

A primeira Escola presencial gratuita de Inteligência Artificial do Brasil



Introdução a Redes Neurais Artificiais (RNAs)

DIY com Keras/Tensorflow!

Agenda

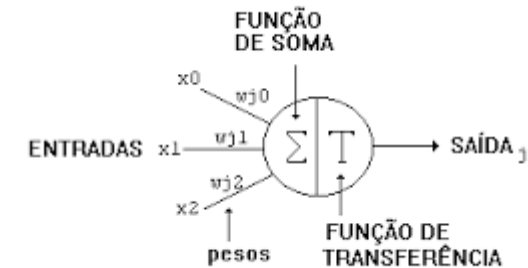
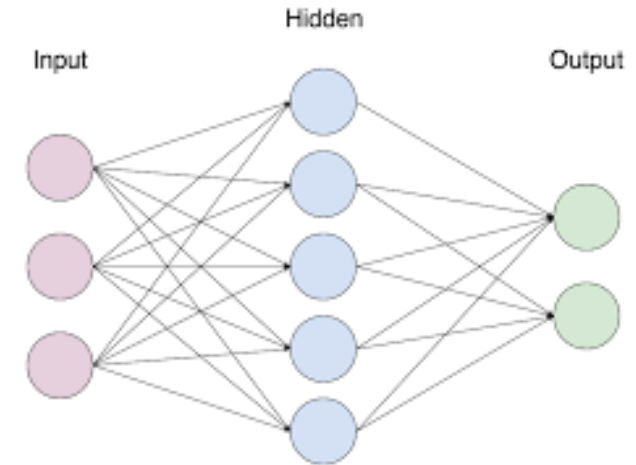
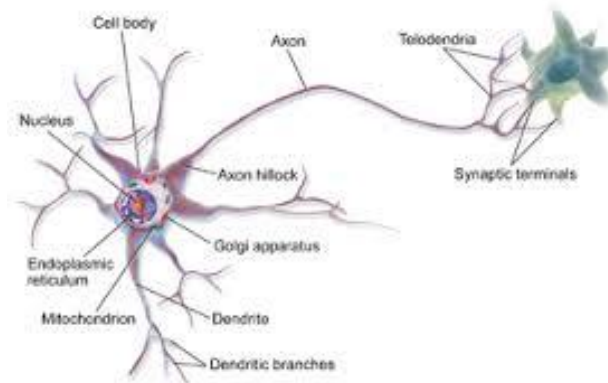
- Motivação
- Breve Histórico
- O Neurônio
- Arquiteturas Básicas
- Aprendizagem
- Aplicações: MLP e CNN com Keras-TF

Motivação: 2W1H

WHAT, WHY e HOW

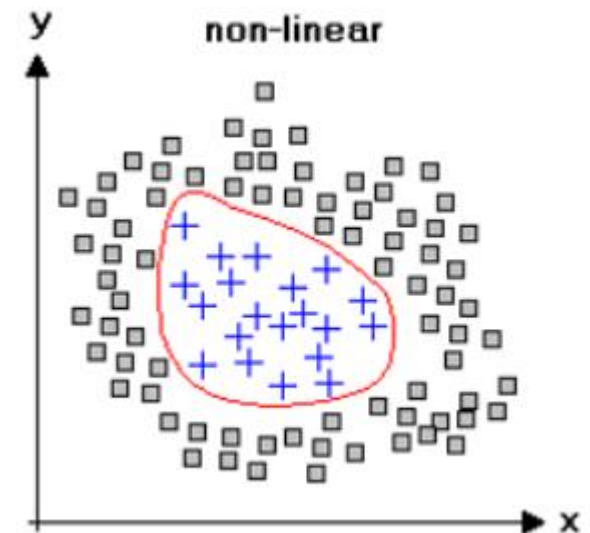
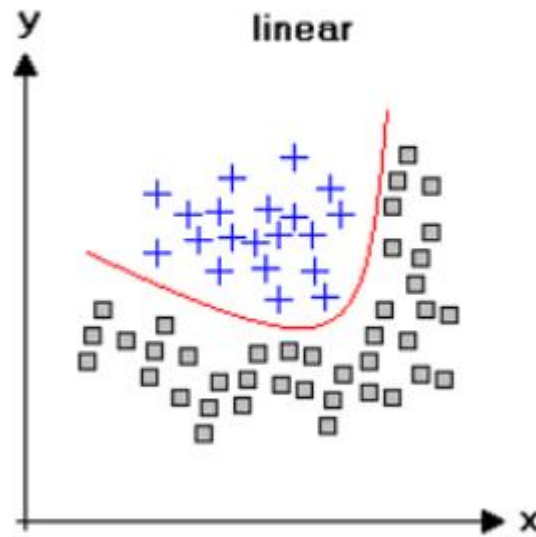
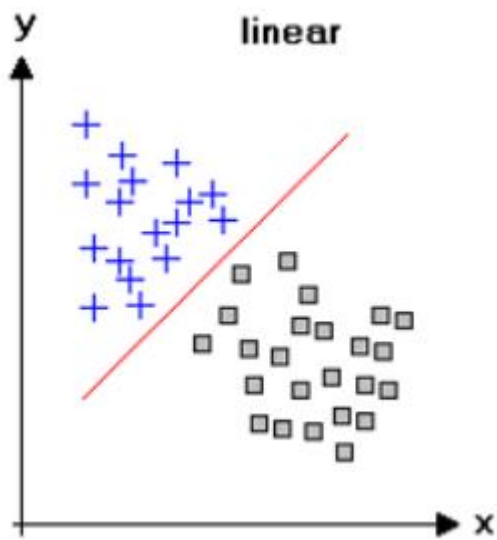
Motivação

- What: O que são Redes Neurais Artificiais?



Motivação

- Why: Por que eu devo usar RNAs?



Quando eu lido com informações de natureza não-linear, é muito difícil tentar organizá-las ou separá-las com apenas uma linha ou reta (filtros lineares)

Motivação

- Why: Por que eu devo usar RNAs?

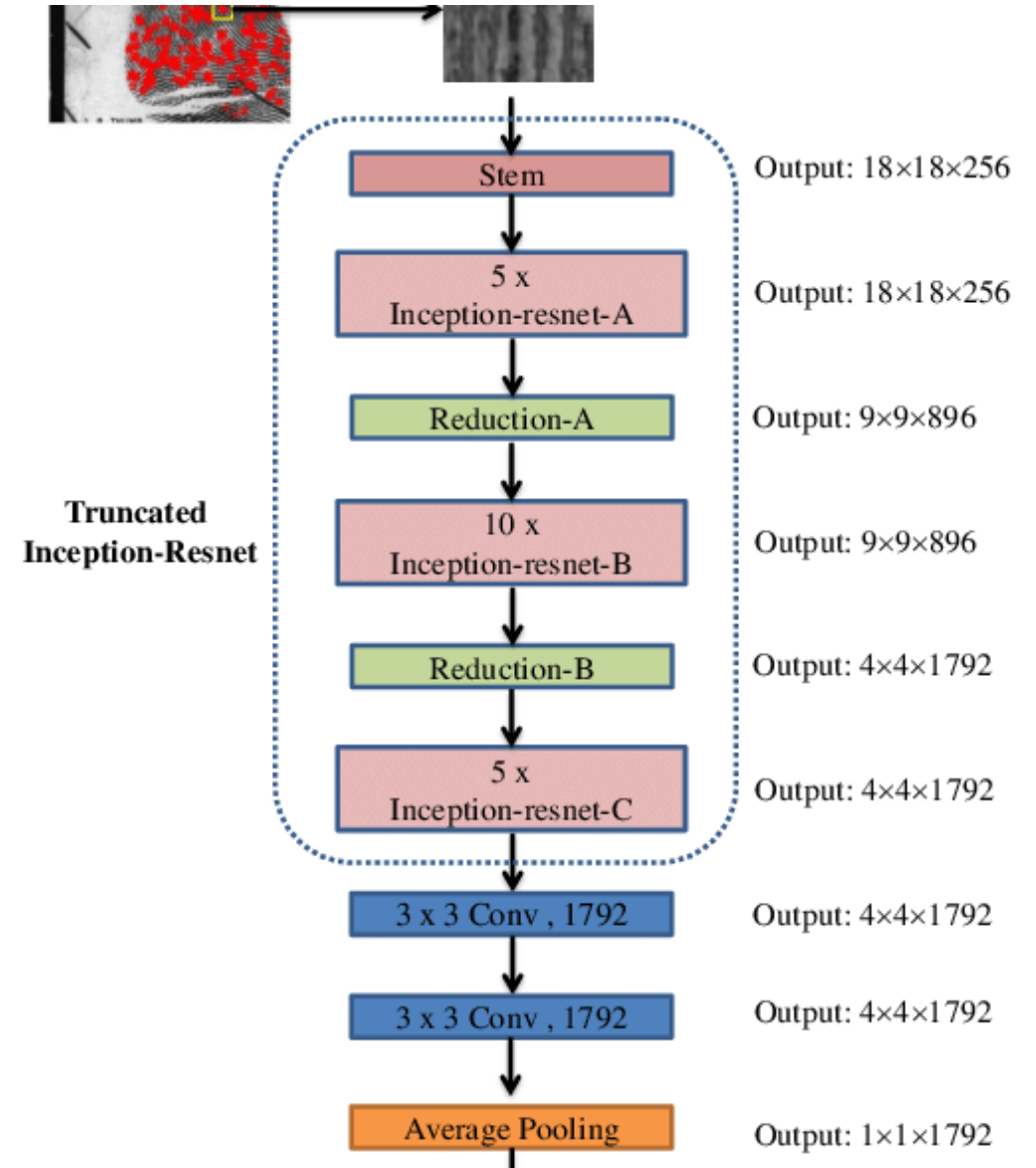


O que diferencia uma imagem da outra? Quais critérios ou padrões são necessários para que se possa distinguir essas informações usando um modelo matemático?

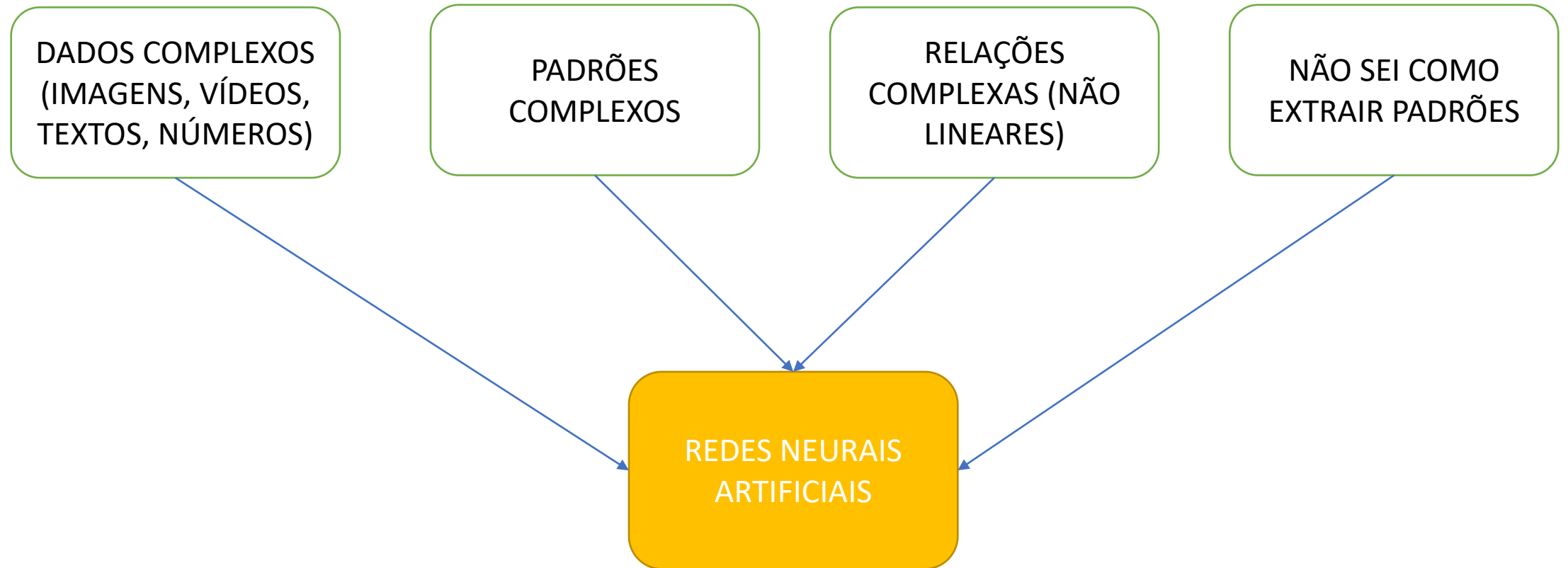
Existem padrões implícitos que dificultam a tarefa de classificar cada uma dessas imagens corretamente.

Motivação

- HOW?????
- Múltiplas Camadas: Extração de Características
- Lida com dados de maneira direta: ela faz todo o processamento, basta deixar com ela!
- Aprendizagem: calibragem de extração e processamento de características



Motivação



Histórico

Uma breve história dessa tecnologia “avançada”

Histórico

- 1943 – Warren McCulloch e Walter Pitts = Neurônios como circuitos elétricos!
- 1949 – Donald Hebb: Conceito de aprendizagem!
- 1950 – Primeira RNA rodando em um PC!!!! (Nathaniel Rochester, IBM)
- 1959 – Redes Neurais ADALINE (Adaptative Linear)
- 1975 – Primeira Rede Multi-Camadas!
- 1986 – Rumelhart: Backpropagation!!!

Histórico

- Muitos artigos foram criticados ou até refutados, deixando um grande hiato na linha do tempo.
- Ainda mais que outras tecnologias chamavam muito mais atenção, pelo fato de darem muito certo: Computadores!!!
- Por fim, esses modelos já eram computacionalmente custosos de se executar, por isso o seu questionamento! Até que...
- 2006: Redes Neurais de Convolução: O melhor casamento entre Redes Neurais e Processamento de Imagens!!!!

O Neurônio

Unidade Fundamental, O Tico e o Teco!

Neurônio

- Elemento chave que conduz as informações pela rede
- Se conecta com outros neurônios através de SINAPSES
- As sinapses regulam como a informação deve propagar para o neurônio, através dos PESOS SINÁPTICOS
- Dentro do neurônio, todas as informações são INTEGRADAS e vão compor o POTENCIAL de disparo de informação.
- Caso esse potencial exceda um certo LIMIAR, uma nova informação é propagada pela saída do neurônio

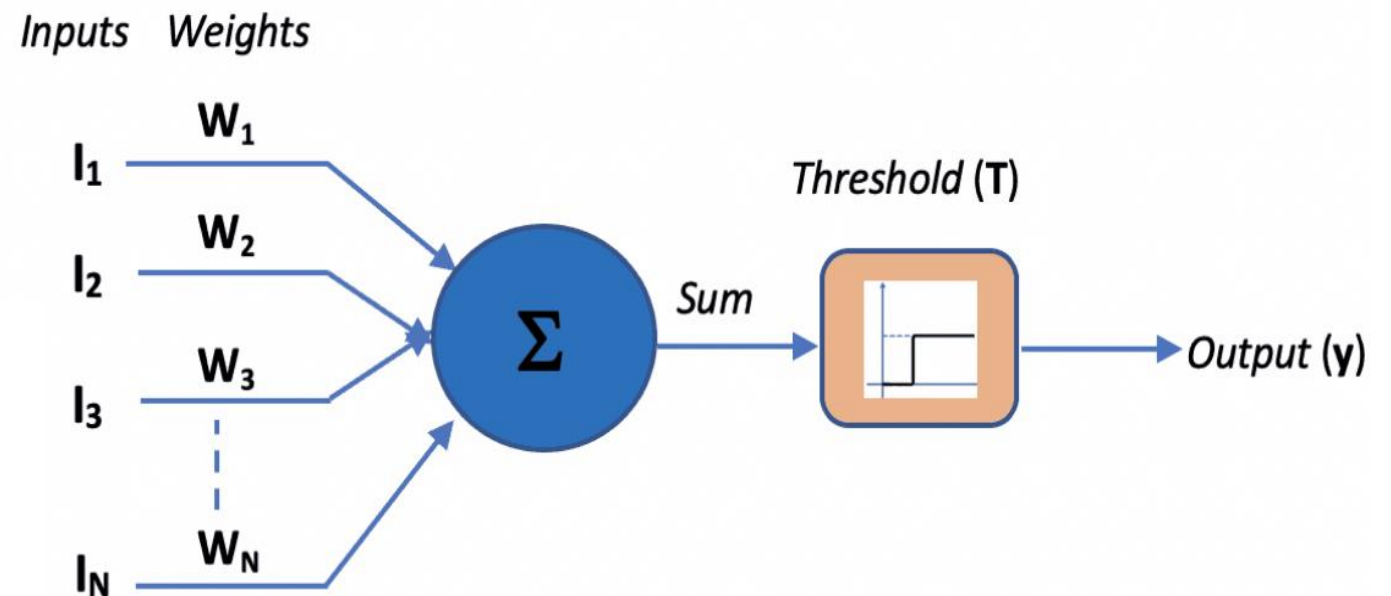
As Gerações dos Neurônios Artificiais (até então...)

- Primeira Geração: McCulloch & Pitts

- Valores Binários (entrada e saída)
- Pesos Sinápticos Unitários
- Integração: Somatório
- Função Limiar (Degrau)

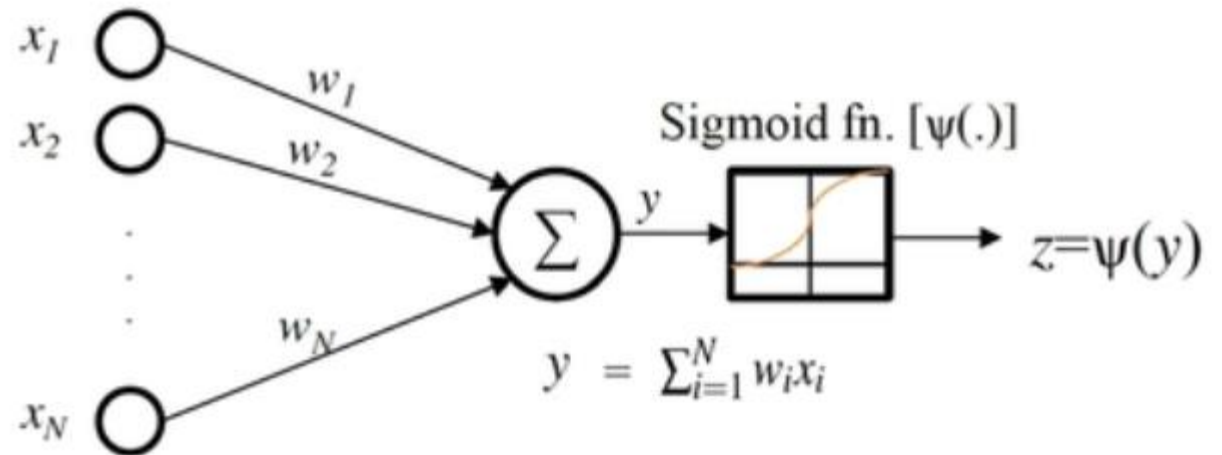
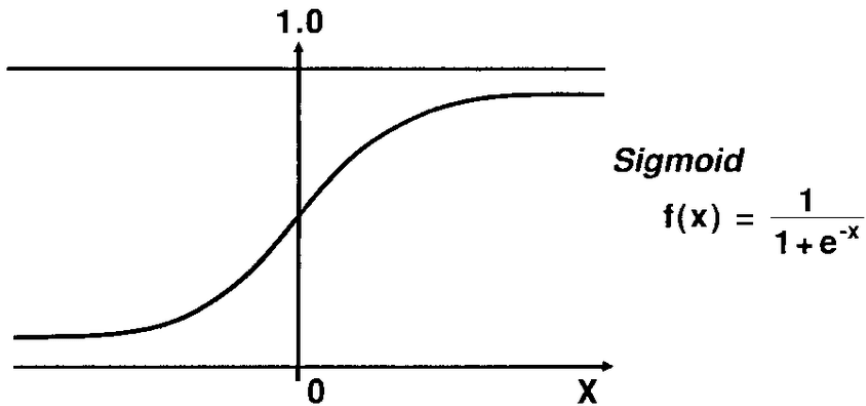
$$g(x_1, x_2, x_3, \dots, x_n) = g(\mathbf{x}) = \sum_{i=1}^n x_i$$

$$y = f(g(\mathbf{x})) = \begin{cases} 1 & \text{if } g(\mathbf{x}) \geq \theta \\ 0 & \text{if } g(\mathbf{x}) < \theta \end{cases}$$



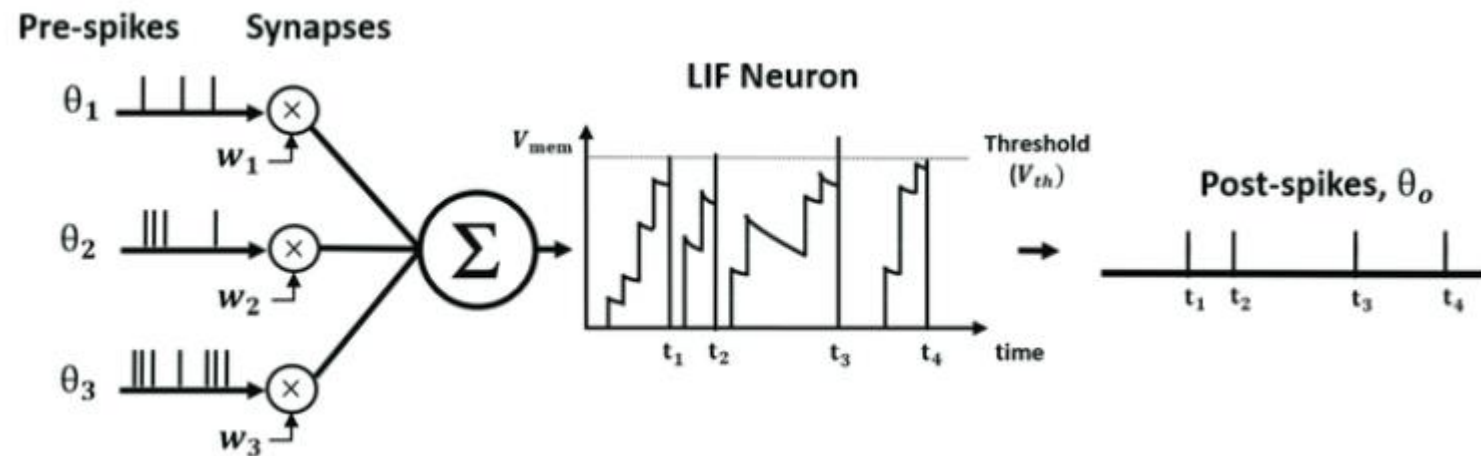
As Gerações dos Neurônios Artificiais (até então...)

- Segunda Geração: Perceptron
 - Valores Reais (entrada e saída)
 - Pesos Sinápticos Reais
 - Integração: Somatório
 - Função Sigmóide



As Gerações dos Neurônios Artificiais (até então...)

- Terceira Geração: Spiking Neuron
 - Entrada Pré-sináptica: Spikes (Impulsos)
 - Pesos Sinápticos Reais
 - Potencial da Membrana do Neurônio
 - Sistema Dinâmico
 - Função Limiar
 - Saída: Spike Train



Arquiteturas de RNAs

Modelos clássicos

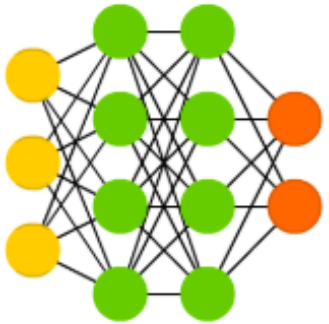
Arquiteturas

- Dependendo da forma como os neurônios são conectados, algumas arquiteturas podem ser montadas
- Nessas arquiteturas, os neurônios ficam organizados em camadas, que se conectam com outras camadas
- Quanto mais camadas, mais profunda é a RNA
- 1) **Feed Forward**: os neurônios somente se conectam com neurônios de camadas mais profundas. A direção da conexão segue uma única direção: Esquerda -> Direita ou Entrada -> Saída
- 2) **Recorrente**: os neurônios podem se conectar com camadas anteriores. Ainda existe o sentido Entrada -> Saída, só que com novas ligações

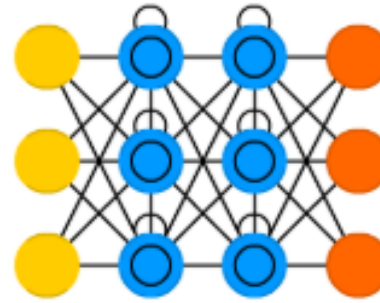
Arquiteturas

- 3) **Populacional:** Dentro da camada, os neurônios podem se conectar. Isso cria um ambiente competitivo entre eles.

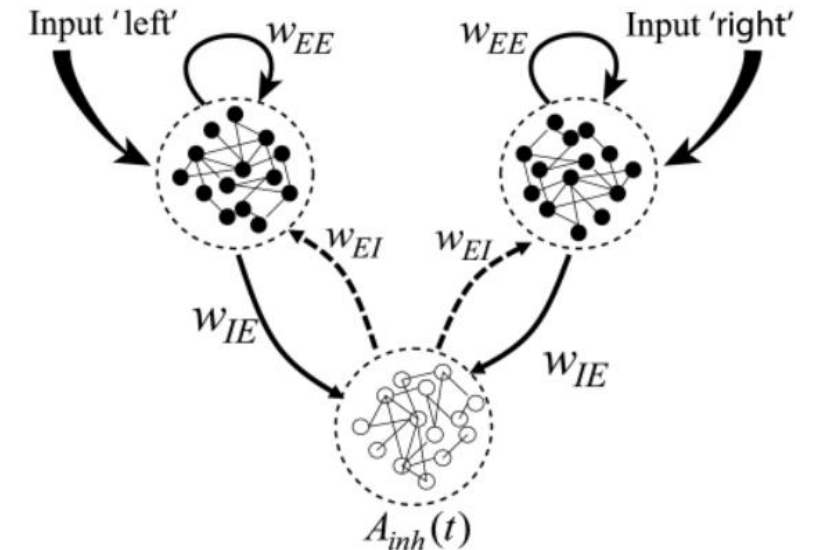
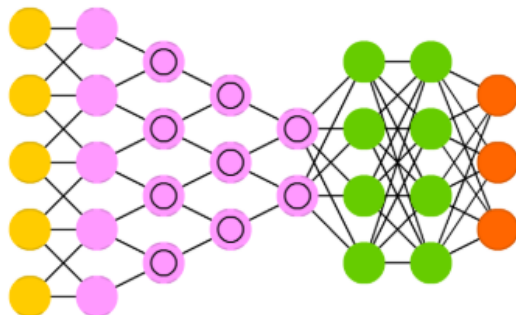
Deep Feed Forward (DFF)



Long / Short Term Memory (LSTM)



Deep Convolutional Network (DCN)



Aprendizagem em SNAs

Já ouviu falar dos Algoritmos pela Internet?

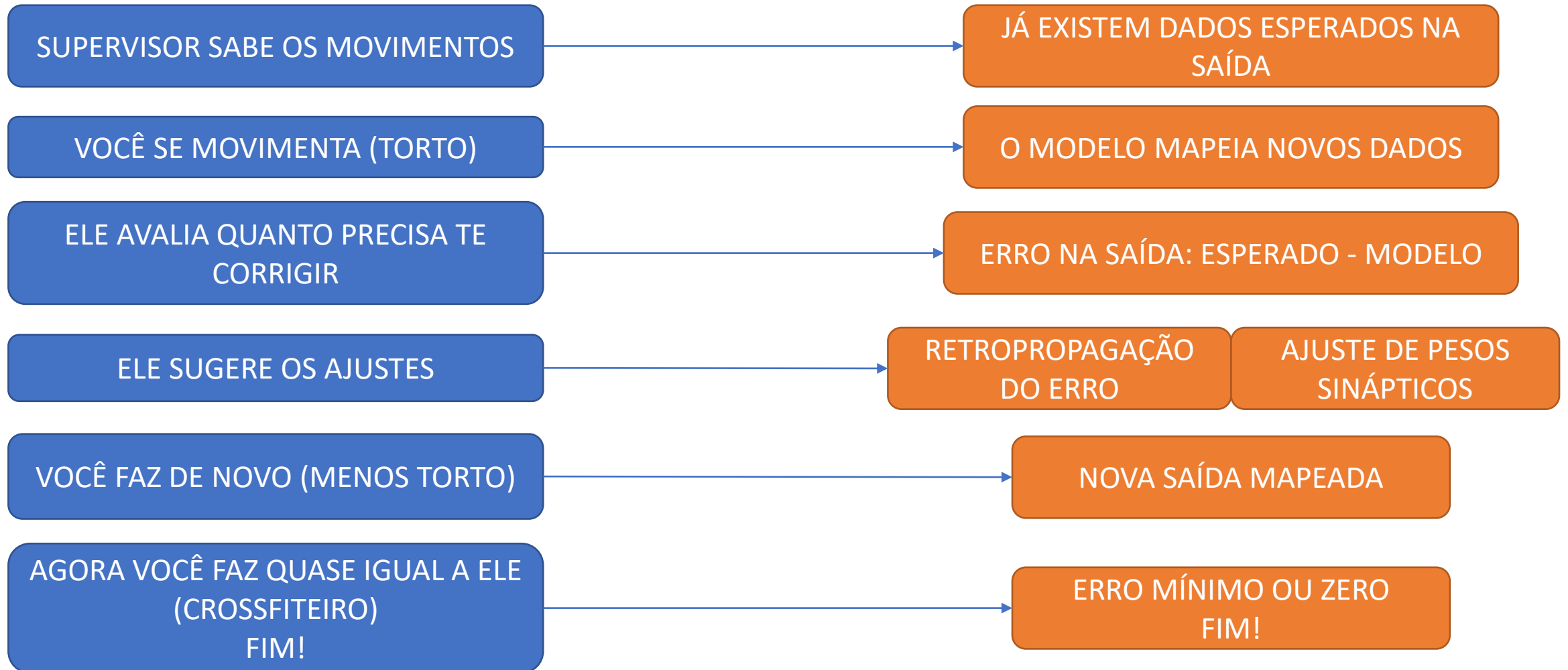
Chegou o momento de olhar por baixo dos panos!!!!

Aprendizagem



Personal Trainer

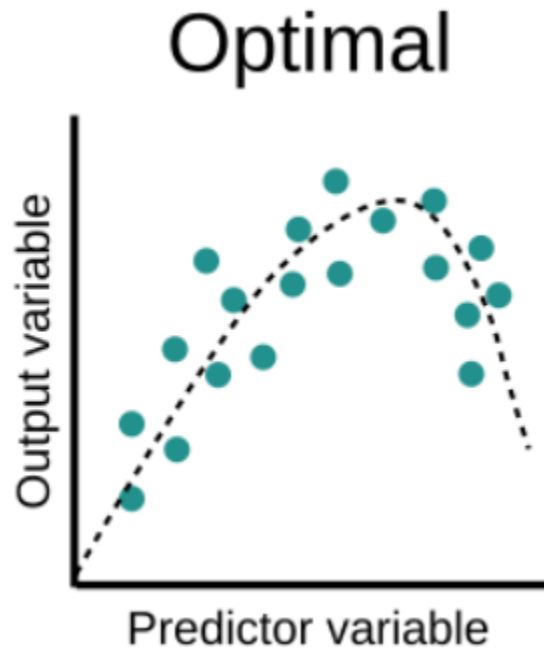
Aprendizagem: Vida X Modelo



Aprendizagem

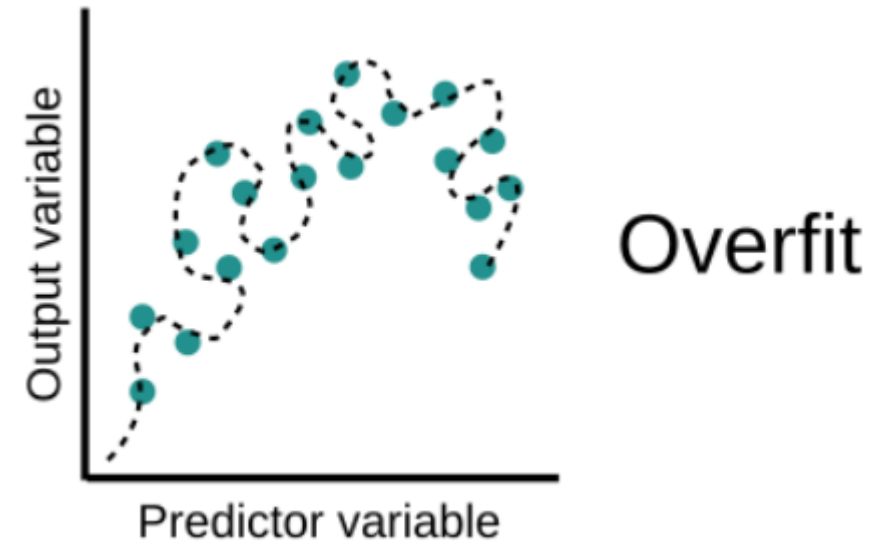
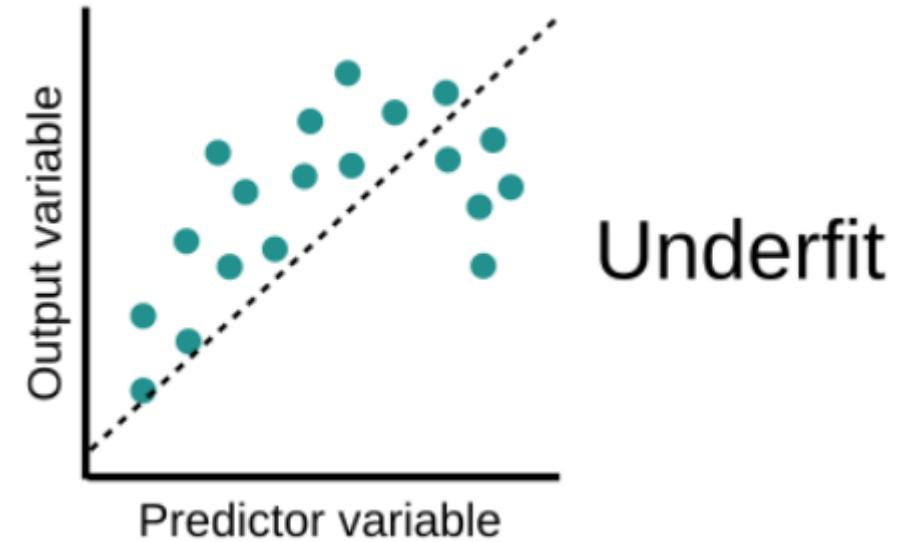
- Aprendizagem Supervisionada: BackPropagation
 - Problema de Otimização não-linear irrestrita
 - Parâmetros a serem Otimizados: Pesos Sinápticos
 - Minimização da Função Erro/Objetivo/Custo
 - Etapa de Treinamento da RNA
- Aprendizagem Não Supervisionada?
 - Sim! >> Spiked Neurons >> Relação temporal entre os Spikes
 - STDP: Spiked Timing Dependent Plasticity
 - Dependendo dos tempos dos Spikes, os pesos são amplificados ou atenuados
 - Quem ajusta os pesos são os dados!
 - O modelo se adapta aos dados!

Aprendizagem



No final das contas, é
isso que queremos!

MAS....e se:



Aprendizagem

- UnderFitting:
 - Seu modelo é menos complexo que a realidade
 - Sua base de dados de entrada não retrata a sua realidade completamente
 - Necessita de um treino melhor
- Overfitting:
 - Seu modelo é mais complexo que a realidade
 - Sua base de dados de entrada retrata muito bem uma realidade específica
 - Contudo se qualquer nova informação chegar ao modelo, ele vai errar feio

PERDA DA CAPACIDADE DE GENERALIZAÇÃO – ADAPTABILIDADE AO NOVO

Aprendizagem

- Validação do modelo após aprendizagem:
- Base de Treino x Base de Teste
- Treino: Otimização do Modelagem do problema
- Teste: Otimização da Capacidade de Generalizar
- Pode ir bem no treino, mas vai mal no teste: Reveja seu modelo!
- Não busque o perfeito, e sim o ótimo!

Vamos ver na prática?

Duas formas de resolver o mesmo problema usando Keras Tensorflow