

**MBA  
USP  
ESALQ**

# Conceitos Básicos para Introdução ao Deep Learning

Prof. Dr. Jeronymo Marcondes

# Introdução

- Plano de ataque:
  - 1) Explicar o conceito
  - 2) Como os dados fluem
  - 3) Como é o processo de otimização
  - 4) Função custo e descida do gradiente
  - 5) Parte prática

# Introdução

- Problema de classificação e regressão

$$y = f(X1, X2, \dots X3)$$

- Machine Learning é uma tecnologia onde os computadores tem a capacidade de aprender de acordo com as respostas esperadas por meio associações de diferentes dados, os quais podem ser imagens, números e tudo que essa tecnologia possa identificar.

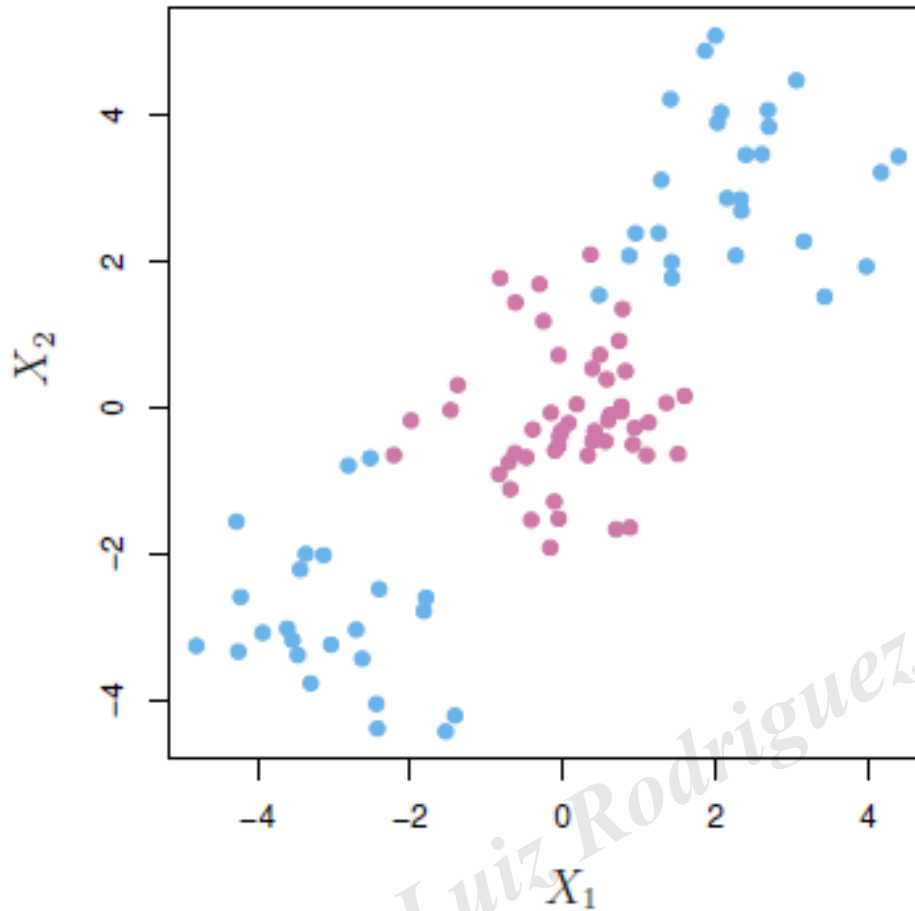
# Exemplos

É possível fazer modelos para:

classificar clientes  
por chance de  
não pagar;

prever o valor de  
uma ação;

reduzir um  
conjunto de  
dados.



Morettin e Singer – Introdução à Ciência de Dados

# AE Supervisionado

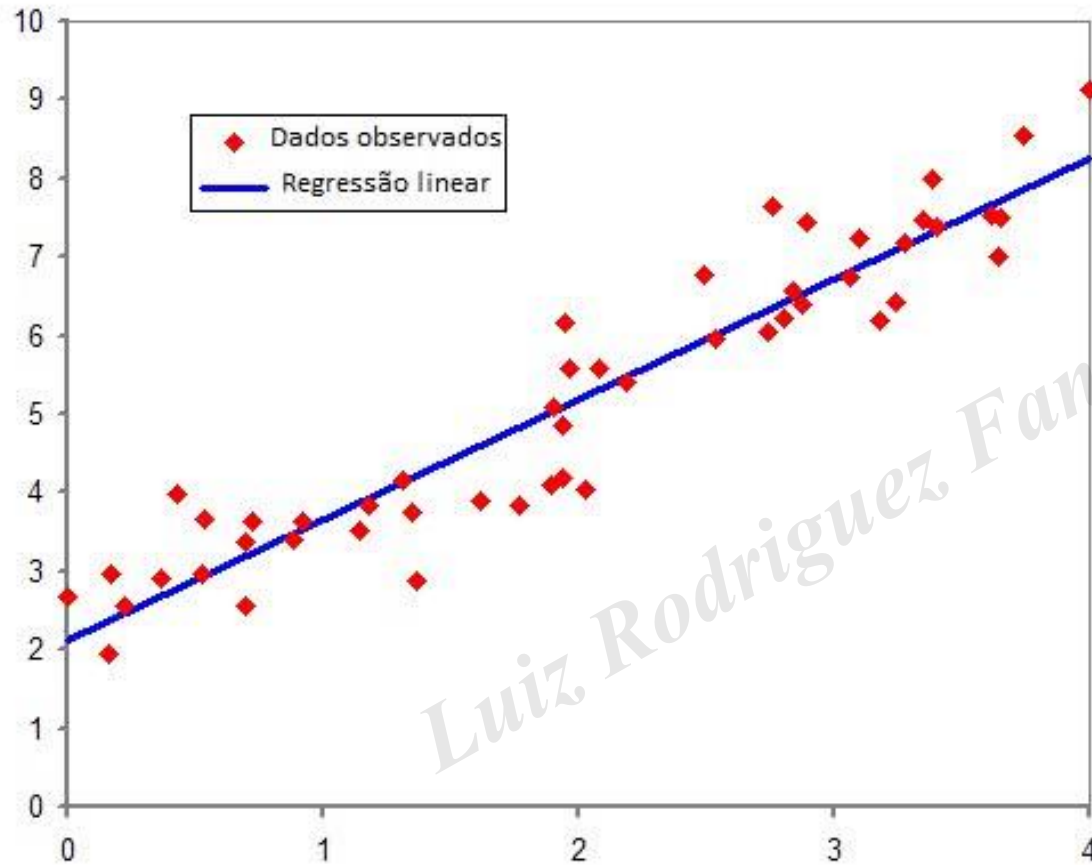
- Problema supervisionado
- Regressão:

$$y = f(X) + e$$

$$y = a + bX + e$$

$$y = 0,2 + 0,1X$$

# Problema Supervisionado



Luiz Rodriguez Fantini 005.374.619-81

# AE Não Supervisionado

- Temos apenas um conjunto de variáveis preditoras (inputs) e o objetivo é descrever associações e padrões entre essas variáveis. Nesse caso, não há uma variável resposta.
- Análise de Cluster e Análise de Componentes Principais.



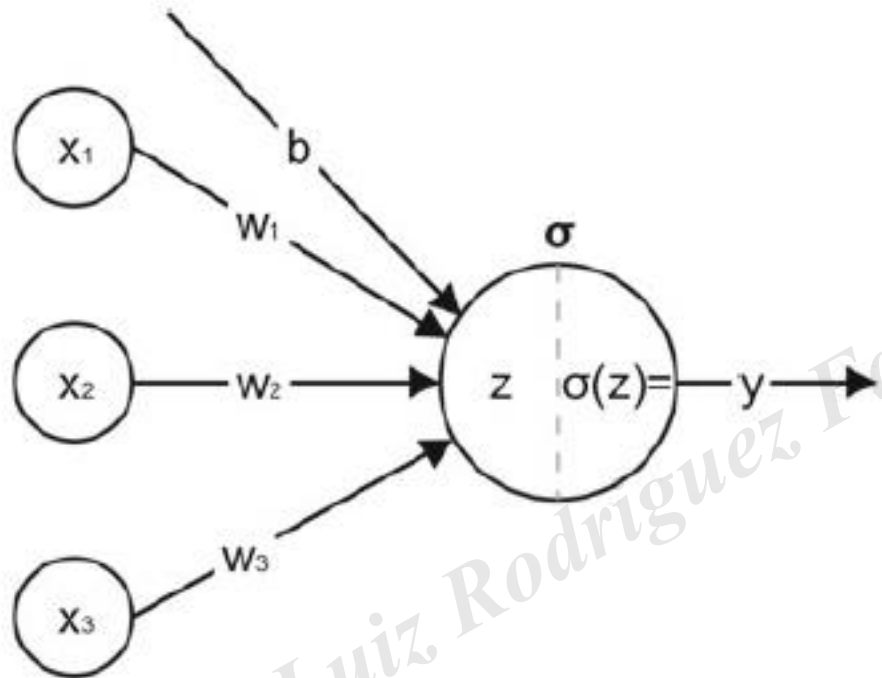
# Classificação

- Resposta do modelo é uma variável qualitativa.
- Exemplo do risco de default.

$$\textit{Default} = f(\textit{Risco do Indivíduo})$$

Luiz Rodrigues Fantini 005.374.619-81

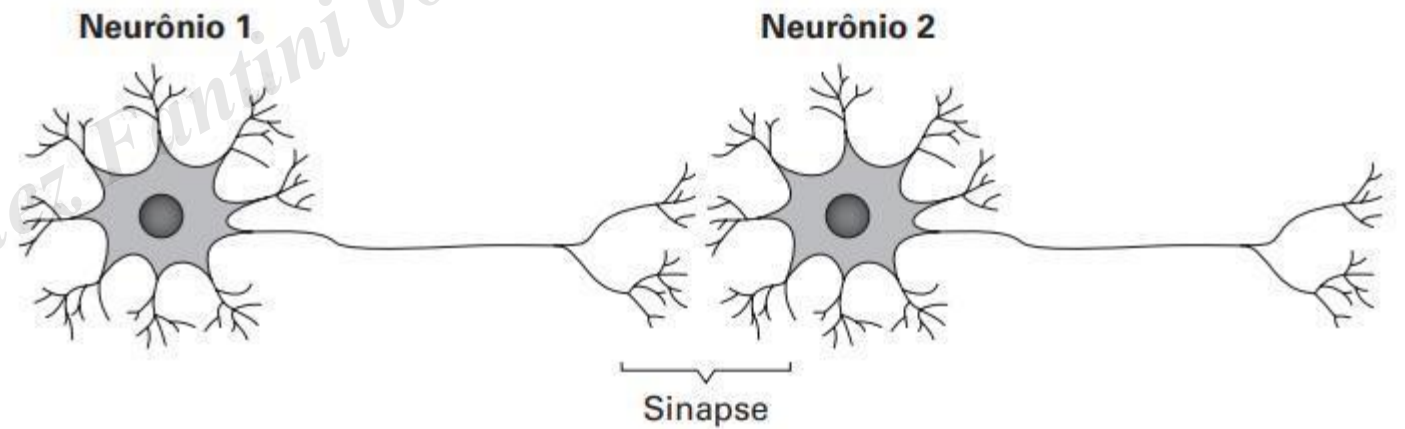
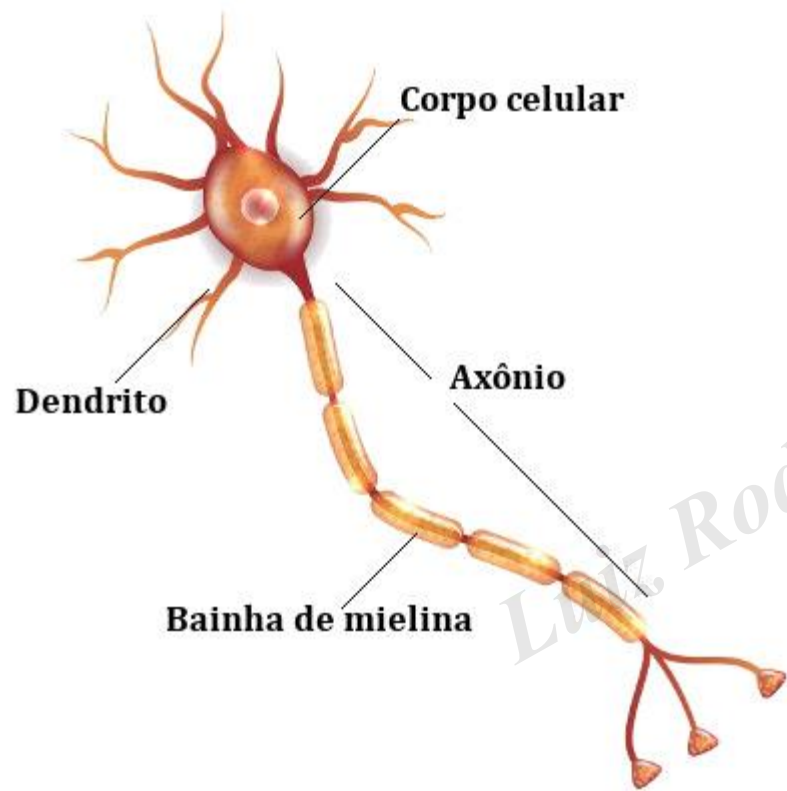
# Regressão como ANN



# Redes Neurais Artificiais

- Deep learning – conceito
- Nessa aula começaremos com a rede neural “rasa”.
- As contribuições pioneiras para a área de Redes Neurais (RN) foram as de McCulloch e Pitts (1943), que introduziram a ideia de RN como máquinas computacionais e de Hebb (1949), por postular a primeira regra para aprendizado organizado.

# Redes Neurais Artificiais

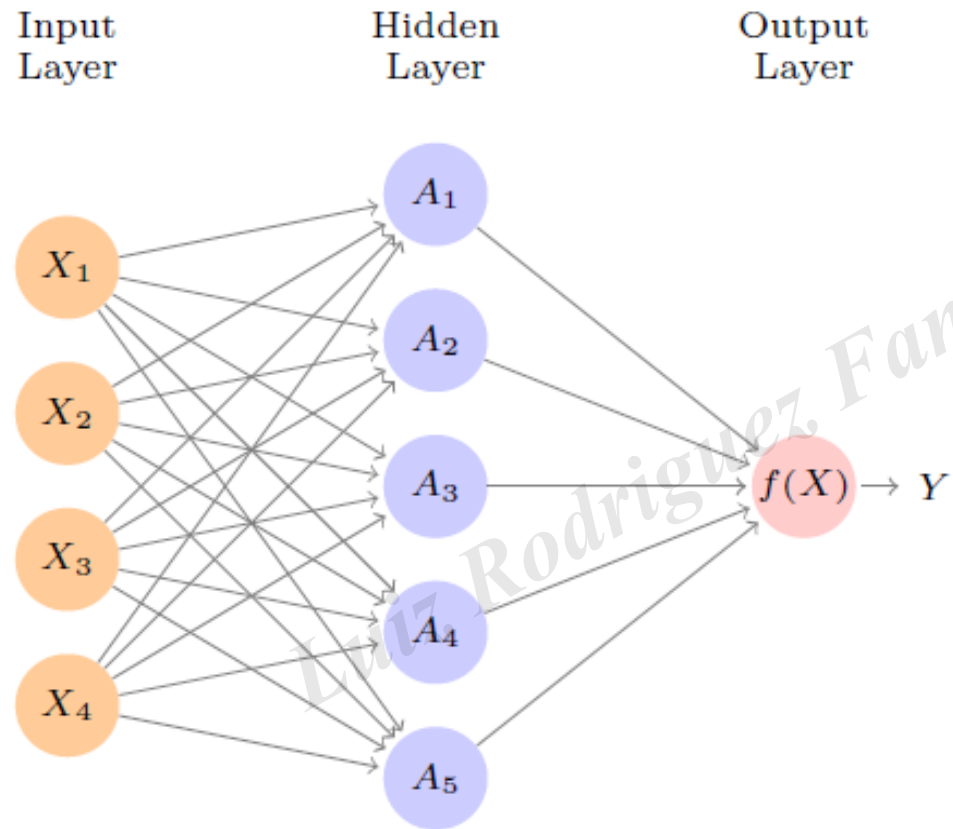


# Redes Neurais Artificiais

- A ideia do perceptron de Rosenblatt.
- Aprende respostas binárias
- Pesos treinados para produzir vetor alvo

Shiffman (2012) demonstra o Perceptron como um modelo computacional de um simples neurônio, o qual consiste em possuir diversas entradas de dados, um núcleo de processamento e uma saída, o qual o possibilita apenas a saída de valores lógicos.

# Redes Neurais Artificiais



# Redes Neurais Artificiais

- Rede neural do tipo feedforward.
- Neurônios não se interconectam.
- Camadas de entrada não pode ser menor do que a quantidade de variáveis que estamos usando para explicar Y.

Luiz Roberto Fantini 005.374.619-81

# Como funciona

- Cada Neurônio recebe dados de entrada ou da camada anterior.
- Diferente do caso de regressão — só tinha uma camada de entrada e saída.
- Cada Neurônio ao enviar informação para o próximo multiplica a informação por **pesos**.



# Matemática

- Dado o valor de  $n$  entradas:

$$X_1 = [X_{11}, X_{12} \dots X_{1n}]$$

Essas são nossas variáveis explicativas! Por exemplo, o risco de default do cliente:

$$X = [\textit{Pagou no passado?}, \textit{Renda Atual}, \textit{etc}]$$

# Matemática

- Dado o valor de saídas para  $t$  clientes:

$$Y = [Y_1, Y_2, \dots, Y_t]$$

Essas são nossas variáveis explicadas. Ou seja, se um determinado cliente efetivamente pagou.

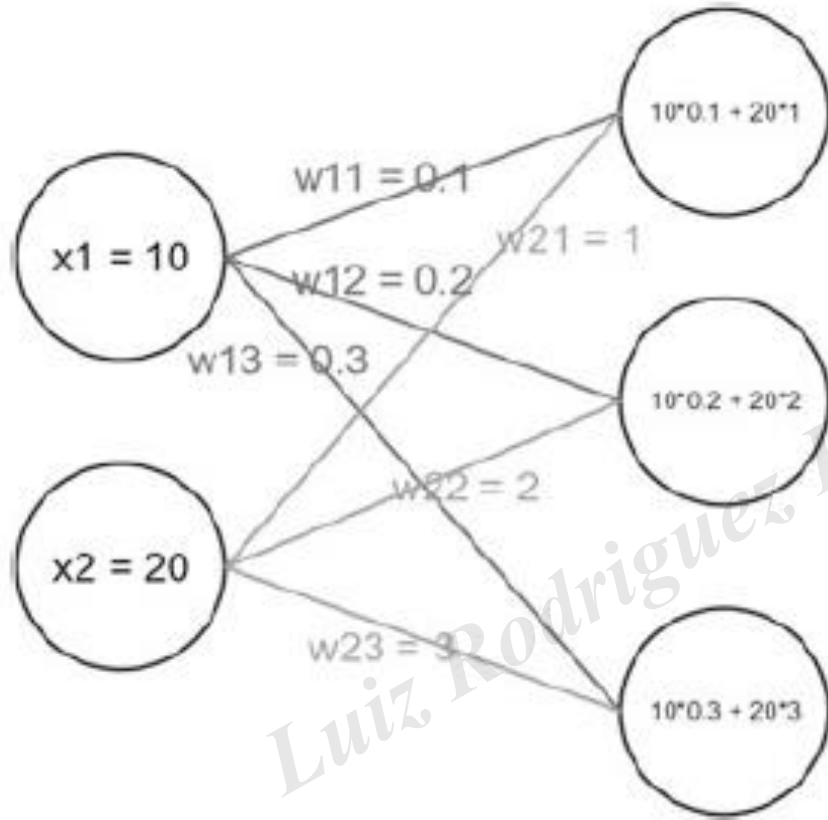
# Matemática

- Dado o valor de pesos para  $h$  neurônios da próxima camada:

$$w = [w_{11}, w_{12}, \dots, w_{nh}]$$

Esses são os valores que serão multiplicados por cada variável de entrada até a próxima camada escondida.

# Visualização



# Matemática

- Para cada uma dos neurônios:

$$z = b + \sum_i w_i X_i$$

O que é equivalente, no caso do primeiro neurônio, a:

$$z_1 = b + w_{11}X_1 + w_{21}X_2$$

# Função de ativação

- Suponha que estejamos em problema de classificação de default com resultados possíveis 1 ou 0.
- 1 vai dar calote e 0 não.
- Nossa variável explicada é dicotômica e trata-se de um problema de classificação.

# Função de ativação

A função de ativação é aquela que processa o sinal gerado pela combinação linear das entradas e dos pesos das sinapses, para gerar o sinal de saída do neurônio.

Ou seja, ela que faz o “processamento” da informação.

Podemos ter respostas lineares e não lineares.

# Função de ativação

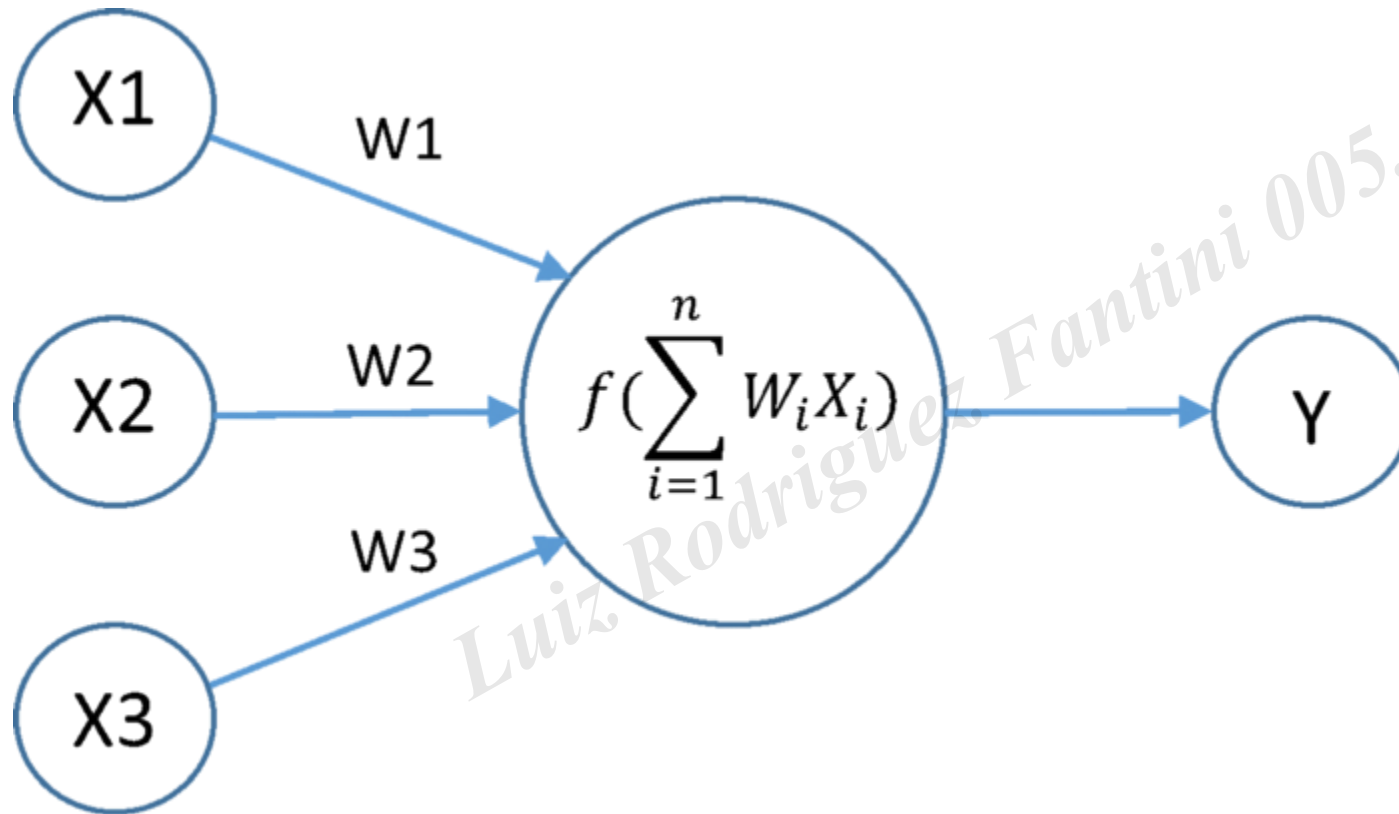
- Para decidir o resultado advindo do neurônio anterior temos (ReLU):

$$f(z) = \begin{cases} 1 & \text{se } z \geq 0 \\ 0 & \text{caso contrário} \end{cases}$$

Qual a diferença para um regressão?



# Função de ativação



# Função de ativação

$$z_1 = w_{11}X_1 + w_{21}X_2 = 1.2$$

$$\text{Relu}(z_1) = 1$$

Comprando com a tabela:

$$y - \text{Relu}(z_1)$$

O que fazer se deu errado??

# Função de ativação

- Um outro exemplo:
- Função Sigmoid:

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

E se fosse uma função linear?

# Como funciona

- Peso a ser multiplicada a informação depende do par origem\destino.
- Resultado é inserido dentro de uma função de ativação.
- A depender do resultado, tal como em neurônio, o sinal é disparado para o próximo neurônio.

# Como funciona

- Como determinar os pesos?
- Iniciado de forma randômica ou com alguma regra pré-especificada.
- Caso do chute de valores das compras.

# Explicação

- Você sabe a quantidade que comprou.
- Você não sabe o preço por kg.
- Você sabe o total que gastou.

$$gasto = ppk_{uva} q_{uva} + ppk_{maça} q_{maça}$$

# Explicação

- Suponha que você gastou 10.
- Recebeu 1 kg de cada.
- Você tinha chutado que o ppk da uva é de R\$ 4.00 e o da maçã é de R\$ R\$ 5.00. Você chuta antes de saber o valor total.
- Você errou por  $10 - 9 = \text{R\$}1.00$ . Erro residual!

# Explicação

- Suponha que você gastou 10.
- Recebeu 1 kg de cada.
- Você tinha chutado que o ppk da uva é de R\$ 4.00 e o da maçã é de R\$ R\$ 5.00. Você chuta antes de saber o valor total.
- Você errou por  $10 - 9 = \text{R\$}1.00$ . Erro residual!



# Ajuste dos Pesos

- $\text{erro residual}_j = \text{valor real}_j - \text{valor estimado}_j$
- $\Delta w_j = \eta \nabla E$

Luiz Rodriguez Fantini 005.374.619-81

# Função Custo

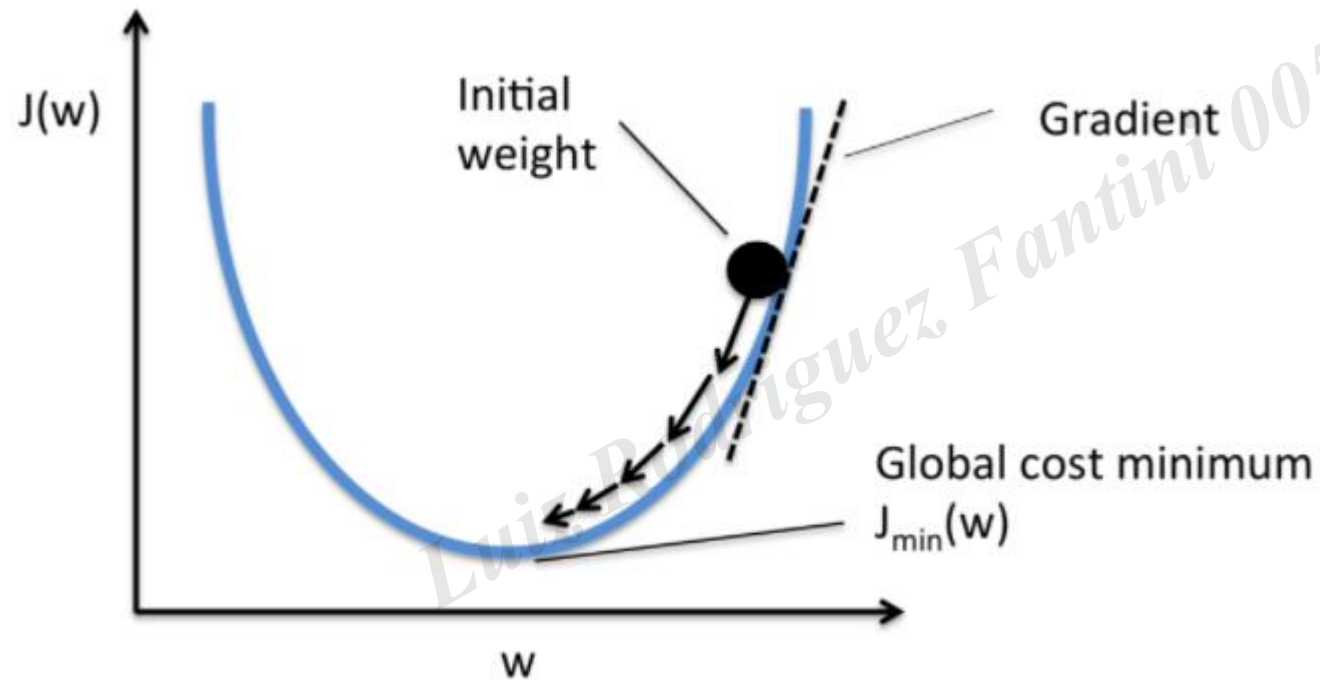
- Função que mensura o quanto estamos acertando.
- Por exemplo: erro quadrático médio.

$$E = \frac{1}{2} \sum_{\text{treinamento}} (t^n - y^n)^2$$

# Função Custo

	valor estimado	valor real	Erro	Quadrado do erro
preço 1	2	2	0	0
preço 2	3	5	-2	4
preço 3	2	4	-2	4
preço 4	5	1	4	16
Soma				24
				6

# Como é feito o ajuste?



Fonte: Deep Learning book

# Como é feito o ajuste?

- Chute um valor inicial para o peso
- Calcule os valores de saída para esses pesos
- Use a fórmula de ajuste para obter os novos pesos:

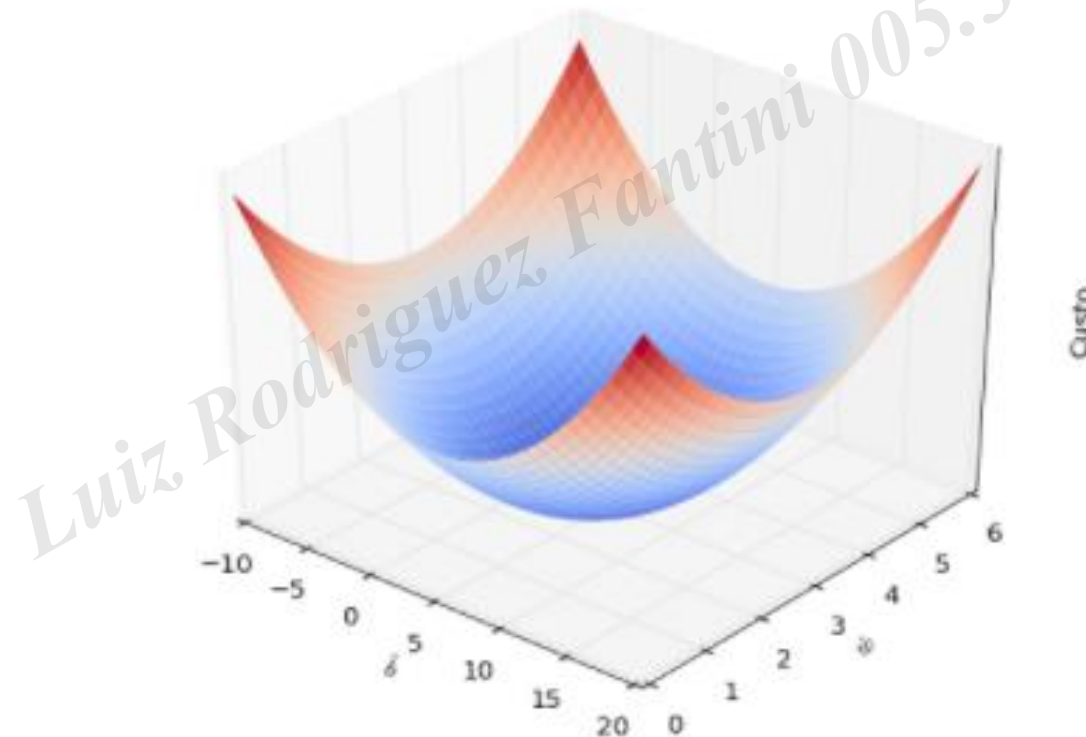
$$\text{ novo peso } = \text{ peso anterior } - \eta \nabla E$$

# Descida do gradiente

- Processo utilizado para encontrar o mínimo da função custo com base em variação nos pesos.
- Através do processo de backpropagation os erros obtidos após os cálculos dos pesos são retroalimentados na rede até otimizarmos o resultado.

# Descida do gradiente

- Matematicamente: qual a variação nos pesos que mais reduz minha diferença entre o valor estimado e o valor real?



Fonte: <https://matheusfacure.github.io/2017/02/20/MQO-Gradiente-Descendente/>

# Descida do gradiente

- Uso do cálculo diferencial:

O gradiente vai nos dar as variações na função perda para pequenas alterações nos parâmetros:

$$\nabla(L) = \left[ \frac{\partial L}{\partial w}, \frac{\partial L}{\partial b} \right]$$

Luiz Rodriguez Fantini 005.374.619-81



# Parte Prática

Luiz Rodriguez Fantini 051 374.619-81