
Tera

Aula 30: Introdução a Deep Learning

Instrutora: Patrícia Pampanelli

Experiência:

- ➔ Cientista da Computação (UFJF)
- ➔ Mestre em Modelagem Computacional (UFJF)
- ➔ Doutora em Processamento de Imagens e Visão Computacional (PUC-Rio)
- ➔ Sênior Data Scientist - Grupo ZAP

Interesses:

Carros Autônomos

Visão Computacional

Processamento de Linguagem Natural

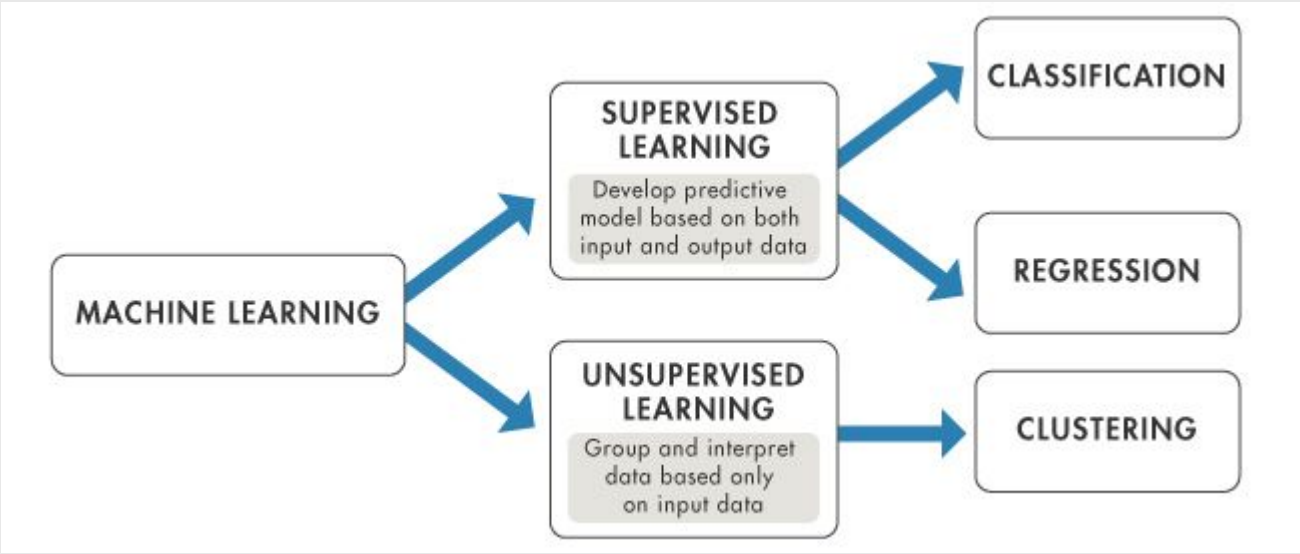
Intelligent Video Analytics

Recapitulando...

- Manipulação, visualização e tratamento dos dados

Recapitulando...

- Algoritmos supervisionados e não-supervisionados



Inteligência Artificial mudando nossas vidas...

Inteligência Artificial nos dias de hoje

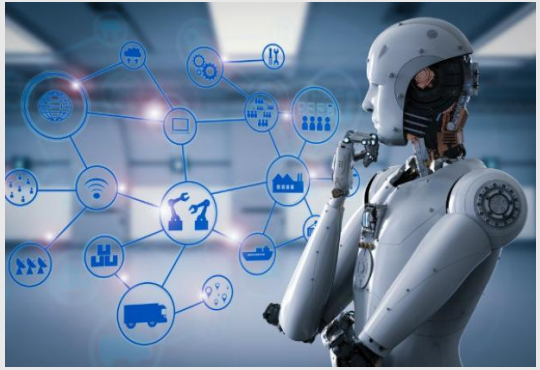
Tay: Microsoft issues apology over racist chatbot fiasco

Dave Lee
North America technology reporter

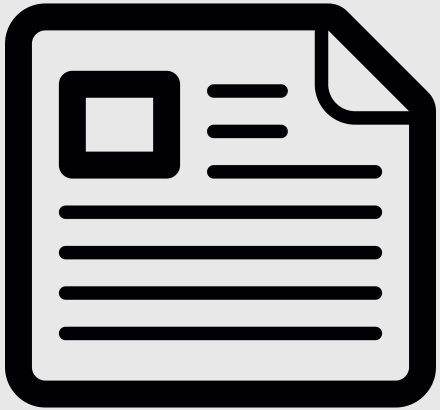
25 March 2016 | Share

Inteligência Artificial nos dias de hoje

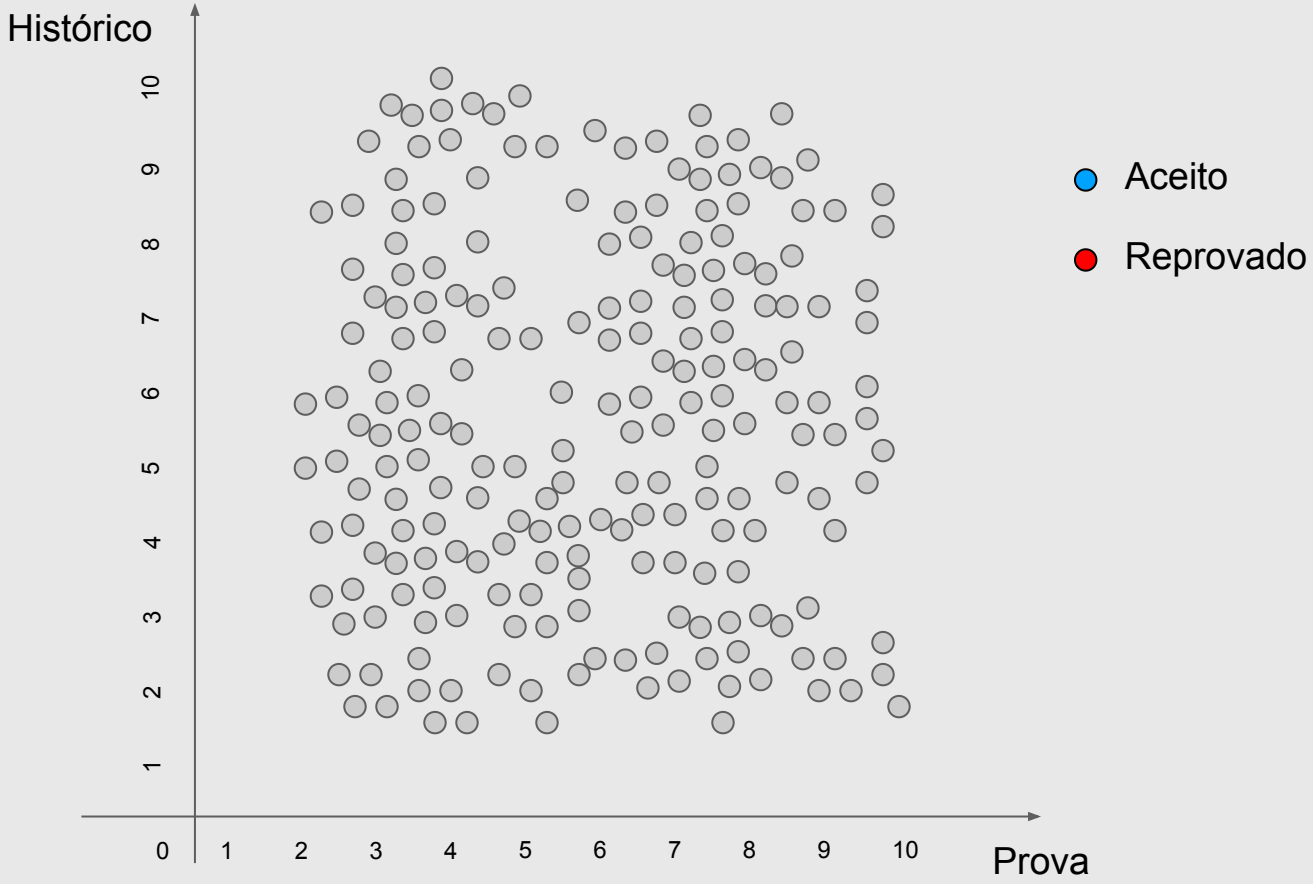
- Implicações do uso dos métodos de Machine Learning
- Google Principles
  - Be socially beneficial
  - Avoid creating or reinforcing unfair bias
  - Be built and tested for safety
  - Be accountable to people
  - Incorporate privacy design principles
  - Uphold high standards of scientific excellence
  - Be made available for uses that accord with these principles



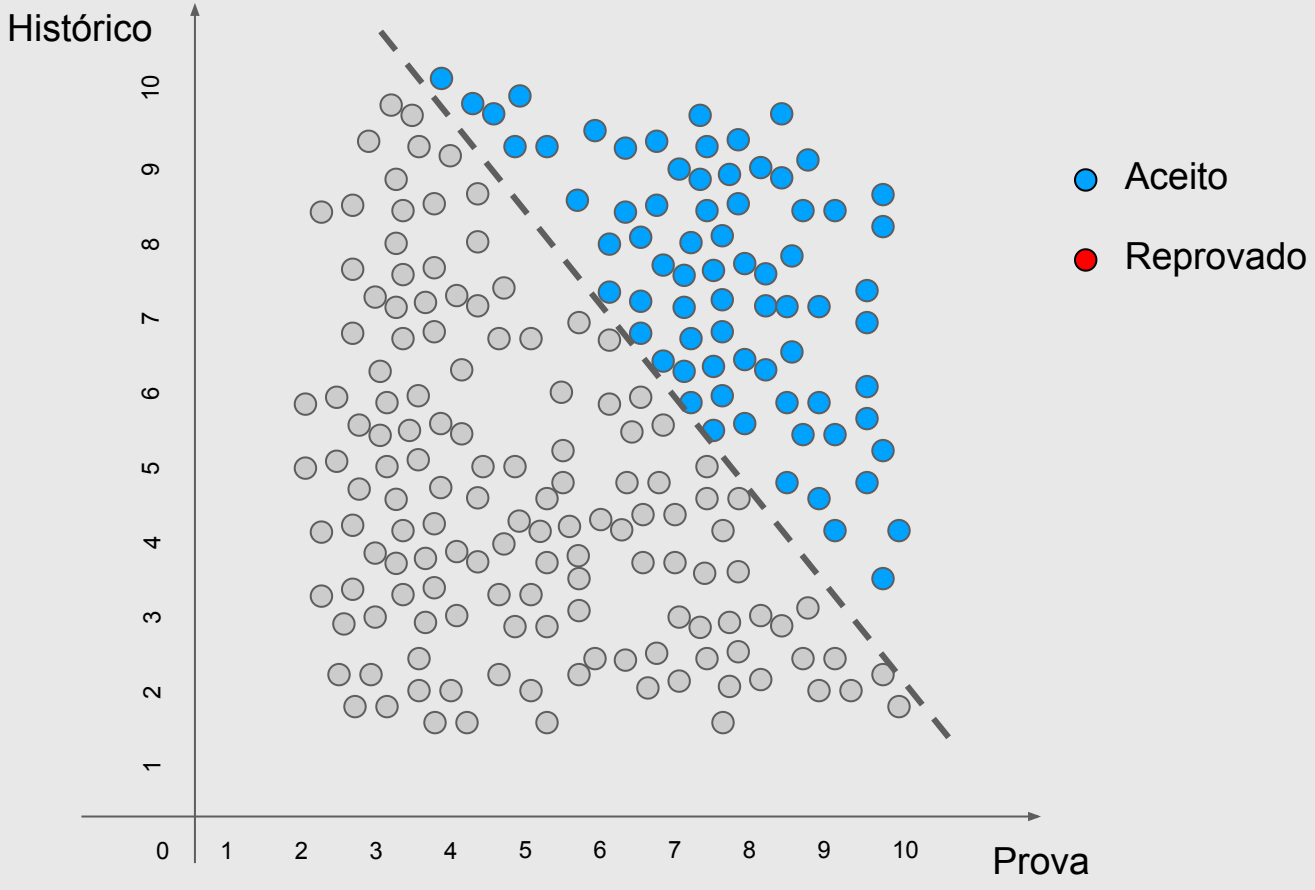
Processo de admissão de uma pós-graduação



Processo de admissão de uma pós-graduação

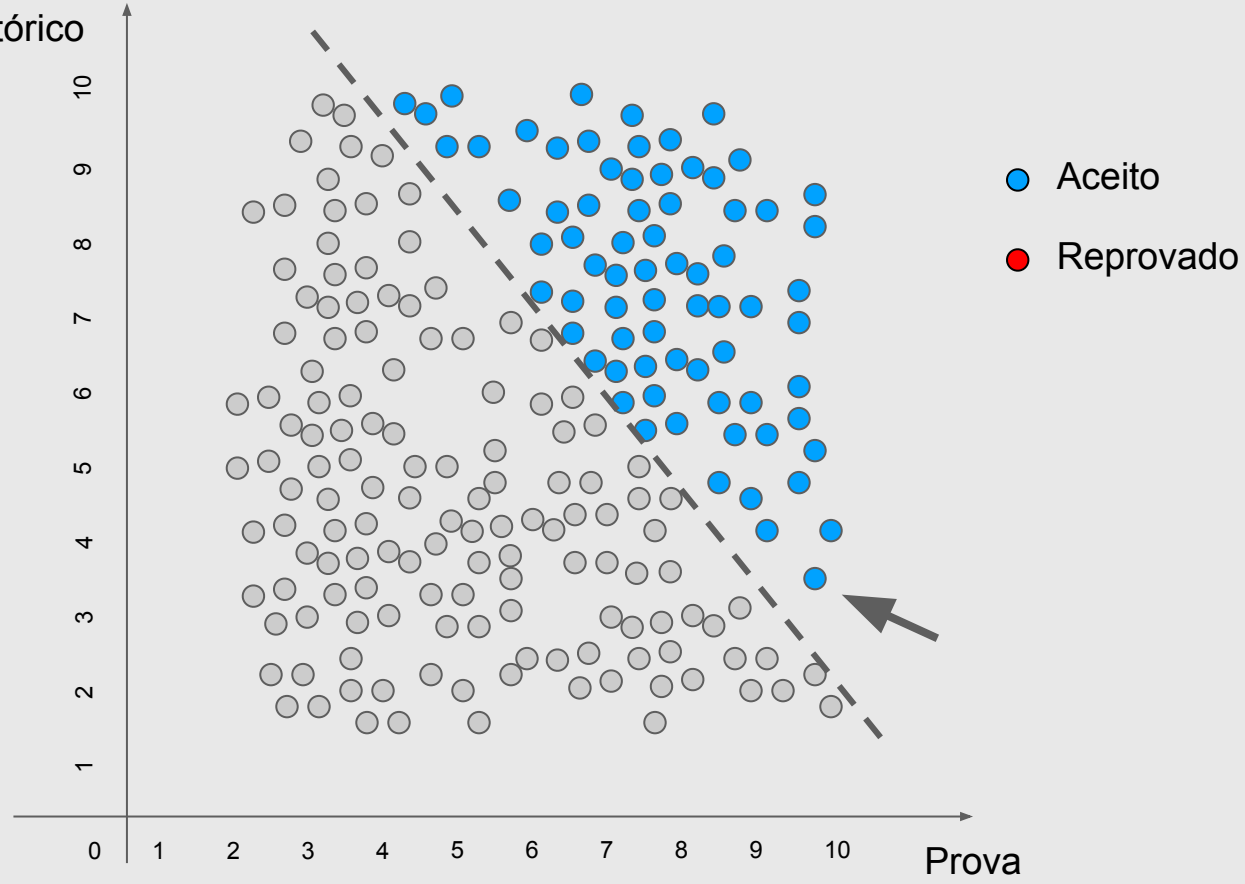


Processo de admissão de uma pós-graduação

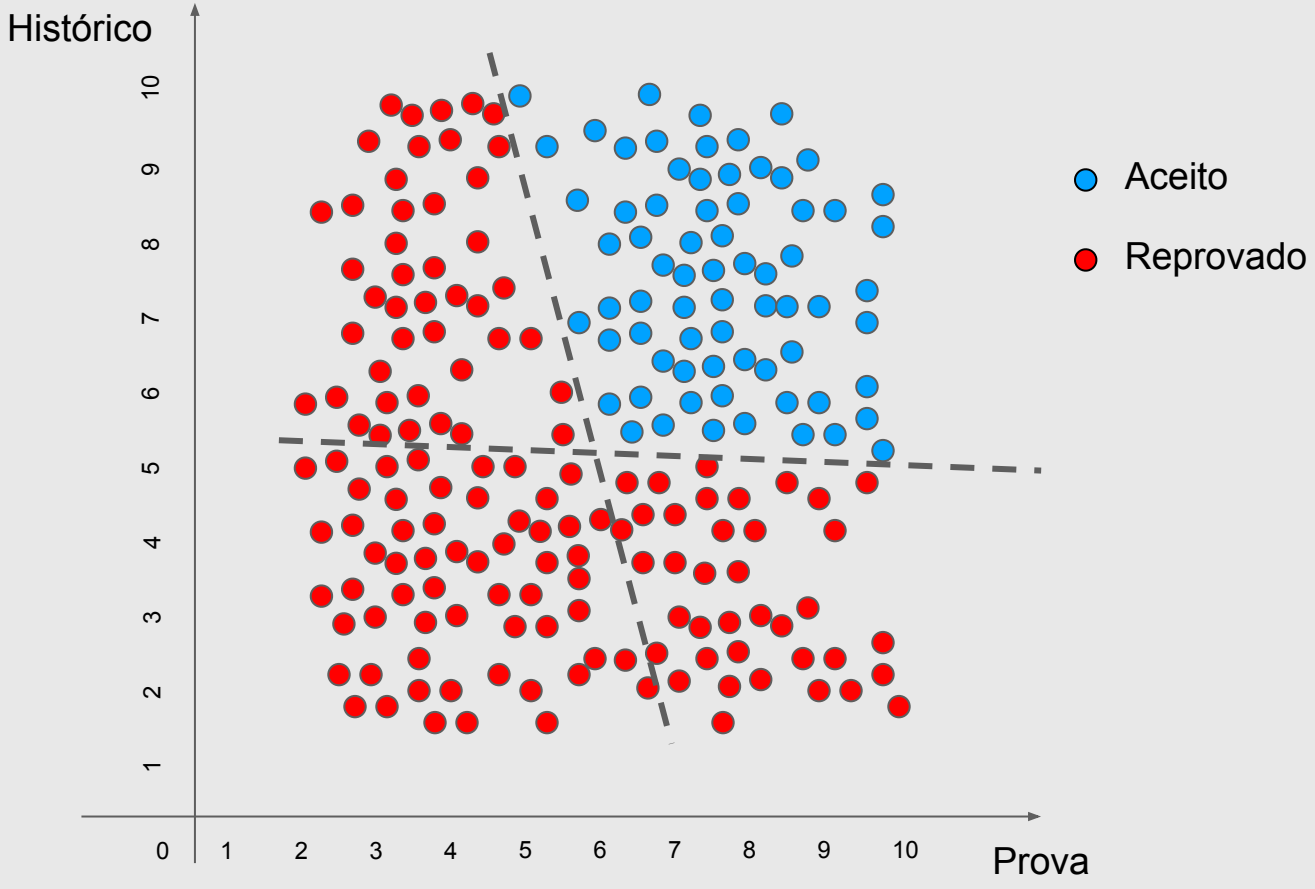


Processo de admissão de uma pós-graduação

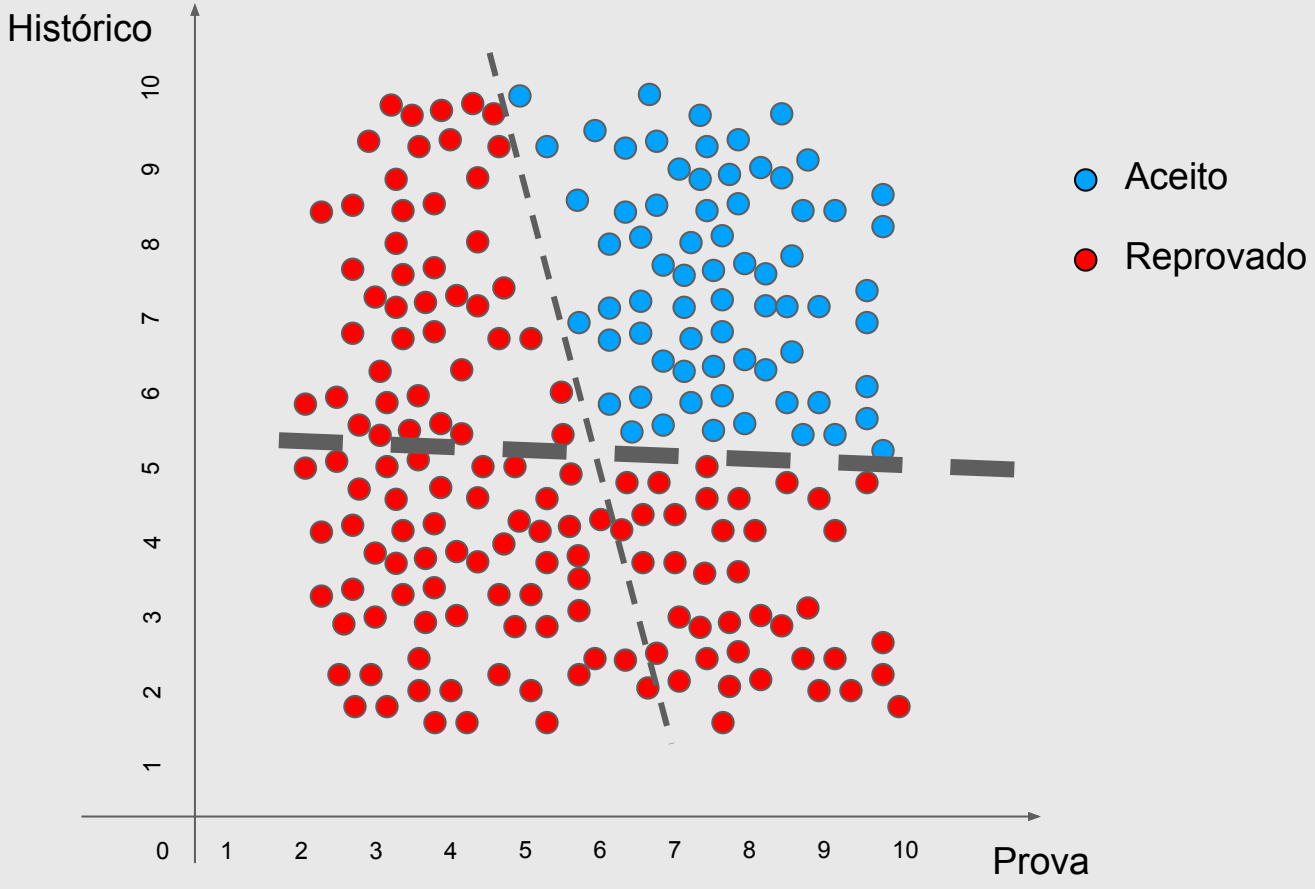
Estes dados são linearmente separáveis?



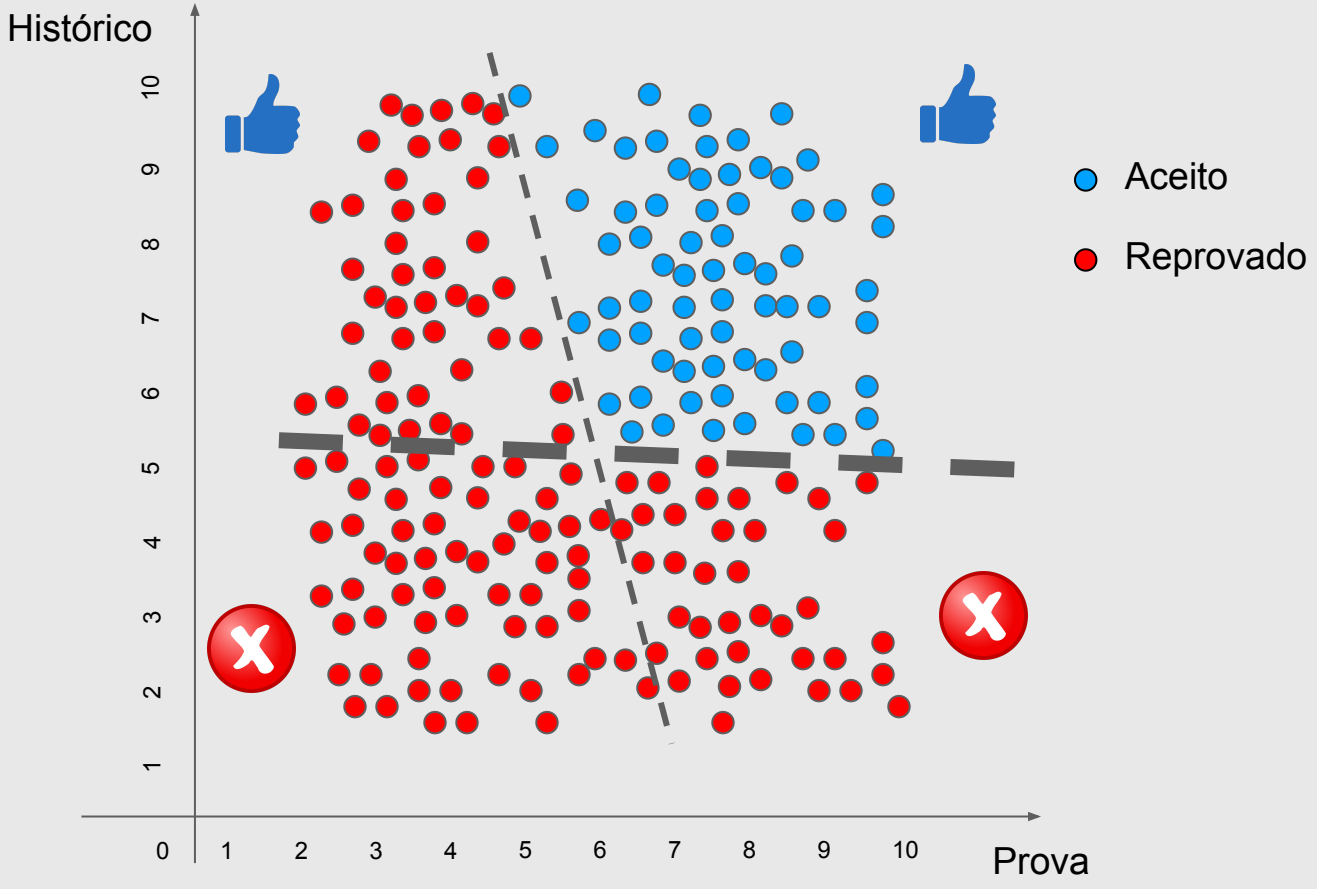
Processo de admissão de uma pós-graduação



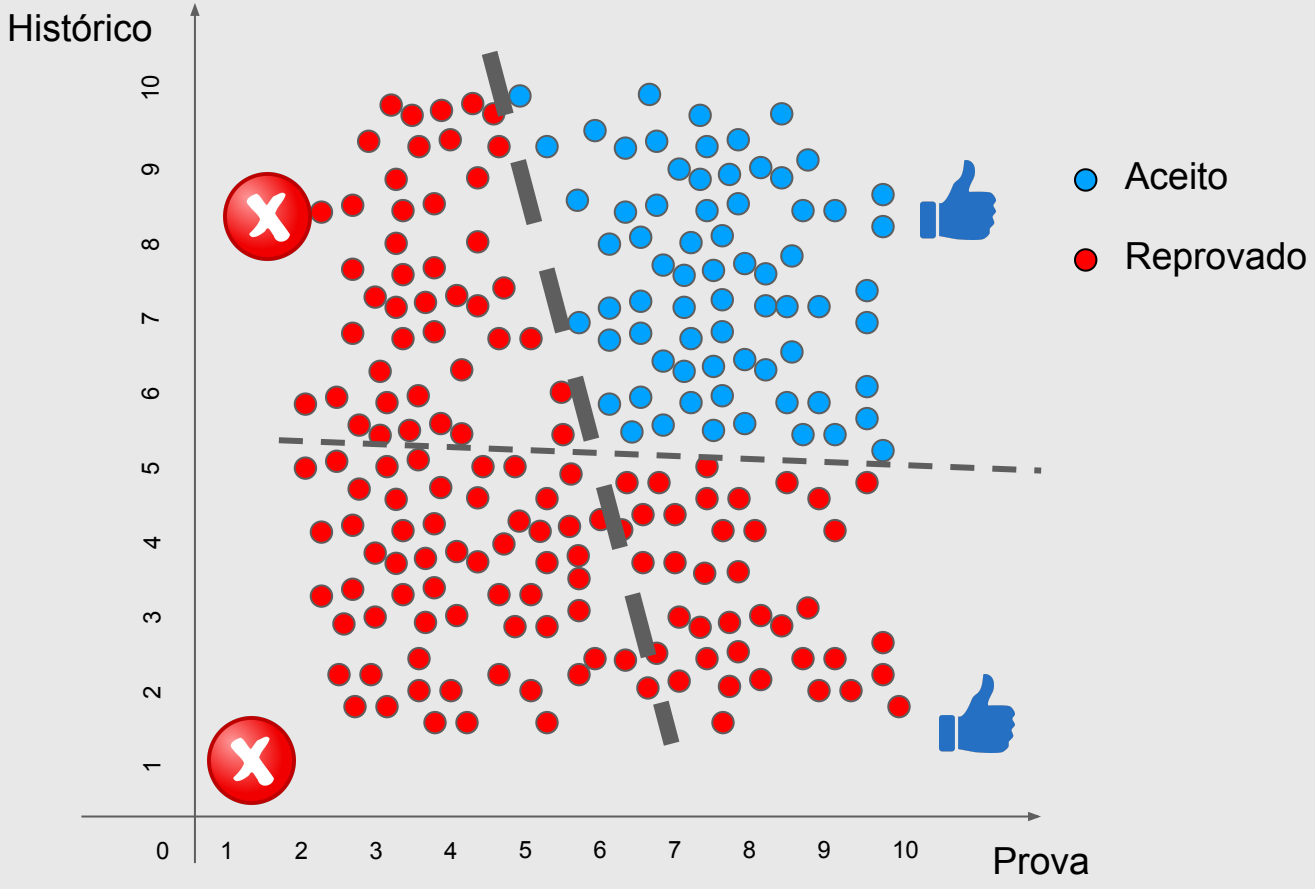
Processo de admissão de uma pós-graduação



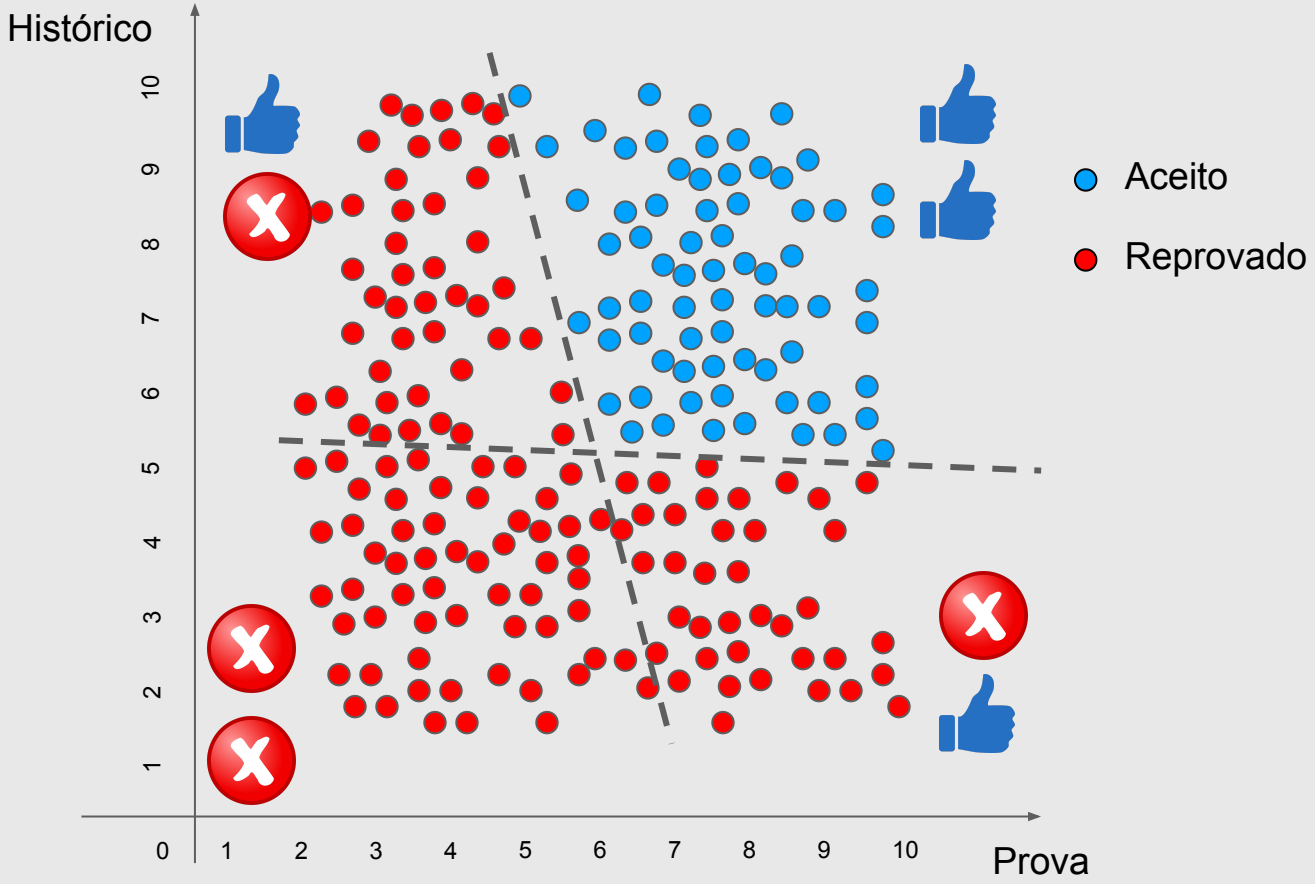
Processo de admissão de uma pós-graduação



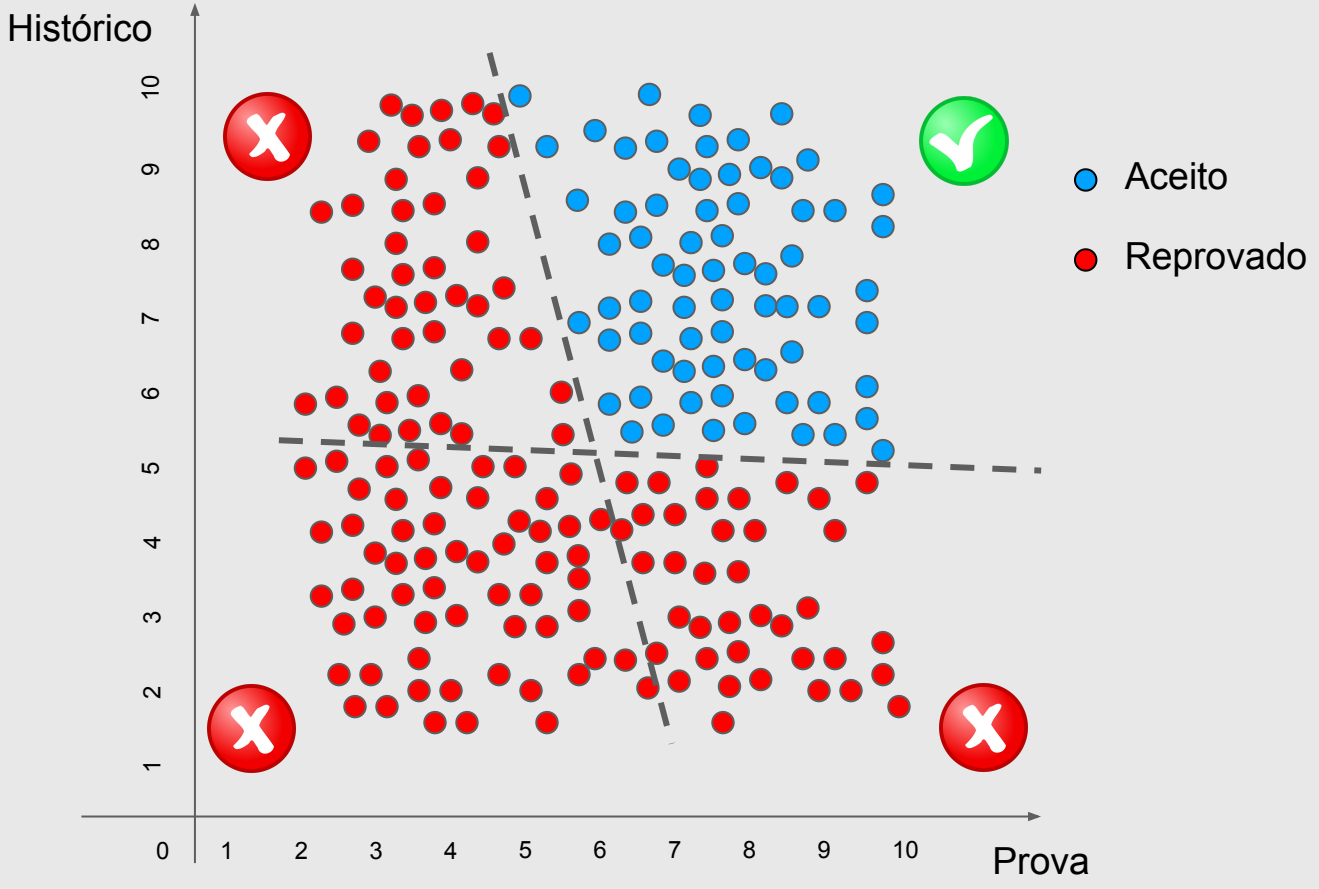
Processo de admissão de uma pós-graduação



Processo de admissão de uma pós-graduação



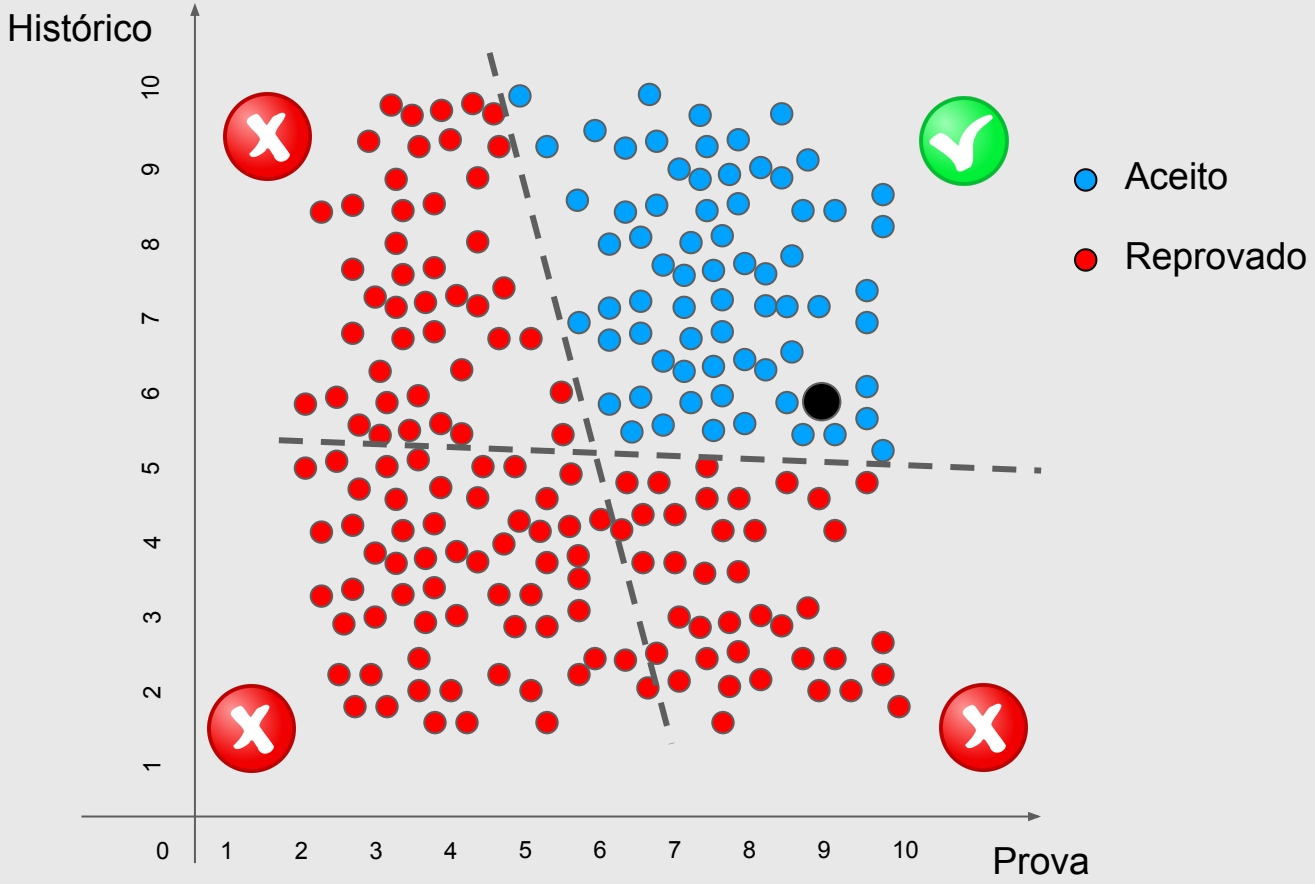
Processo de admissão de uma pós-graduação



Processo de admissão de uma pós-graduação

Entrada [prova, histórico]:

- [9, 6]



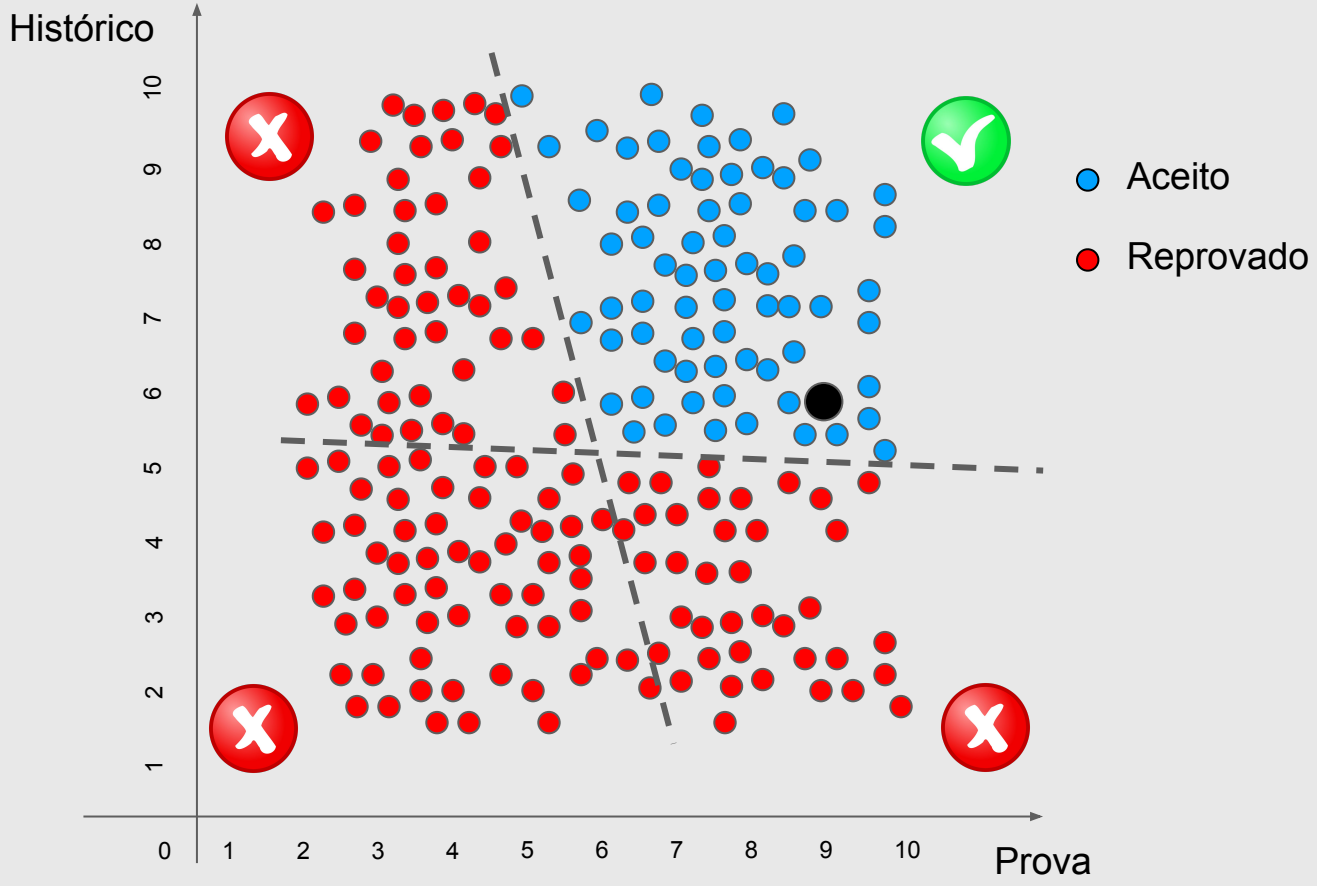
Processo de admissão de uma pós-graduação

Entrada [prova, histórico]:

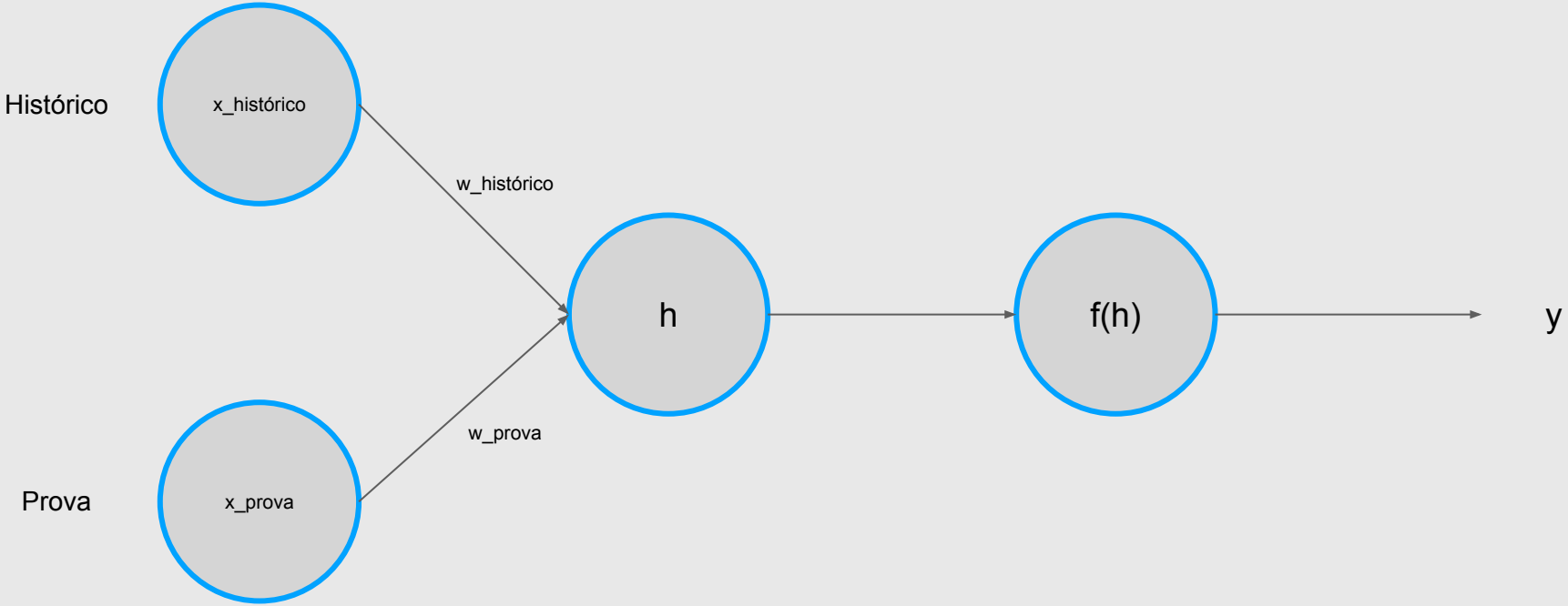
- [9, 6]

Saída:

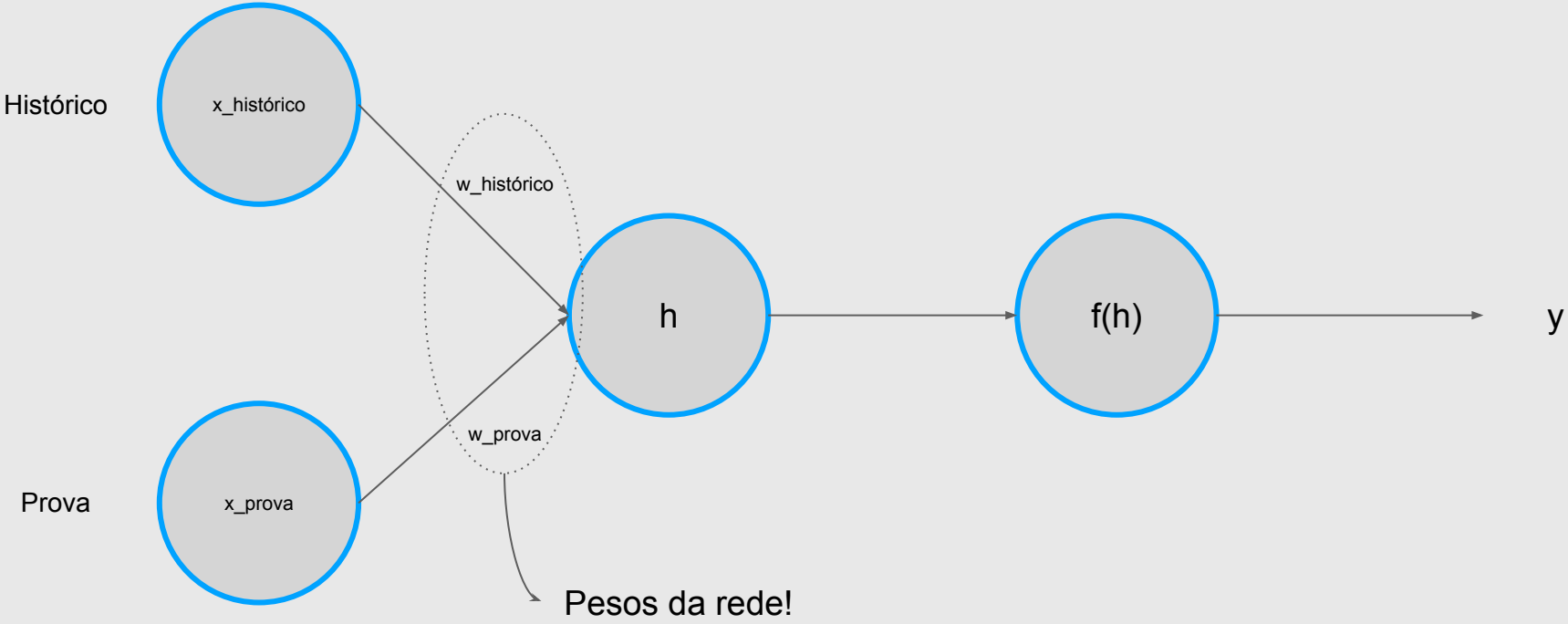
- Aceito



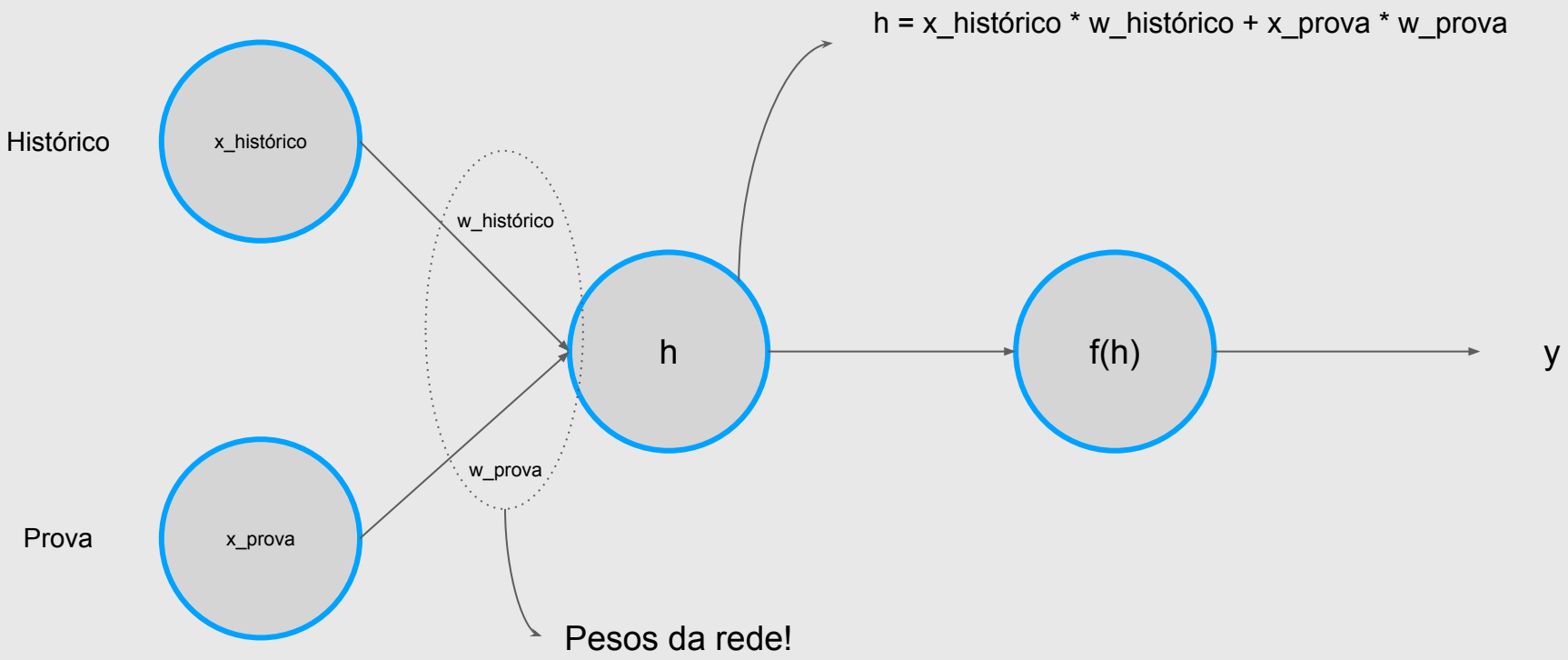
Perceptron


Histórico
x_histórico
w_histórico
h
f(h)
y
Prova
x_prova
w_prova

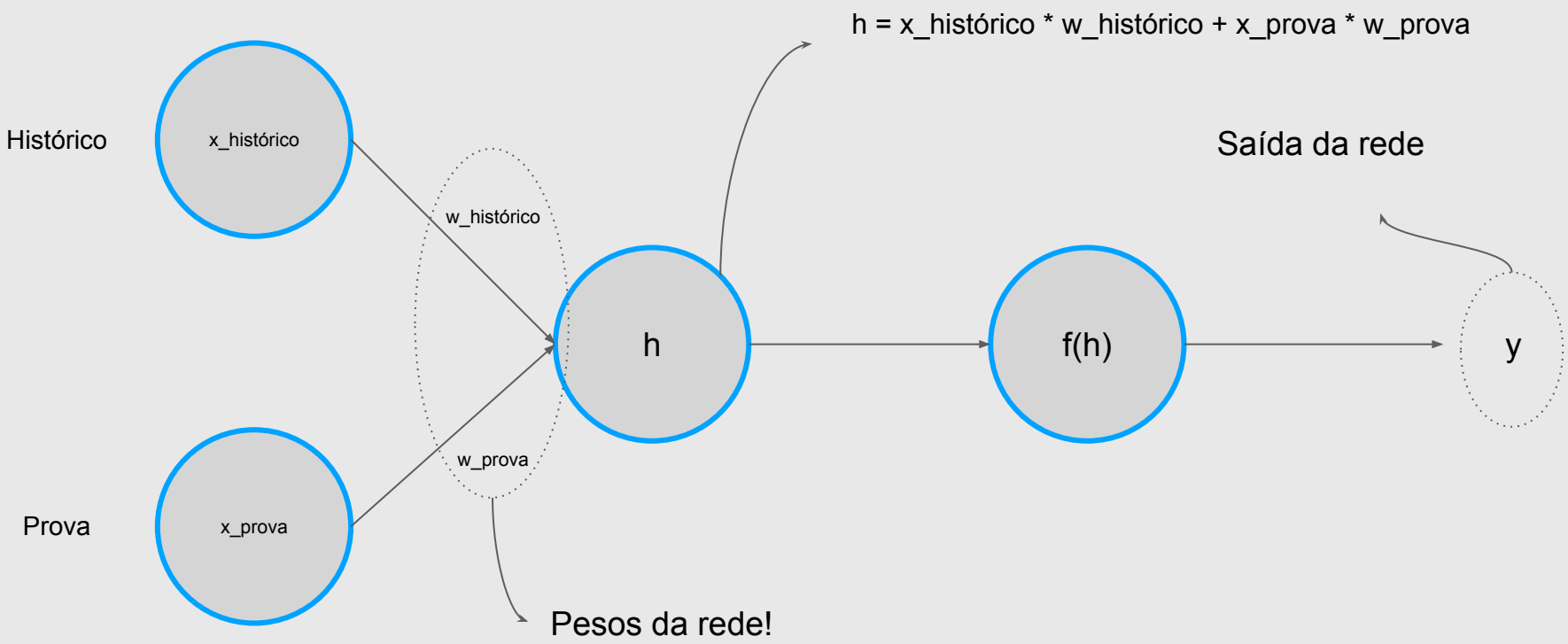
Perceptron


Histórico
x_histórico
w_histórico
h
f(h)
y
Prova
x_prova
w_prova
Pesos da rede!

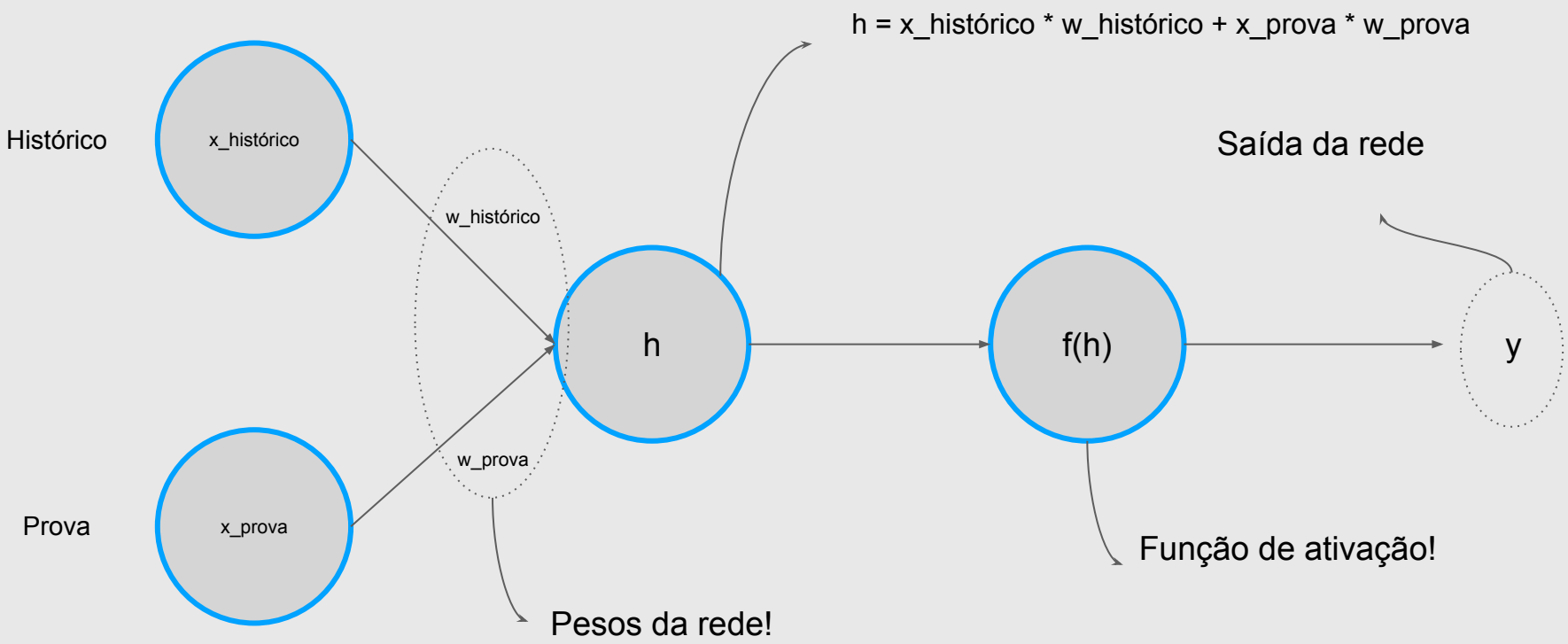
Funções de ativação



Funções de ativação



Funções de ativação



Funções de ativação

- Aplicam a não linearidade nos dados já que cada neurônio representa a equação de uma reta:

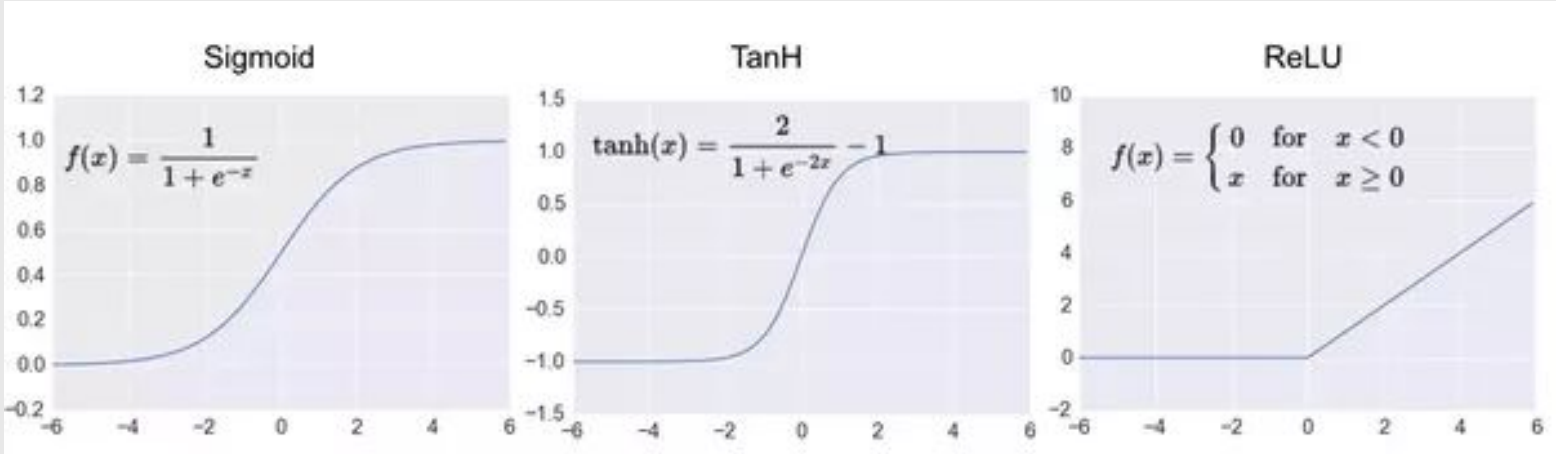
$$xw + b = y$$

Funções de ativação

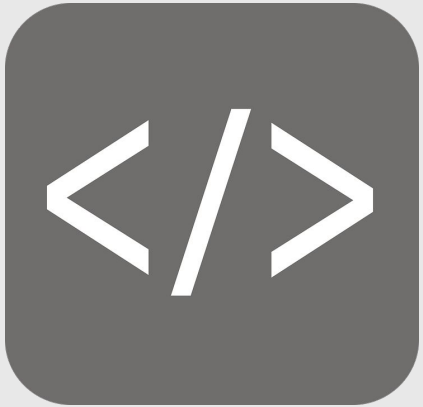
- Aplicam a não linearidade nos dados já que cada neurônio representa a equação de uma reta:

$$xw + b = y$$

- Funções de ativação são responsáveis por decidir qual deve ser a saída do neurônio dada a entrada e os pesos



Exemplo - Jupyter notebook



Como os pesos *w* são aprendidos?

- Para aprendermos os pesos a partir dos dados precisamos primeiro calcular uma medida de erro entre o resultado previsto e o resultado real
- Soma quadrática dos erros:

$$E = \frac{1}{2} \sum_{\mu} \sum_{j} \left[y_j^{\mu} - \hat{y}_j^{\mu} \right]^2$$

- ŷ: é o resultado previsto
- y: é o resultado real
- j: unidades de saída da rede
- μ: dados de entrada

Como os pesos *w* são aprendidos?

- A saída da rede sempre depende dos pesos:

$$\hat{y}_j^\mu = f\left(\sum_i w_{ij} x_i^\mu\right)$$

- Erro também depende dos pesos:

$$E = \frac{1}{2}\sum_\mu \sum_j \left[y_j^\mu - f\left(\sum_i w_{ij} x_i^\mu\right)\right]^2$$

- Nosso objetivo: encontrar os **pesos que minimizem o erro quadrático *E***
- São estes pesos que permitem que a rede neural “aprenda” com base nos dados de entrada

Como os pesos *w* são aprendidos? **Gradiente descendente!**

- Gradiente descendente
- Atualizando os pesos da rede:

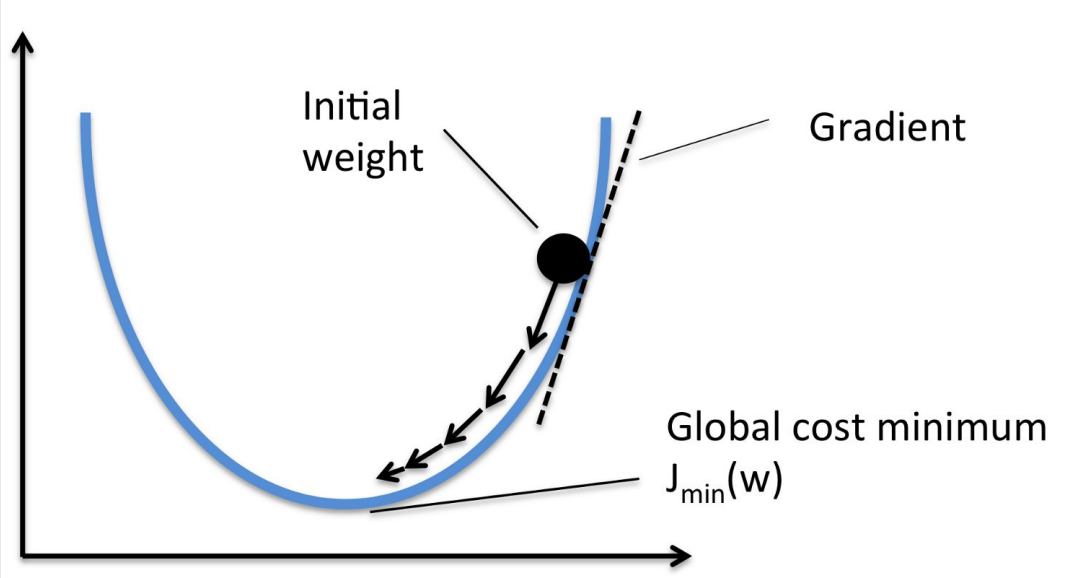
$$w_i = w_i + \Delta w_i$$

$$\Delta w_i = -\eta \frac{\delta E}{\delta w_i}$$

η velocidade de treinamento

$$\frac{\delta E}{\delta w_i} = -\left(y - \hat{y}\right) f'\left(h\right) x_i$$

$$\Delta w_i = \eta \left(y - \hat{y}\right) f'\left(h\right) x_i$$



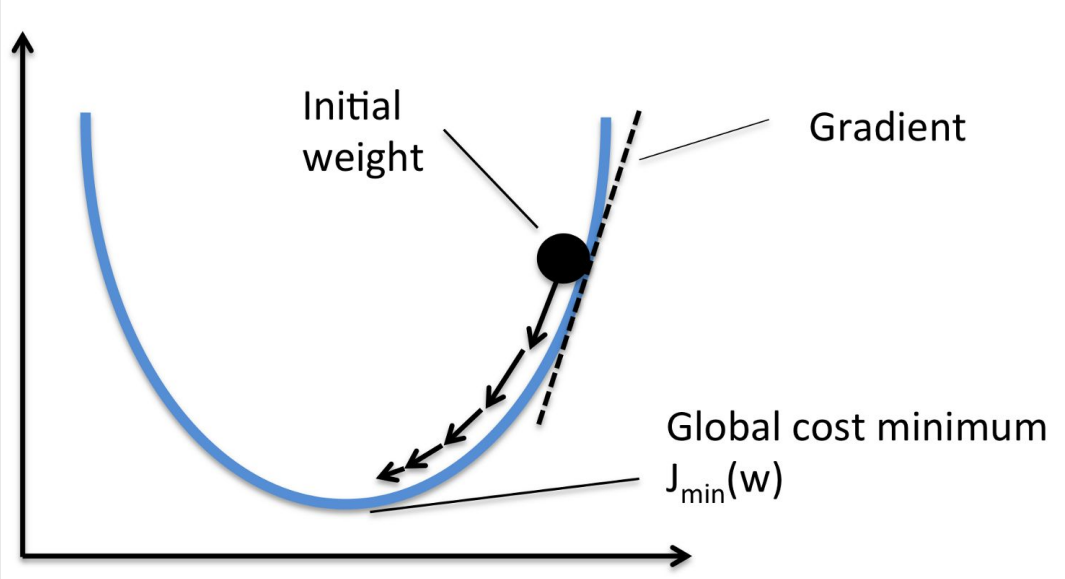
Como os pesos *w* são aprendidos? **Gradiente descendente!**

- Gradiente descendente
- Simplificando...

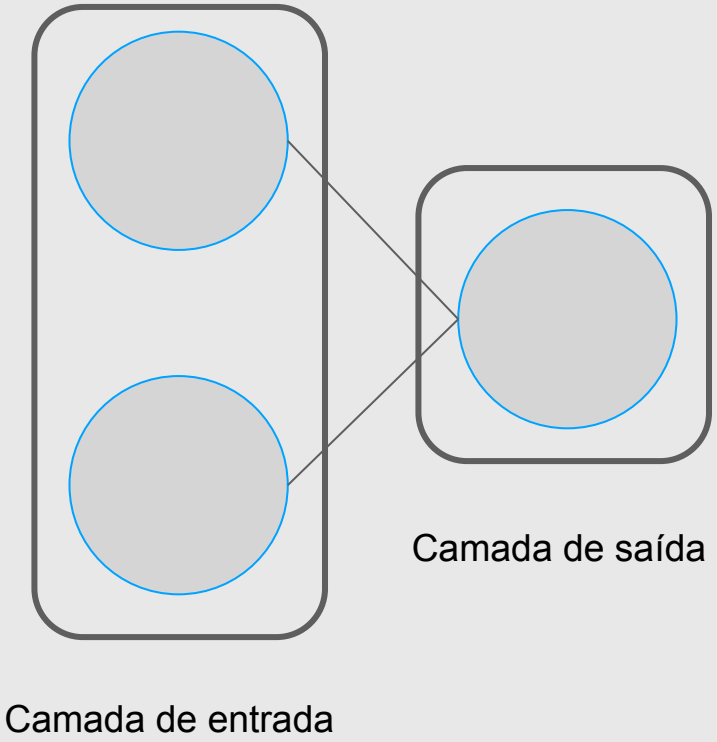
η velocidade de treinamento

$$\delta = (y - \hat{y})\, f'(h)$$

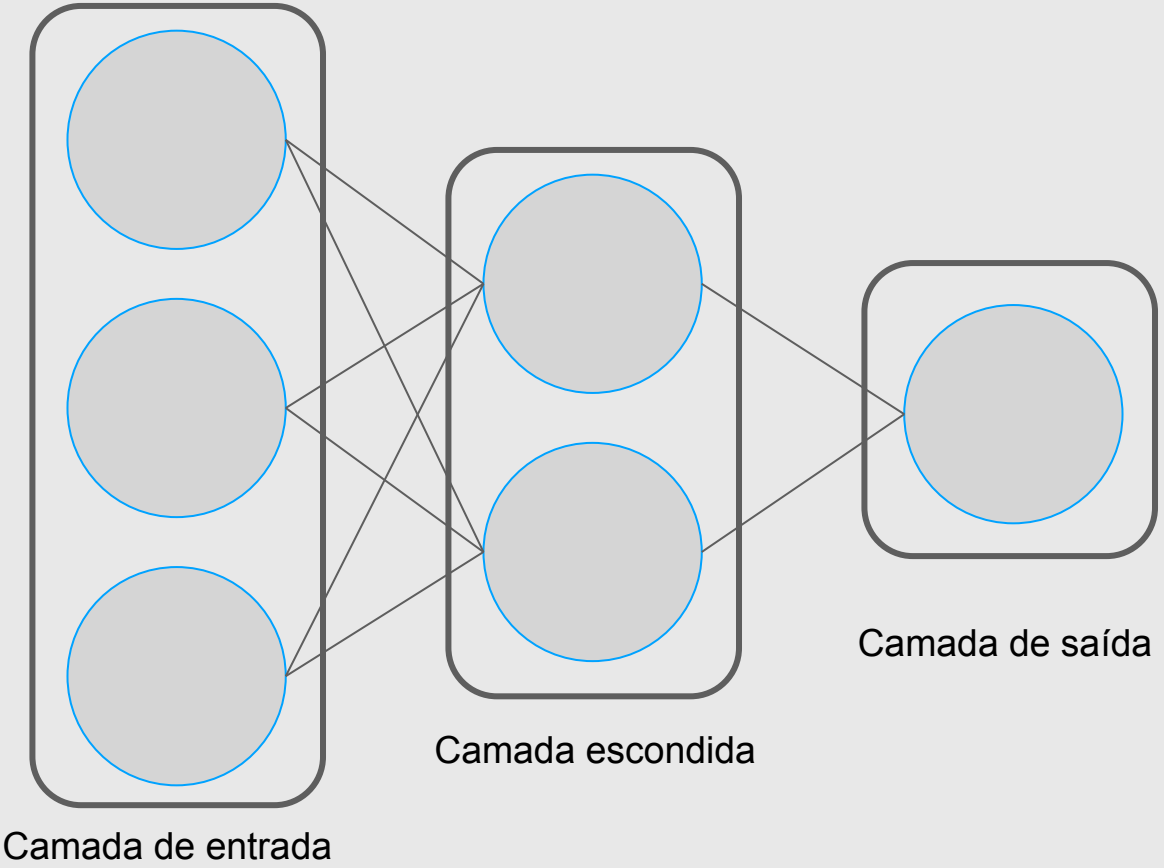
$$w_i = w_i + \eta \delta x_i$$



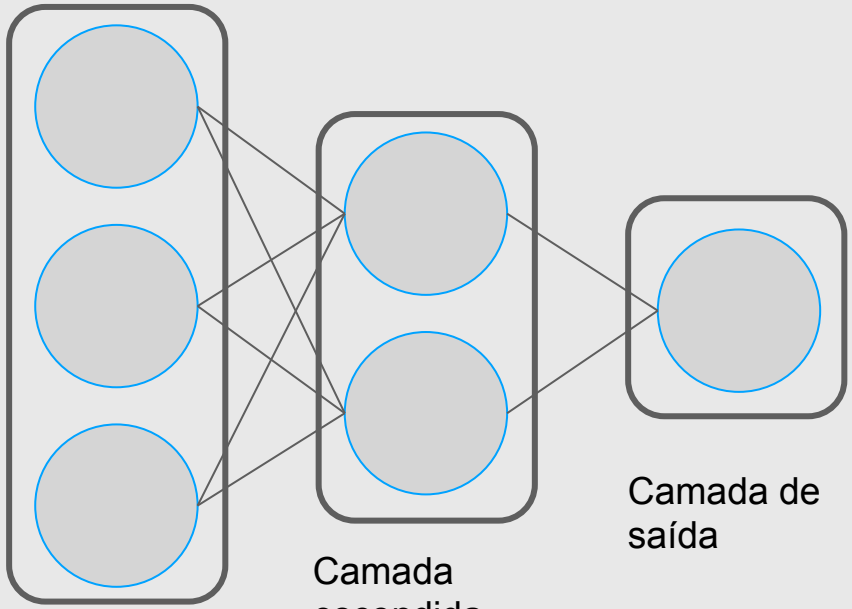
Rede neural uma única camada



Rede neural com múltiplas camadas



Como ajustar os pesos de acordo com o erro encontrado e para mais camadas de rede?
Backpropagation!

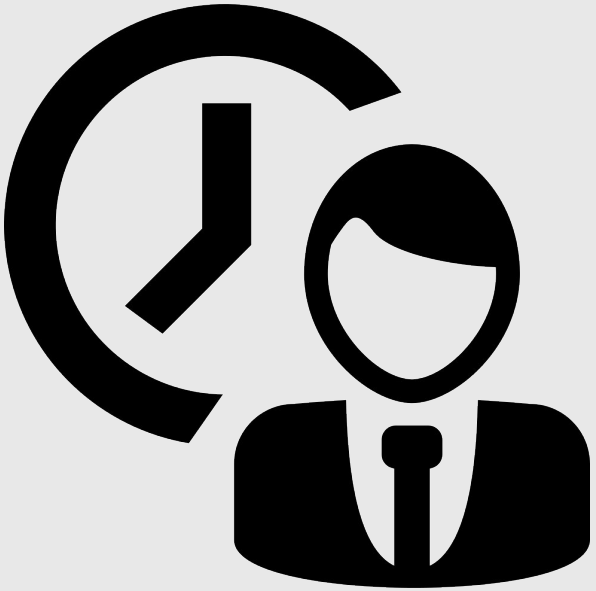


Erro: $\delta = (y - \hat{y})\, f'(h)$

Perguntas?

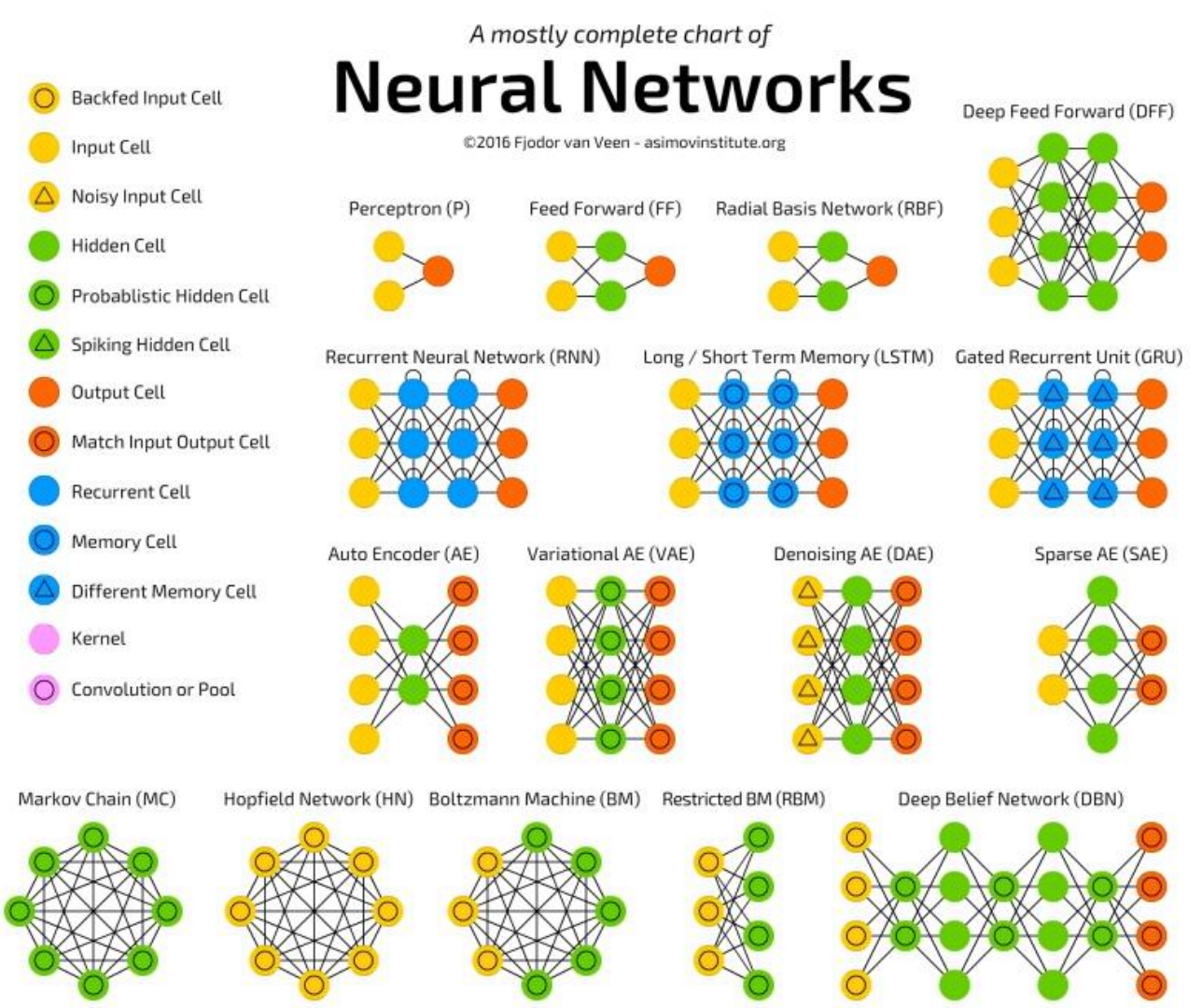
Intervalo

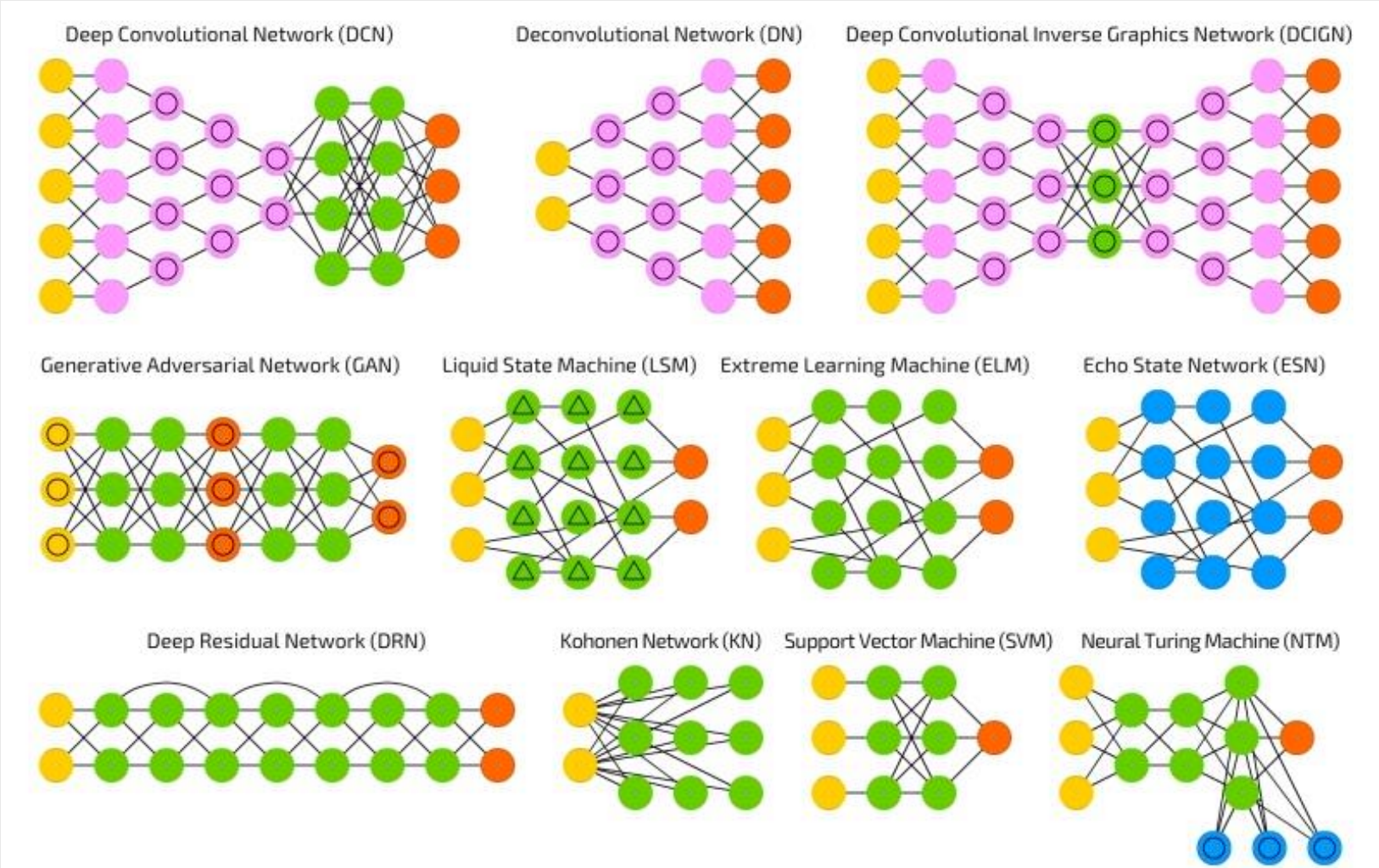
Problema real: Medir qualidade de um atendimento

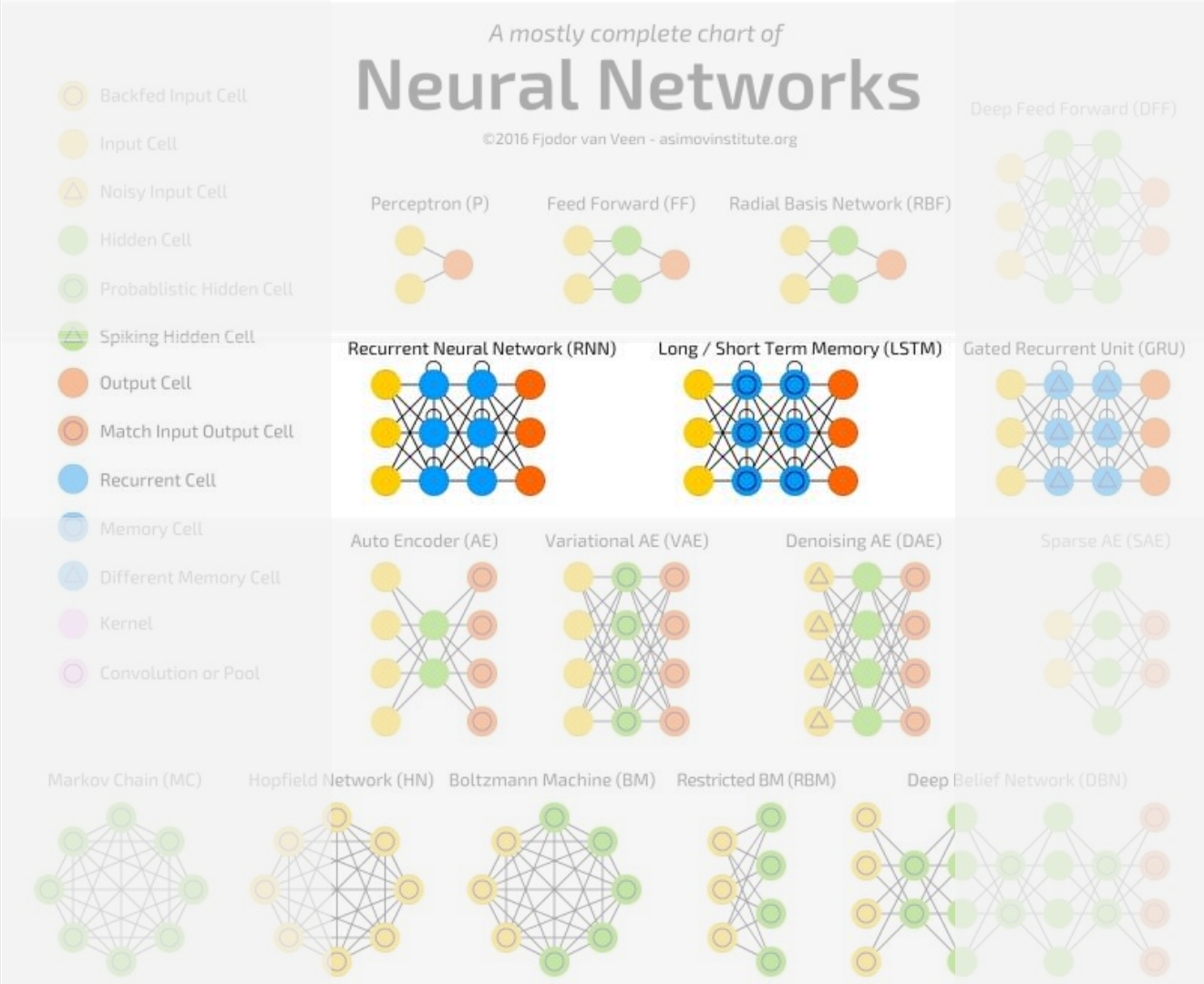


Problema real: Medir qualidade de um atendimento

- **Objetivo:** medir a satisfação do cliente
- **Abordagem:** analisar o sentimento (positivo/negativo) das respostas enviadas por e-mail para nossa central de atendimento
- **Observação importante:** a ordem que as palavras são utilizadas é importante

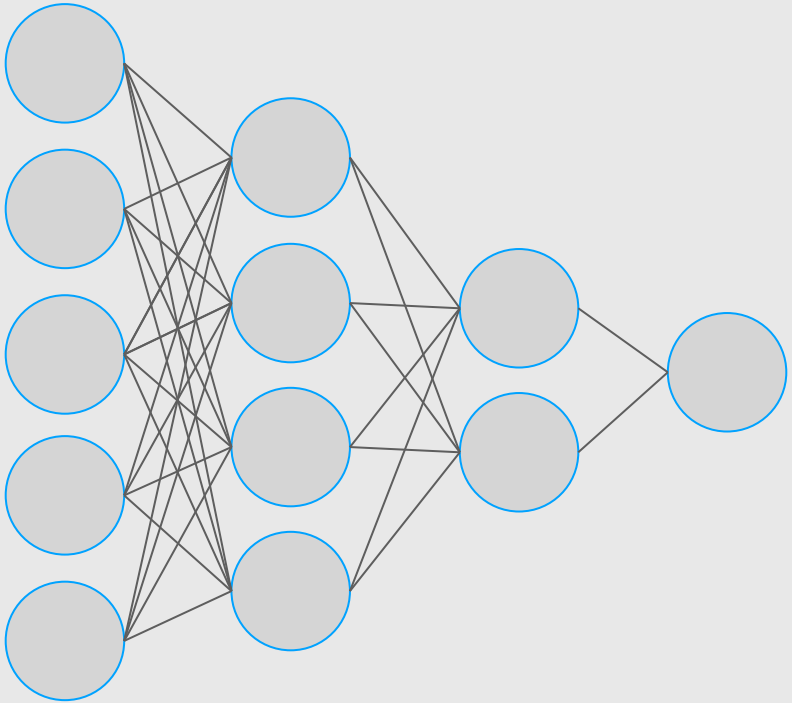

A mostly complete chart of
Neural Networks
©2016 Fjodor van Veen - asimovinstitute.org
Backfed Input Cell
Input Cell
Noisy Input Cell
Hidden Cell
Probablistic Hidden Cell
Spiking Hidden Cell
Output Cell
Match Input Output Cell
Recurrent Cell
Memory Cell
Different Memory Cell
Kernel
Convolution or Pool
Perceptron (P)
Feed Forward (FF)
Radial Basis Network (RBF)
Deep Feed Forward (DFF)
Recurrent Neural Network (RNN)
Long / Short Term Memory (LSTM)
Gated Recurrent Unit (GRU)
Auto Encoder (AE)
Variational AE (VAE)
Denoising AE (DAE)
Sparse AE (SAE)
Markov Chain (MC)
Hopfield Network (HN)
Boltzmann Machine (BM)
Restricted BM (RBM)
Deep Belief Network (DBN)


Deep Convolutional Network (DCN)
Deconvolutional Network (DN)
Deep Convolutional Inverse Graphics Network (DCIGN)
Generative Adversarial Network (GAN)
Liquid State Machine (LSM)
Extreme Learning Machine (ELM)
Echo State Network (ESN)
Deep Residual Network (DRN)
Kohonen Network (KN)
Support Vector Machine (SVM)
Neural Turing Machine (NTM)


A mostly complete chart of
Neural Networks
©2016 Fjodor van Veen - asimovinstitute.org
Backfed Input Cell
Input Cell
Noisy Input Cell
Hidden Cell
Probablistic Hidden Cell
Spiking Hidden Cell
Output Cell
Match Input Output Cell
Recurrent Cell
Memory Cell
Different Memory Cell
Kernel
Convolution or Pool
Perceptron (P)
Feed Forward (FF)
Radial Basis Network (RBF)
Deep Feed Forward (DFF)
Recurrent Neural Network (RNN)
Long / Short Term Memory (LSTM)
Gated Recurrent Unit (GRU)
Auto Encoder (AE)
Variational AE (VAE)
Denoising AE (DAE)
Sparse AE (SAE)
Markov Chain (MC)
Hopfield Network (HN)
Boltzmann Machine (BM)
Restricted BM (RBM)
Deep Belief Network (DBN)

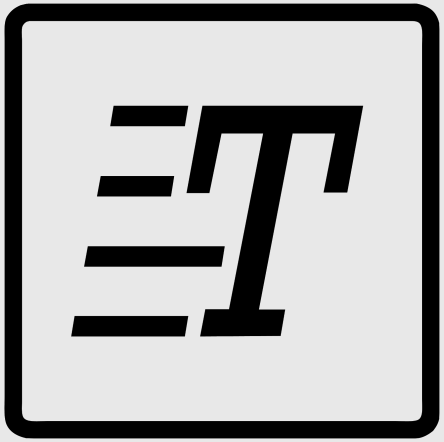
Tipos de arquiteturas de rede

- Como vimos anteriormente, as redes neurais tradicionais são neurônios densamente conectados (*fully connected*)



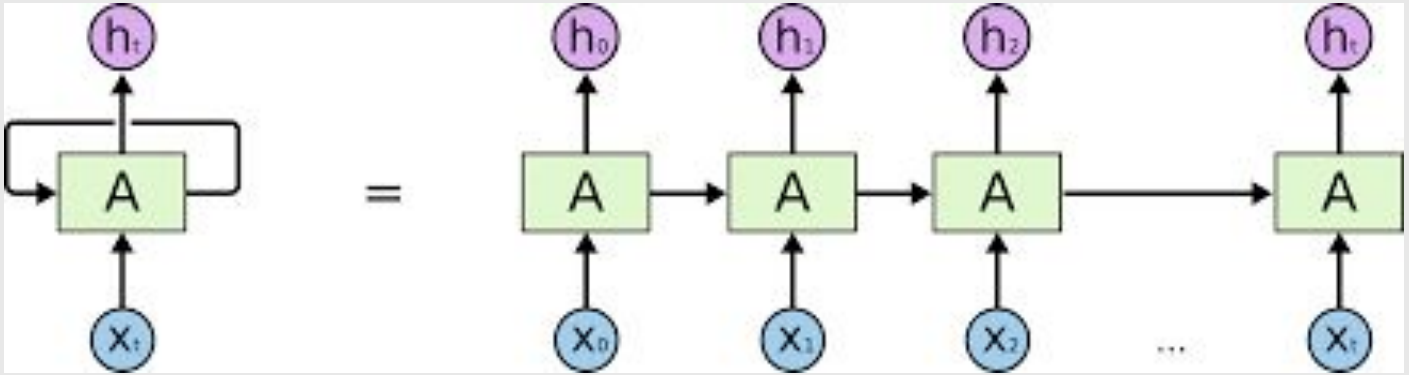
Uma nova arquitetura de rede: Redes Neurais Recorrentes

- Trabalhar com texto exige a utilização de uma nova arquitetura de redes neurais: Redes Neurais Recorrentes (RNN)
- Estas redes são utilizadas para trabalhar com sequências, como: texto, previsão do preço de ações, voz e etc

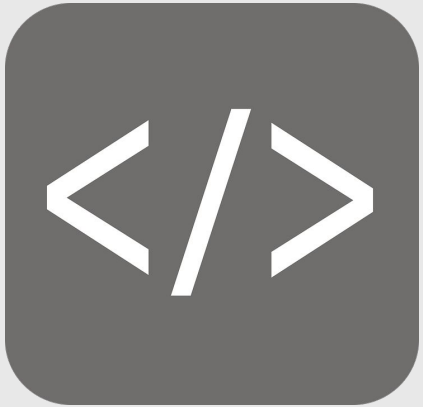


Uma nova arquitetura de rede: Redes Neurais Recorrentes

- São redes de apresentam um “loop” interno



Exemplo - Jupyter notebook



Perguntas?

Feedback