Ministério da Ciência, Tecnologia e Inovações

Centro Brasileiro de Pesquisas Físicas

Redes Neurais profundas e aplicações Deep Learning

Clécio Roque De Bom – debom@cbpf.br

CBPF

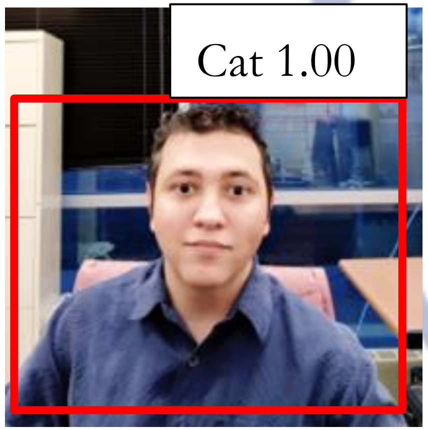
clearnightsrthebest.com

The Modelling ….

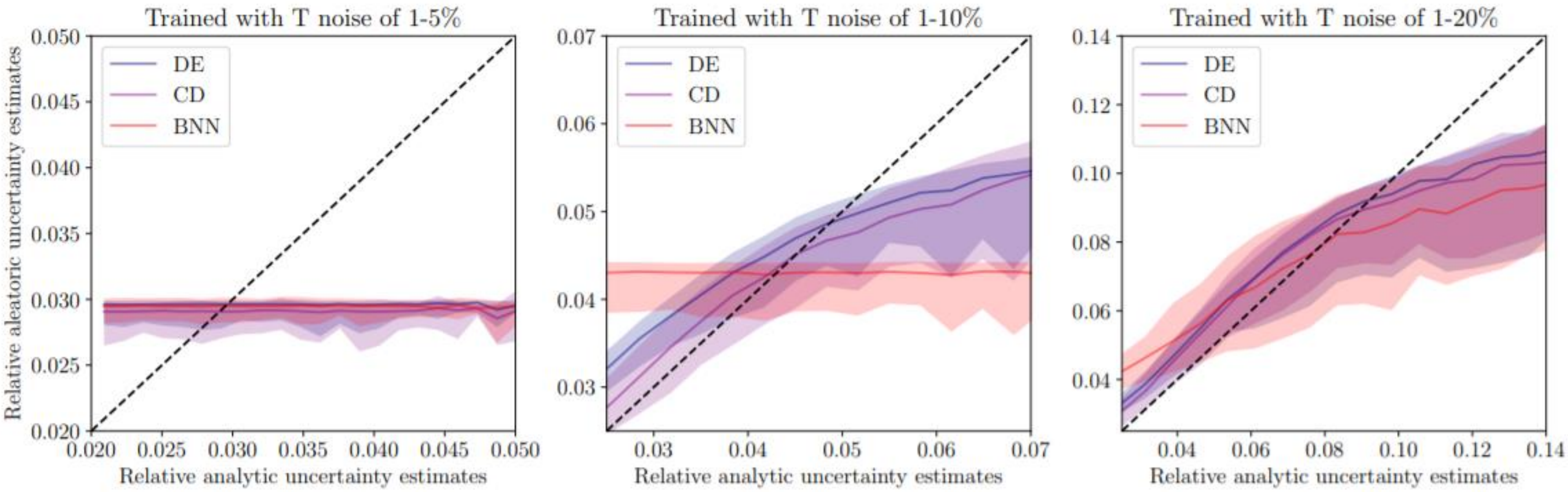
How sure is my NN ?

In a Deep Learning classification problem that classifies dogs and cats would classify a human as a dog or a cat anyway. It would not be possible to know that the image is not a dog, neither a cat.

It would be interesting to have a framework in which one could derive a PDF on the predictions ….



Deeply Uncertain



https://arxiv.org/pdf/2004.10710.pdf

Can we make a reliable model of Strong lensing systems with Deep Neural Networks?

We simulate the Lens population from DES and add constraints to derive if they would be observable, e.g. SNR >20. Images resolved.

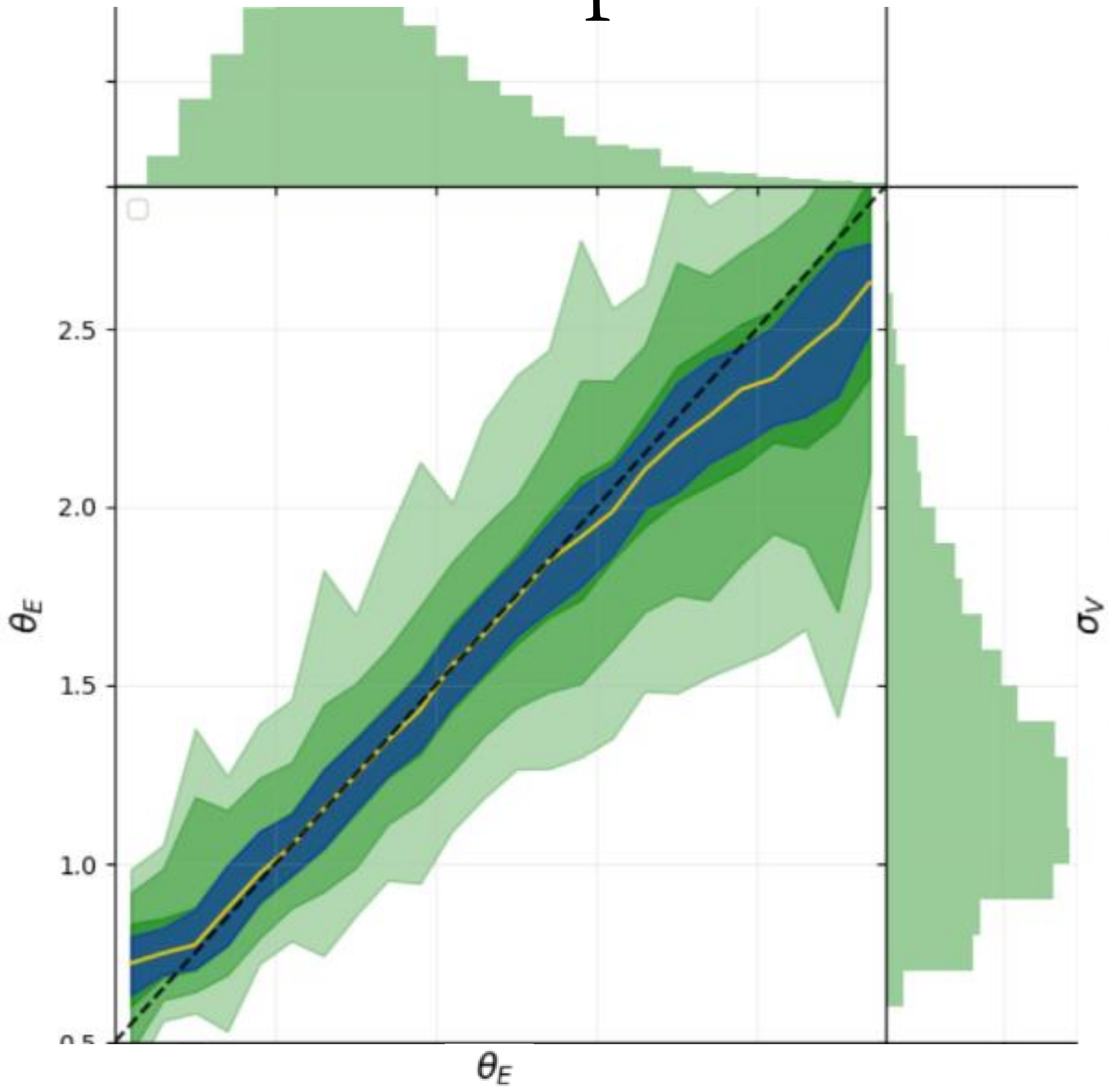
We ended up with 18600 galaxy scale lens systems.

The Lens model is a Singular Isothermal Ellipsoids (SIEs)

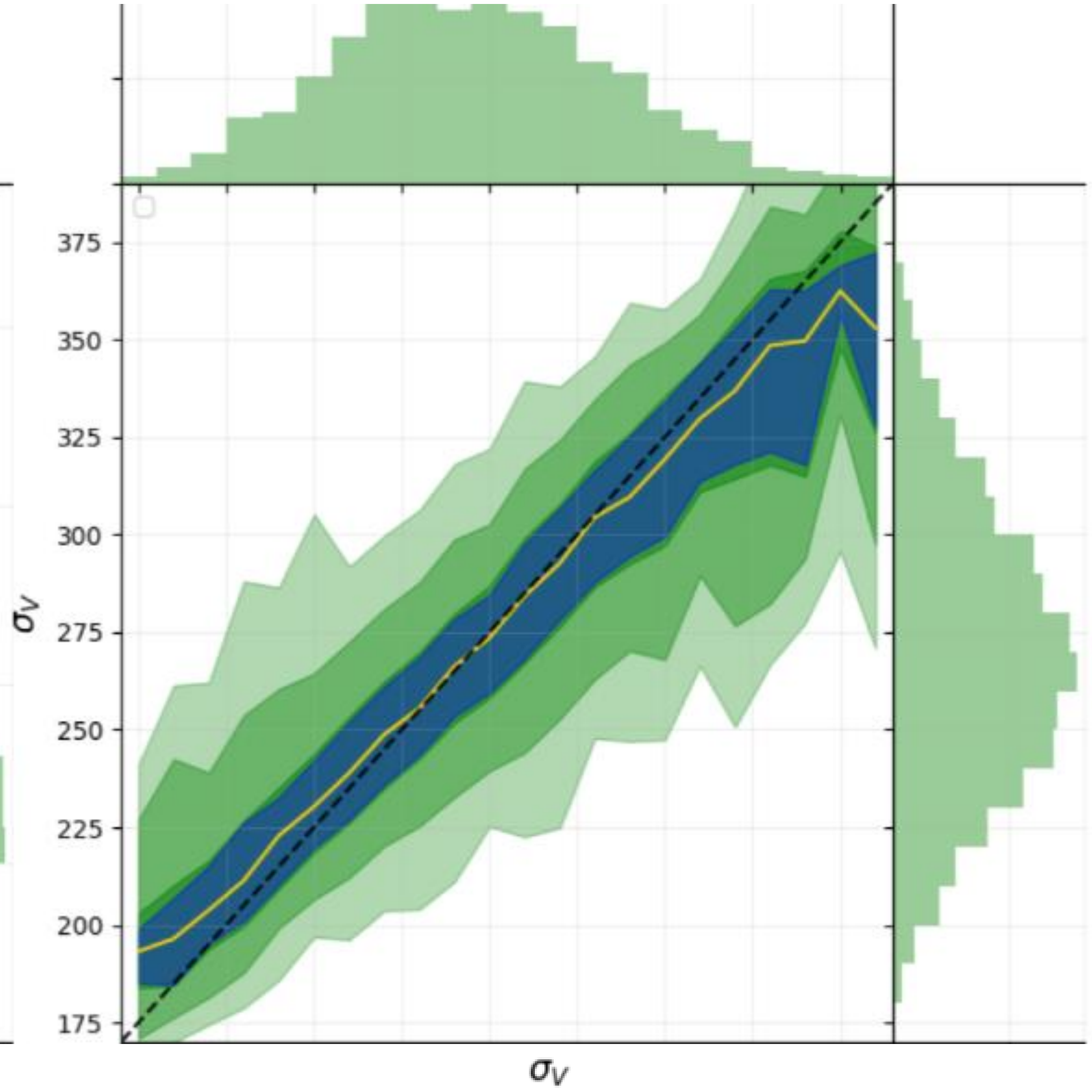
$$\rho(\tilde{r}) = \frac{{\sigma_V}^2}{2\pi G \tilde{r}^2}.$$

$$\theta_{\mathrm{E}}^{\mathrm{SIS}} = 4\pi \frac{\sigma_V^2}{c^2} \frac{D_{\mathrm{ls}}}{D_{\mathrm{s}}}.$$

One to one line plot



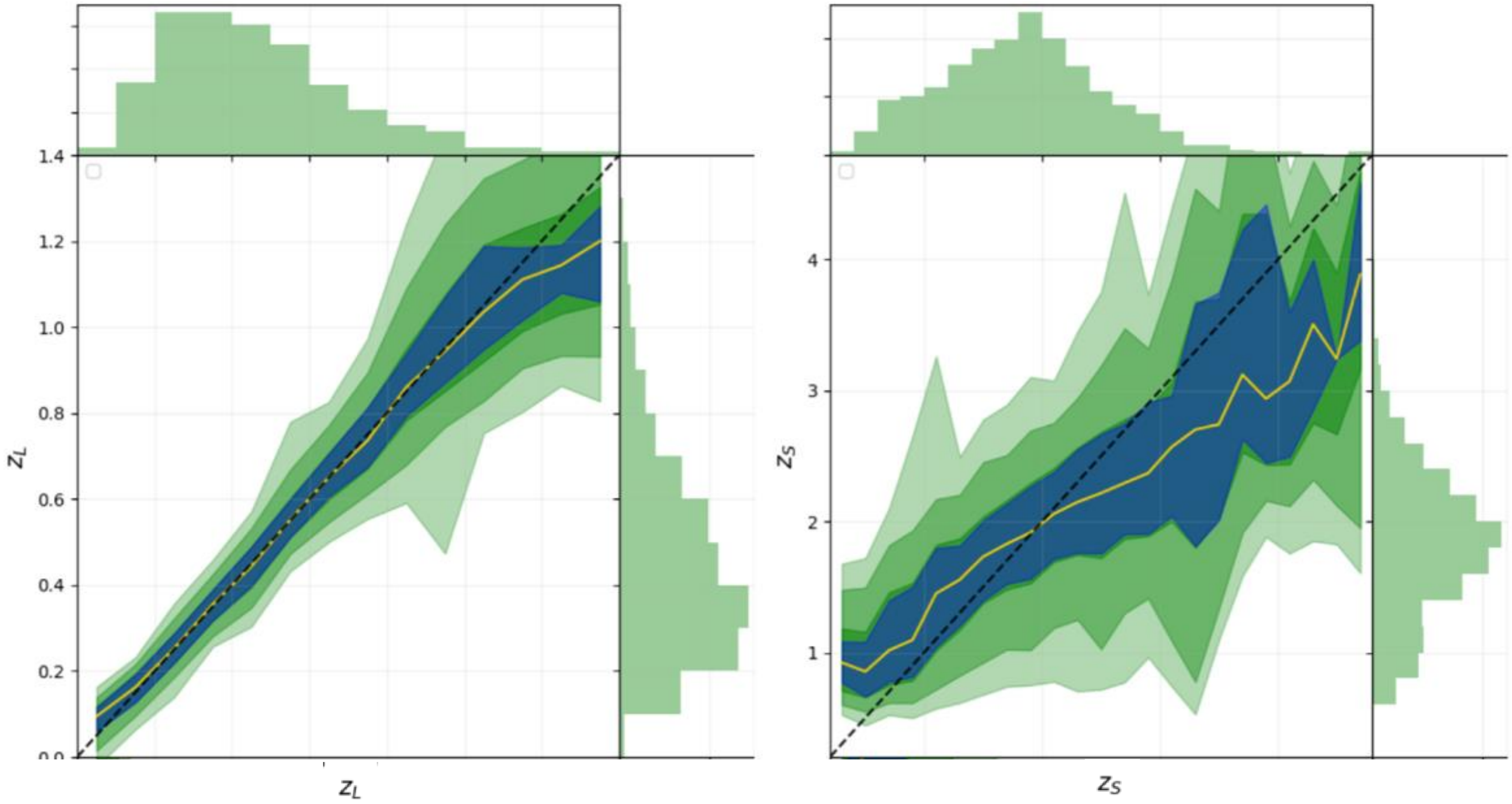
Intuitively correlated with curvature radius



Connected with mass distribution

$$\rho(\tilde{r}) = \frac{{\sigma_V}^2}{2\pi G\tilde{r}^2}.$$

One to one line plot



Connected with the distance through cosmological model

Einstein Rings (idealized case)

Singular Isothermal Sphere: $\rho(r) = \frac{\sigma_v^2}{2\pi G r^2}$

slip parameter

$$\gamma = \frac{\Phi}{\Psi}$$

Einstein Ring in modified gravity $\theta_E = 4\pi\sigma_{\rm obs}^2 \left(\frac{1+\gamma}{2}\right) \frac{D_{LS}}{D_S}$

If $\gamma =$ 1, GR is correct. In our simulated sample we assume GR

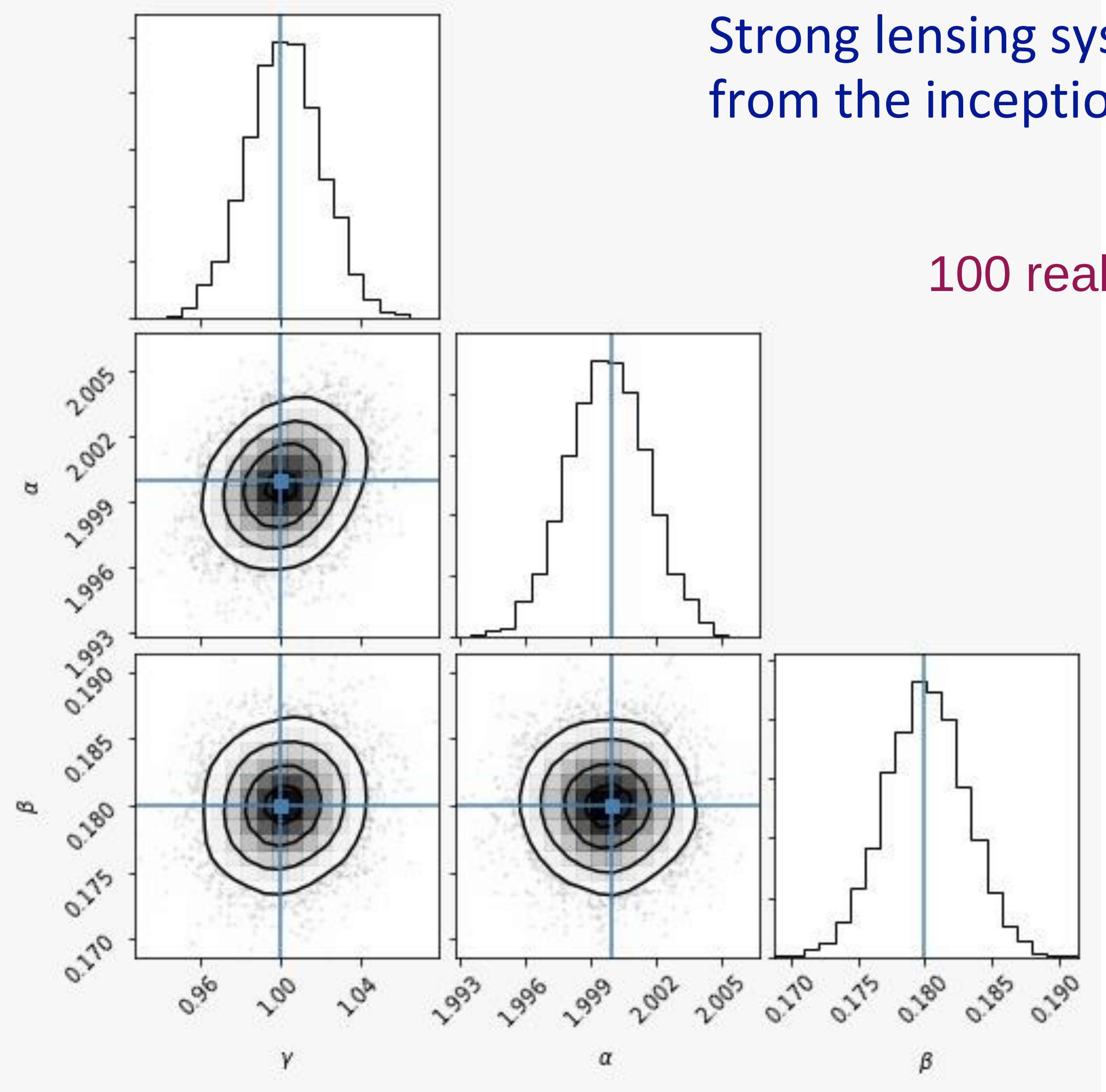
Measure velocity dispersion + Einstein Radius ➔ Test of Einstein General Relativity

Current results need detailed modelling of the lens + z_S, which is hard to obtain

Schwab, Bolton, Rappaport, arXiv:0907.4992

Results

Crisnejo, MM, CB



Strong lensing systems with parameters recovered from the inception neural network

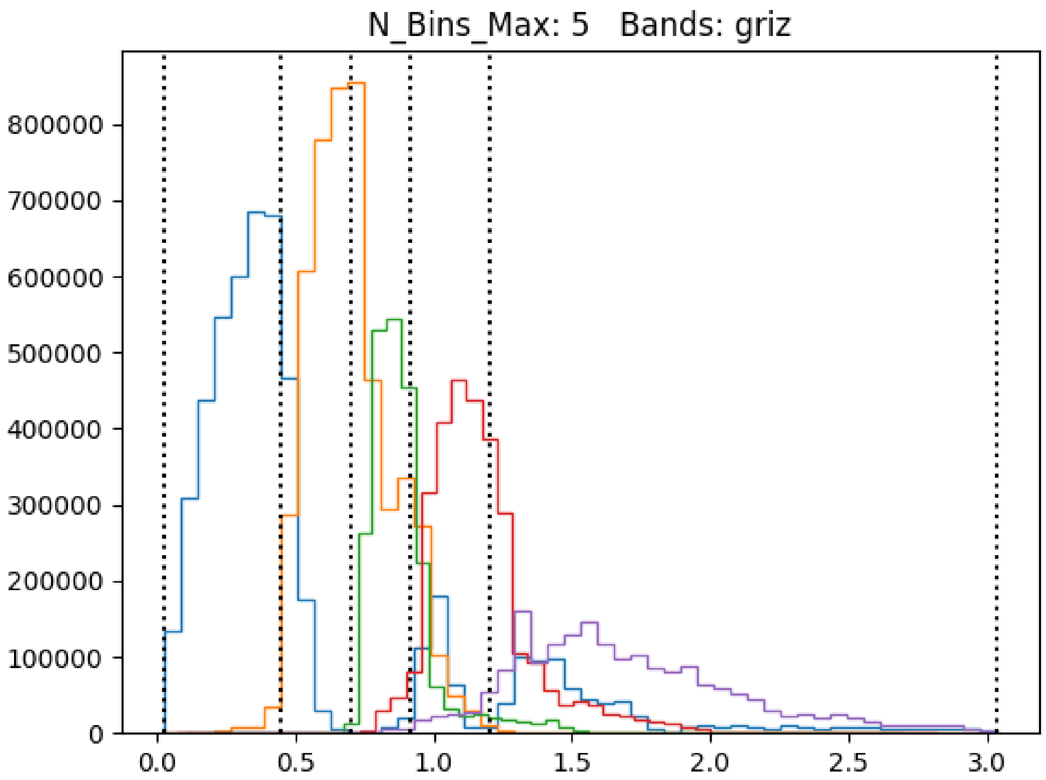
100 realizations of the simulated sample

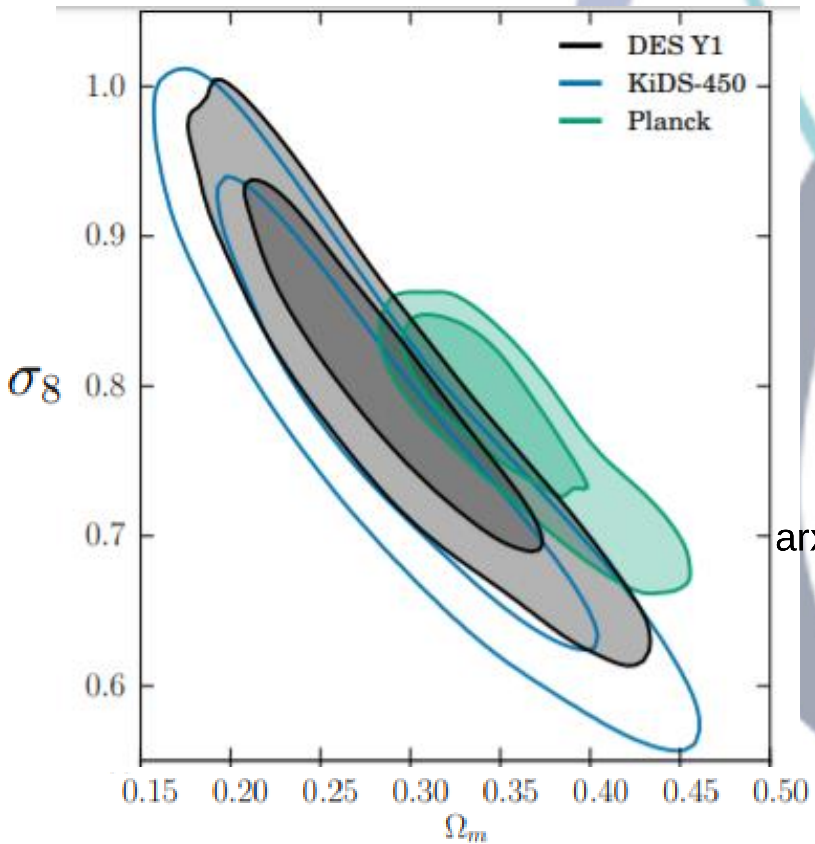
Mean slip parameter

$$\gamma = 1.003 \pm 0.019$$

Redshift Bias Challenge

In this challenge you are asked to group galaxies into tomographic bins using only the quantities we generate using the metacalibration method. These quantities are the only ones for which we can compute a shear bias correction associated with the division.

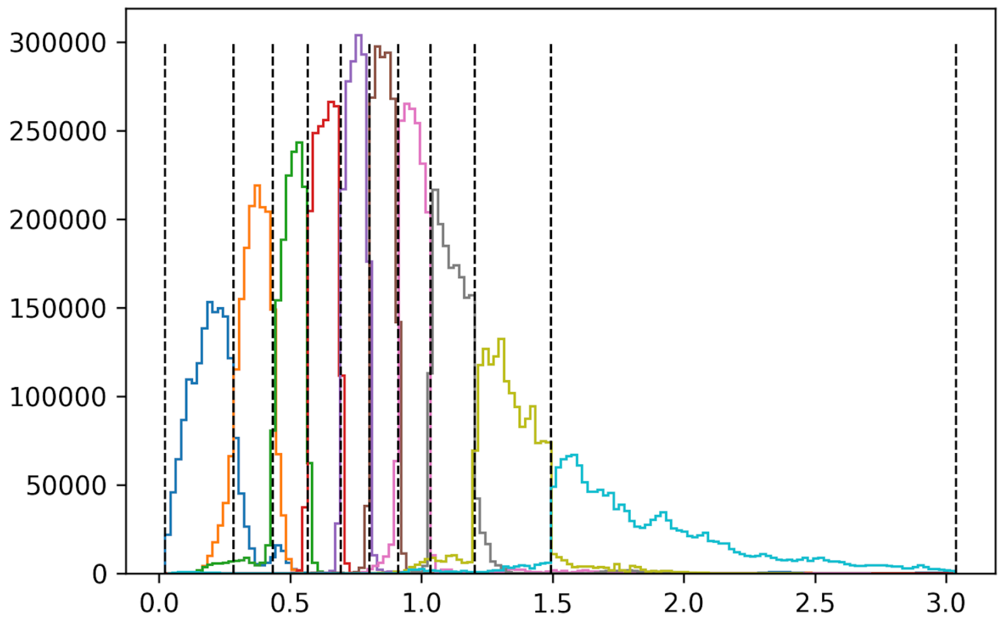




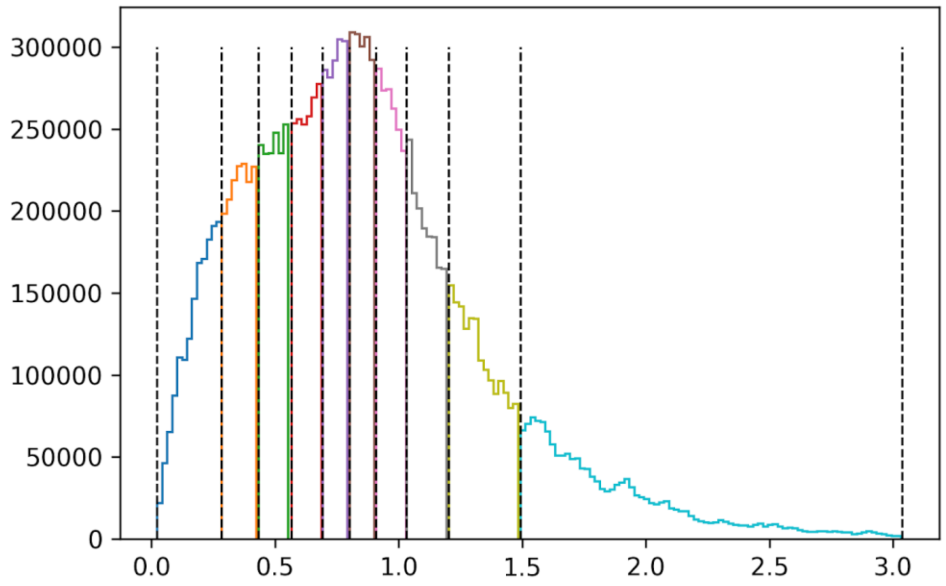
arxiv:1708.01538

Redshift Bias Challenge

In this challenge you are asked to group galaxies into tomographic bins using only the quantities we generate using the metacalibration method. These quantities are the only ones for which we can compute a shear bias correction associated with the division.

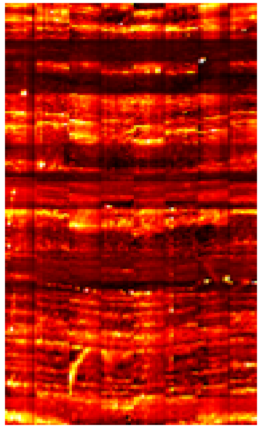


LSTM model with AutoML

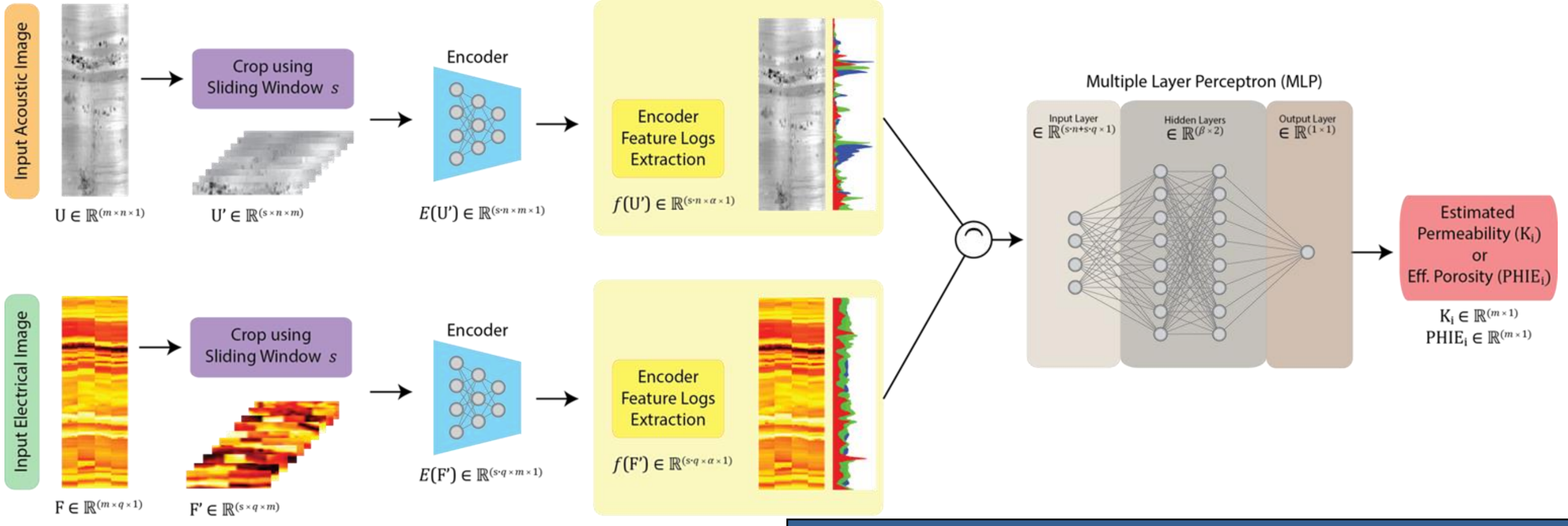


Truth Table

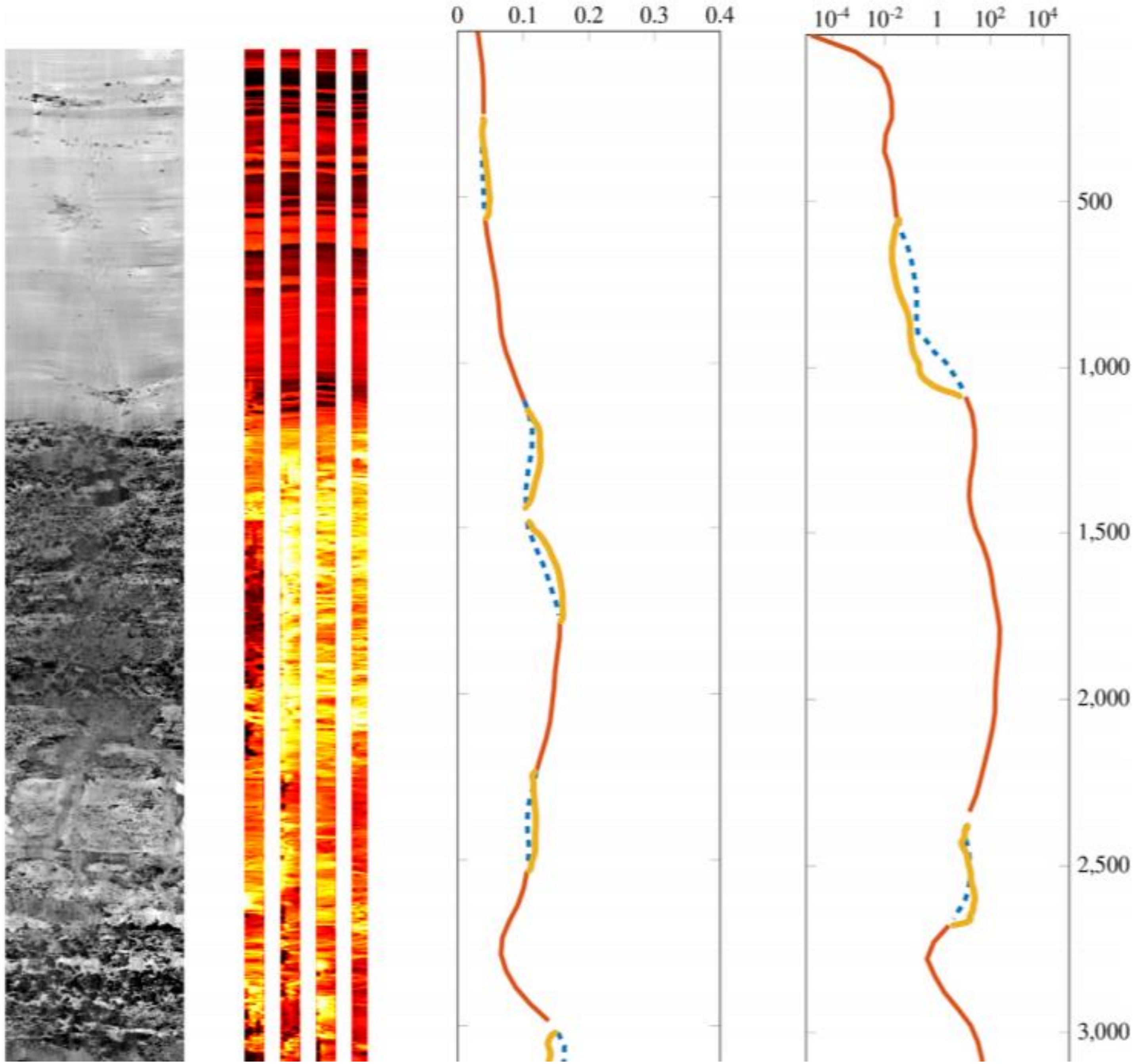
Deep Learning aplicado a perfis de imagens



Como primeira abordagem nós desenvolvemos uma técnica que estiva a permeabilidade e porosidade no poço utilizando perfis de imagens (processados) acústicas e resistivas, recuperando resultado de NMR com Timur-Coates. Blanco-Valentim et al. 2018

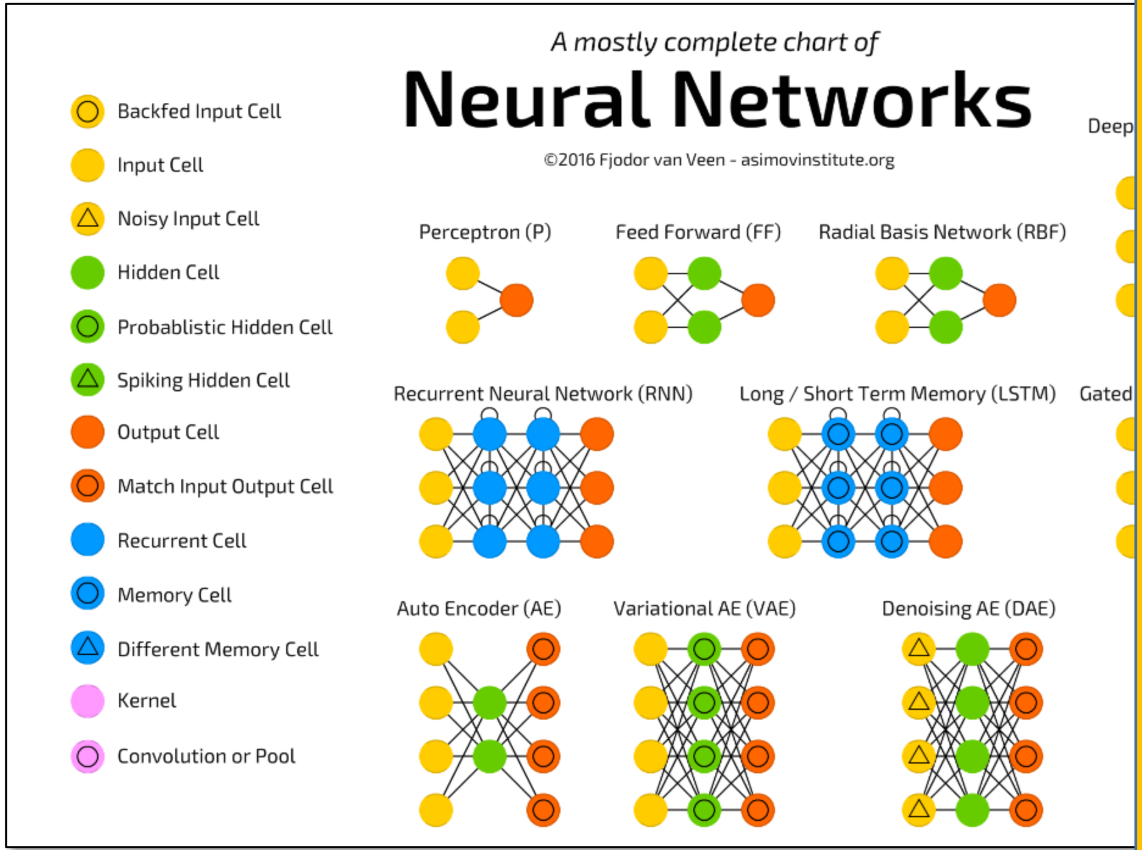


Com Maury Duarte, Anelise de Lima, André Compan, Candida Menezes de Jesus



0
0.1
0.2
0.3
0.4
10-4
10-2
1
10²
10⁴
500
1,000
1,500
2,000
2,500
3,000

Alguns termos importantes



Asimov Institute
http://www.asimovinstitute.org/neural-network-zoo/

Alguns termos básicos

Deep Learning é uma Rede Neural Artificial
Deep Learning é uma área de Machine Learning

Alguns Termos de Redes Neurais Artificiais

- **MLP: Multi-layer Perceptron**
- **DNN: Deep Neural Networks**
- **RNN: Recurrent Neural Networks**
  - **LSTM: Long Short-Term Memory**
- **CNN ou ConvNet: Convolution Neural Network**

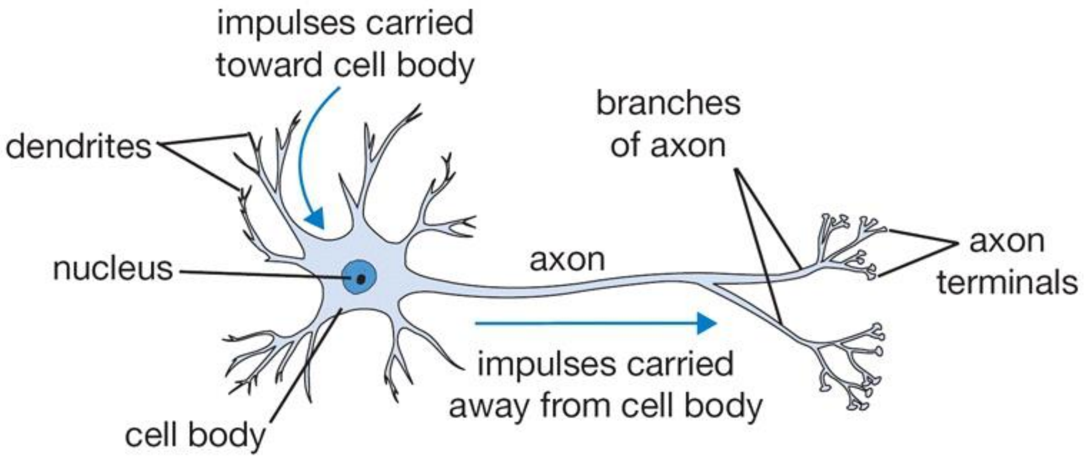
Operações das Redes Neurais Artificiais

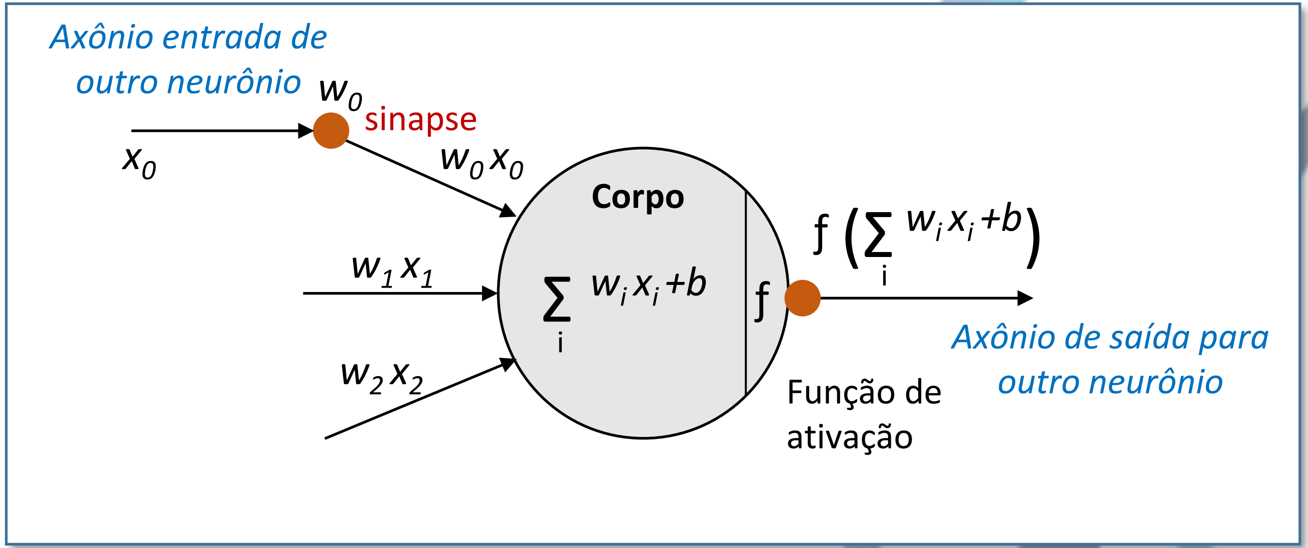
- **Convolução**
- **Pooling**
- **Função de Ativação**
- **Backpropagation**

Redes Neurais Artificiais (RNAs) são modelos computa
particular o cérebro) que são capazes de realizar o aprendizado de máquina bem como o reconhecimento de padrões.

Neurônio

Computação (Neurônio Artificial) → Inspiração na Biologia





Neurônio Biológico: **bloco computacional de processamento do cérebro.**

Cérebro Humano: ~100 – 1.000 trilhões de sinapses

10.000 x

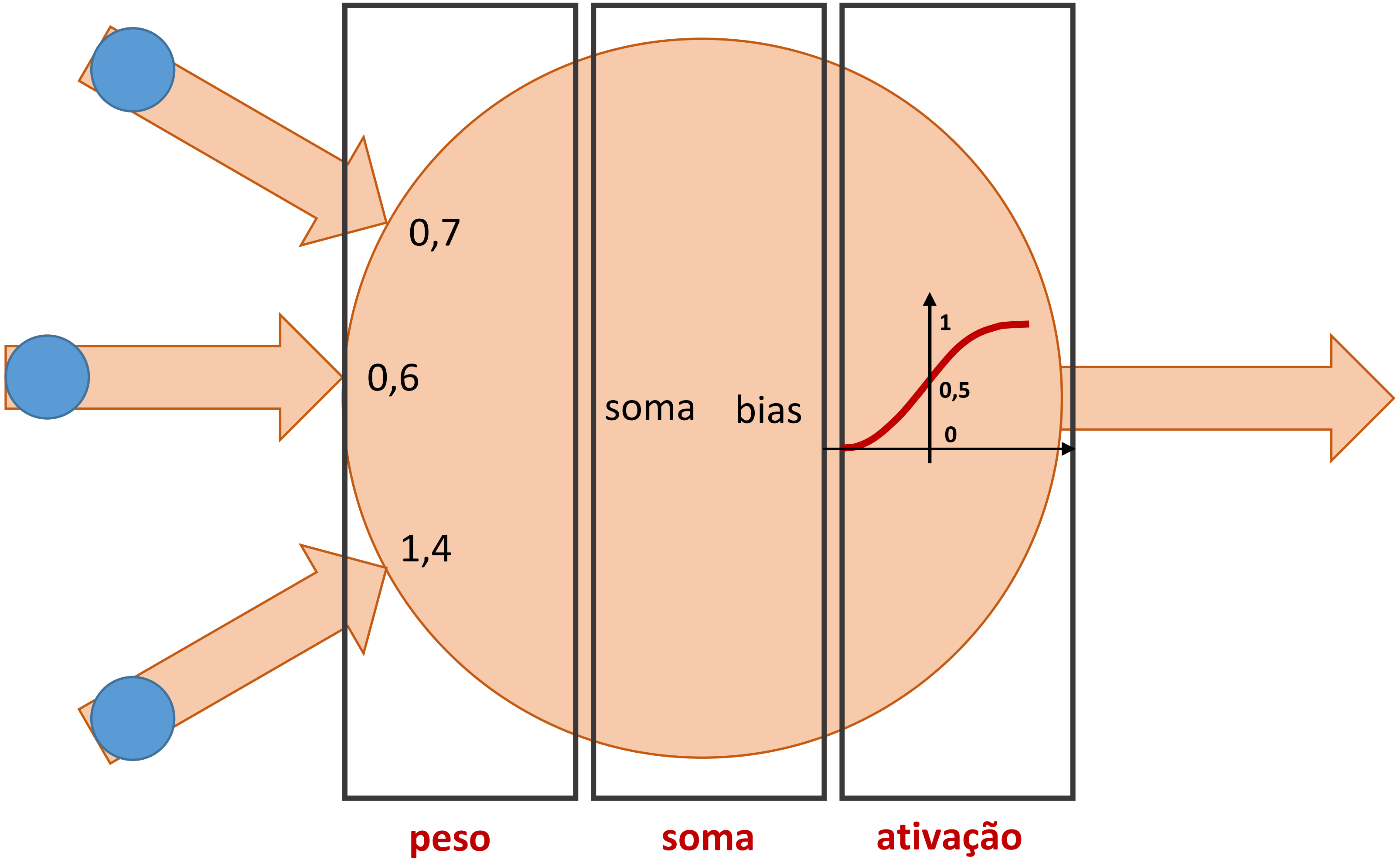
Neurônio Artificial: **bloco computacional de processamento das Redes Neurais Artificiais.**

Rede Neural Artificial : ~1 – 10 bilhões de sinapses.

Rede Neural Artificial

Perceptron

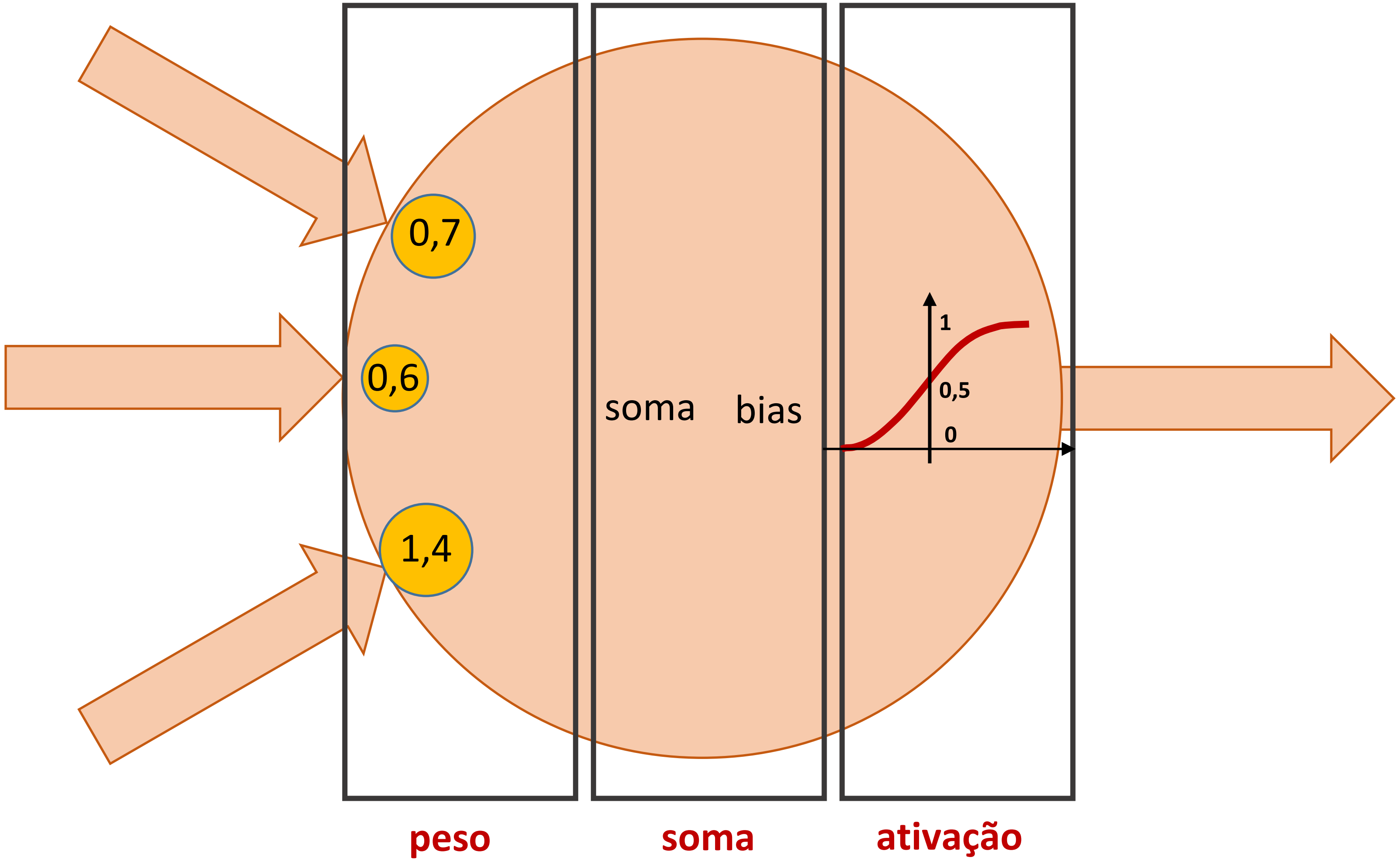
Frank Rosenblatt (1957)



Rede Neural Artificial

Perceptron

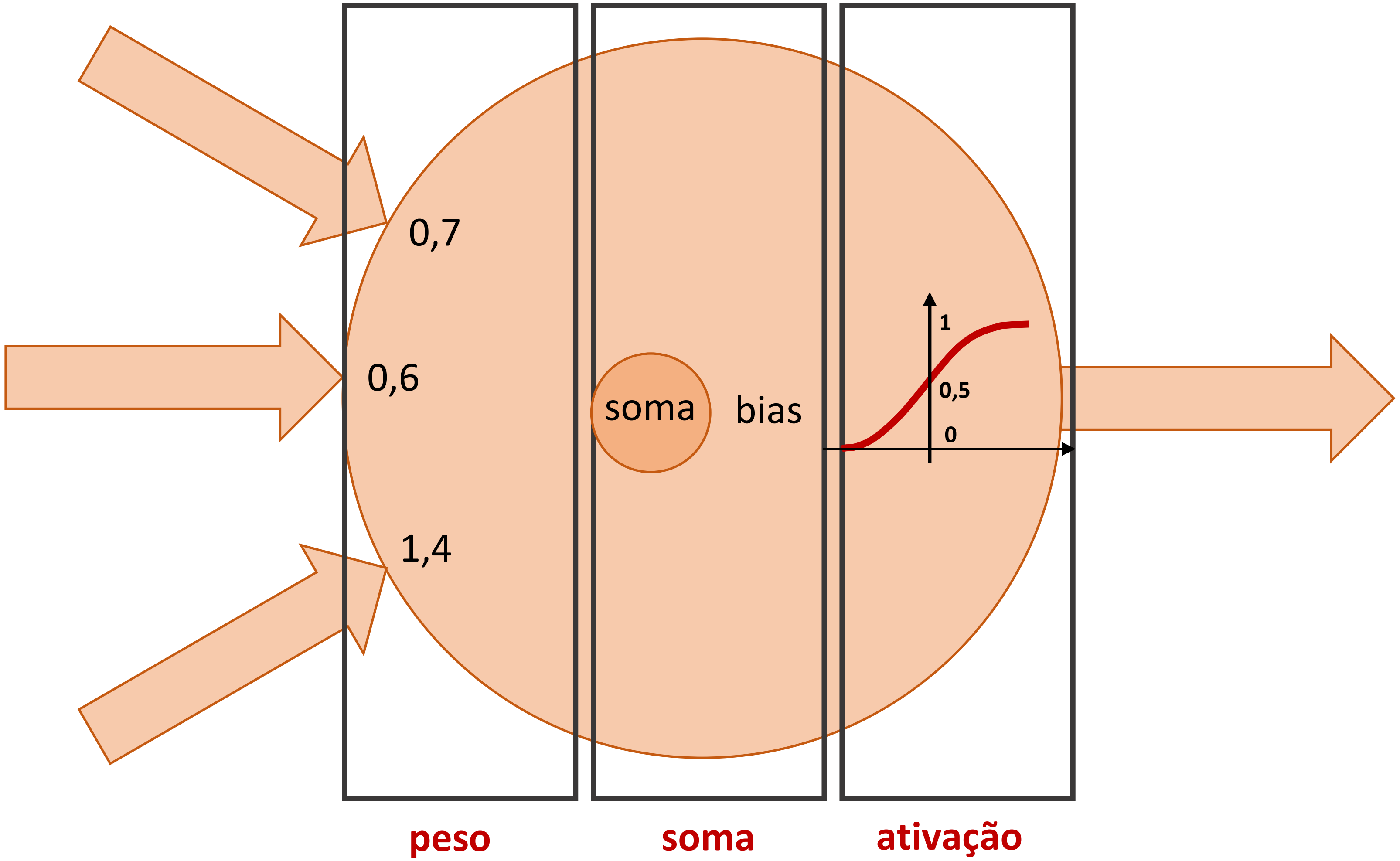
Frank Rosenblatt (1957)



Rede Neural Artificial

Perceptron

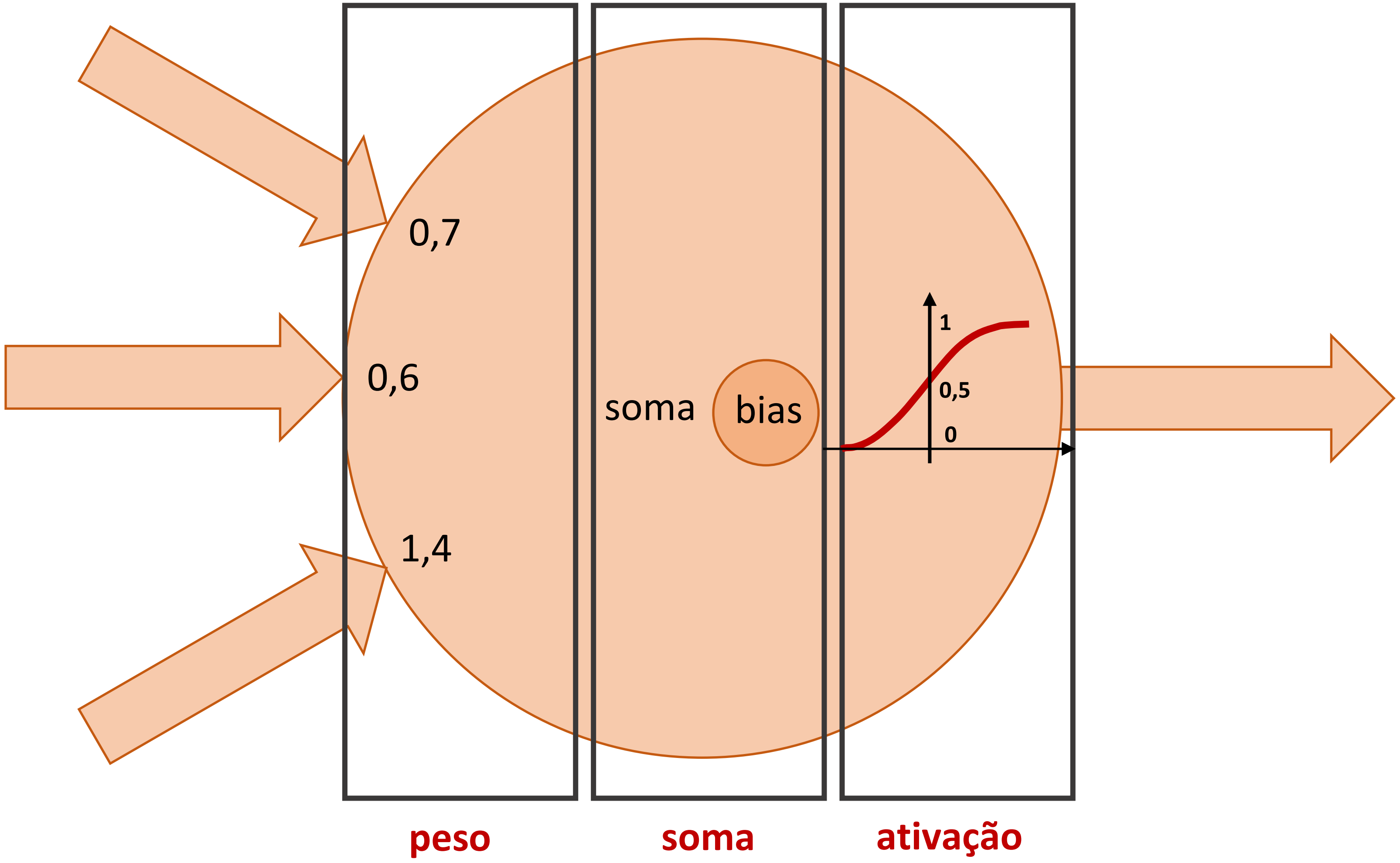
Frank Rosenblatt (1957)



Rede Neural Artificial

Perceptron

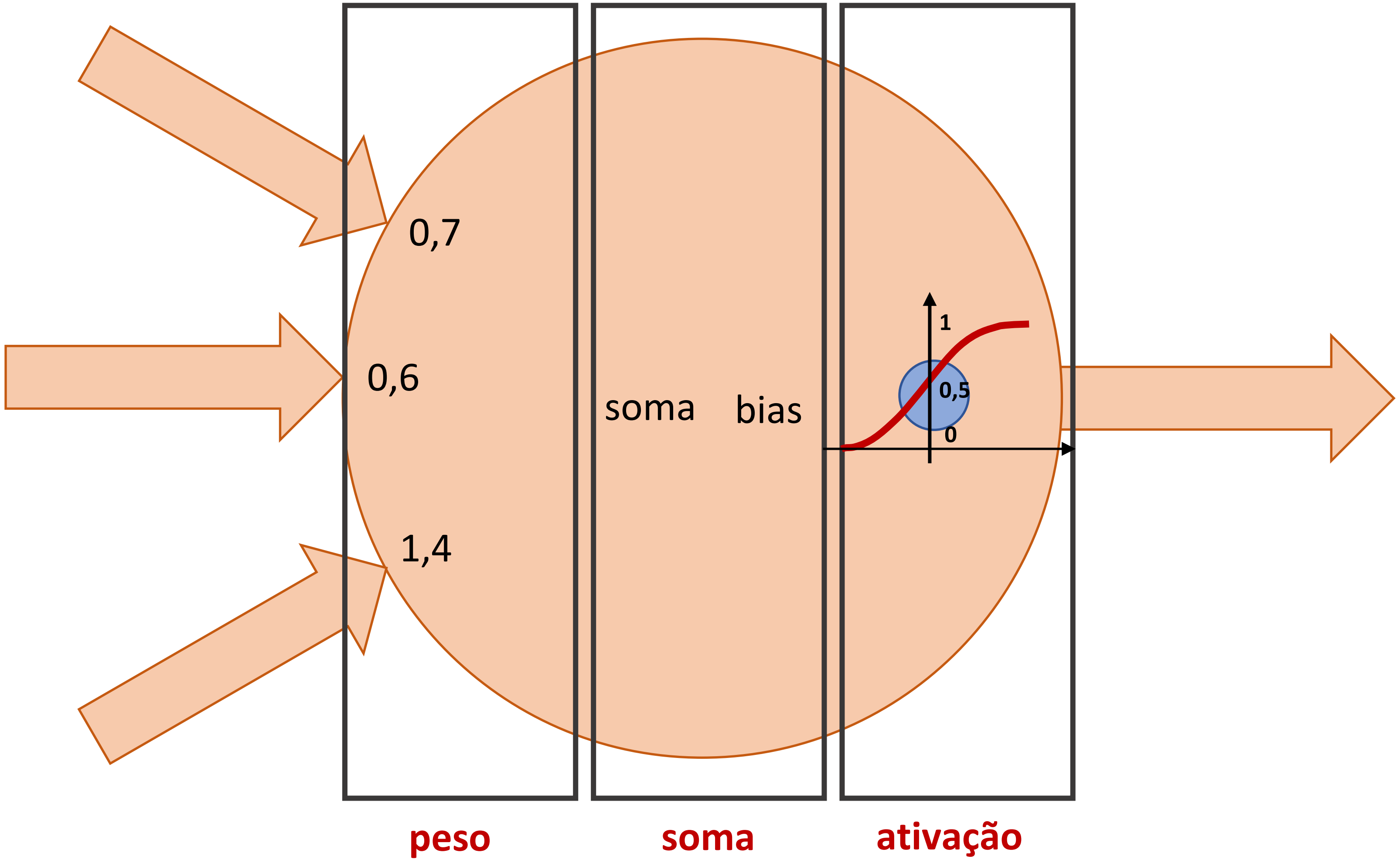
Frank Rosenblatt (1957)



Rede Neural Artificial

Perceptron

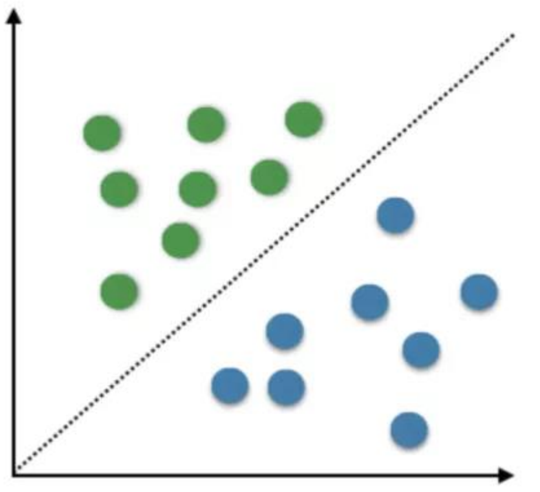
Frank Rosenblatt (1957)

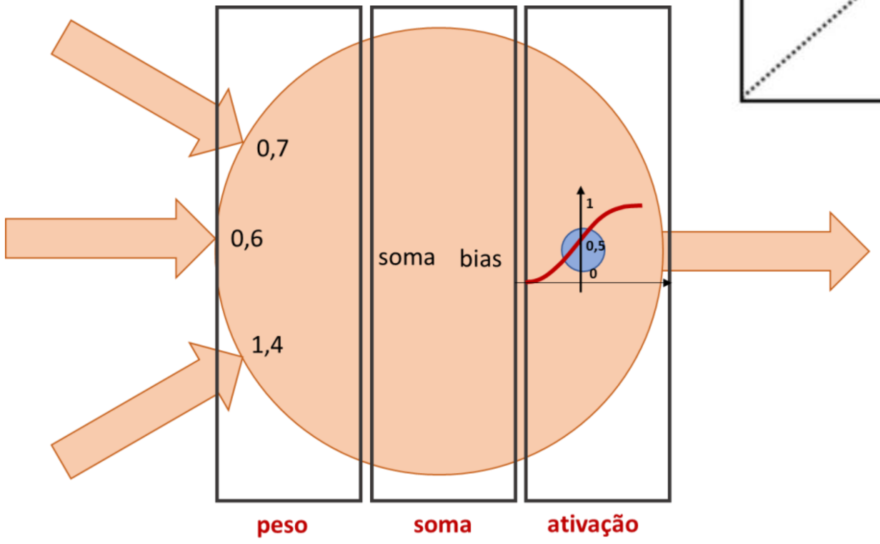


Rede Neural Artificial

Perceptron

Frank Rosenblatt (1957)





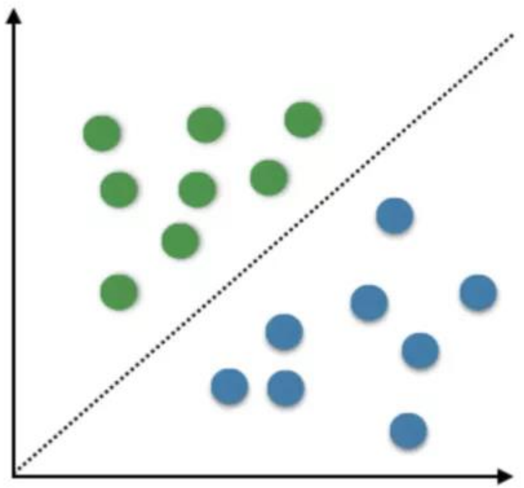
Algoritmo

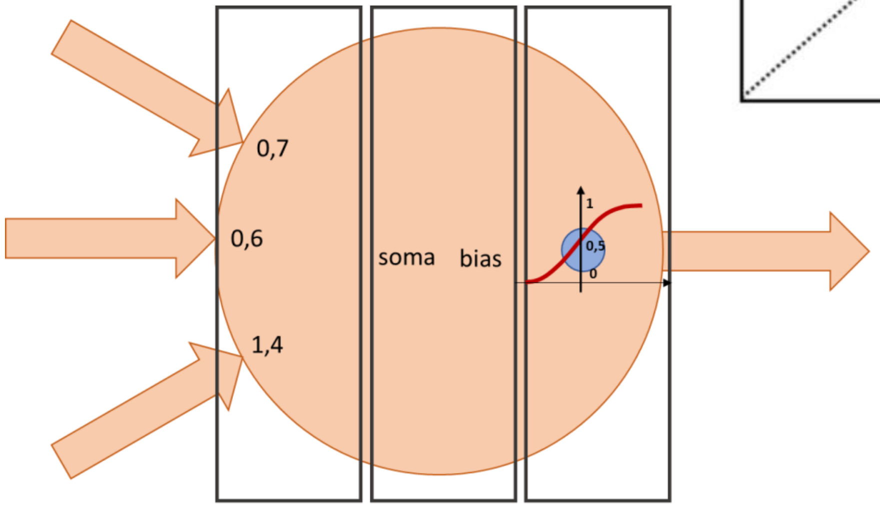
- **Inicialize a rede Perceptron com pesos (w) aleatórios;**
- **Para uma data entrada, processe a saída da rede;**
- **Se a saída da rede não for igual a saída desejada, então a rede deve ser alterada, trocando os valores dos pesos (w) das sinapses;**
- **Repita esse procedimento com todos os dados de treinamento até a rede Perceptron não apresentar mais erros.**

Rede Neural Artificial

Perceptron

Frank Rosenblatt (1957)

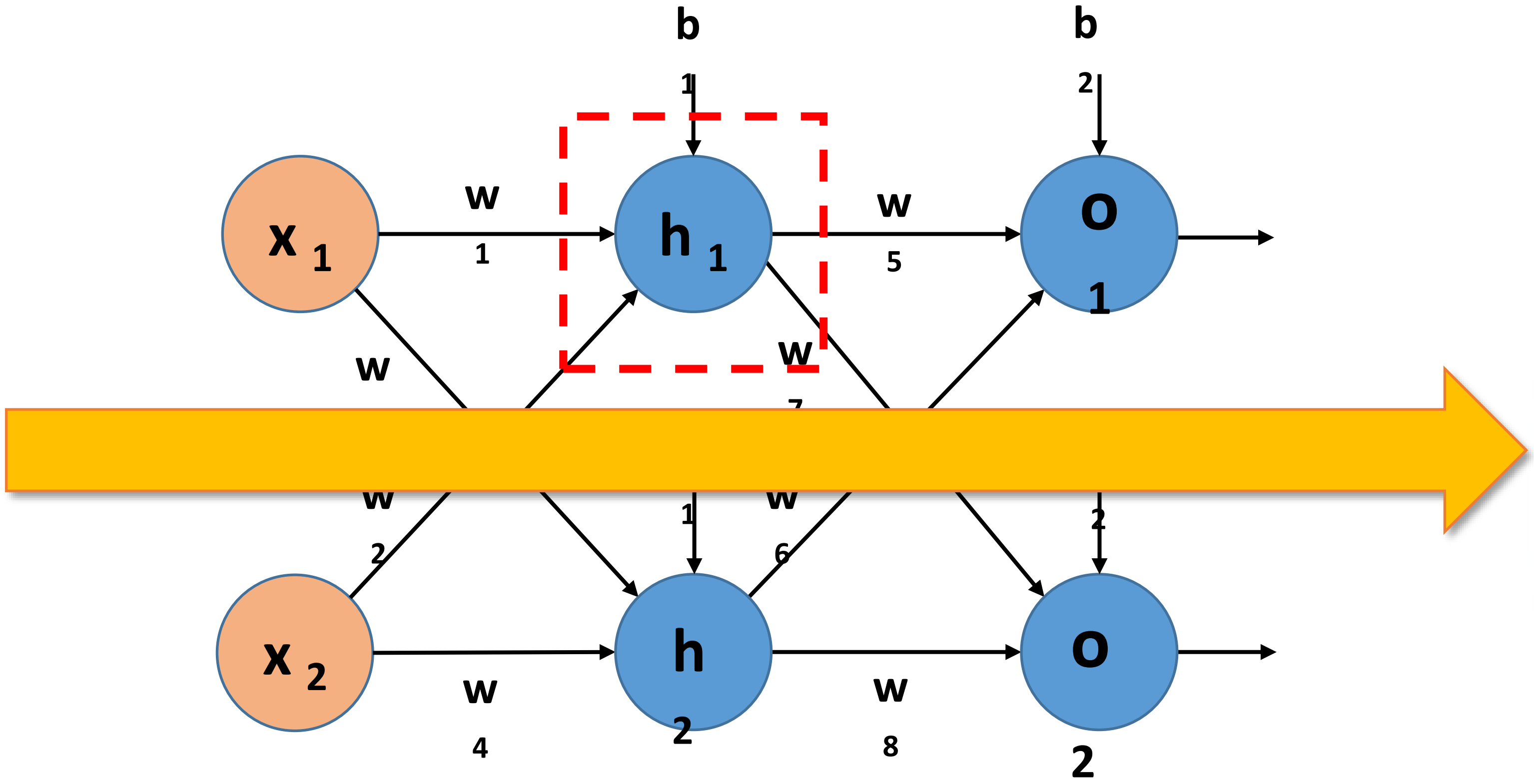




- **Inicialize a rede Perceptron com pesos (w) aleatórios;**
- **Para uma data entrada, processe a saída da rede;**
- **Se a saída da rede não for igual a saída desejada, então a rede deve ser alterada, trocando os valores dos pesos (w) das sinapses;**
- **Repita esse procedimento com todos os dados de treinamento até a rede Perceptron não apresentar mais erros.**

Feedforward and Backpropagation

Fase 1
Propagação

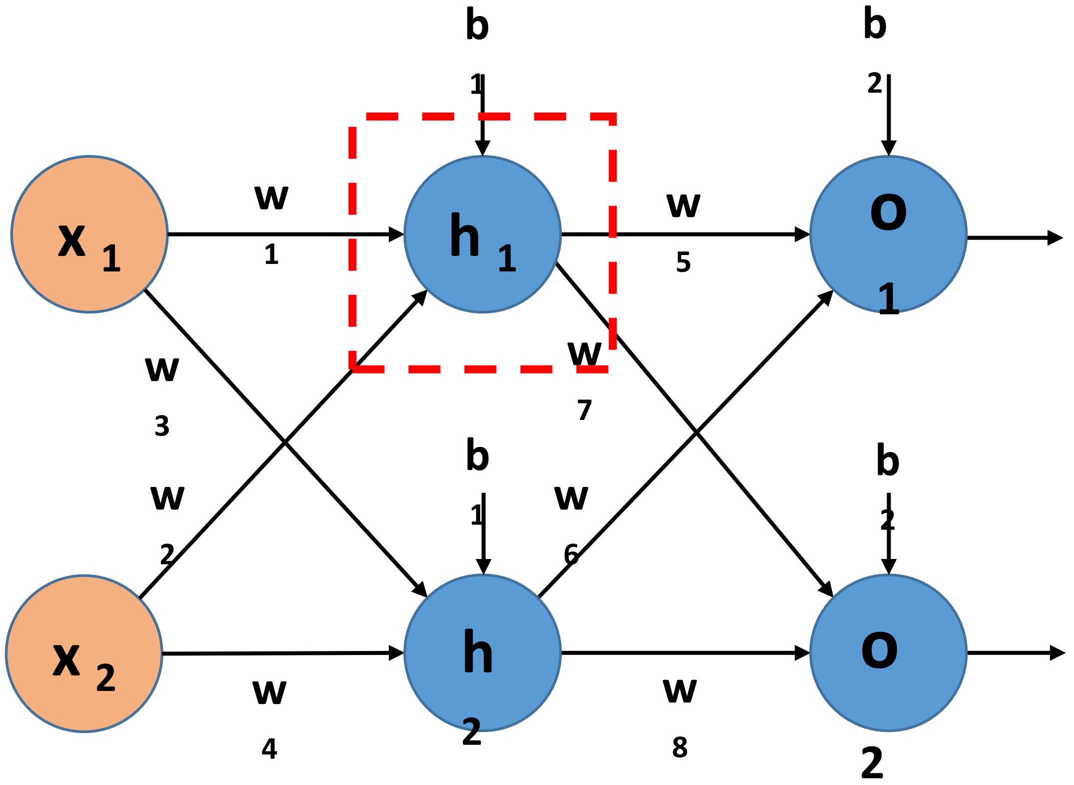


$$uh_1 = x_1 * w_1 + x_2 * w_2 + b_1 * 1$$

$$\mathrm{g}(h_1) = \mathrm{g}(uh_1) = \frac{1}{1+e^{uh_1}}$$

Feedforward and Backpropagation

Fase 1
Propagação

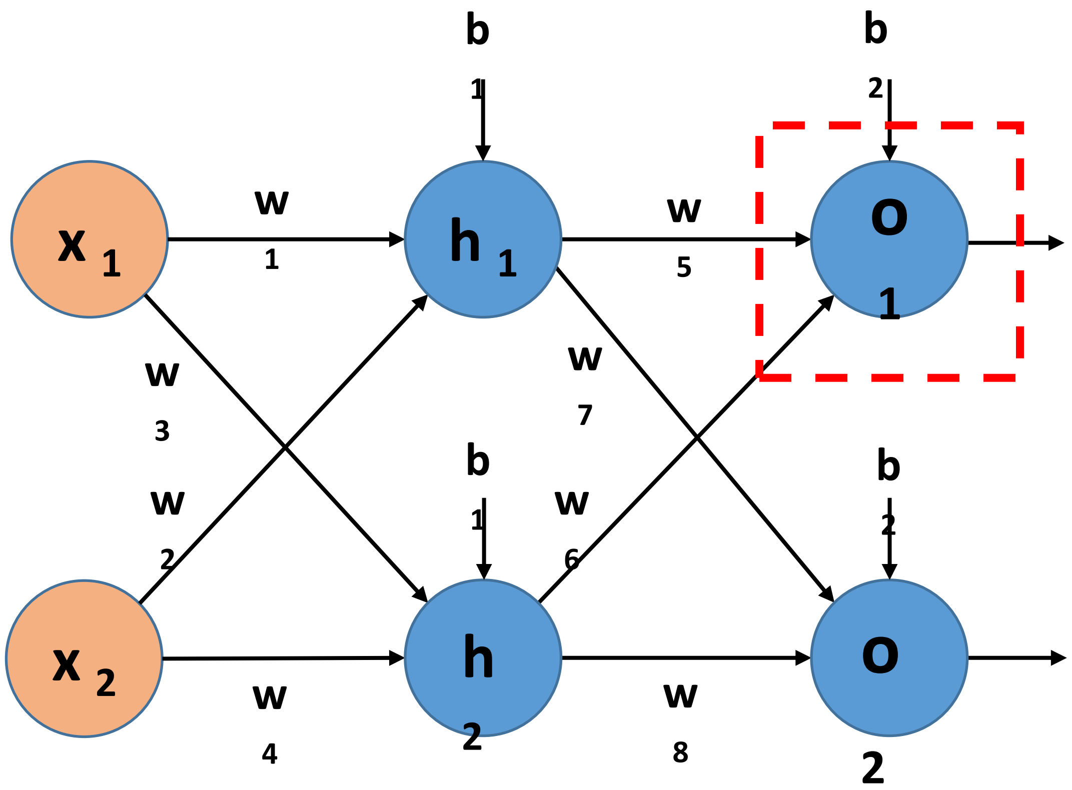


$$uh_2 = x_1 * w_3 + x_2 * w_4 + b_1 * 1$$

$$g(h_2) = g(uh_2) = \frac{1}{1+e^{uh_2}}$$

Feedforward and Backpropagation

Fase 1
Propagação

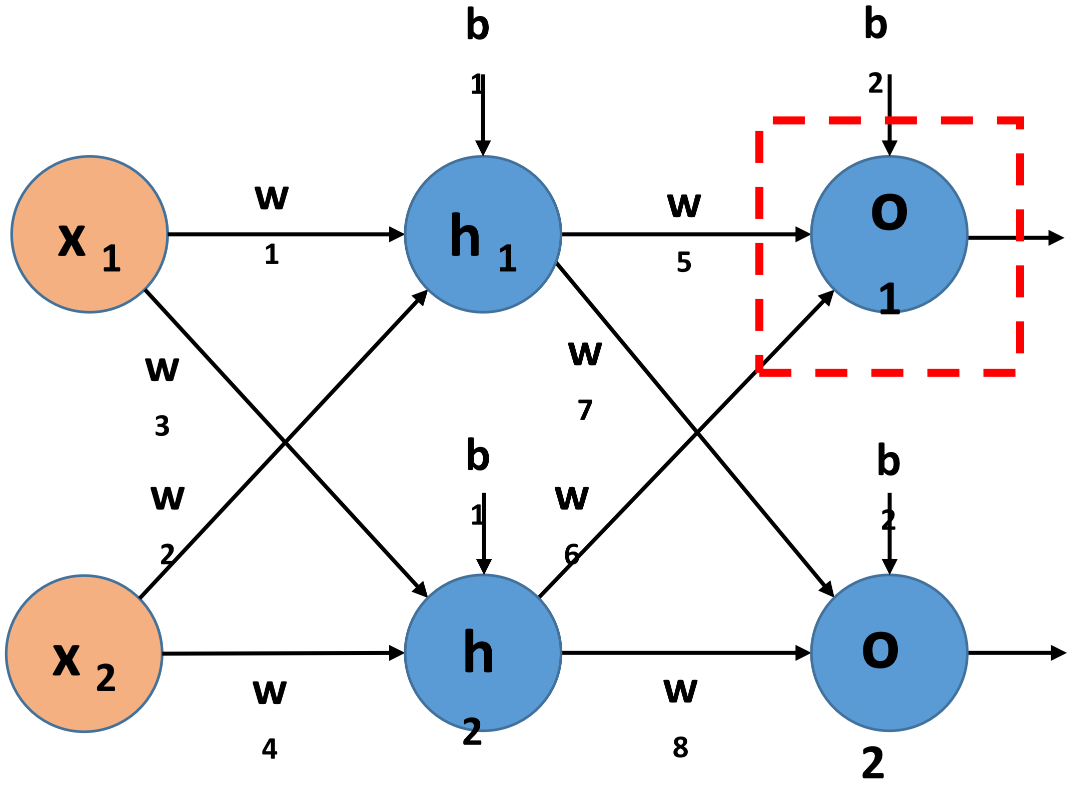


$$uo_1 = \mathrm{g}(h_1) * w_5 + \mathrm{g}(h_2) * w_6 + b_2 * 1$$

$$\hat{y}_1 = \mathrm{g}(o_1) = \mathrm{g}(uo_1) = \frac{1}{1+e^{uo_1}}$$

Feedforward and Backpropagation

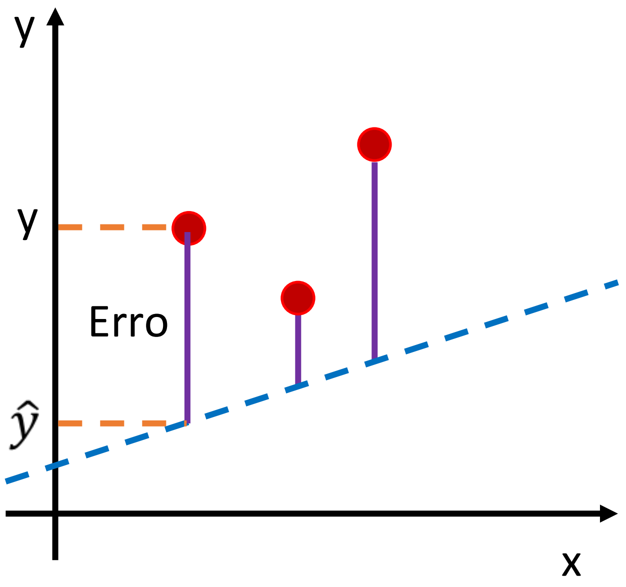
Fase 1
Propagação



$$uo_2 = \mathrm{g}(h_1) * w_7 + \mathrm{g}(h_2) * w_8 + b_2 * 1$$

$$\hat{y}_2 = \mathrm{g}(o_2) = \mathrm{g}(uo_2) = \frac{1}{1+e^{uo_2}}$$

Rede Neural Artificial: Erro/Custo



y = valor original

$\hat{y}$ = valor predito

$$\hat{y} = w_0 + w_1 x$$

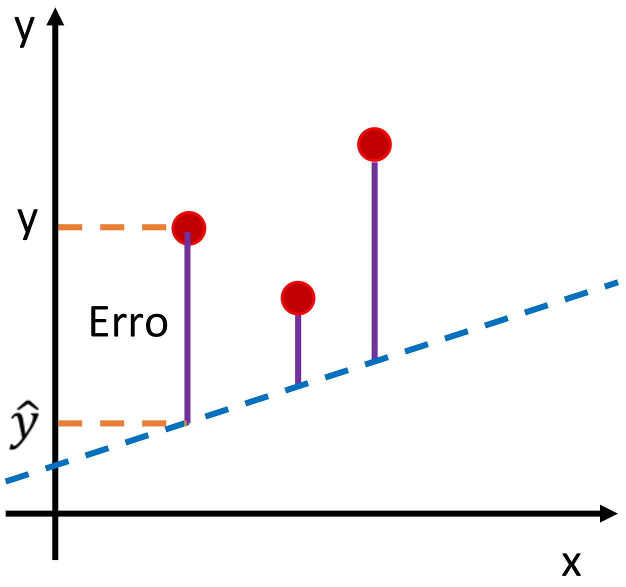
$$J(w_0, w_1) = \frac{\sum_{i=1}^{m} (\hat{y}_i - y_i)^2}{m}$$

(média)

CUST
O COST

MSE (*Mean Square Error* – Erro quadrático médio)

Rede Neural Artificial: Erro/Custo



y = valor original

$\hat{y}$ = valor predito

$$\hat{y} = w_0 + w_1 x$$

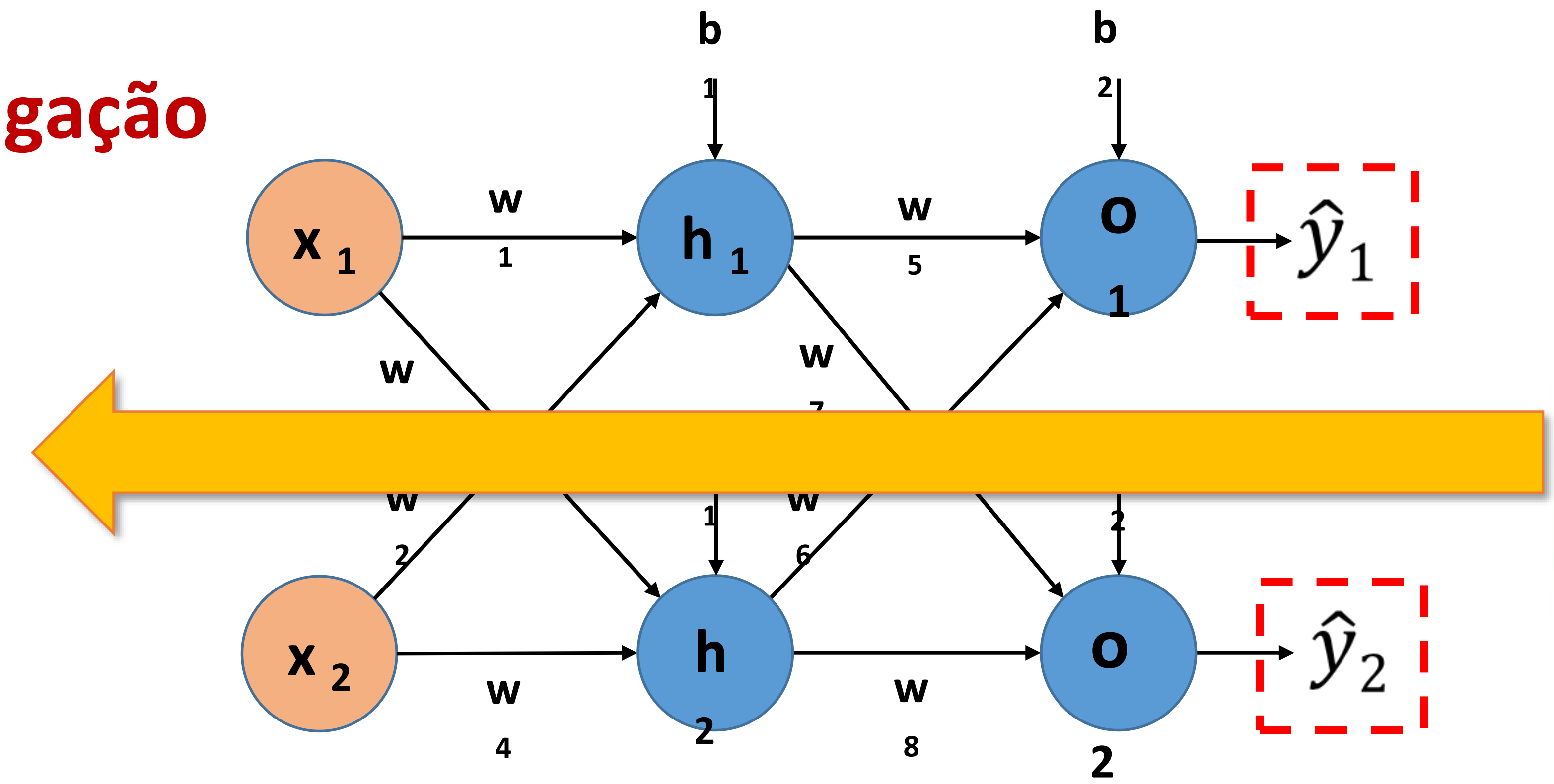
$$J(w_0, w_1) = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2$$

MSE (*Mean Square Error* – Erro quadrático médio)

Como reduzir o custo? $\min\limits_{(w_0, w_1)} J(w_0, w_1)$

Feedforward and Backpropagation

Fase 1
Retropropagação



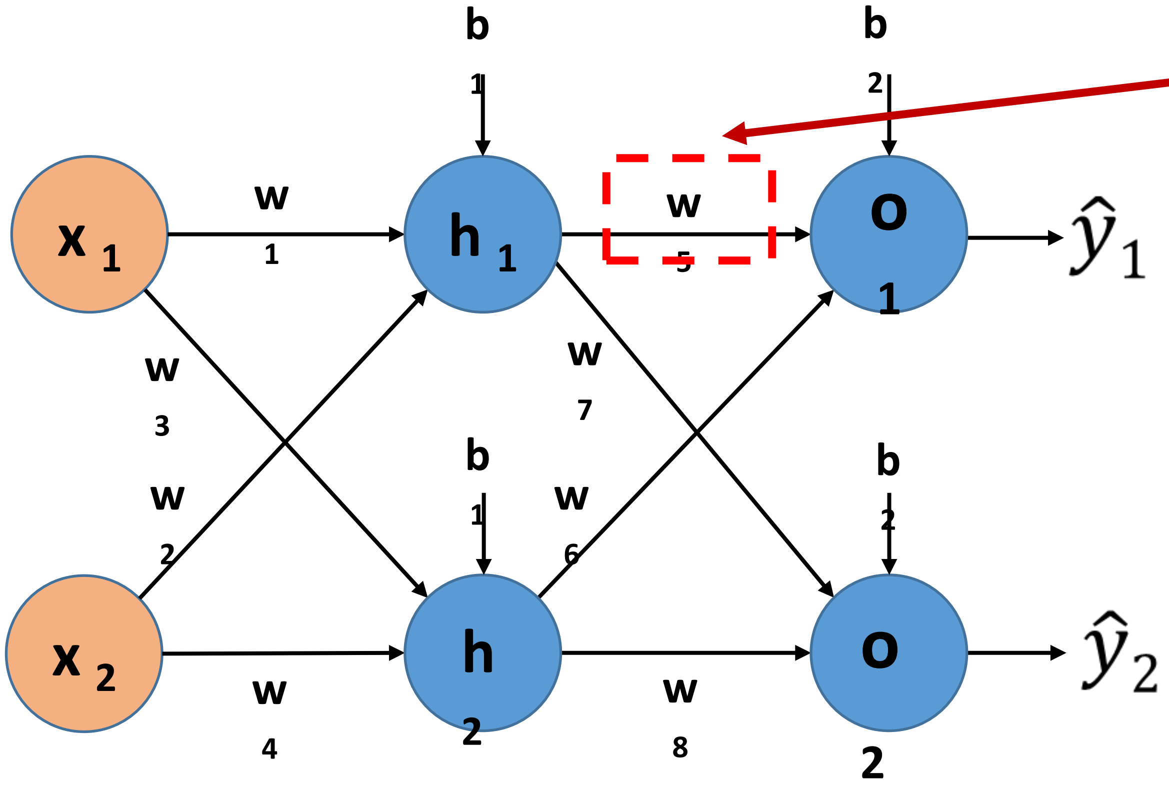
$$E_{total} = \frac{1}{2}\sum_{k=1}^{N}(\hat{y}_k - y_k)^2 = E_{o1} + E_{o2}$$

Feedforward and Backpropagation

Fase 1

Retropropagação

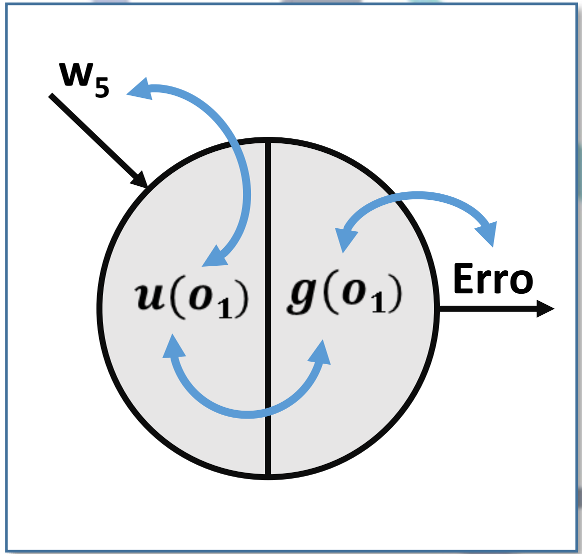
Correção de w5: Queremos estimar quanto w5 afeta o Erro total



$$\boxed{E_{total}}$$

$$\frac{\partial E_{total}}{\partial w_5} = gradiente\ em\ relação\ a\ w_5$$

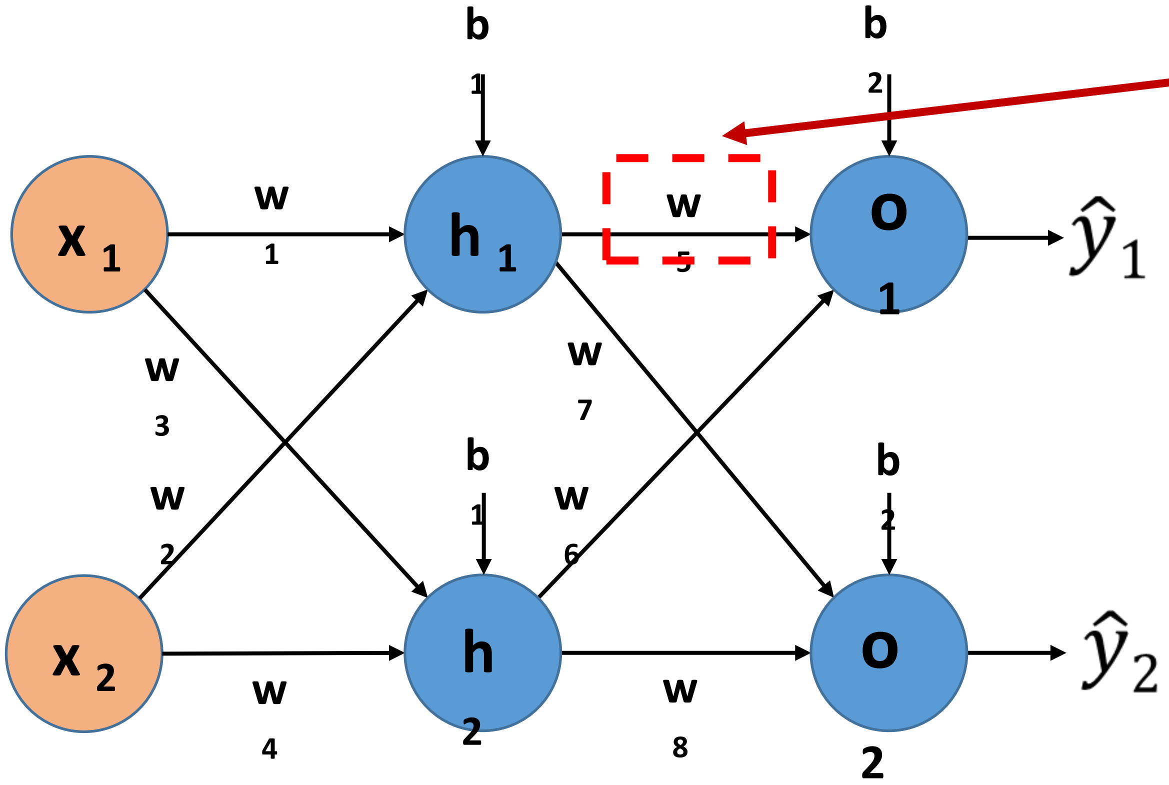
$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial go_1} * \frac{\partial go_1}{\partial uo_1} * \frac{\partial uo_1}{\partial w_5}$$



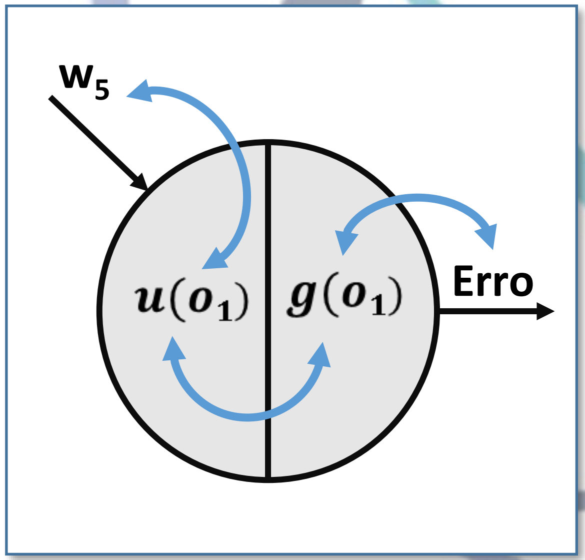
Feedforward and Backpropagation

Fase 1
Retropropagação

Correção de w5: Queremos estimar quanto w5 afeta o Erro total

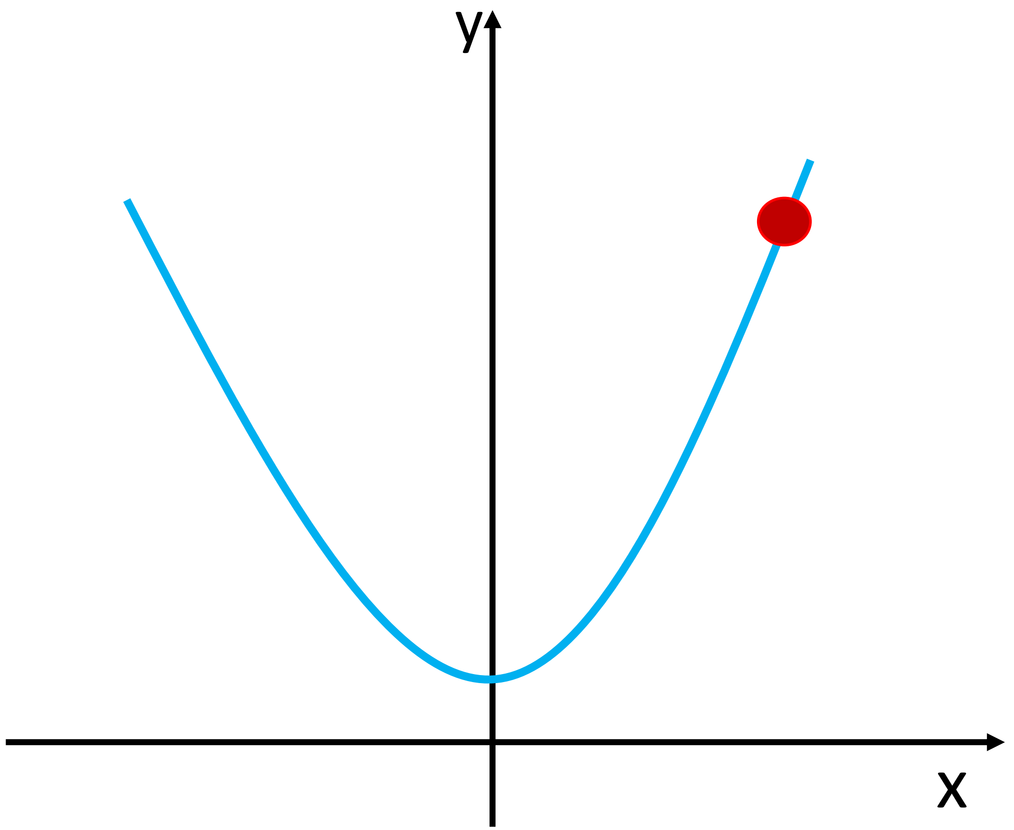


$$\boxed{E_{total}}$$



$$w_5^+ = w_5 - \eta * \frac{\partial E_{total}}{\partial w_5}:$$

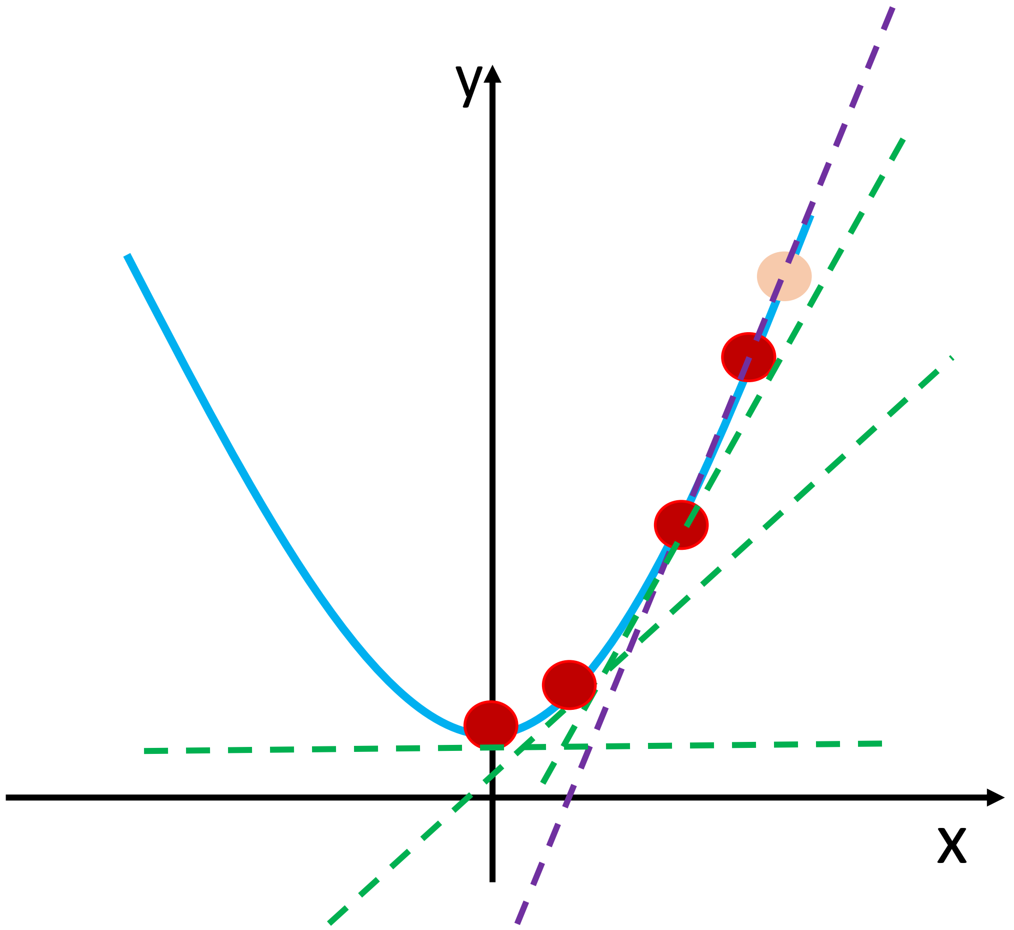
Rede Neural Artificial: Otimização



Função quadrática

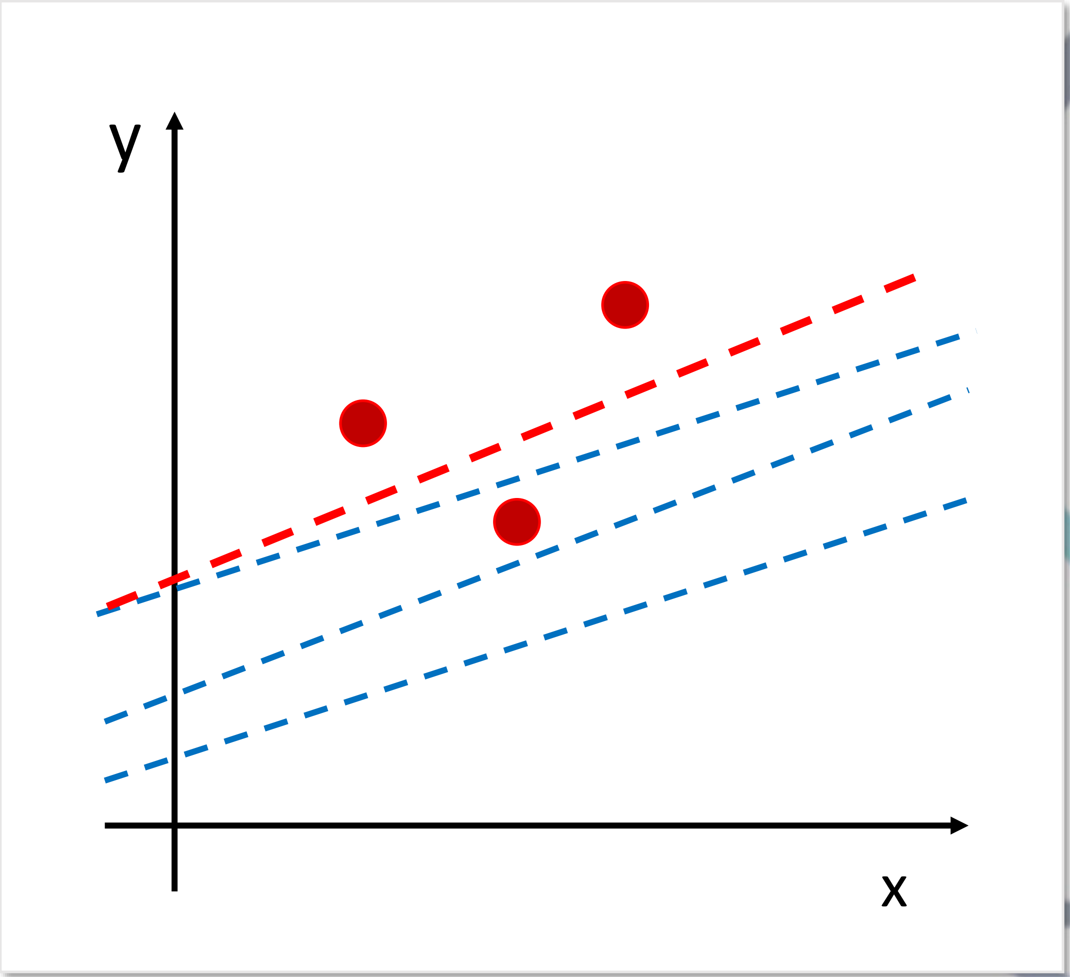
$$(\hat{y}_i - y_i)^2$$

Rede Neural Artificial: Otimização

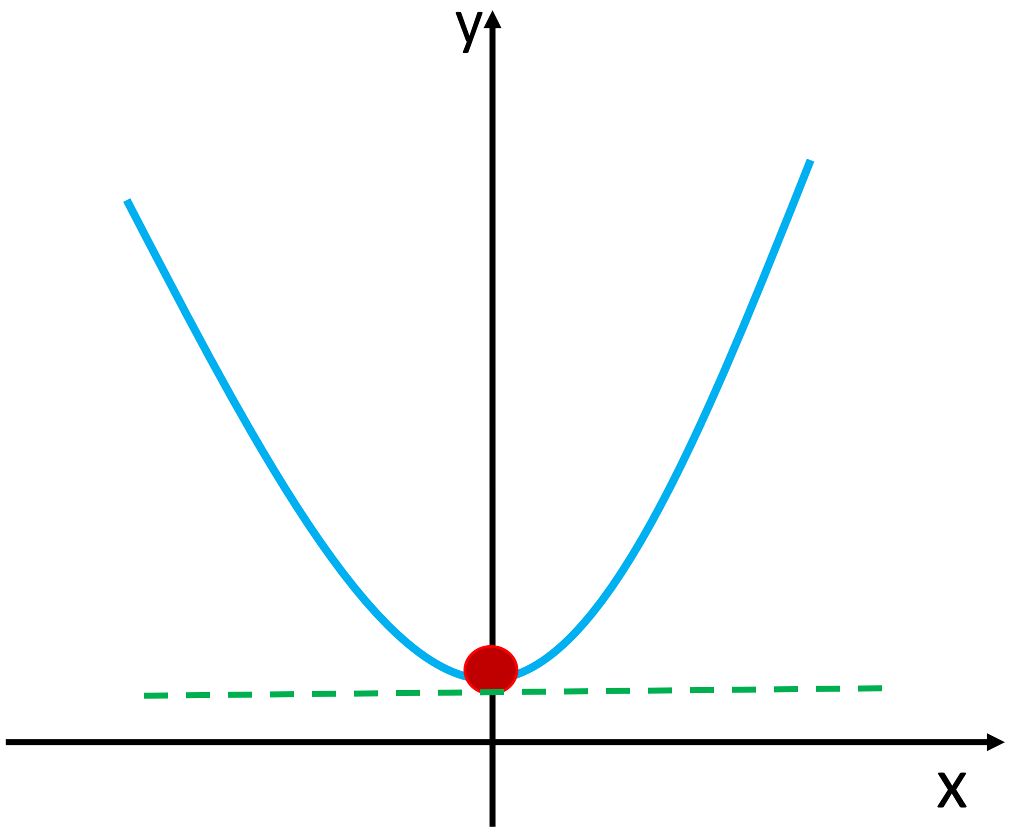


Função quadrática

$$(\hat{y}_i - y_i)^2$$

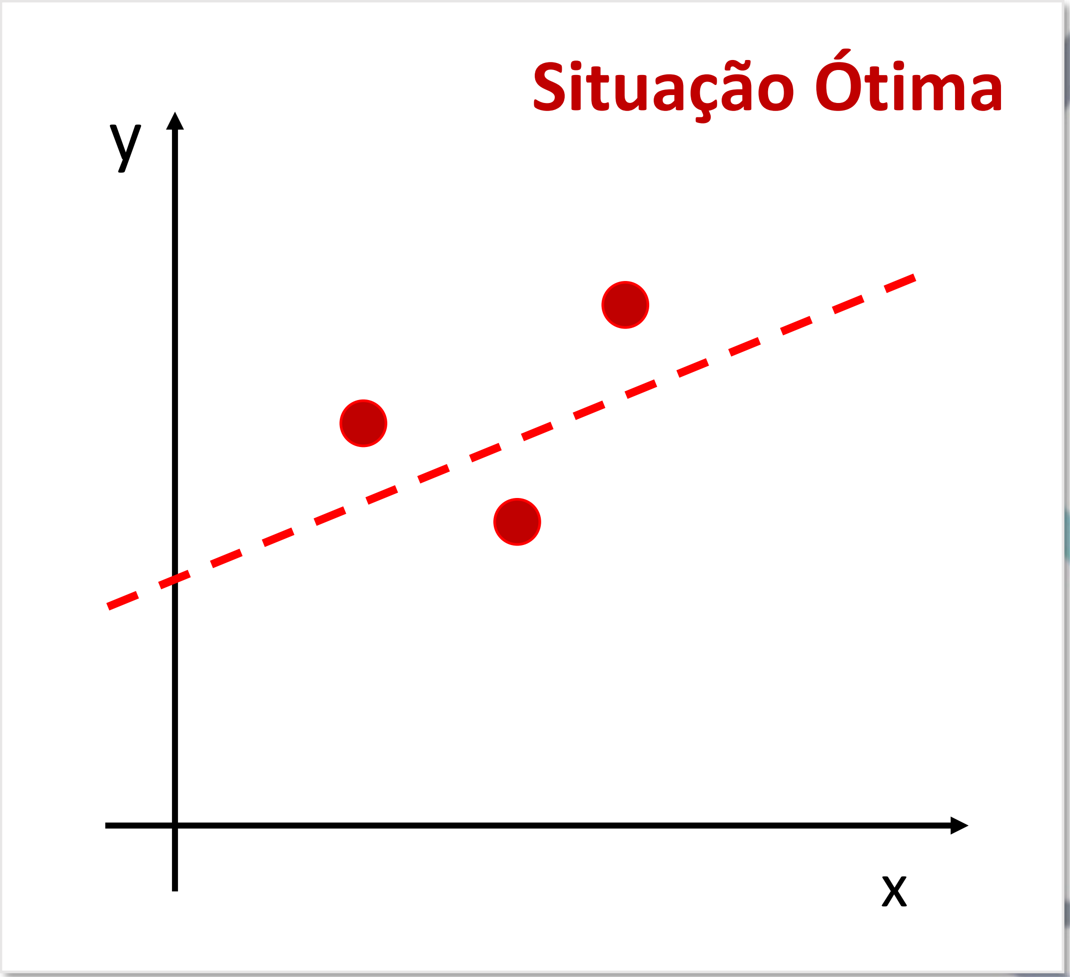


Rede Neural Artificial: Otimização



Função quadrática

$$(\hat{y}_i - y_i)^2$$



Ministério da Ciência, Tecnologia e Inovações

Centro Brasileiro de Pesquisas Físicas

Redes Neurais profundas e aplicações Deep Learning

Clécio Roque De Bom – debom@cbpf.br

CBPF

clearnightsrthebest.com