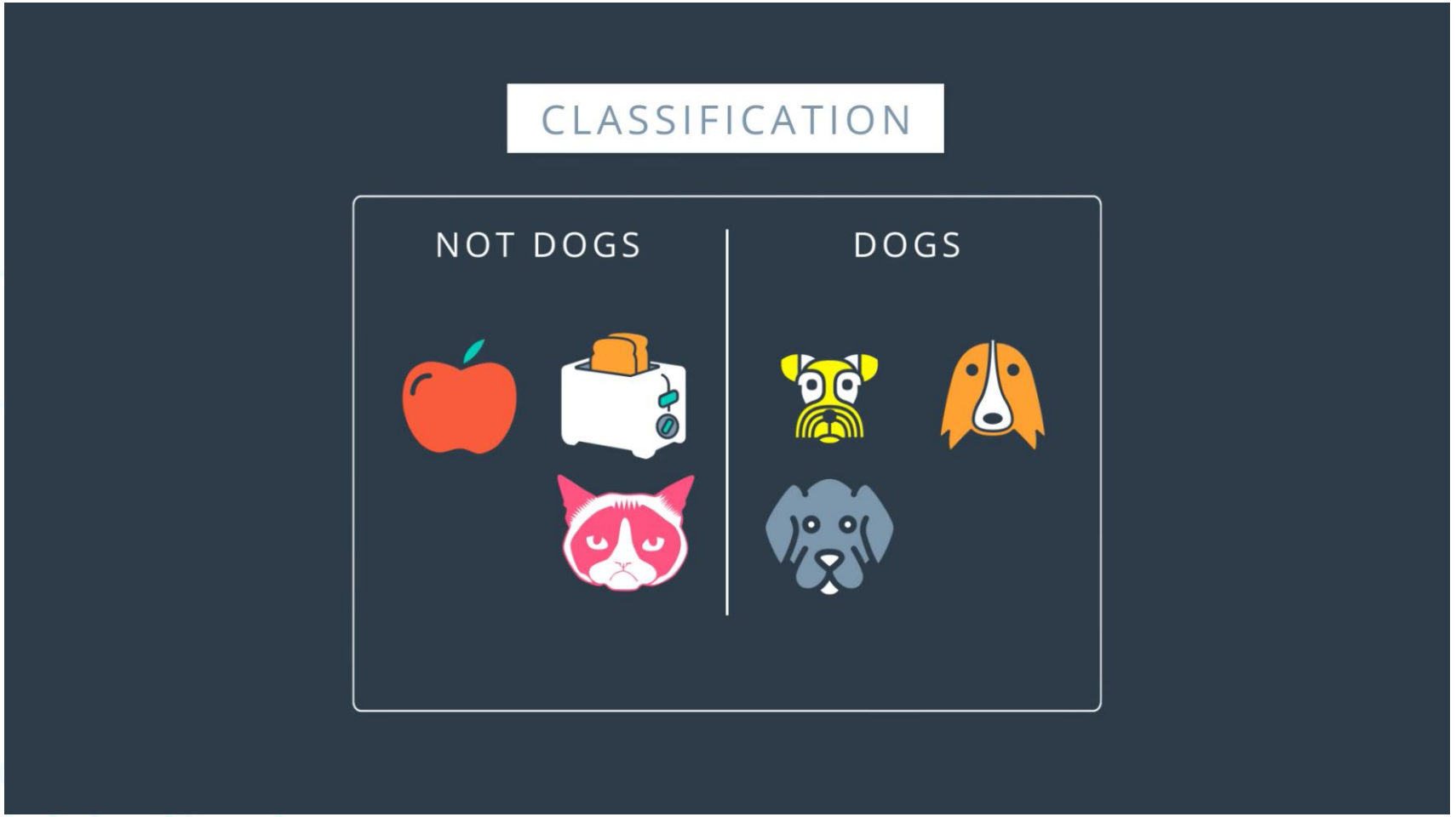
Redes Neurais – Overfitting and Underfitting



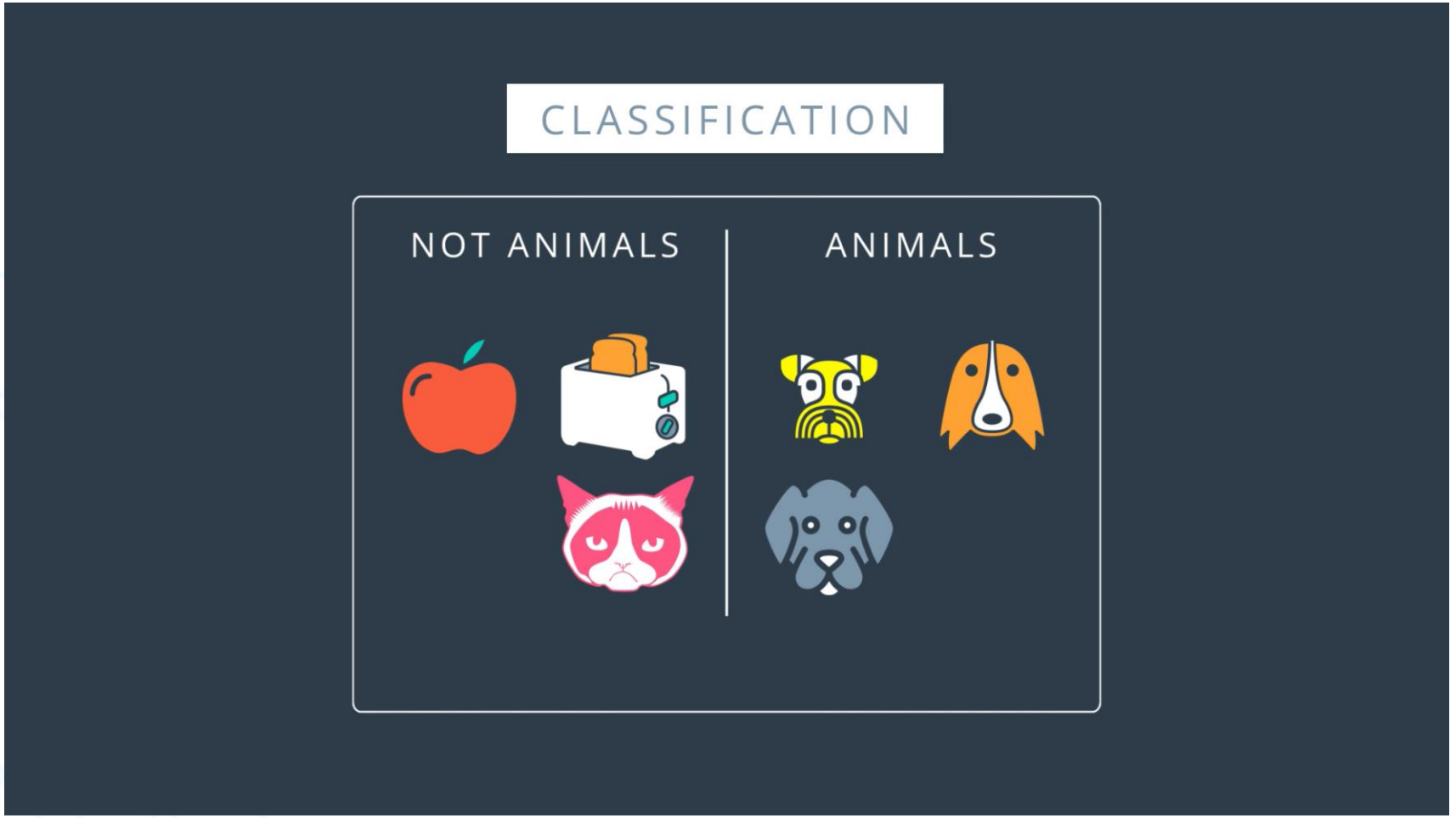
Redes Neurais – Overfitting and Underfitting



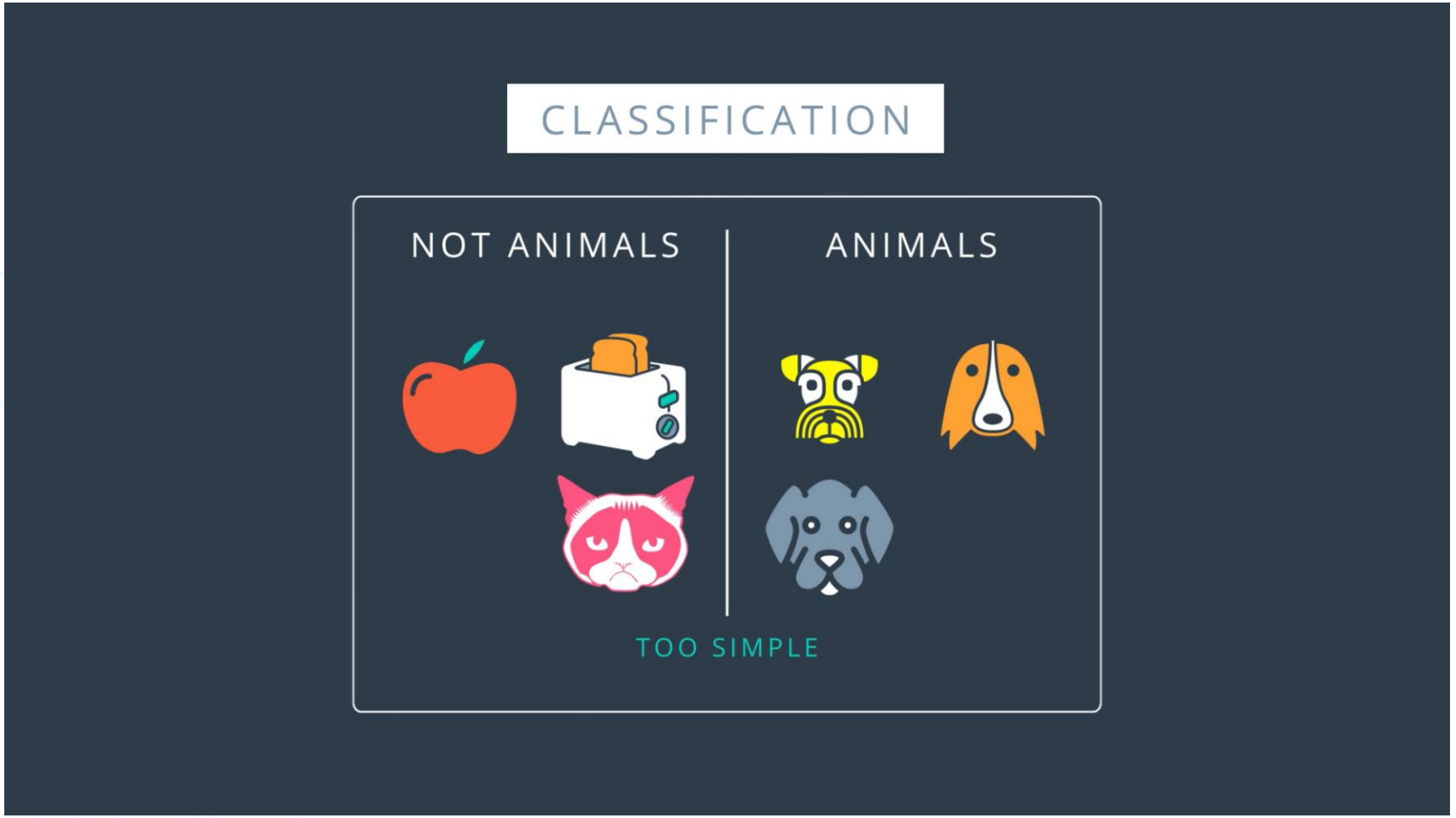
Redes Neurais – Overfitting and Underfitting



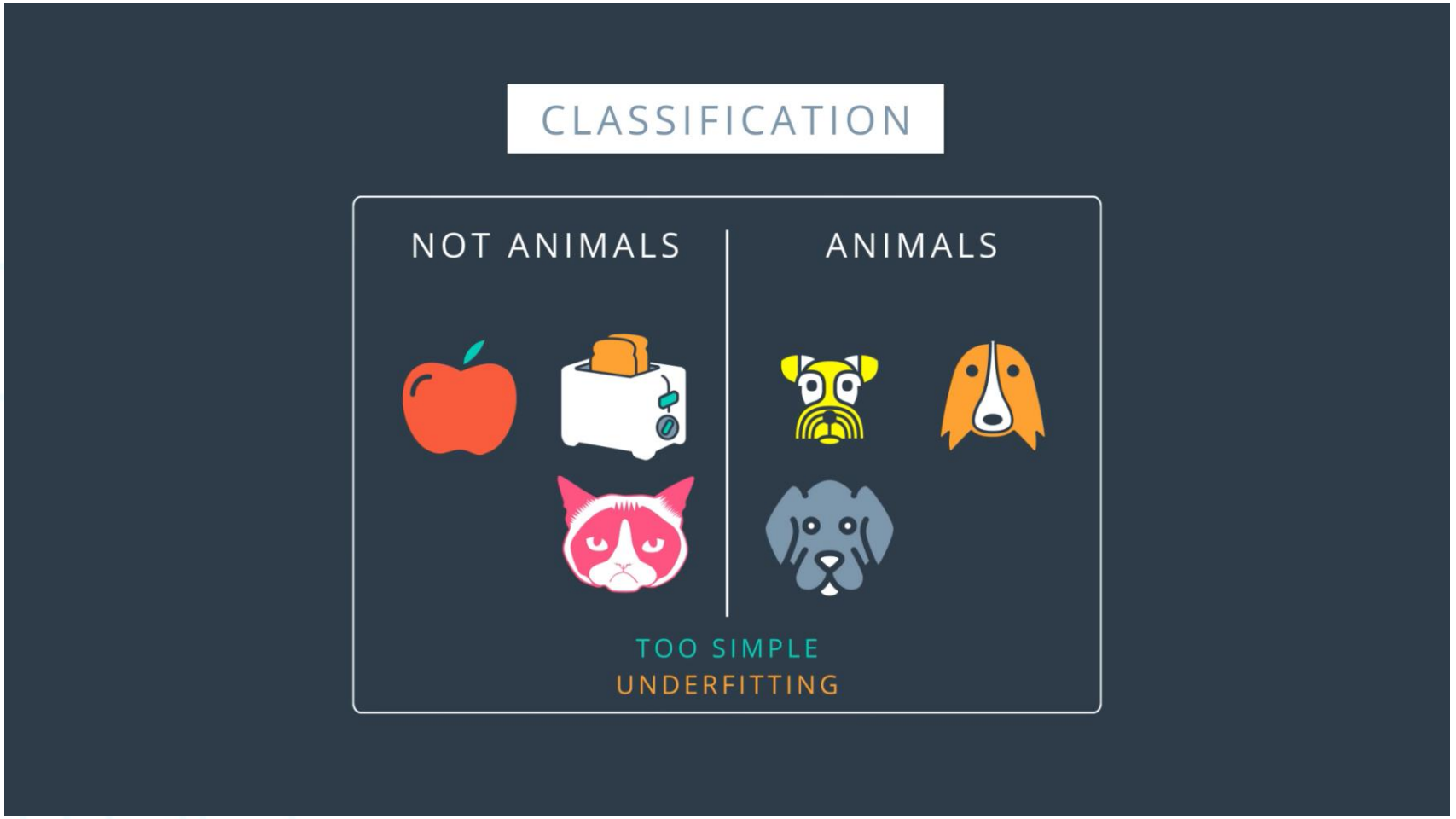
Redes Neurais – Overfitting and Underfitting



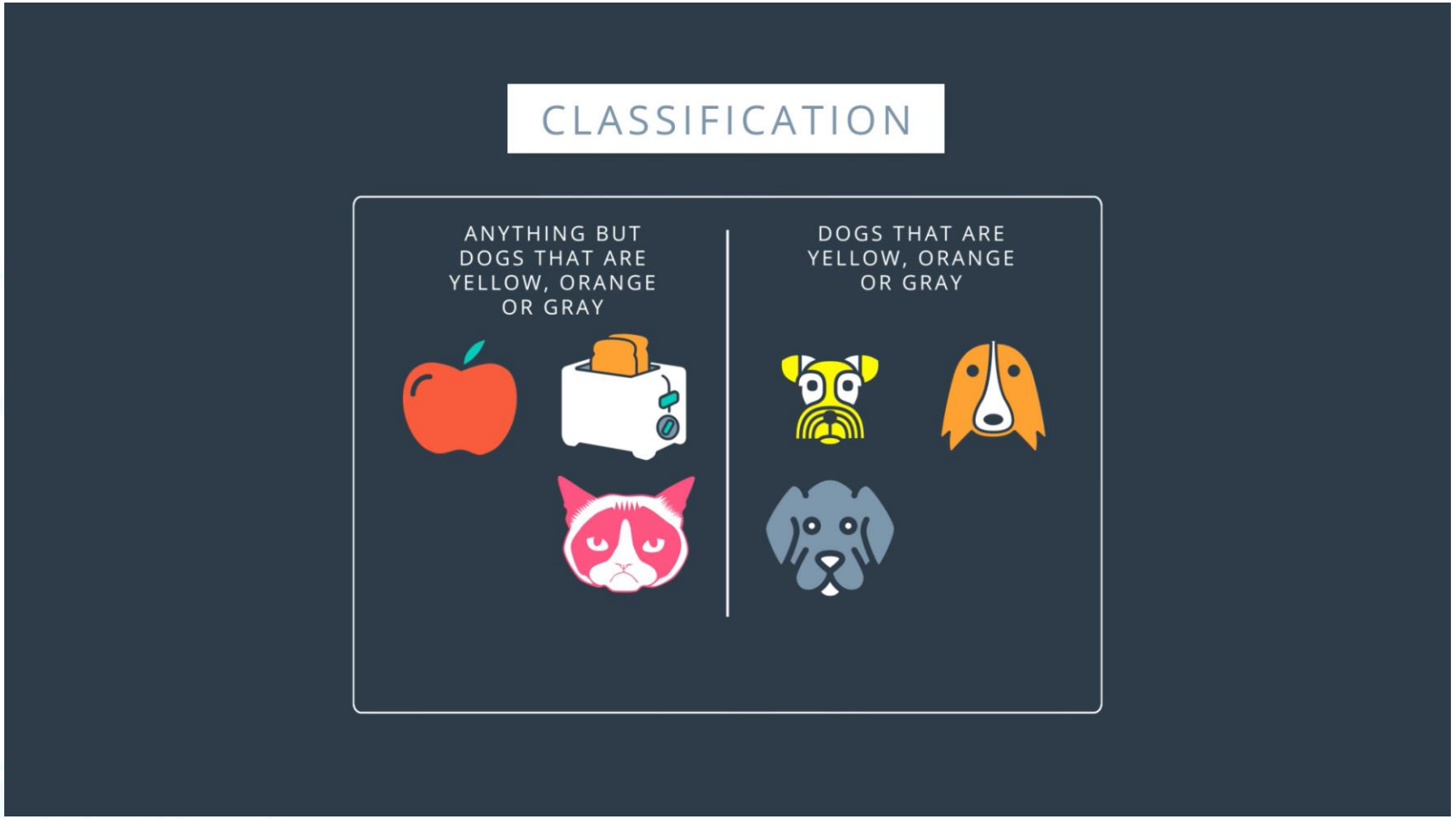
Redes Neurais – Overfitting and Underfitting



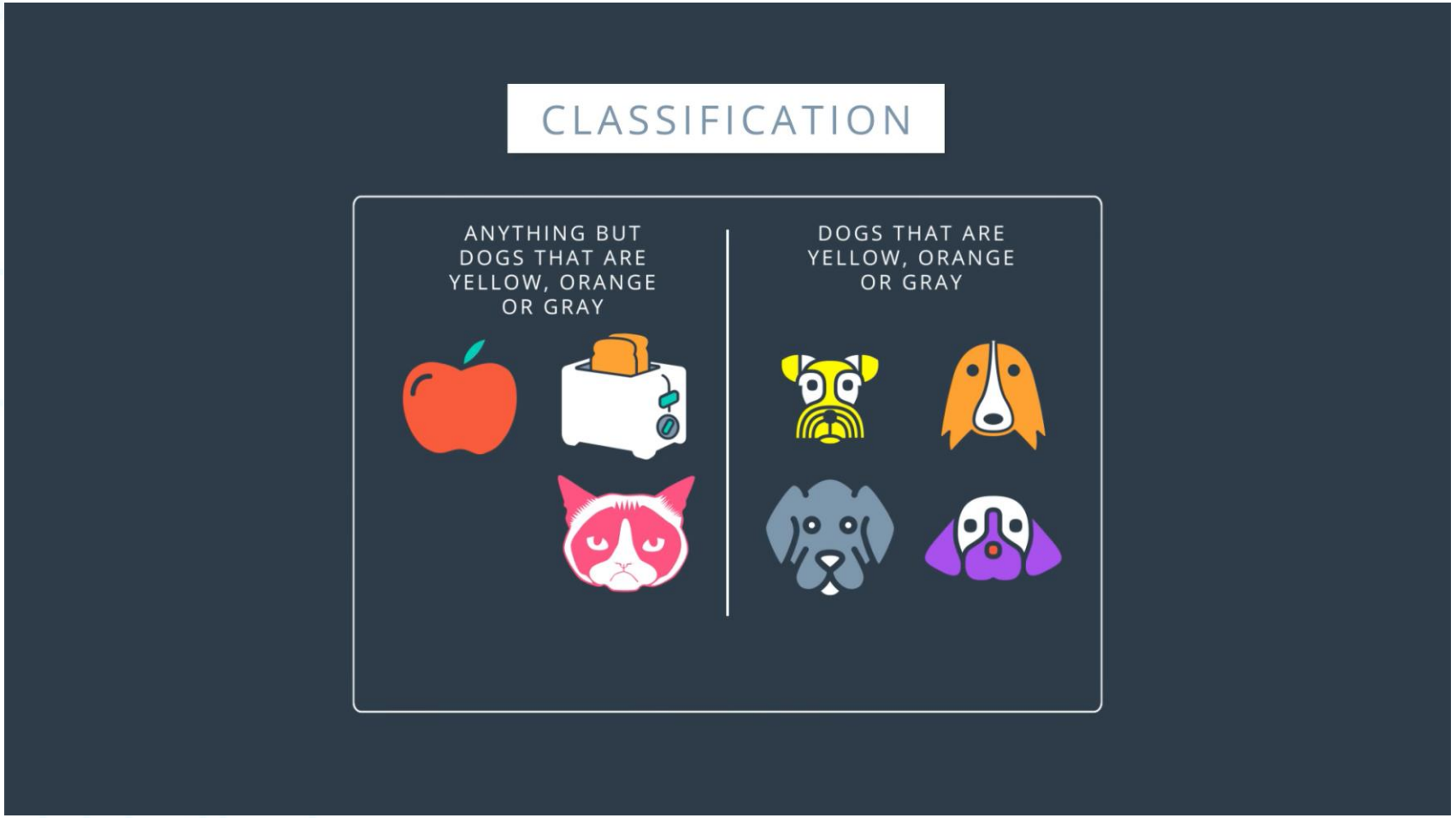
Redes Neurais – Overfitting and Underfitting



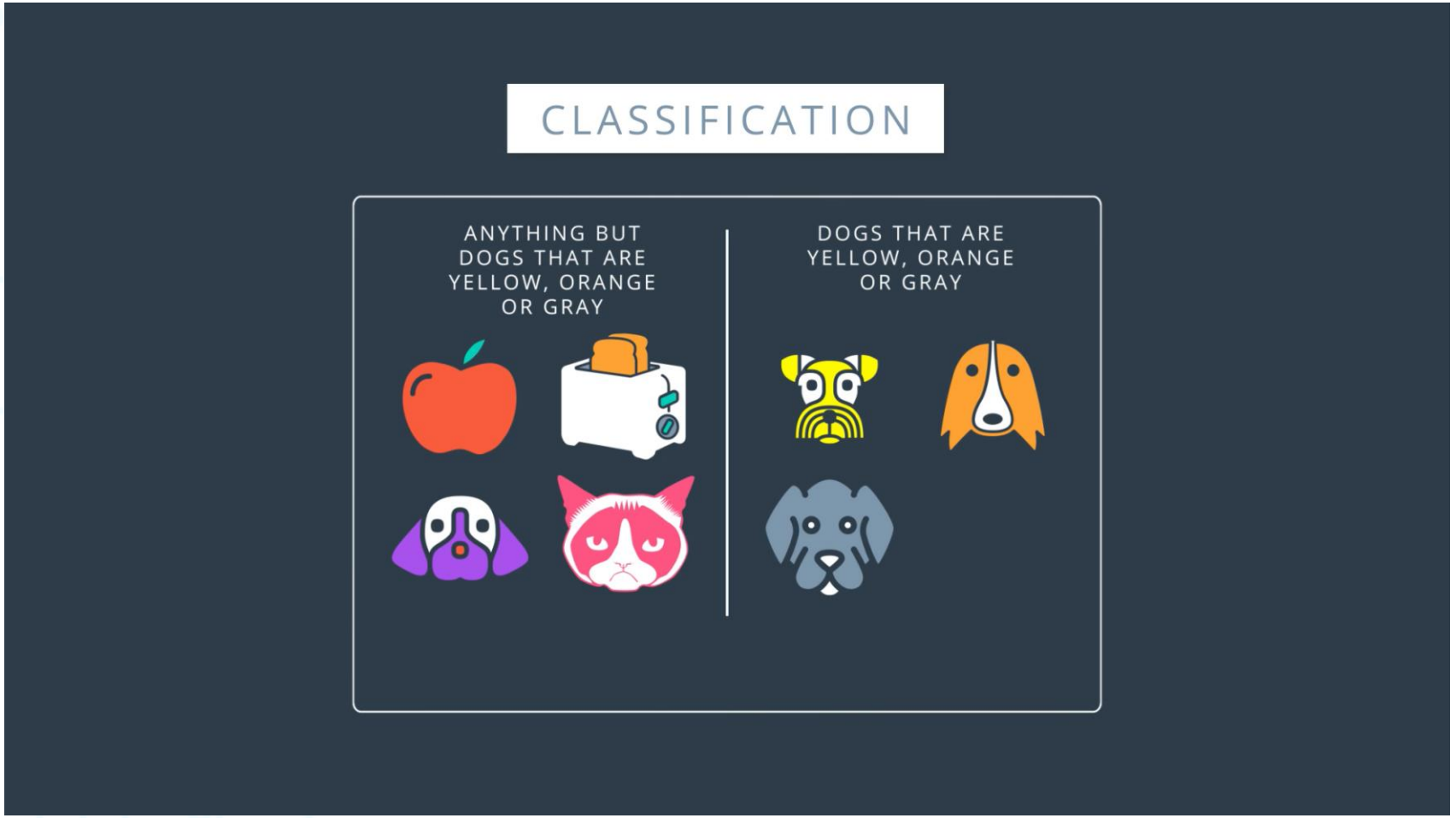
Redes Neurais – Overfitting and Underfitting



Redes Neurais – Overfitting and Underfitting

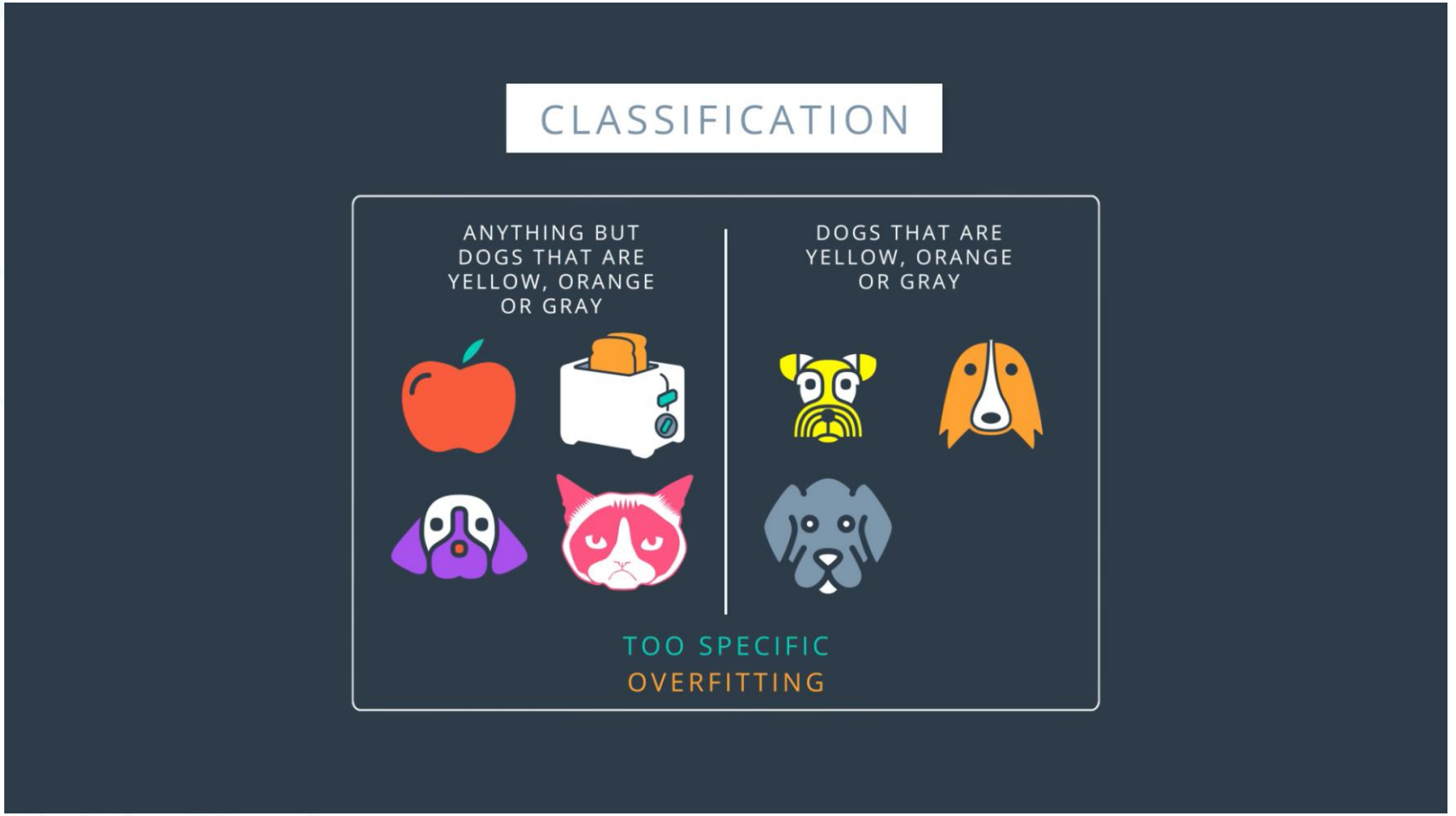


Redes Neurais – Overfitting and Underfitting

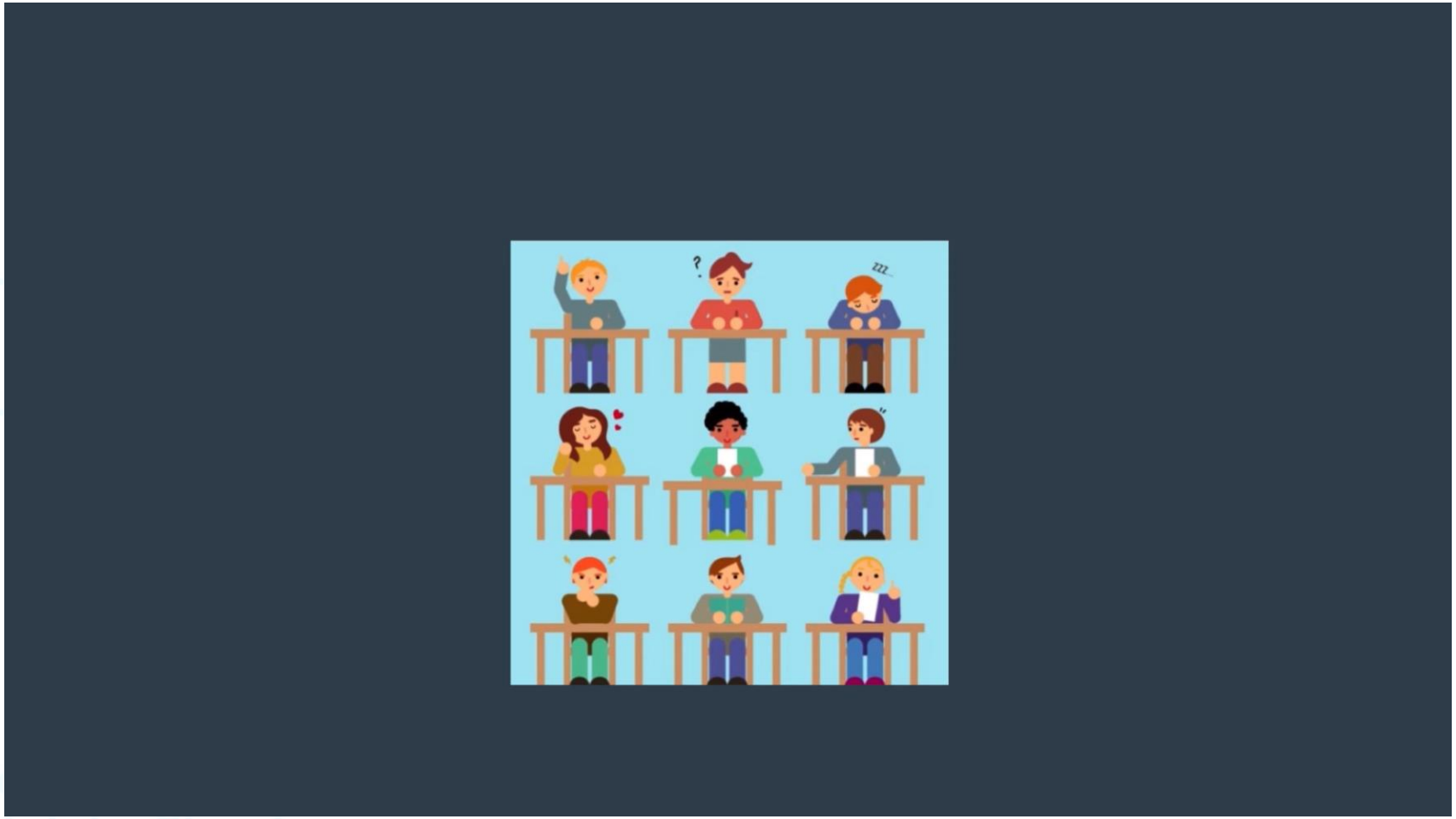


Redes Neurais – Overfitting and Underfitting

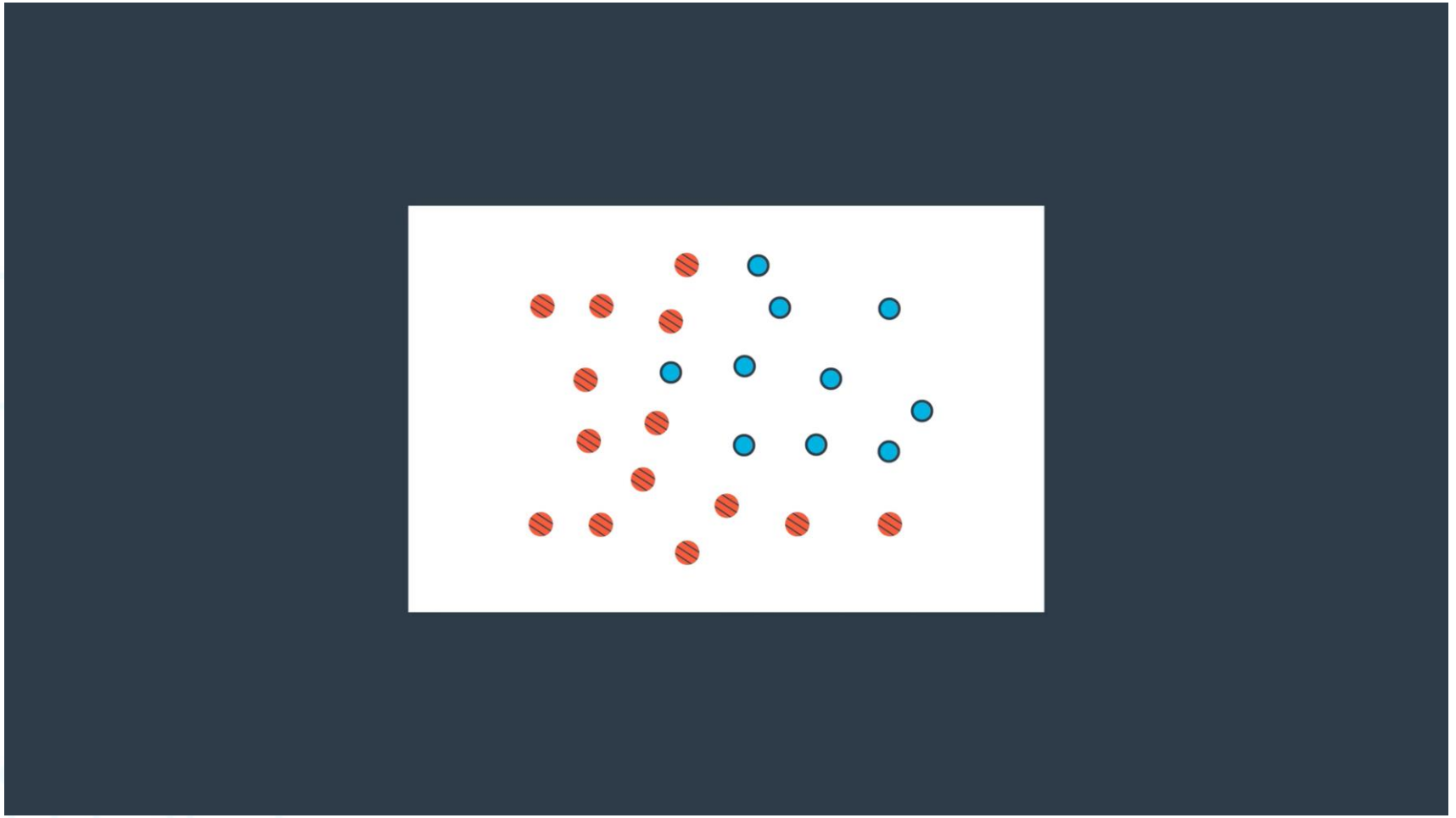
Redes Neurais – Overfitting and Underfitting



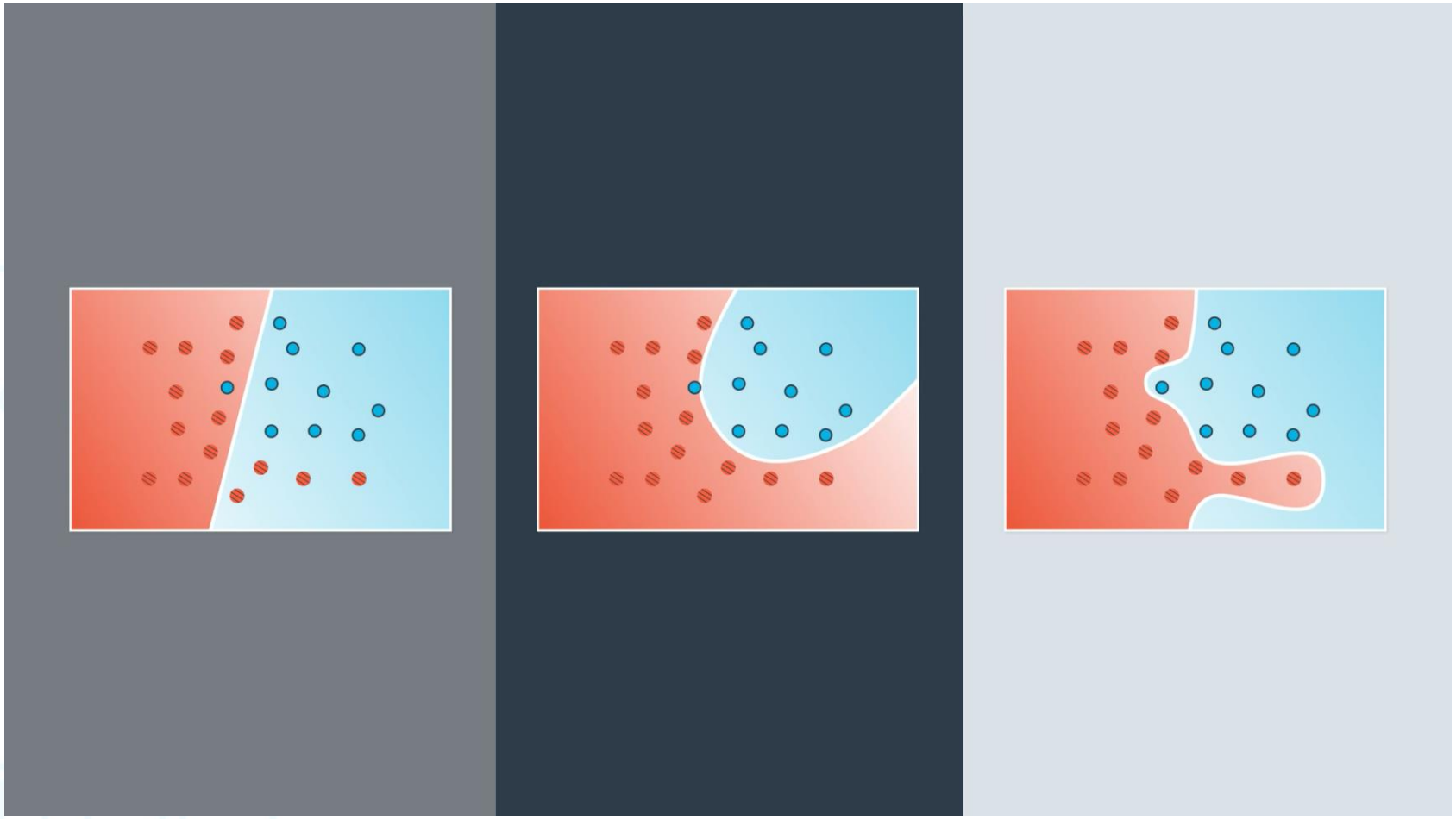
Redes Neurais – Overfitting and Underfitting



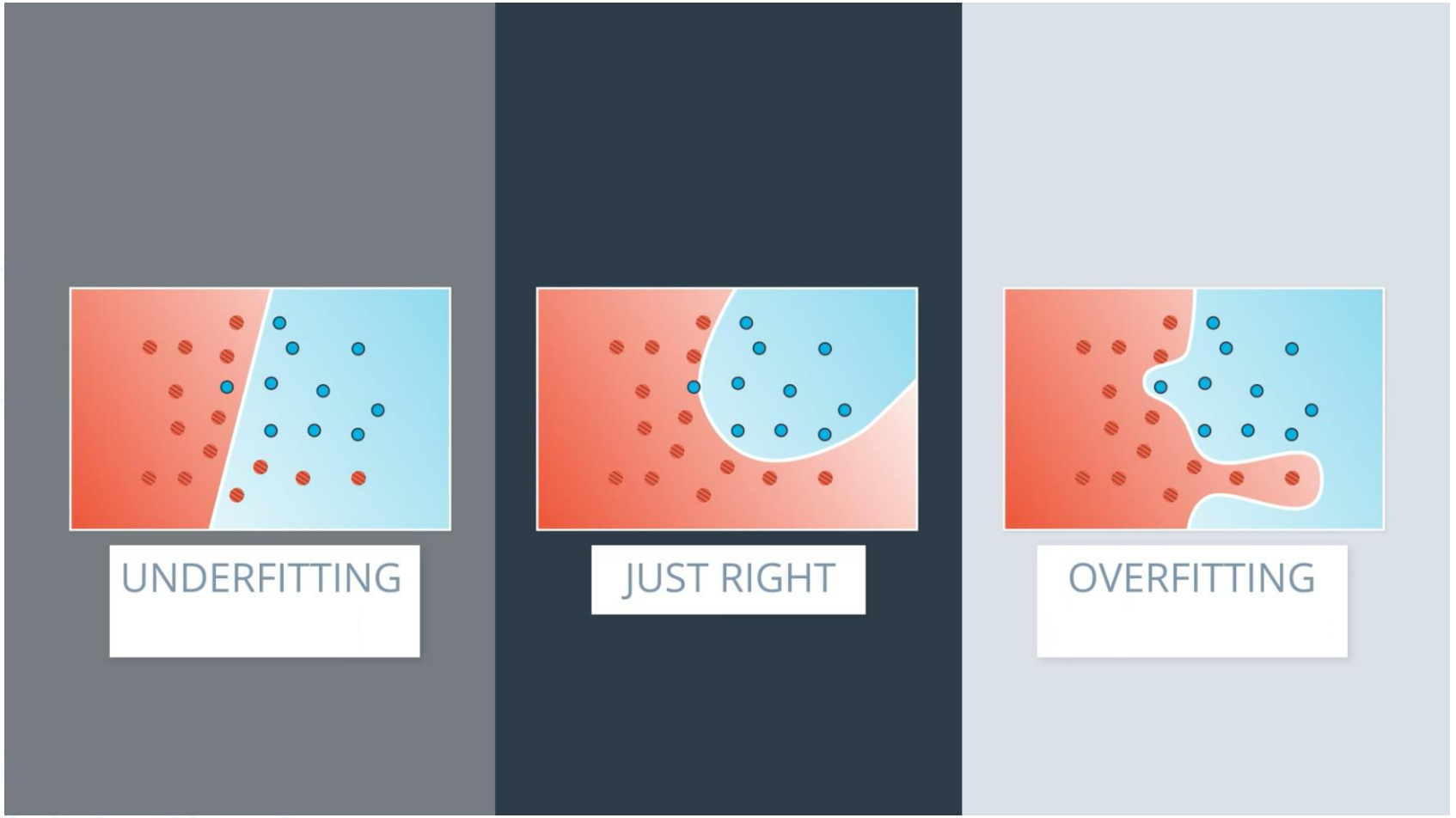
Redes Neurais – Overfitting and Underfitting



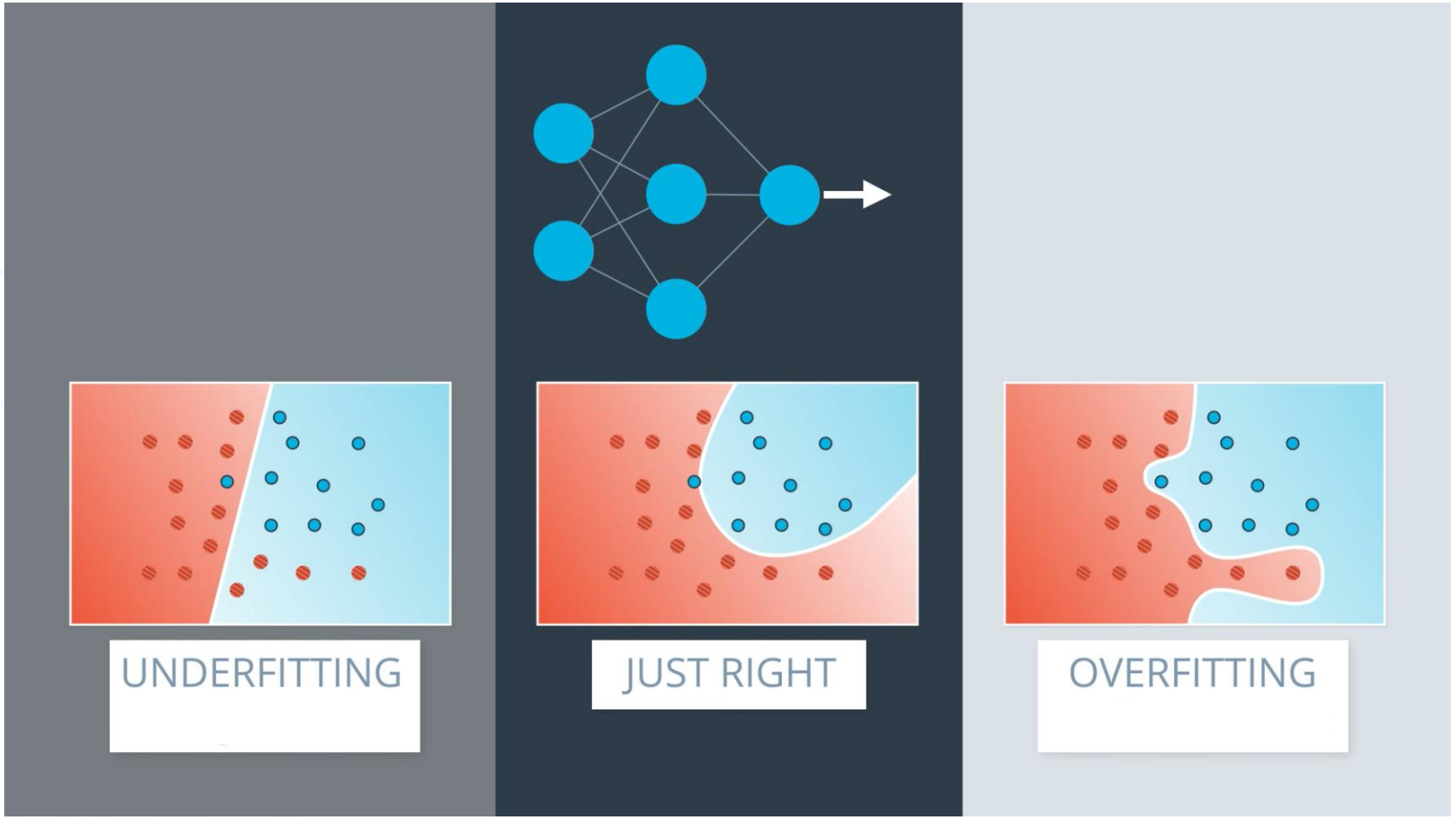
Redes Neurais – Overfitting and Underfitting



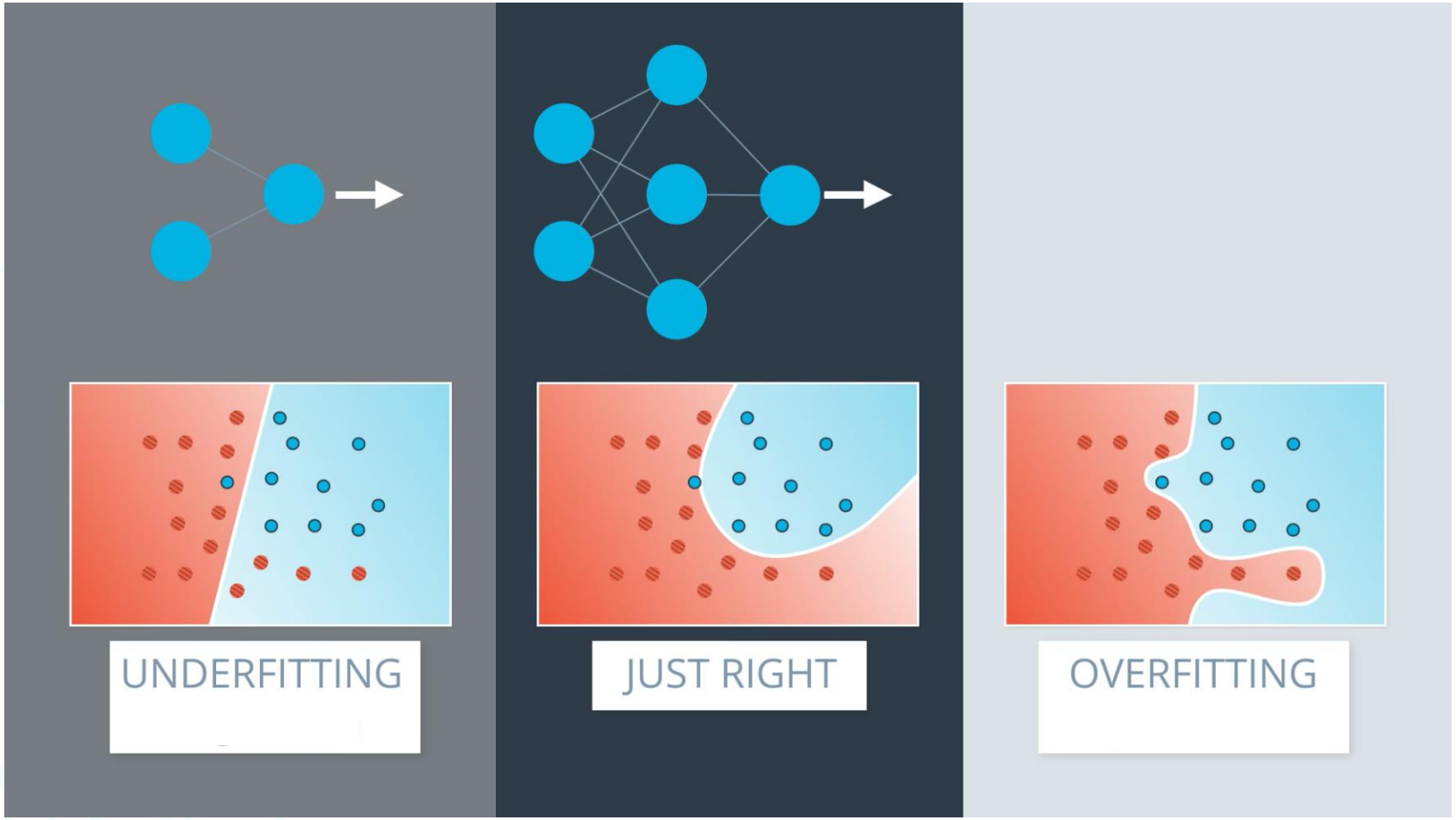
Redes Neurais – Overfitting and Underfitting



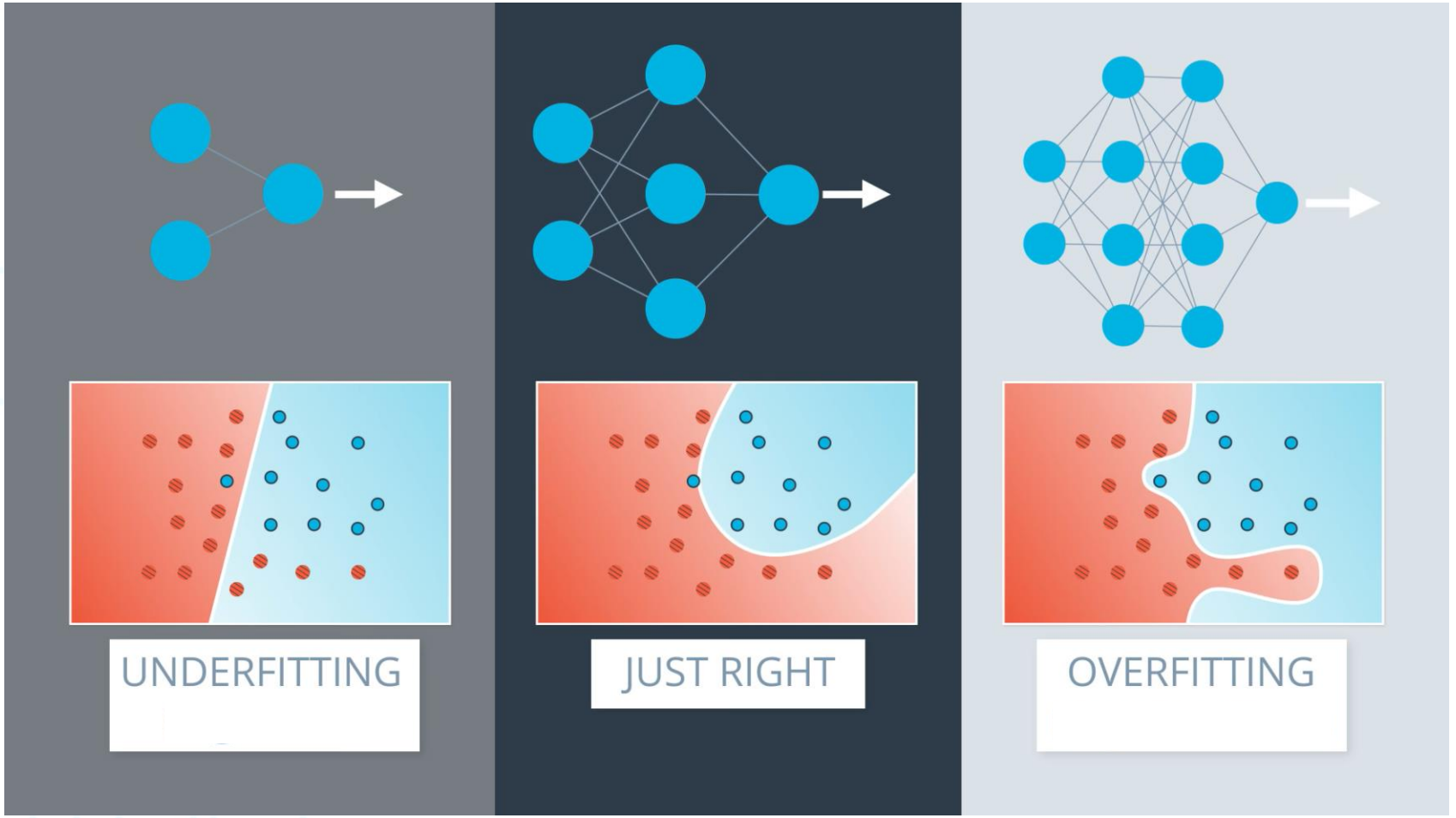
Redes Neurais – Overfitting and Underfitting



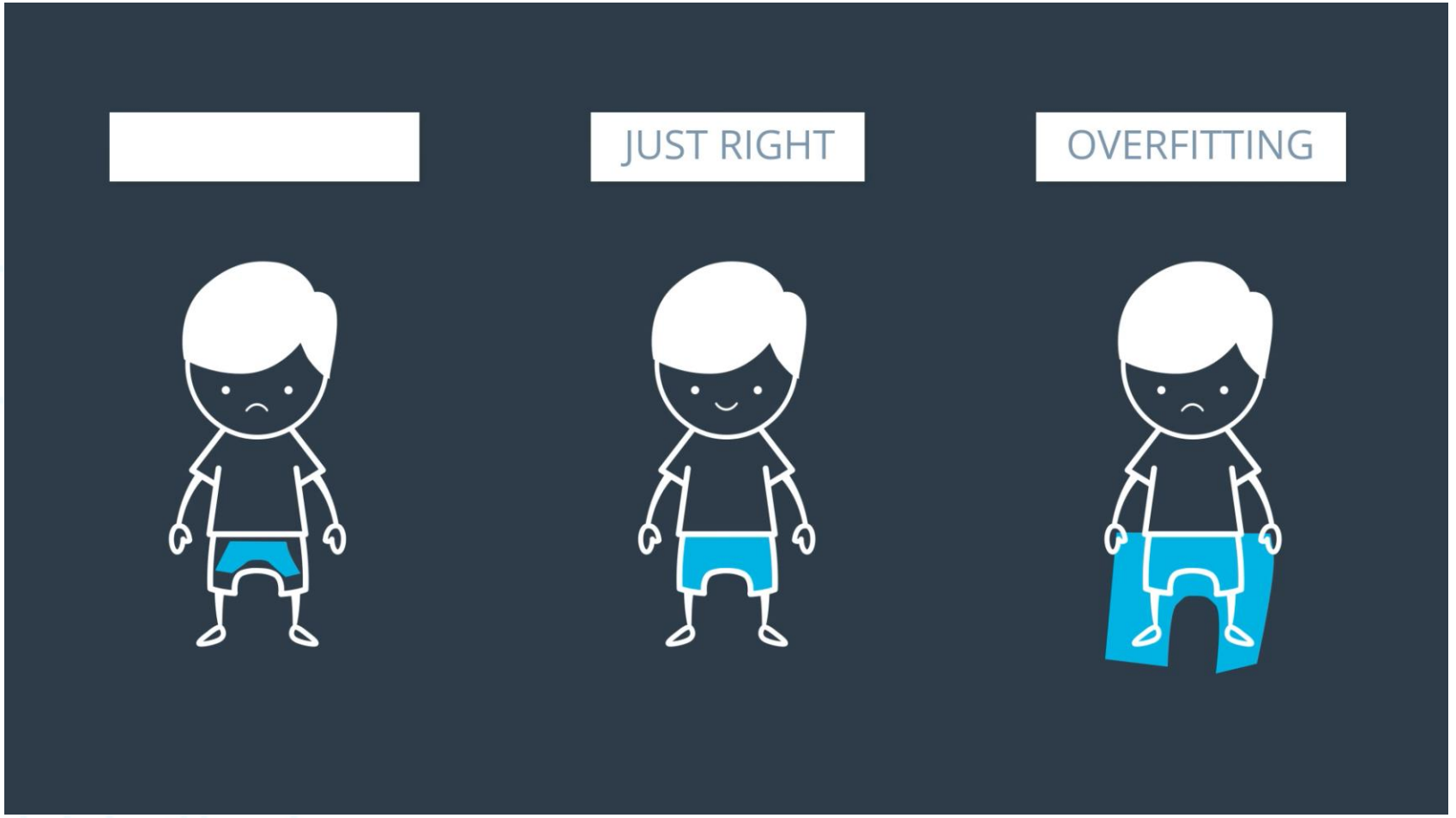
Redes Neurais – Overfitting and Underfitting



Redes Neurais – Overfitting and Underfitting



Redes Neurais – Overfitting and Underfitting



Thanks !

VINICARIDÁ

Human-Centered Data & AI

Vinicius Fernandes Caridá

vfcarida@gmail.com


@vinicius caridá


@vfcarida


@vinicius caridá


@vfcarida


@vinicius caridá


@vfcarida