

SICO70

SISTEMAS INTELIGENTES 2

Aula 06 - Breve Introdução ao *Deep Learning*

Prof. Rafael G. Mantovani



Apucarana - PR, Brasil

Universidade Tecnológica Federal do Paraná (UTFPR)
Engenharia de Computação

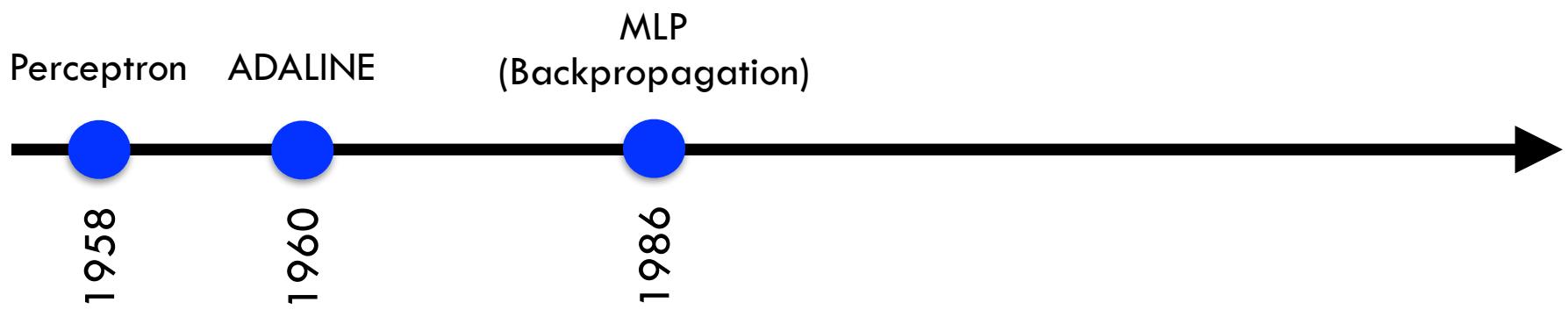
Roteiro

- 1 Introdução**
- 2 Deep Learning (DL)**
- 3 Treinamento de DL**
- 4 Exemplos**
- 5 Referências**

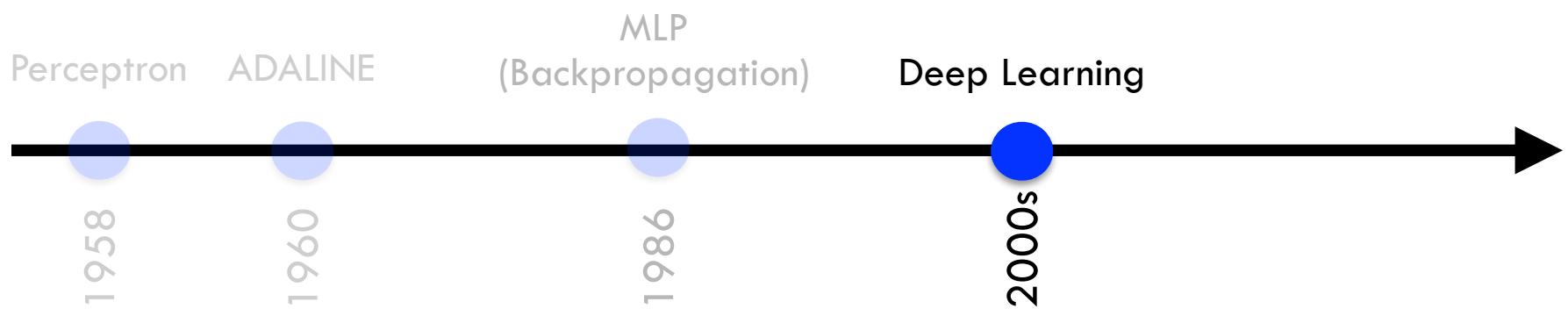
Roteiro

- 1 Introdução**
- 2 Deep Learning (DL)**
- 3 Treinamento de DL**
- 4 Exemplos**
- 5 Referências**

Introdução



Introdução



Introdução

- **Teorema da Aproximação Universal (George Cybenko, 1989)**
 - “... uma RNA com única camada escondida e um número finito de neurônios pode aproximar qualquer função contínua”.

Introdução

- **Teorema da Aproximação Universal (George Cybenko, 1989)**
 - "... uma RNA com única camada escondida e um número finito de neurônios pode aproximar qualquer função contínua".

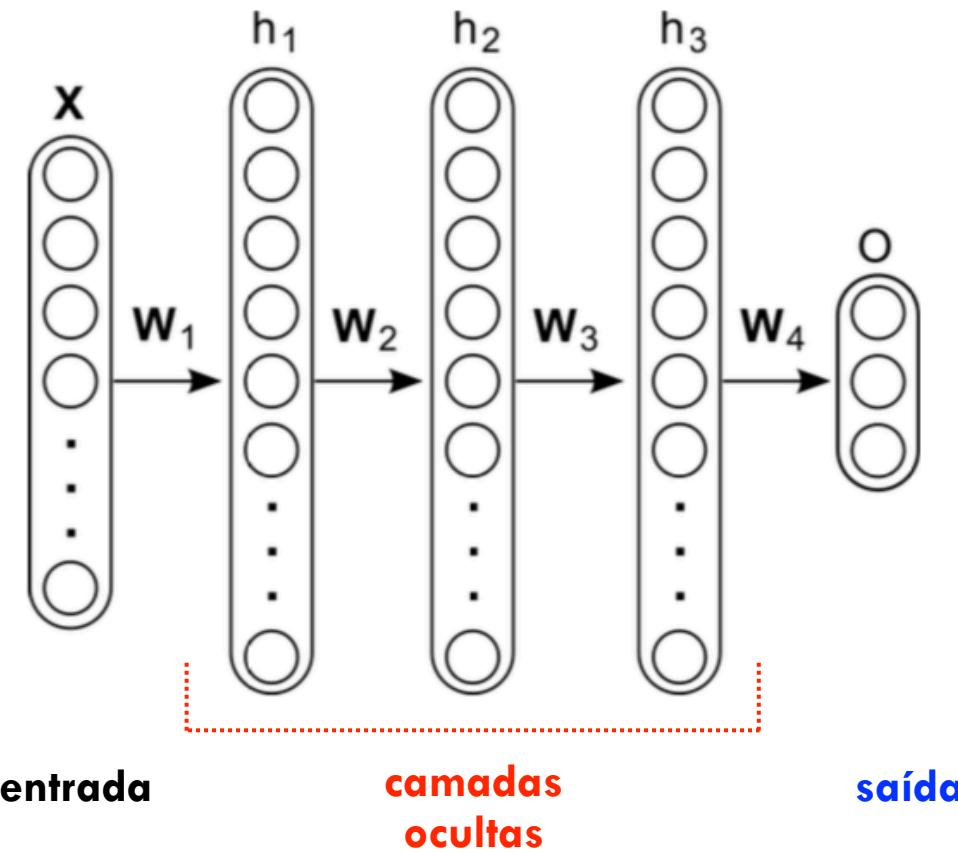
- **Problemas! Esse teorema não diz nada sobre:**
 - tempo de treinamento
 - facilidade de implementação
 - generalização

Introdução

- Em uma **MLP** podemos ter **L** camadas ocultas:

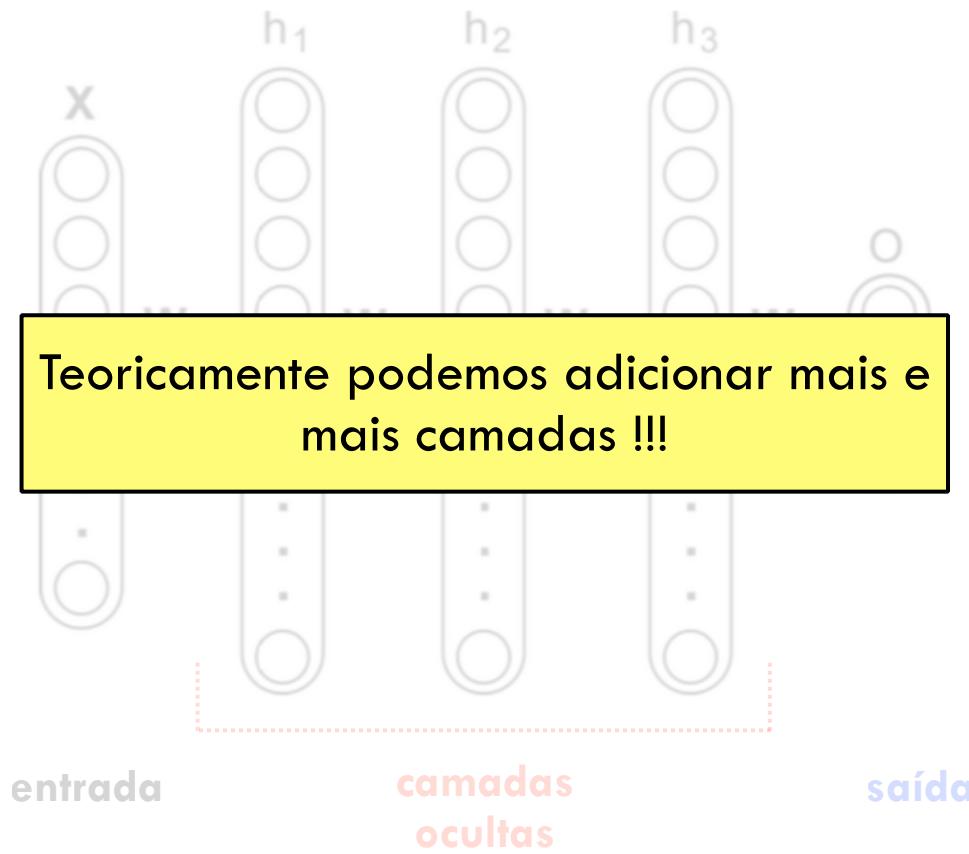
Introdução

- Em uma **MLP** podemos ter **L** camadas ocultas:



Introdução

- Em uma **MLP** podemos ter **L** camadas ocultas:



Introdução

- Deep Learning

Introdução

□ Deep Learning

- **Objetivo:** aprender modelos com múltiplas camadas de representação. Exs:
 - MLPs, DNNs, DBNs, Deep *Autoencoders*,
 - CNNs, RBMs, LSTM, etc

Introdução

□ Deep Learning

□ **Objetivo:** aprender modelos com múltiplas camadas de representação. Exs:

- MLPs, DNNs, DBNs, Deep Autoencoders,
- CNNs, RBMs, LSTM, etc

□ Cada **camada** corresponde a uma forma de representação

- unidade (neurônio) está associada a uma **característica** da entrada
- unidades podem ser ativadas simultaneamente

Introdução

- Algumas implicações de explorarmos arquiteturas profundas (com muitas camadas):

Introdução

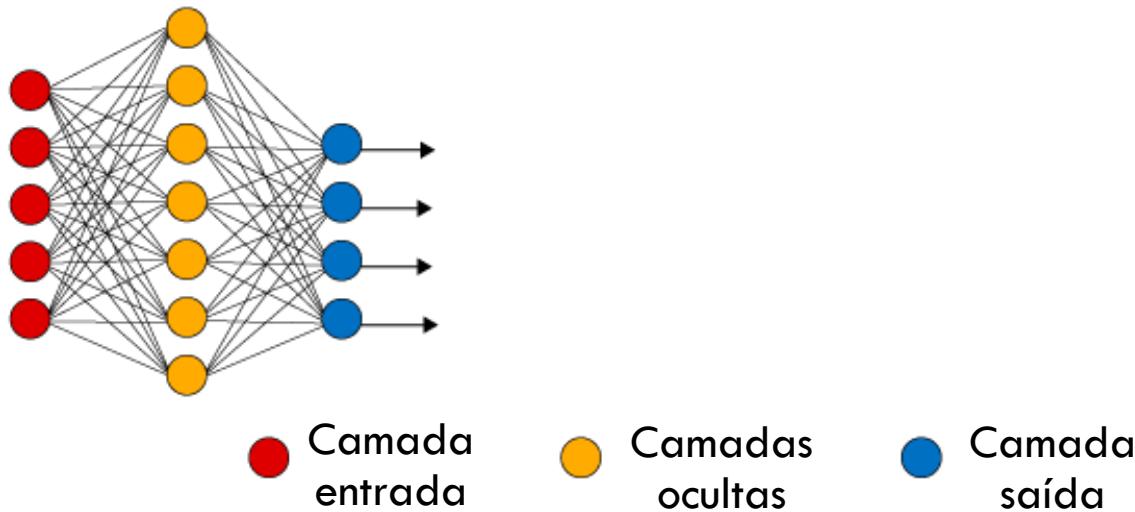
- Algumas implicações de explorarmos arquiteturas profundas (com muitas camadas):
 - podem representar certas funções de maneira mais compacta
 - há funções que podem ser representadas com uma única camada, porém requerem um número exponencial de neurônios
 - porém, pode ser necessário um número polinomial de neurônios, se pudermos aumentar o número de camadas

Fonte: [Larochelle, et. al. 2009] *Exploring Strategies for Training Deep Neural Networks*

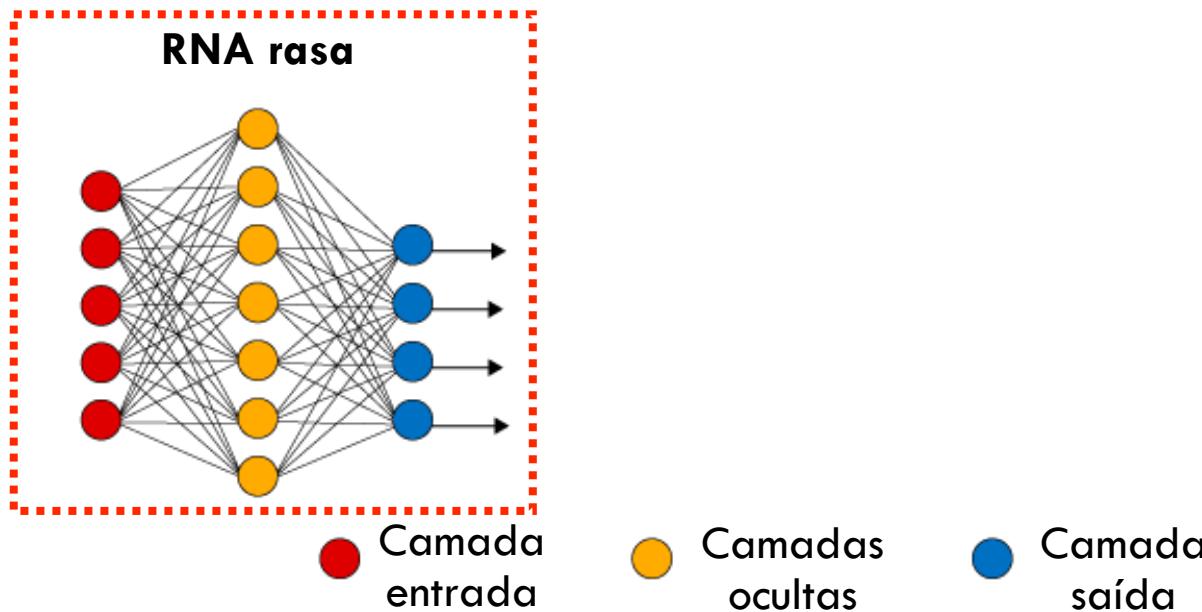
Roteiro

- 1** Introdução
- 2** Deep Learning (DL)
- 3** Treinamento de DL
- 4** Exemplos
- 5** Referências

Deep Learning

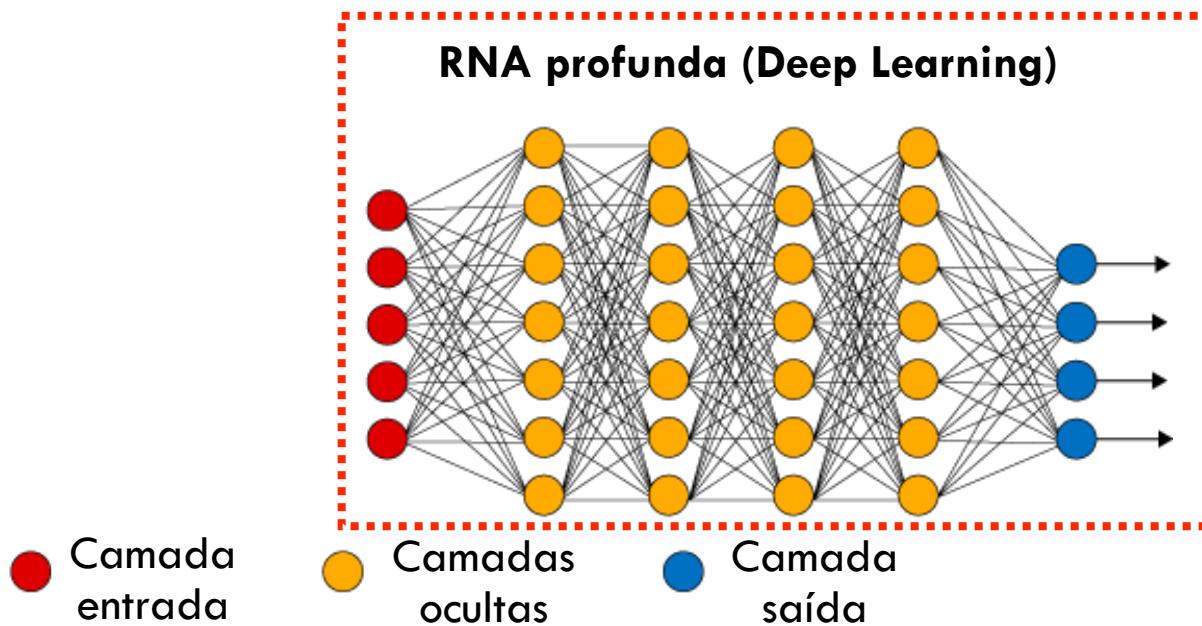


Deep Learning



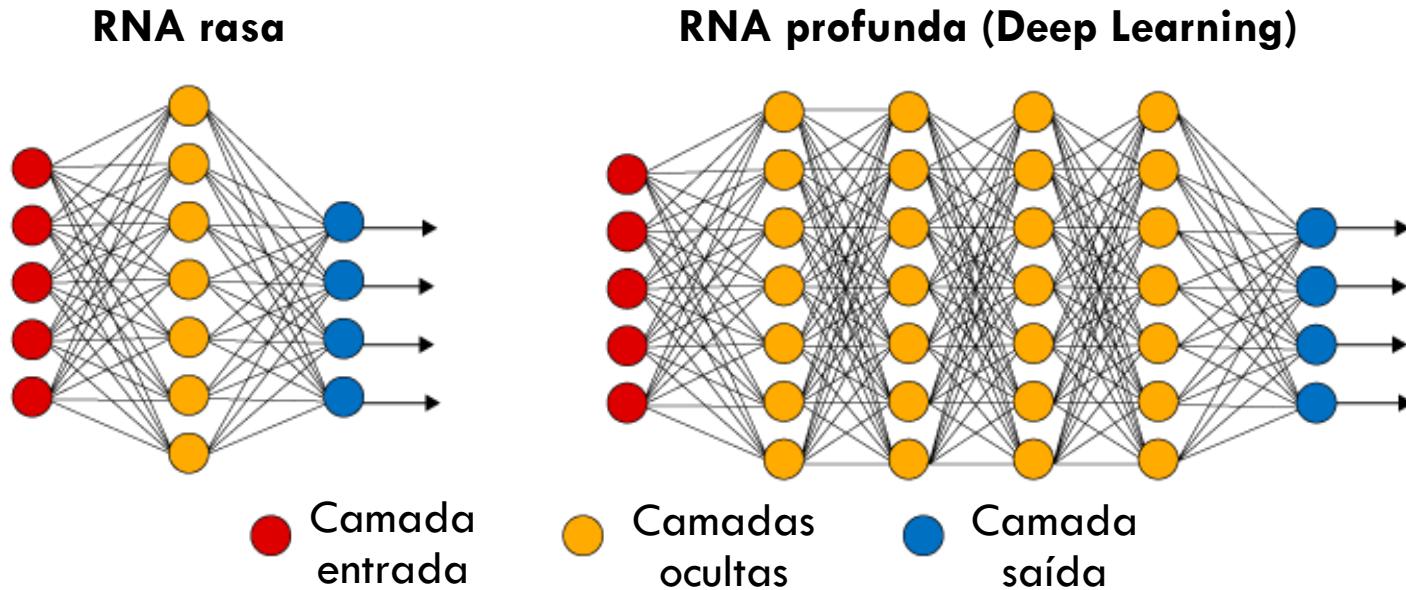
Até agora vimos
modelos "rasos"
(shallow)

Deep Learning



Agora exploraremos
mais camadas
(deep)

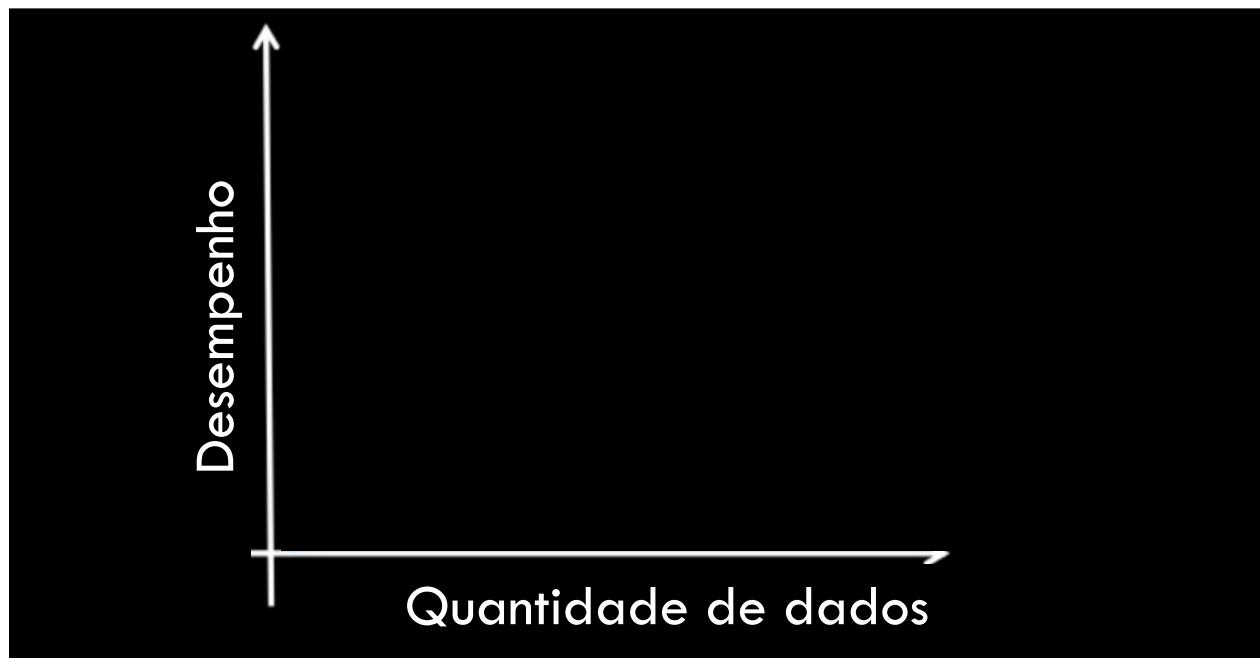
Deep Learning



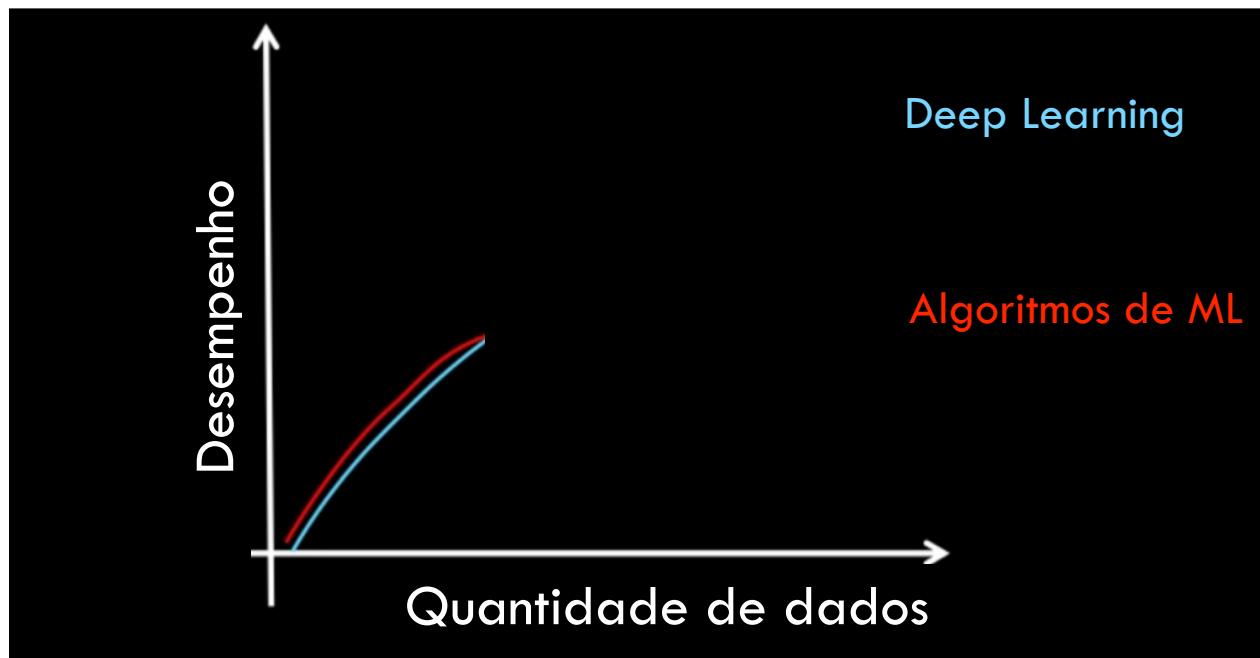
Deep Learning



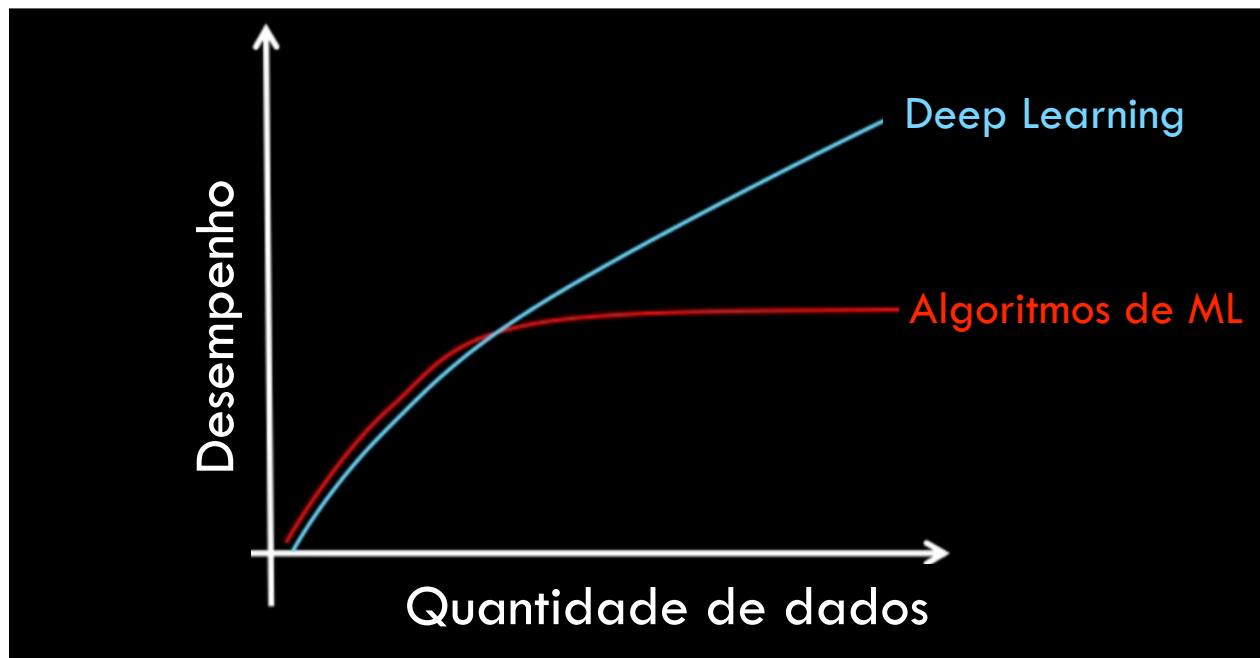
Deep Learning



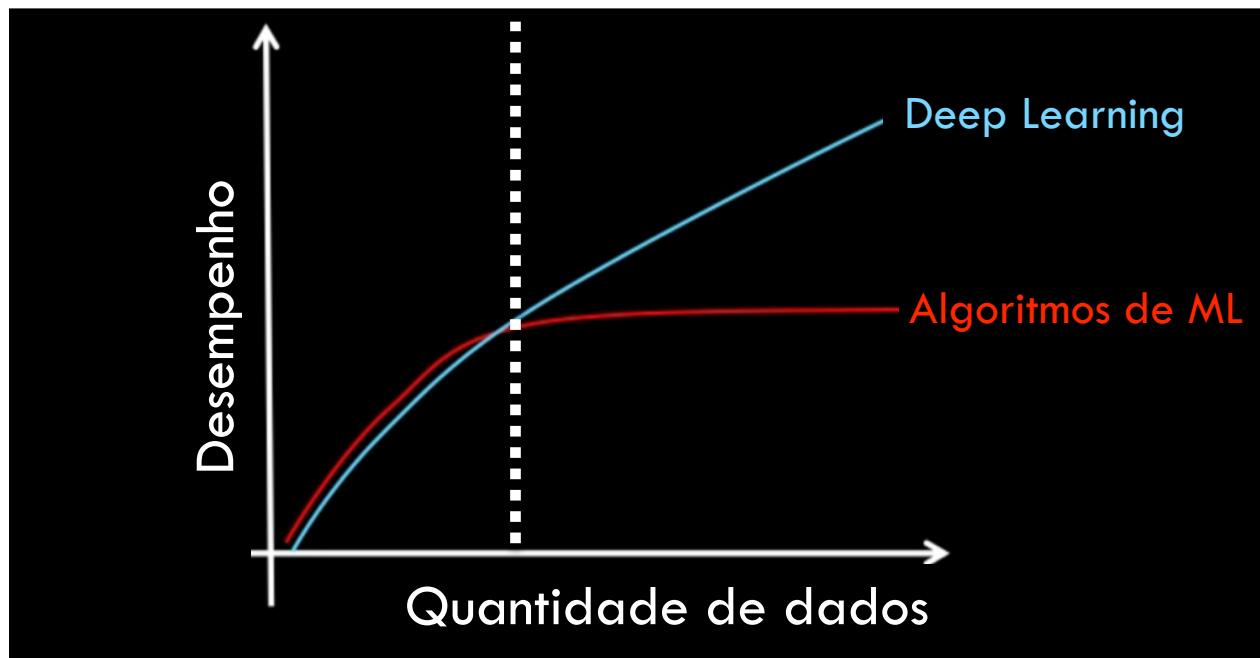
Deep Learning



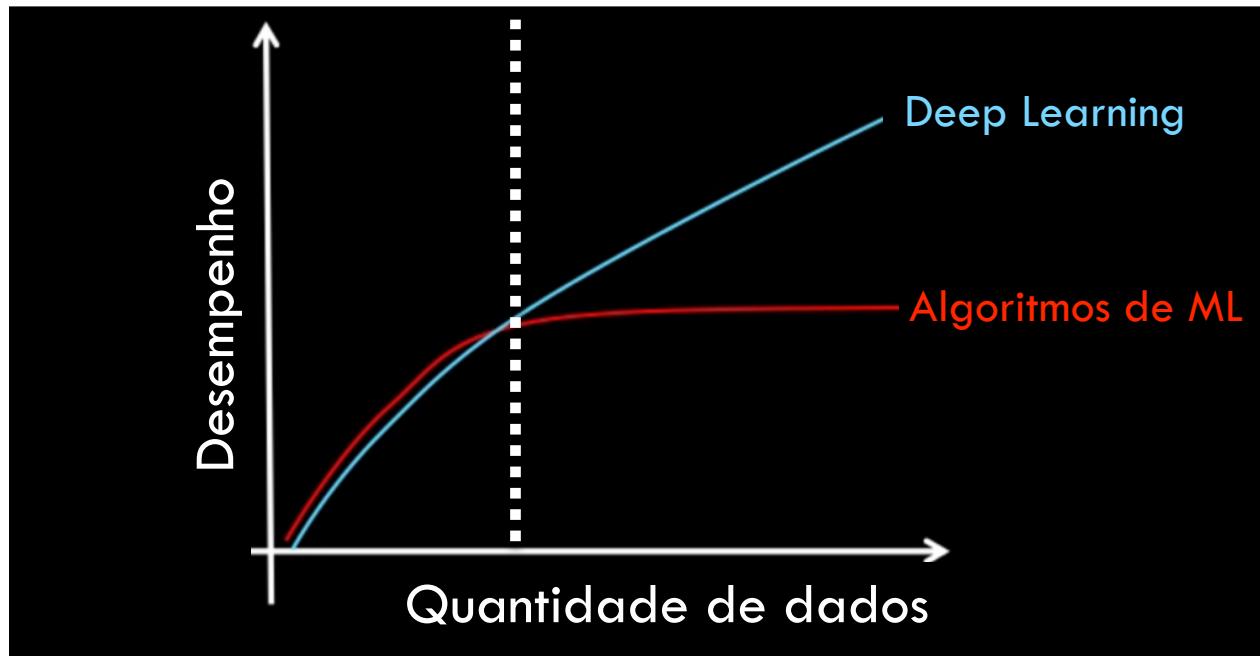
Deep Learning



Deep Learning



Deep Learning



Teoremas e experimentos indicam que DL tem um desempenho melhor quando manipula uma **quantidade maior de dados** ...

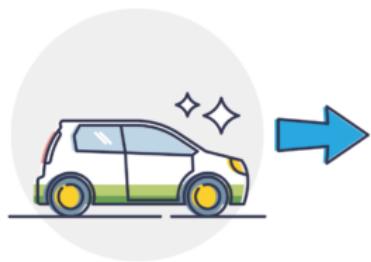
Deep Learning

Deep Learning

Aprendizado de Máquina

Deep Learning

Aprendizado de Máquina



Entrada

Deep Learning

Aprendizado de Máquina



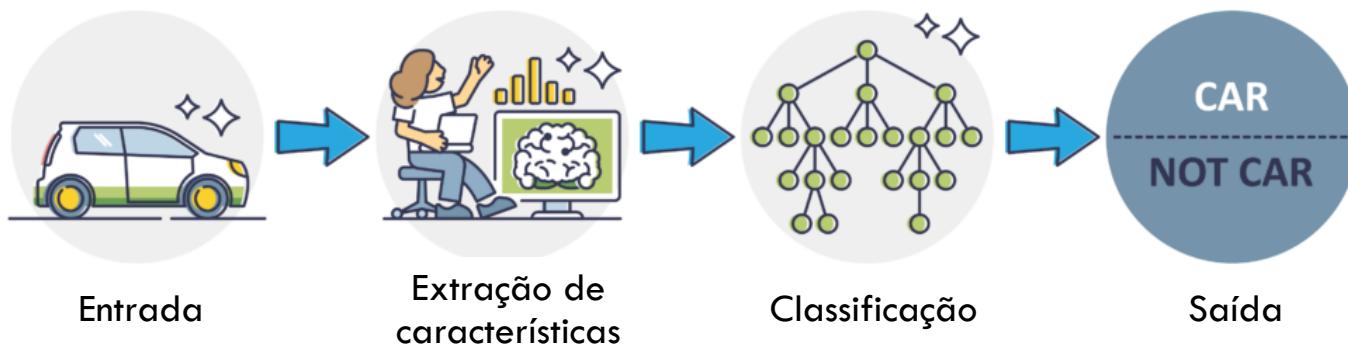
Deep Learning

Aprendizado de Máquina



Deep Learning

Aprendizado de Máquina

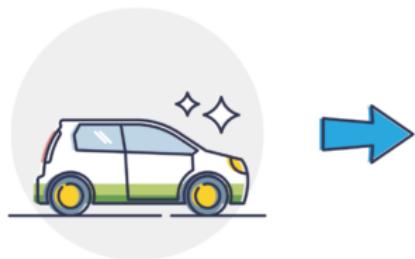


Deep Learning

Deep Learning

Deep Learning

Deep Learning



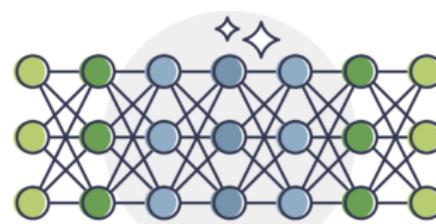
Entrada

Deep Learning

Deep Learning



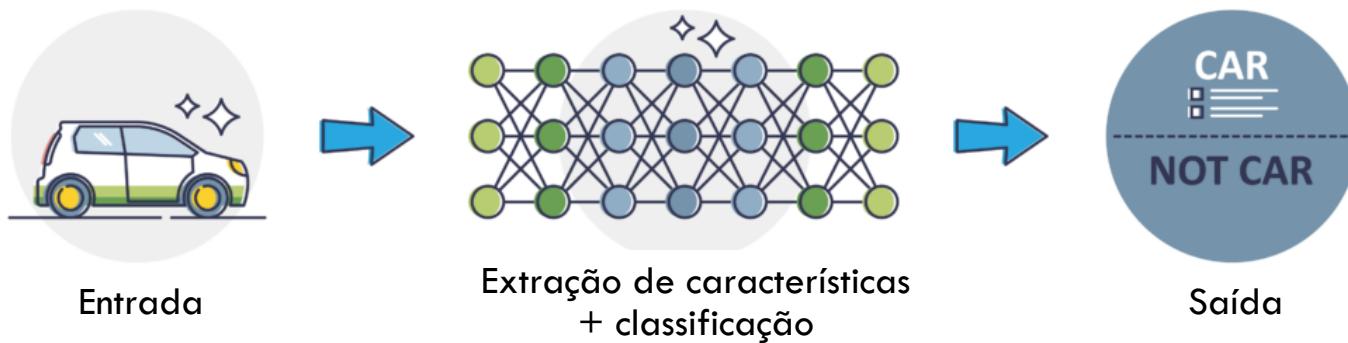
Entrada



Extração de características
+ classificação

Deep Learning

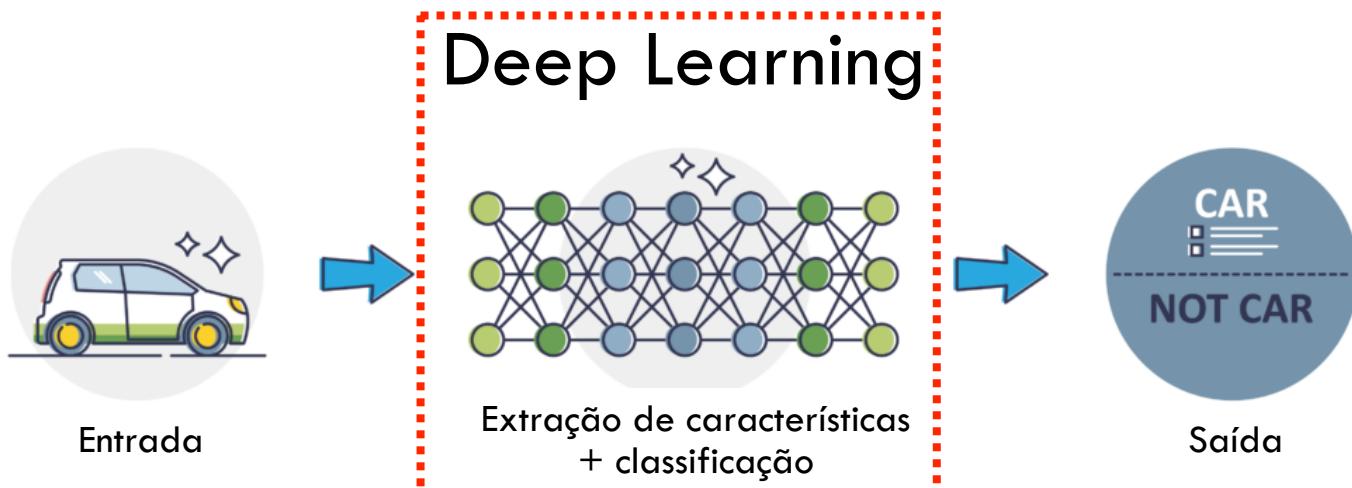
Deep Learning



Deep Learning

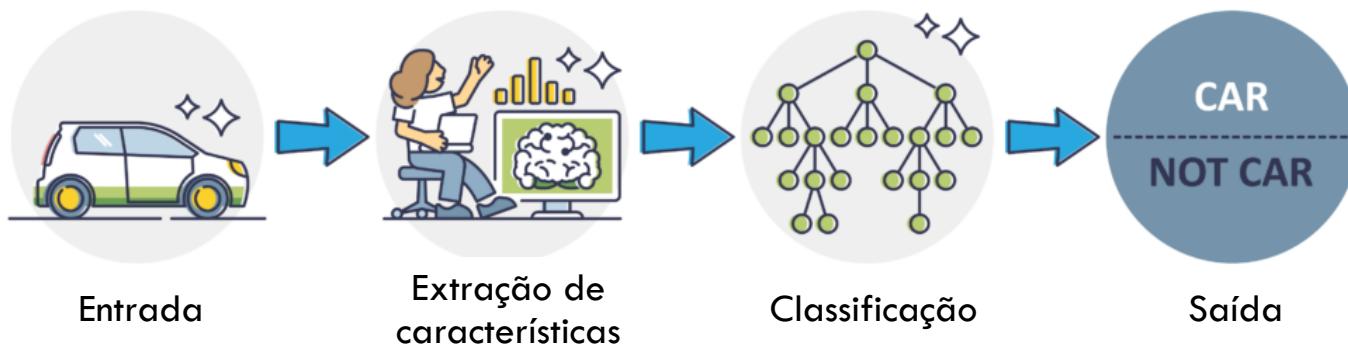
Tanto a extração das características como a geração do modelo preditivo fazem parte do algoritmo de DL.

Até por isso são comumente referenciados como algoritmos de caixa-preta (“*black-box*”)

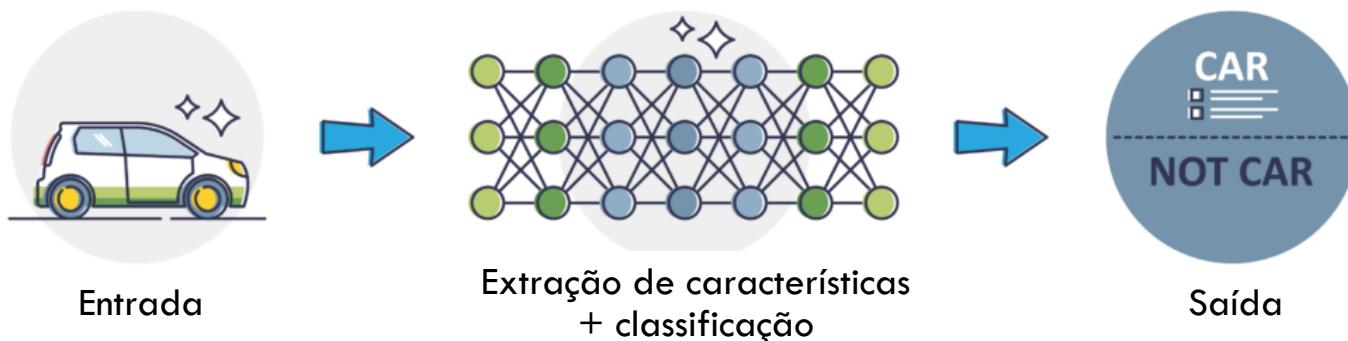


Deep Learning

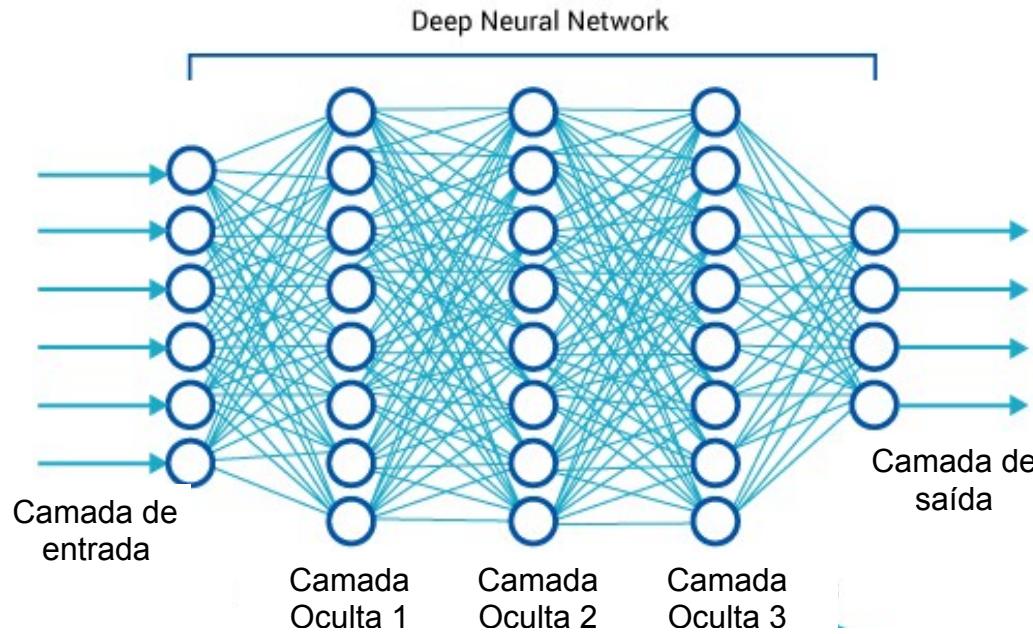
Aprendizado de Máquina



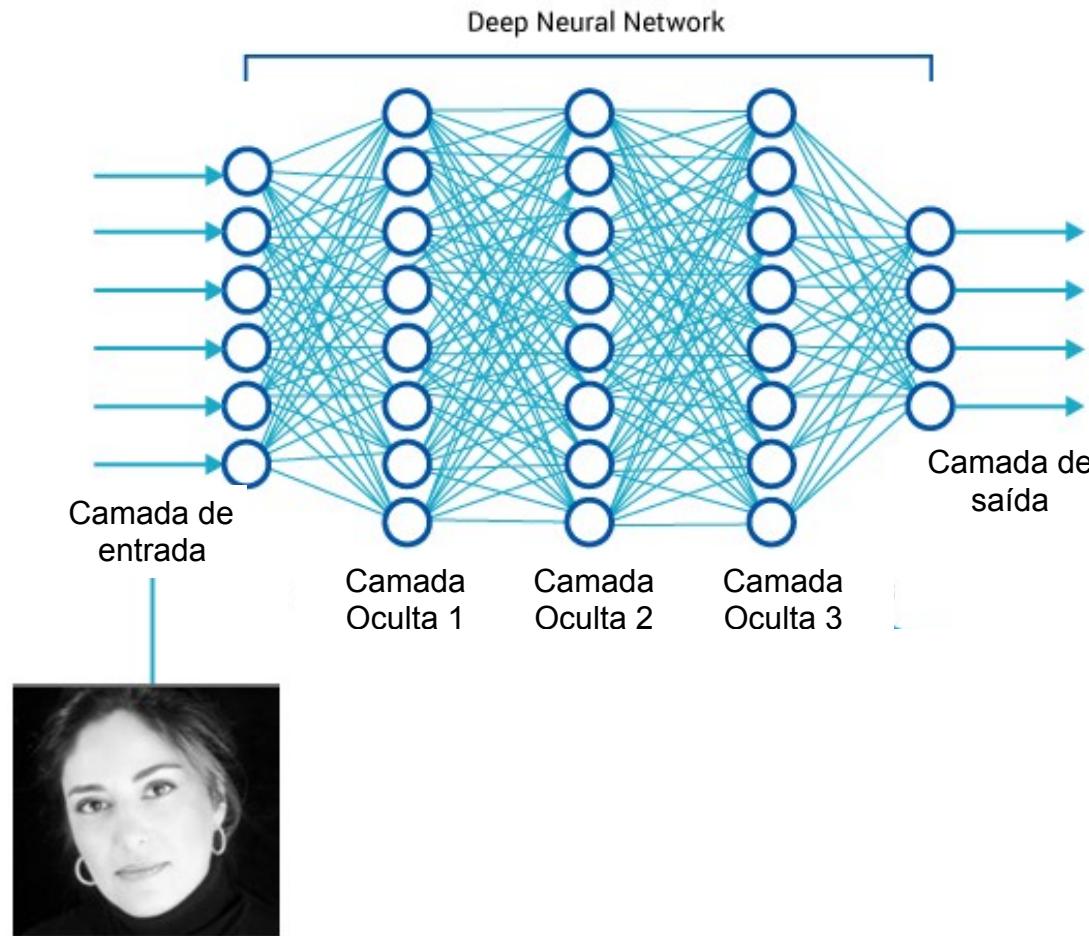
Deep Learning



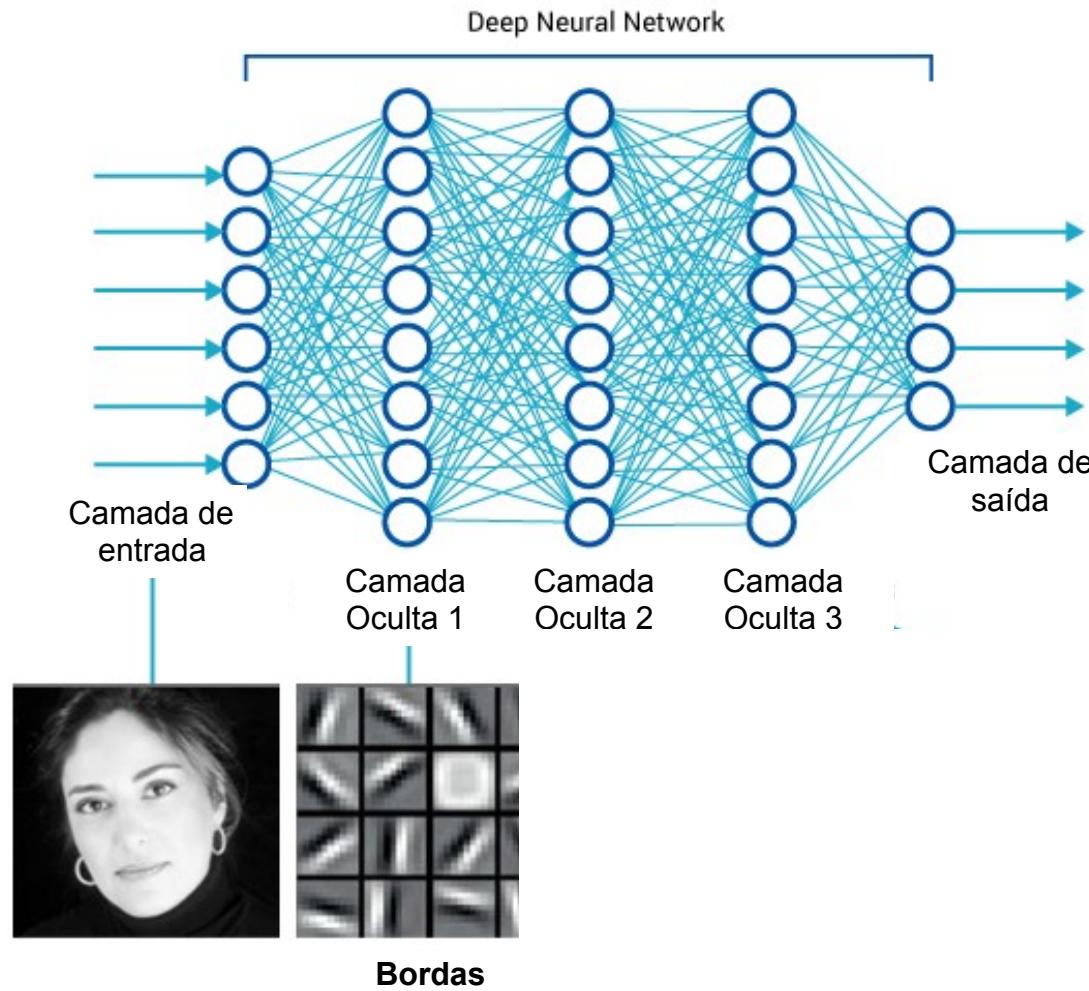
Deep Learning



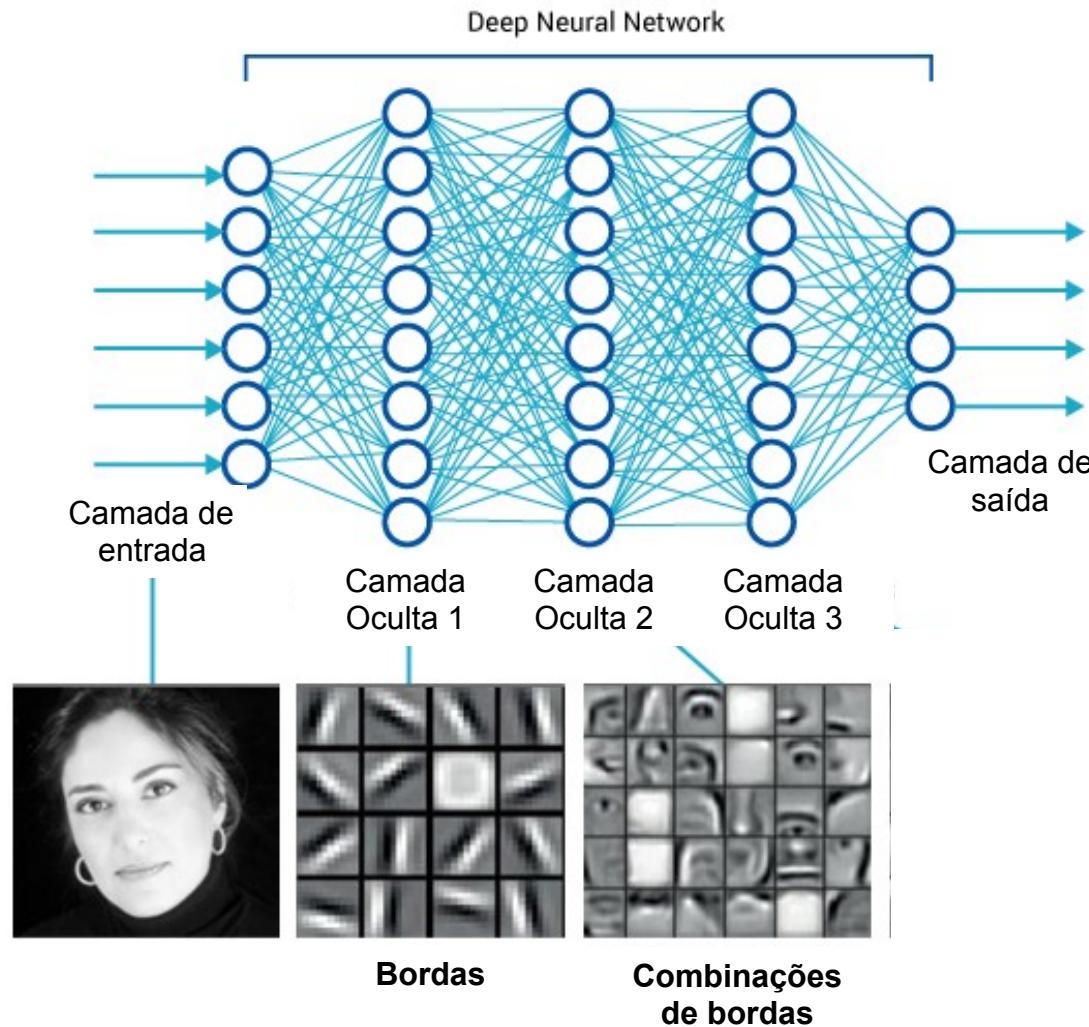
Deep Learning



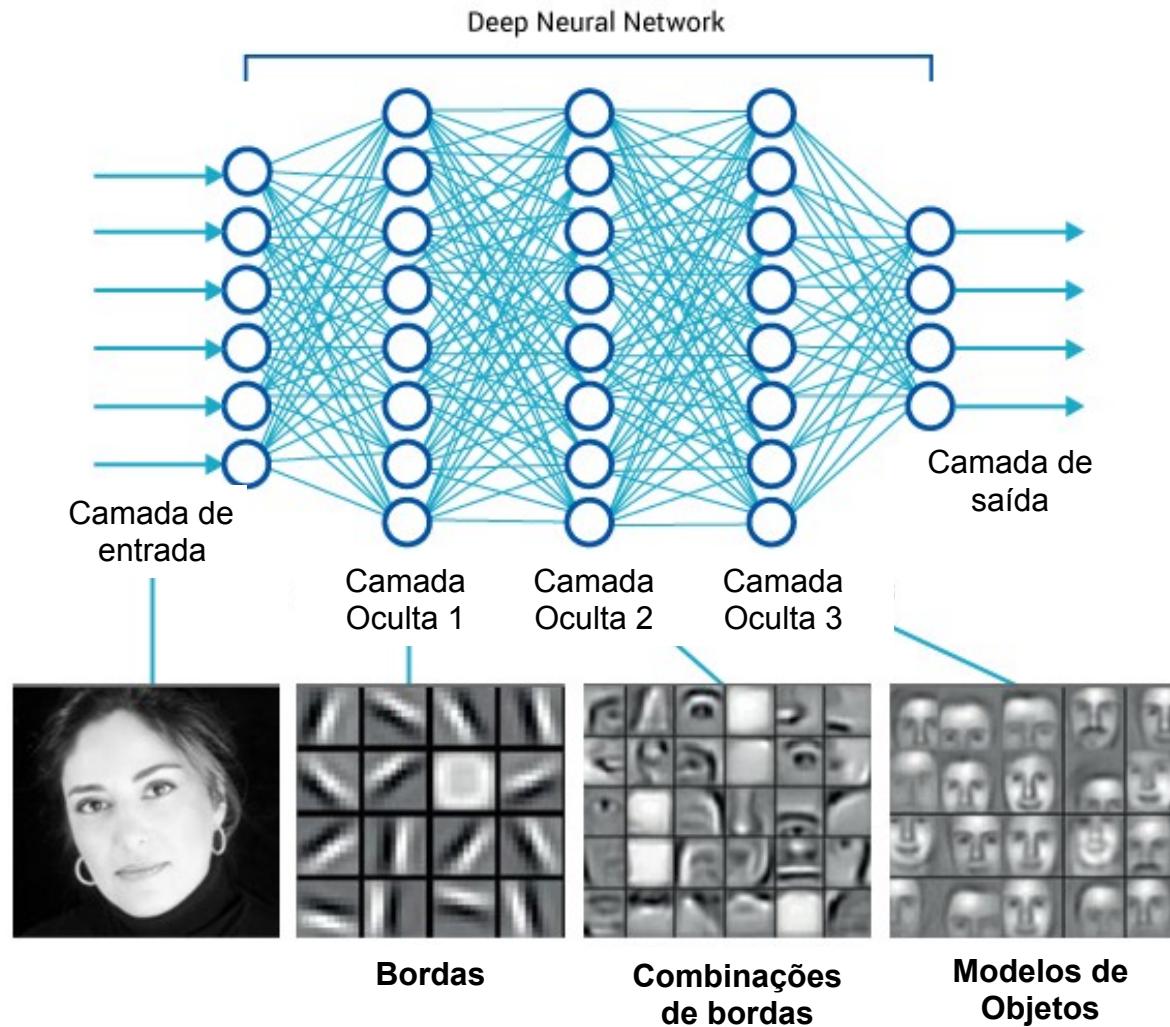
Deep Learning



Deep Learning



Deep Learning



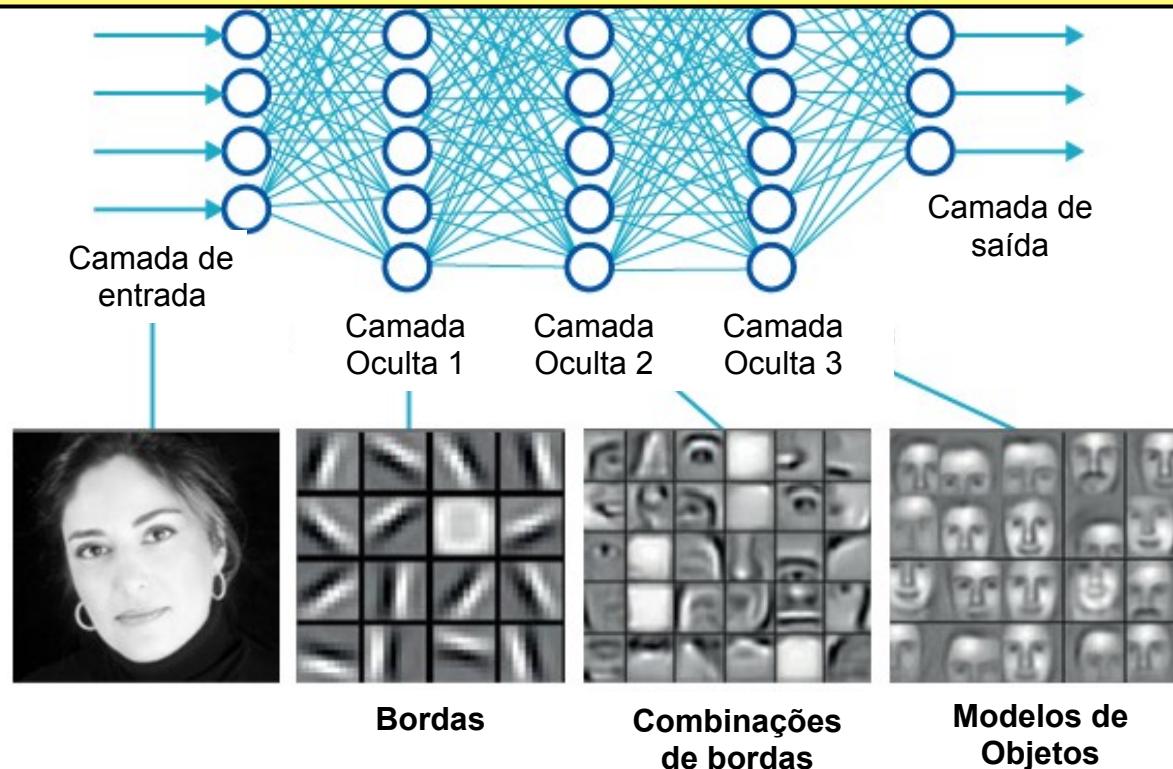
Deep Learning

1a camada extrai **características simples** (bordas)

2a camada combina essas bordas em **formatos mais complexos** (contornos)

3 camada tenta criar padrões **ainda mais complexos** (objetos, etc)

....

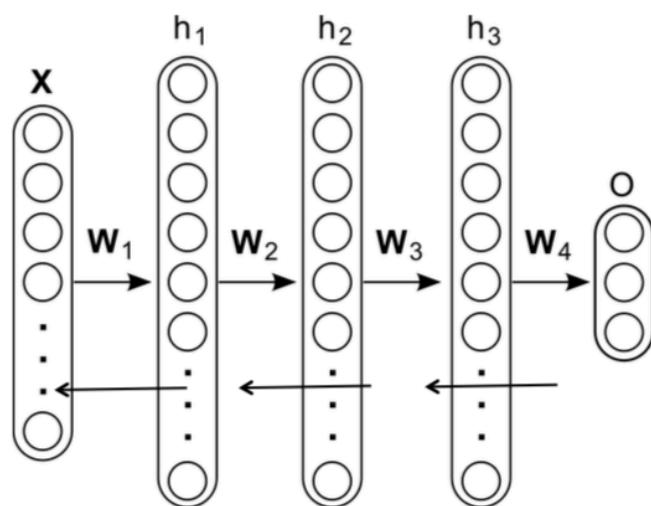


Roteiro

- 1** Introdução
- 2** Deep Learning (DL)
- 3** Treinamento de DL
- 4** Exemplos
- 5** Referências

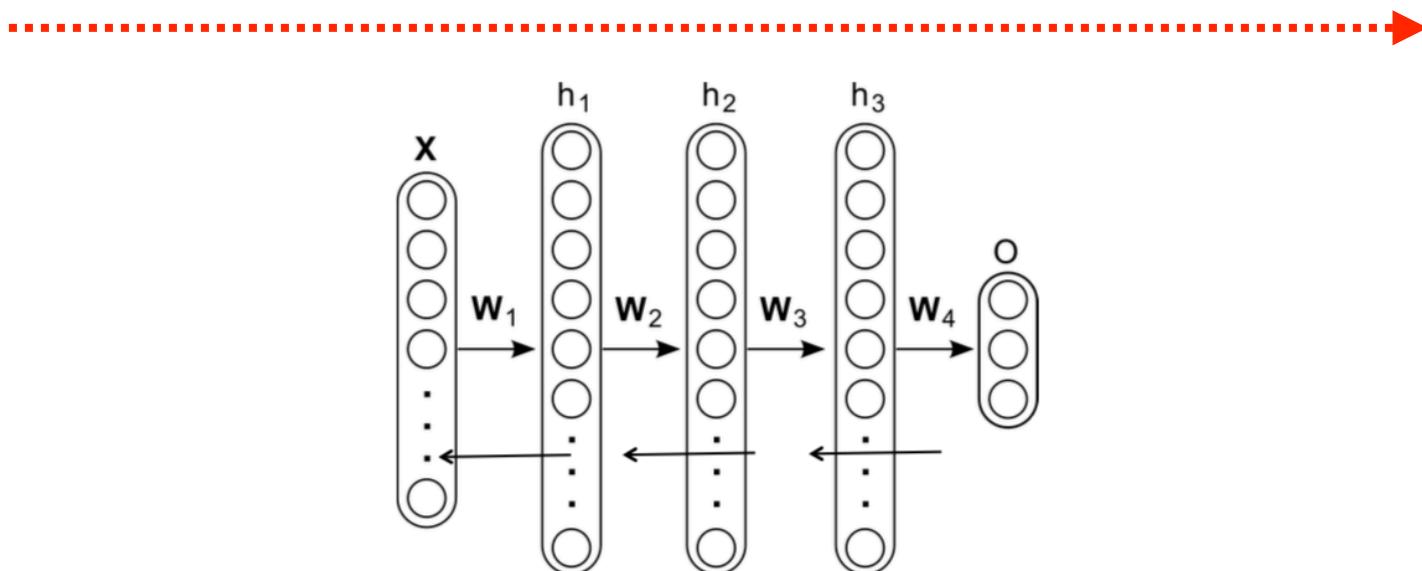
Treinamento

Treinamento

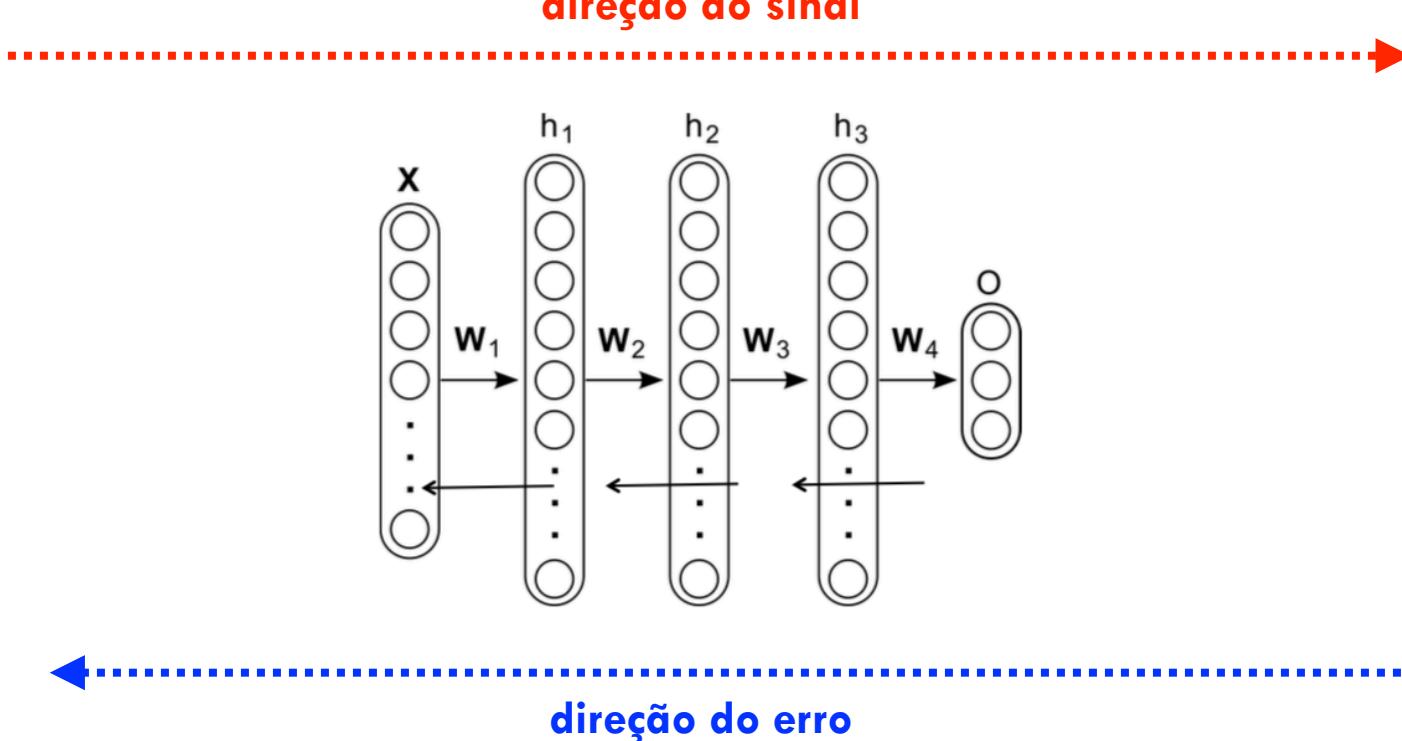


Treinamento

direção do sinal



Treinamento

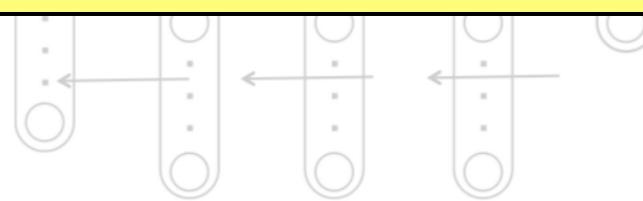


Treinamento

direção do sinal



Treinamento via Backpropagation!



direção do erro

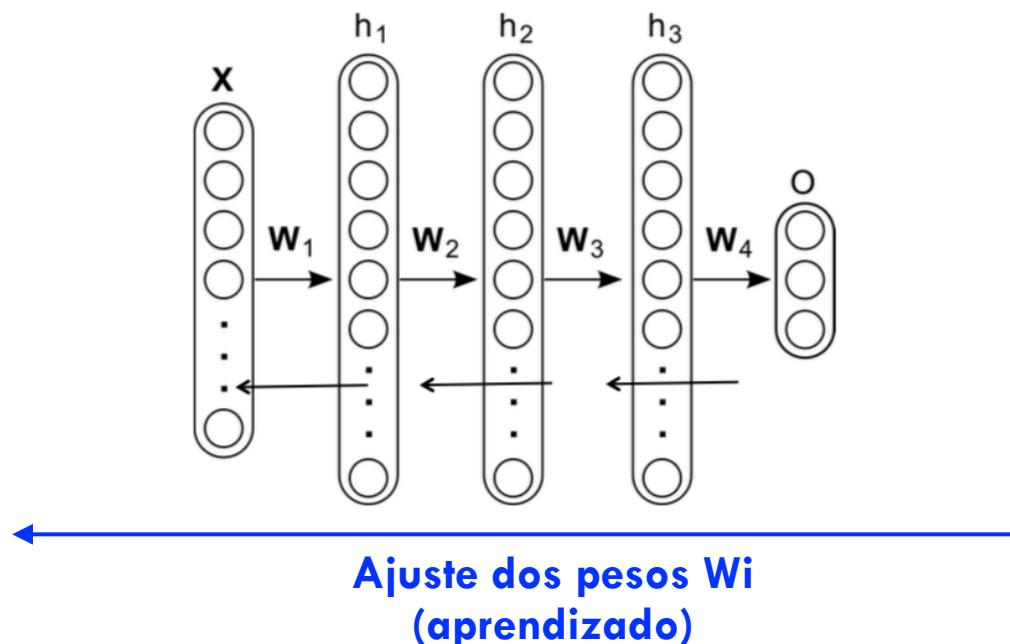
Problemas

Problemas

- *Underfitting* → Gradientes desaparecem ($\Delta w_i = 0$)

Problemas

- *Underfitting* → Gradientes desaparecem ($\Delta w_i = 0$)



Problemas

- **Overfitting** → muitos parâmetros, pois explora um espaço muito maior de funções

Problemas

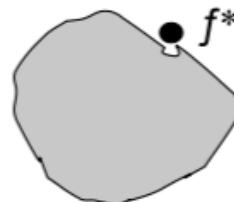
- **Overfitting** → muitos parâmetros, pois explora um espaço muito maior de funções



• f^*

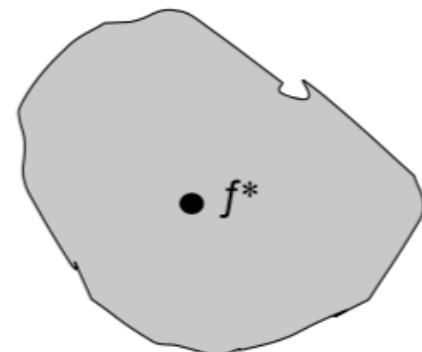
Possível f

Baixa variância
Bias alto



Possível f

Bom tradeoff



• f^*

Possível f

Alta variância
Bias baixo

Soluções

Soluções

- Usar métodos de regularização melhores:

- 1 treinamento não-supervisionado
- 2 treinamento com *dropout* estocástico
- 3 utilização de outros algoritmos de aprendizado
(estimadores)

1. Pré-treino

- Inicializar as camadas escondidas → aprendizado não-supervisionado
 - força a rede a aprender a estrutura da distribuição dos dados de entrada
 - encoraja as camadas escondidas a codificar essa estrutura

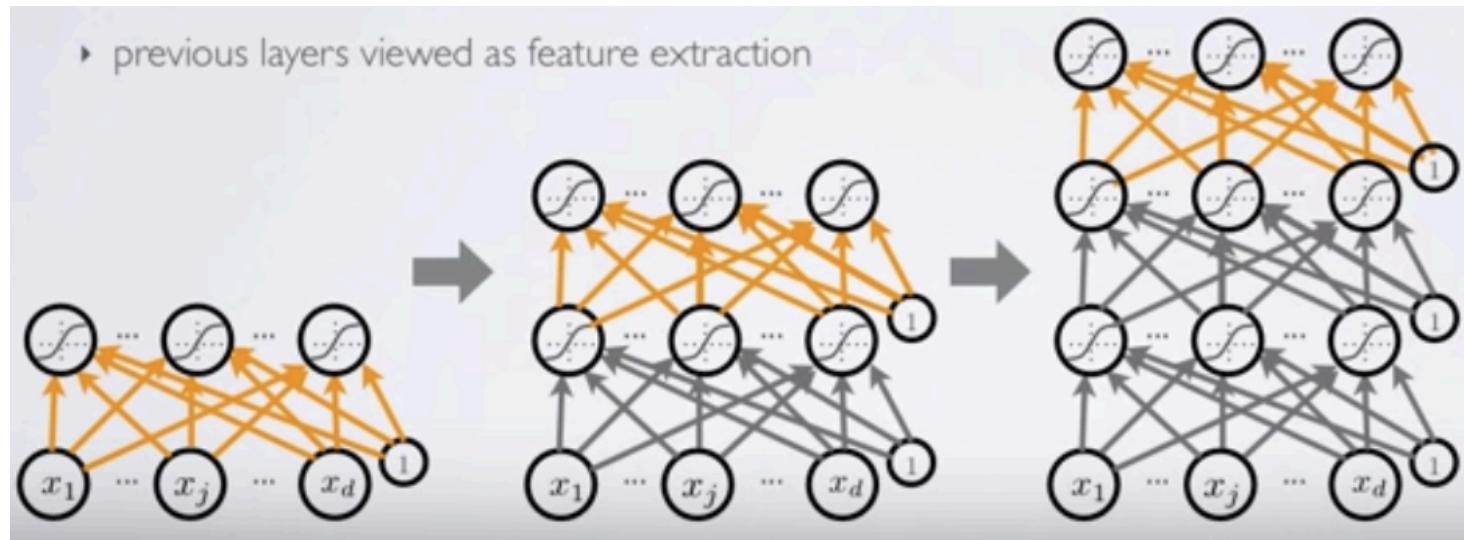
Pré-treino: inicializar os parâmetros de uma região tal que haja menos *overfitting*

1. Pré-treino

- **Estratégia:** procedimento guloso camada a camada
 - treina uma camada por vez, da primeira até a última, utilizando aprendizado não-supervisionado
 - Cada camada ajusta os parâmetros das camadas anteriores
 - Camadas anteriores → extractores de características

1. Pré-treino

- **Estratégia:** procedimento guloso camada a camada



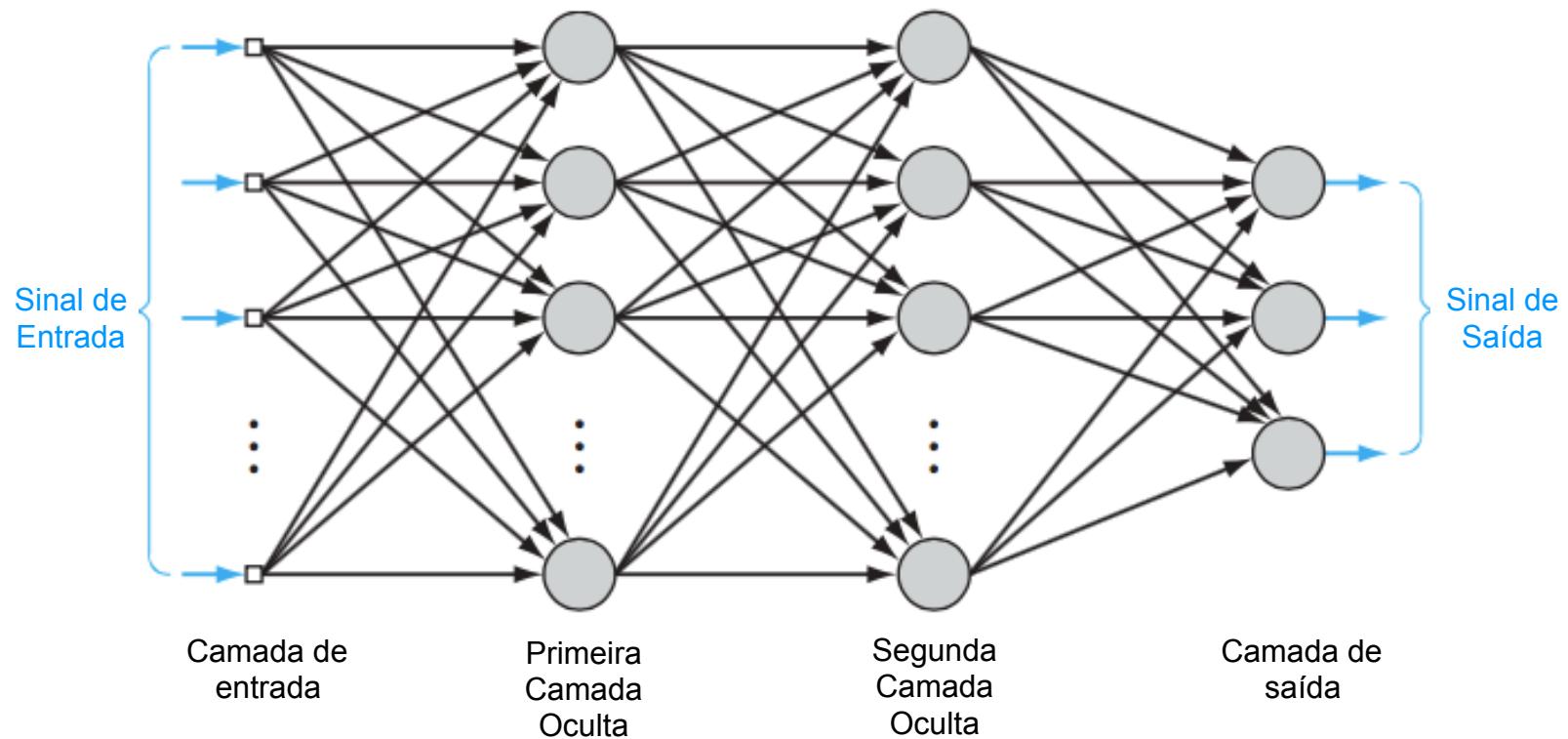
Soluções

- Usar métodos de regularização melhores:
 1. treinamento não-supervisionado
 - 2** treinamento com *dropout* estocástico
 3. utilização de outros algoritmos de aprendizado (estimadores)

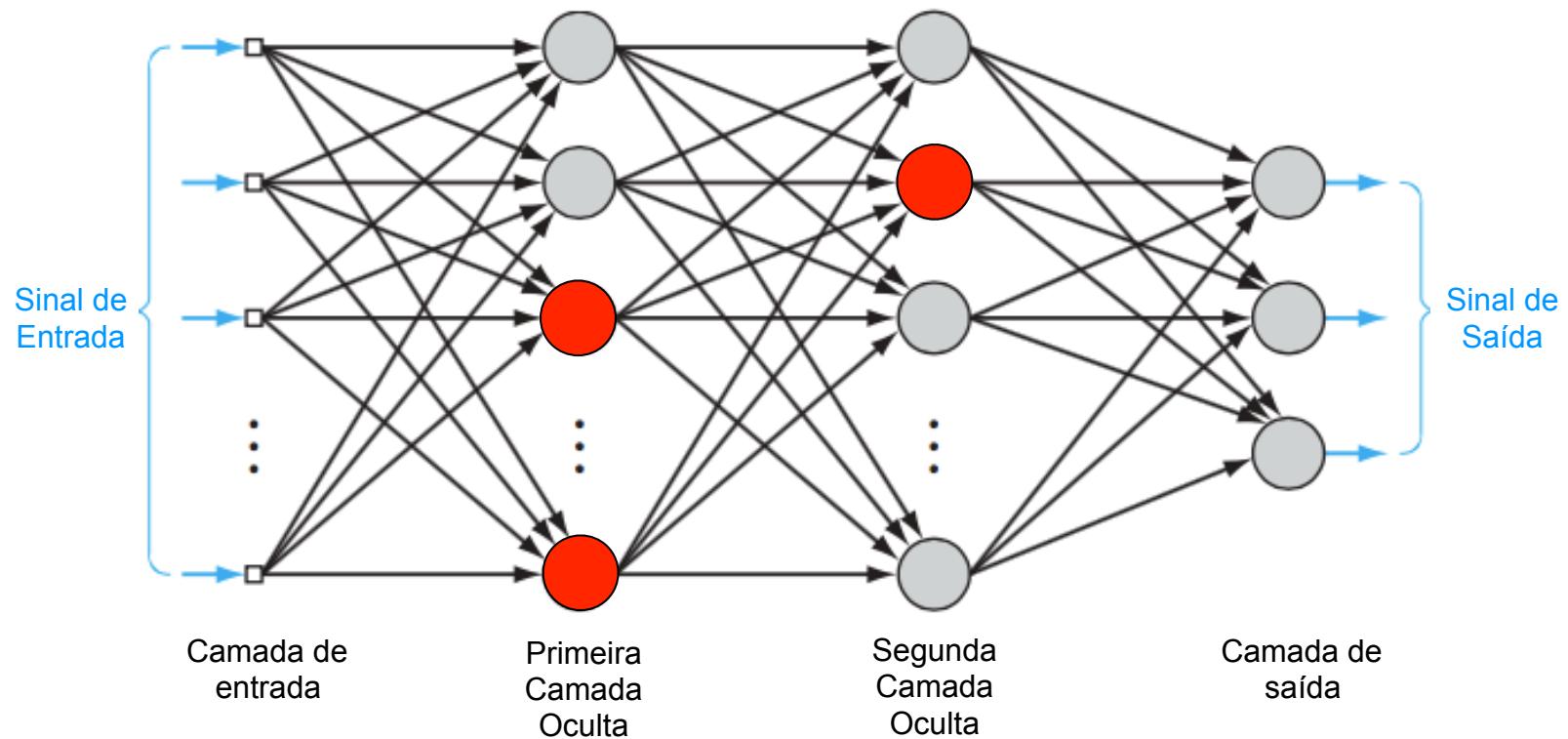
2. Dropout

- **Ideia:** remover aleatoriamente neurônios das camadas ocultas
 - cada neurônio escondido é zerado com probabilidade igual a p ($p = 0.5$)
 - Resultado
 - neurônios escondidos não podem se coadaptar a outros neurônios escondidos
 - cada neurônio escondido é迫使ado a extrair características mais gerais

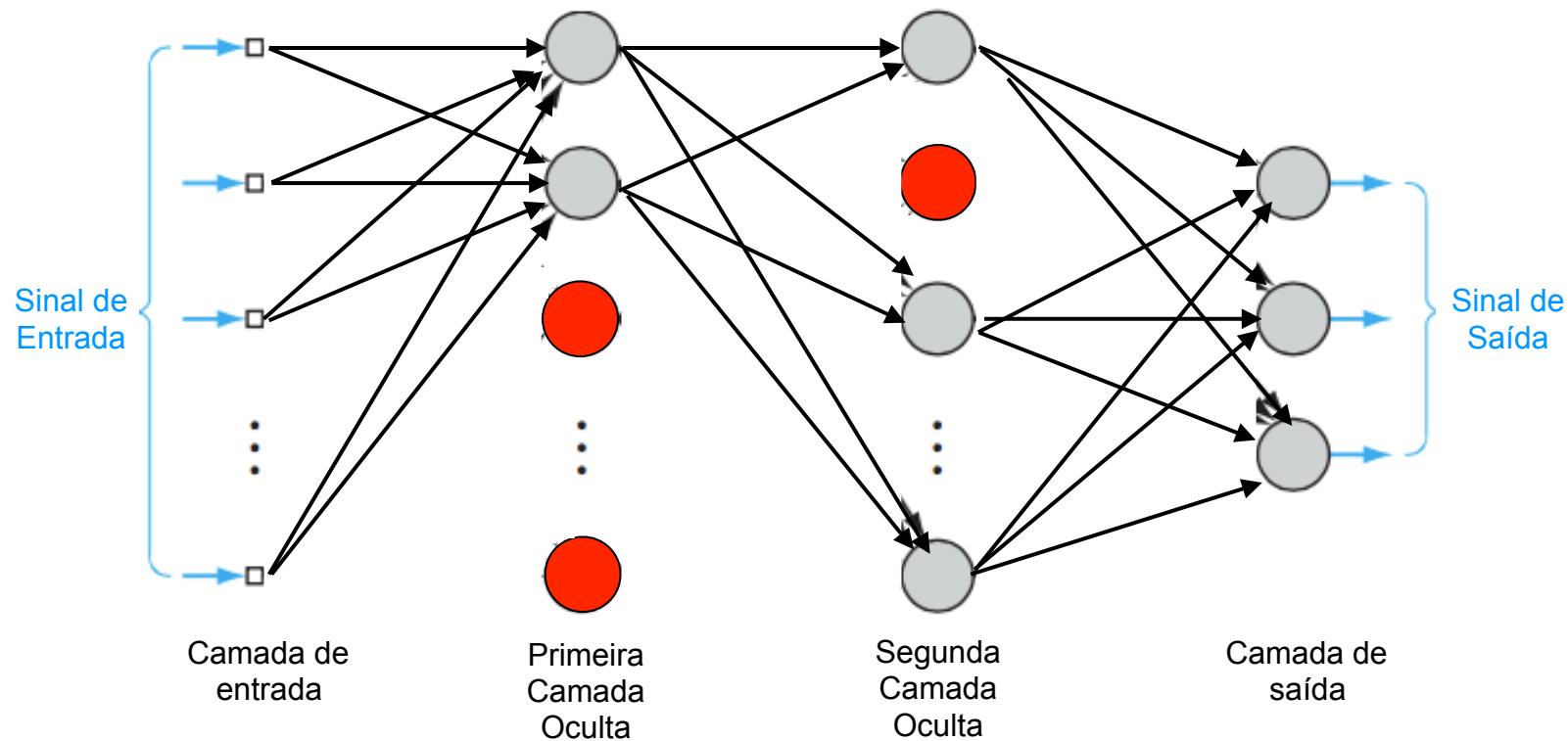
2. Dropout



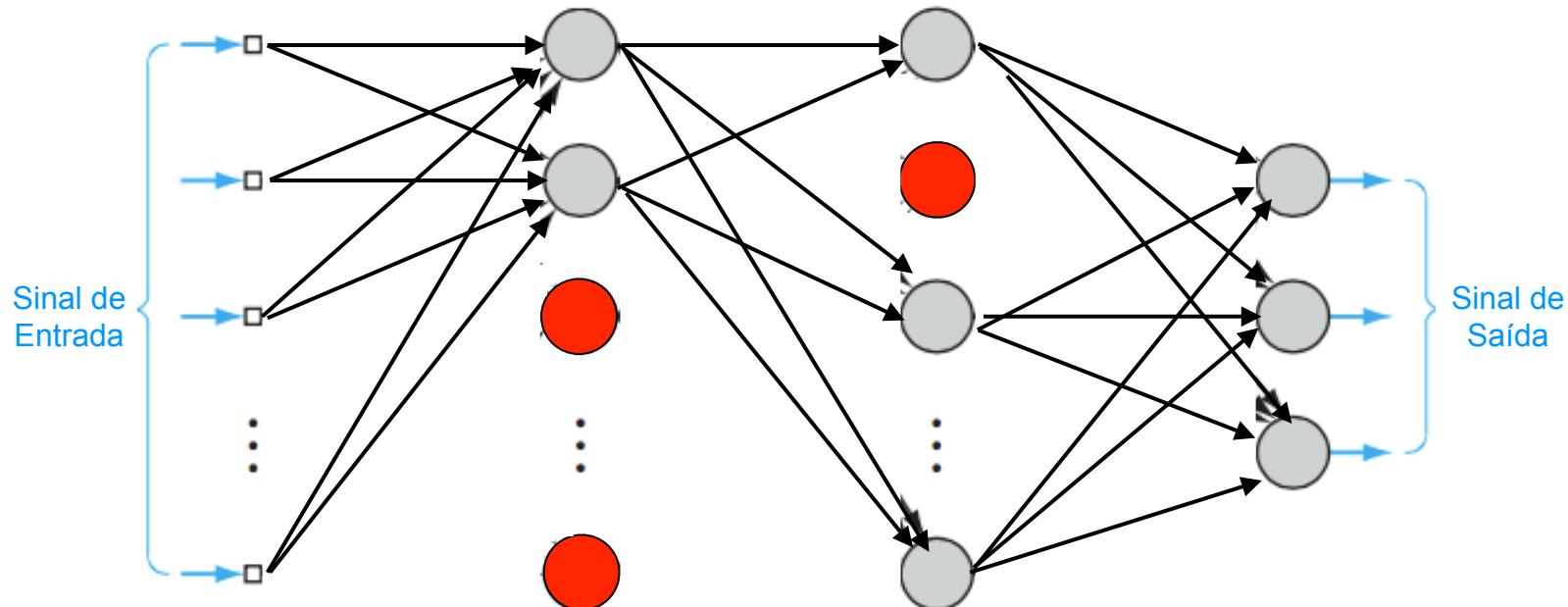
2. Dropout



2. Dropout



2. Dropout



Neurônios selecionados aleatoriamente (vermelho) são **removidos** da rede

Soluções

- Usar métodos de regularização melhores:
 1. treinamento não-supervisionado
 2. treinamento com *dropout* estocástico
 3. utilização de outros algoritmos de aprendizado (estimadores)

3. Outros estimadores

- **Adagrad: Adaptive Subgradient (Duchi et al, 2011)**
 - ajusta a taxa de aprendizado para cada parâmetro, de acordo com os gradientes das iterações anteriores
 - Divide o gradiente atual na atualização pela soma dos gradientes anteriores
 - Quanto mais atualizações um parâmetro (peso) receber, menor é a atualização

3. Outros estimadores

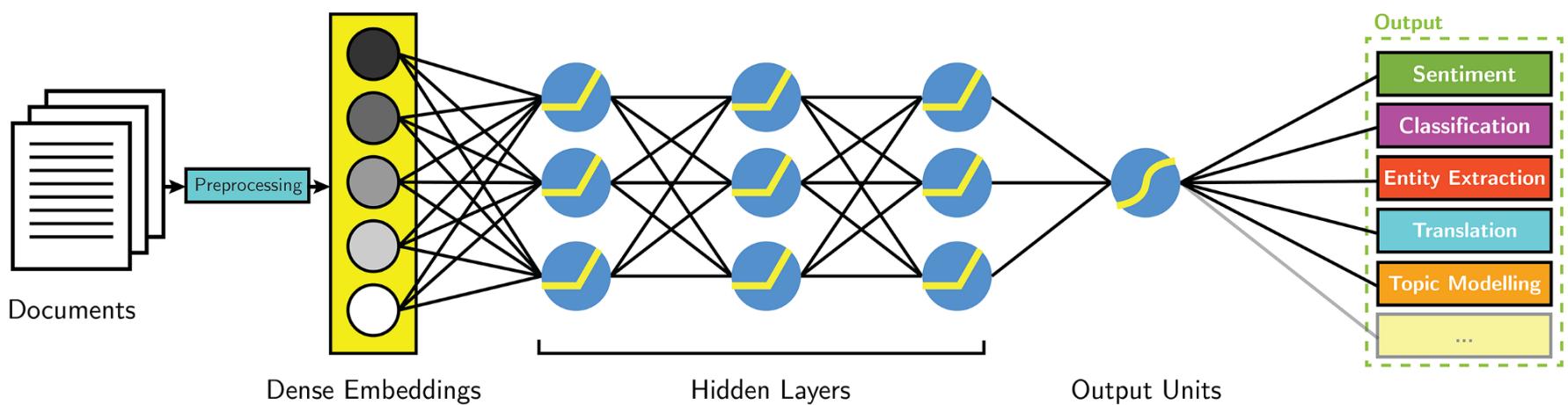
- **AdaDelta (Zeiler, 2012)**
 - extensão do Adagrad, utiliza uma janela de gradientes ao invés da acumulação de todos os gradientes anteriores
 - aprendizado continua mais do que no Adagrad, mesmo quando muitas atualizações já foram aplicadas
- **Adam (Kingma & Ba, 2014)**
 - Combinação do Adagrad e do RMSProp
 - faz decremento dos ***momentums*** de primeira e segunda ordem

Roteiro

- 1** Introdução
- 2** Deep Learning (DL)
- 3** Treinamento de DL
- 4** Exemplos
- 5** Referências

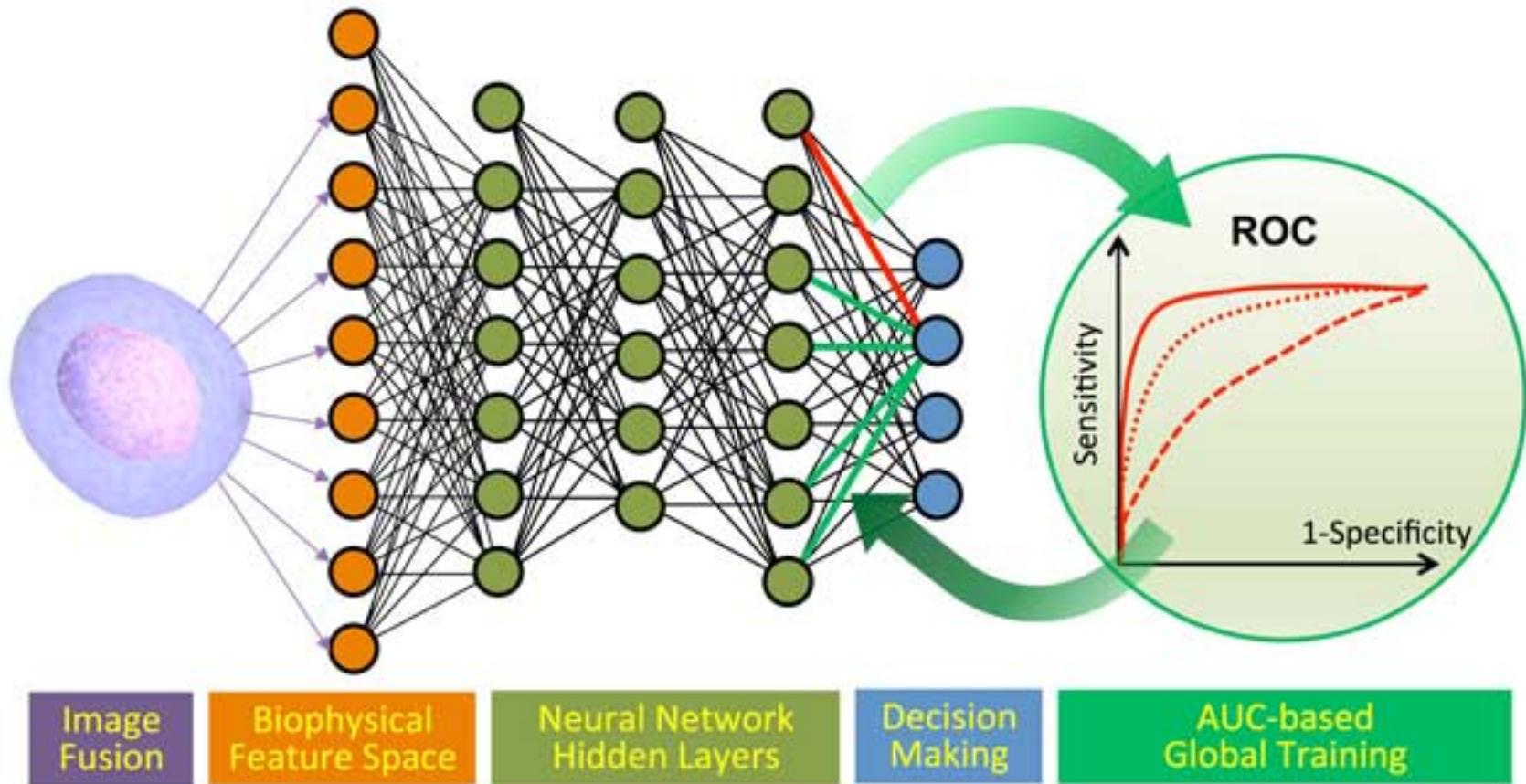
Exemplos de aplicações de DL

- Detecção de sentimentos:



Exemplos de aplicações de DL

- Diagnóstico de cancer:



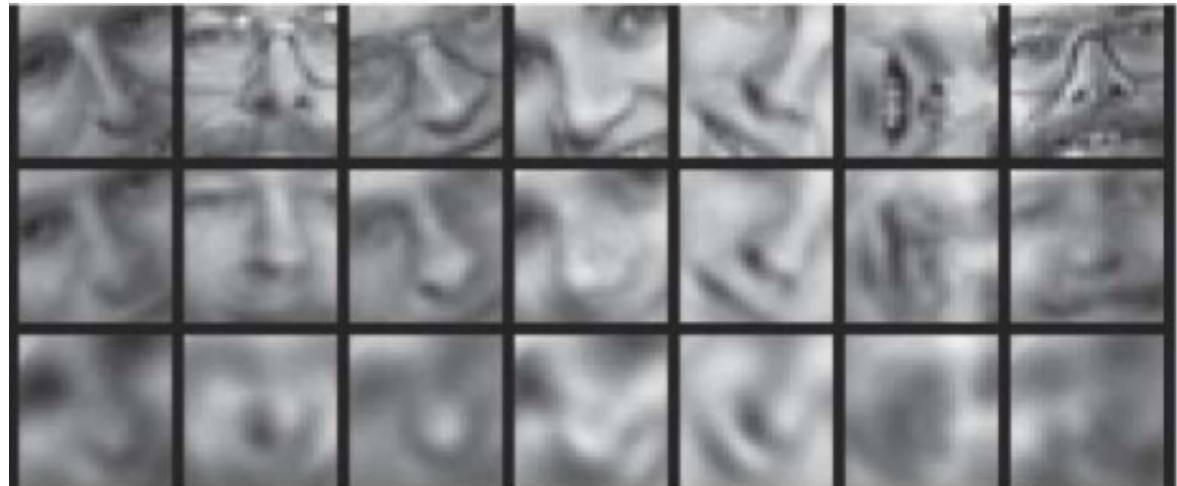
Exemplos de aplicações de DL

- Redução de dimensionalidade (Autoencoders)

Reducing the Dimensionality of Data with Neural Networks

G. E. Hinton* and R. R. Salakhutdinov

dados originais

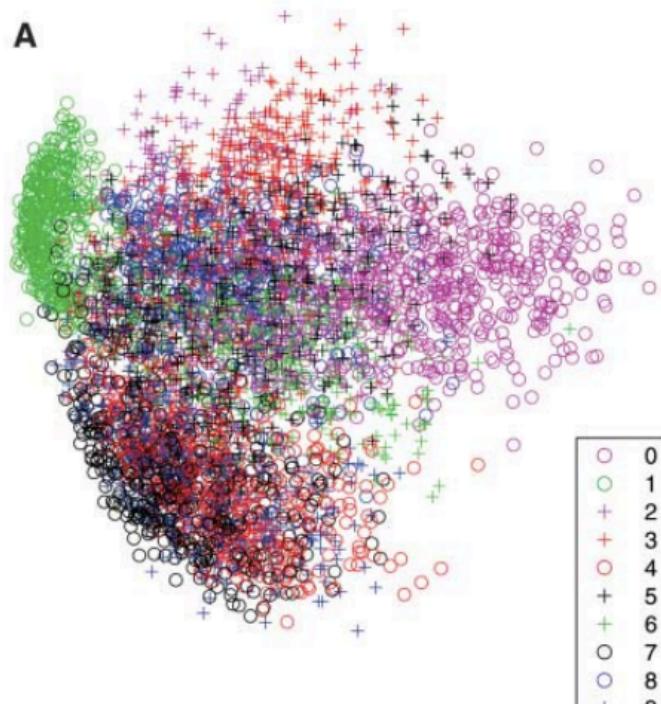


deep autoencoder

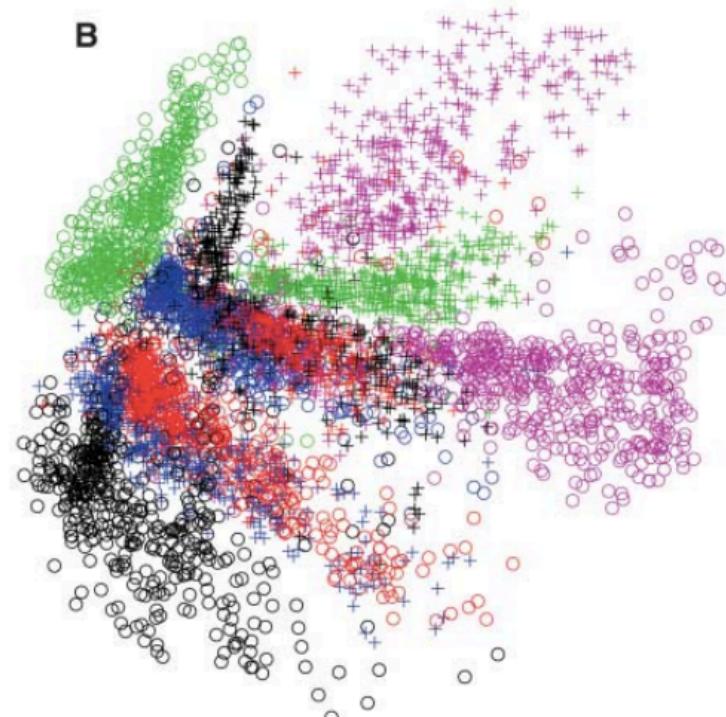
PCA

Exemplos de aplicações de DL

- Reconhecimento de dígitos por imagens



PCA

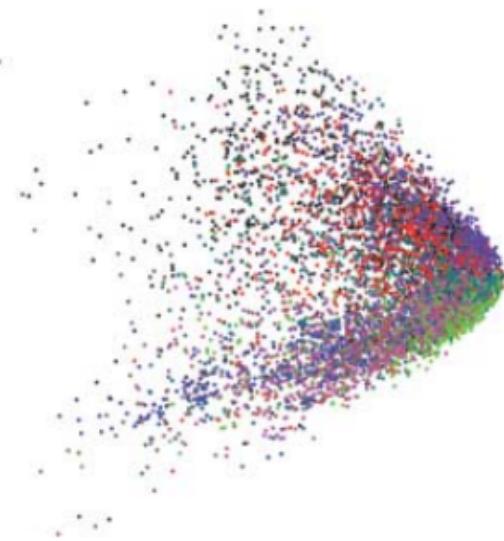


Autoencoder

Exemplos de aplicações de DL

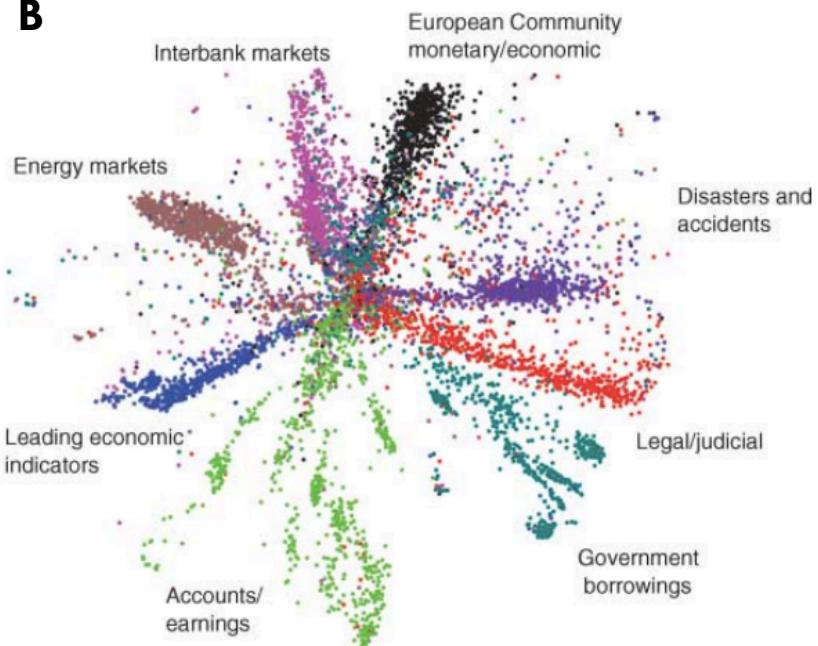
- Classificação de documentos por assunto (tipo)

A



Latent Semantic
Analysis (LSA)

B



Autoencoder

Exemplos de aplicações de DL

BossSensor

Hide your screen when your boss is approaching.

Demo

The boss stands up. He is approaching.



<https://github.com/Hironsan/BossSensor>

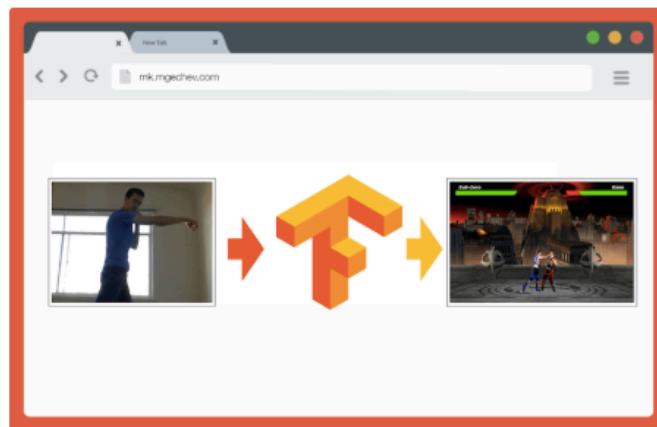
When he is approaching, the program fetches face images and classifies the image.

Exemplos de aplicações de DL

Playing Mortal Kombat with TensorFlow.js. Transfer learning and data augmentation

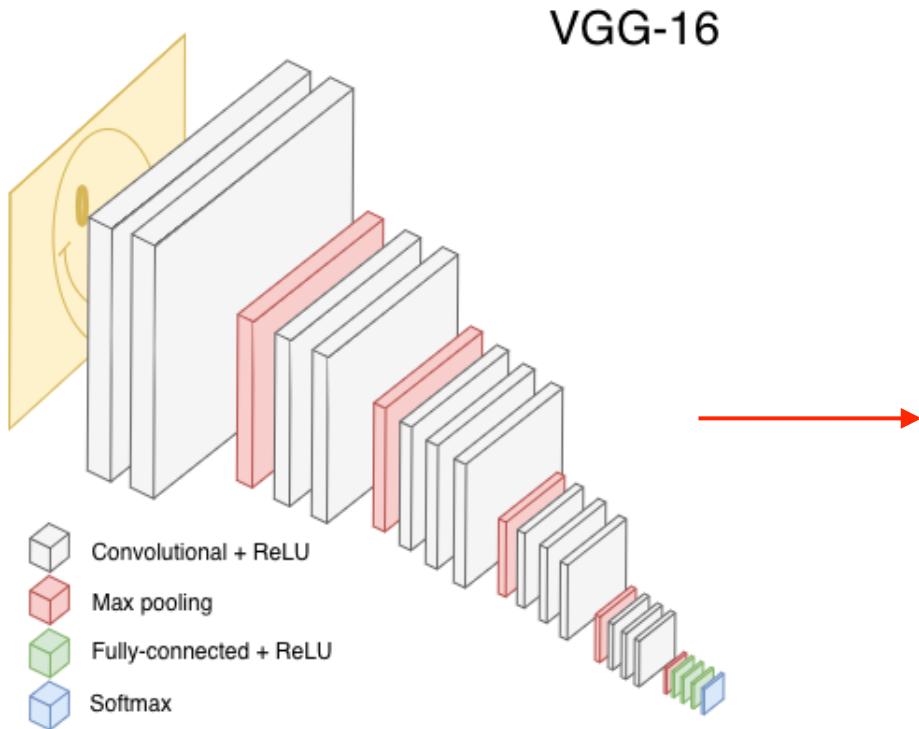
[Edit](#) · Oct 20, 2018 · 25 minutes read · [Follow @mgechev](#) · 10.8K followers
MACHINE LEARNING TENSORFLOW CNN TRANSFER LEARNING DATA AUGMENTATION ML

While experimenting with enhancements of the prediction model of [Guess.js](#), I started looking at deep learning. I've focused mainly on recurrent neural networks (RNNs), specifically LSTM because of their “[unreasonable effectiveness](#)” in the domain of Guess.js. In the same time, I started playing with convolutional neural networks (CNNs), which although less traditionally, are also often used for time series. CNNs are usually used for image classification, recognition, and detection.



<https://blog.mgechev.com/2018/10/20/transfer-learning-tensorflow-js-data-augmentation-mobile-net/>

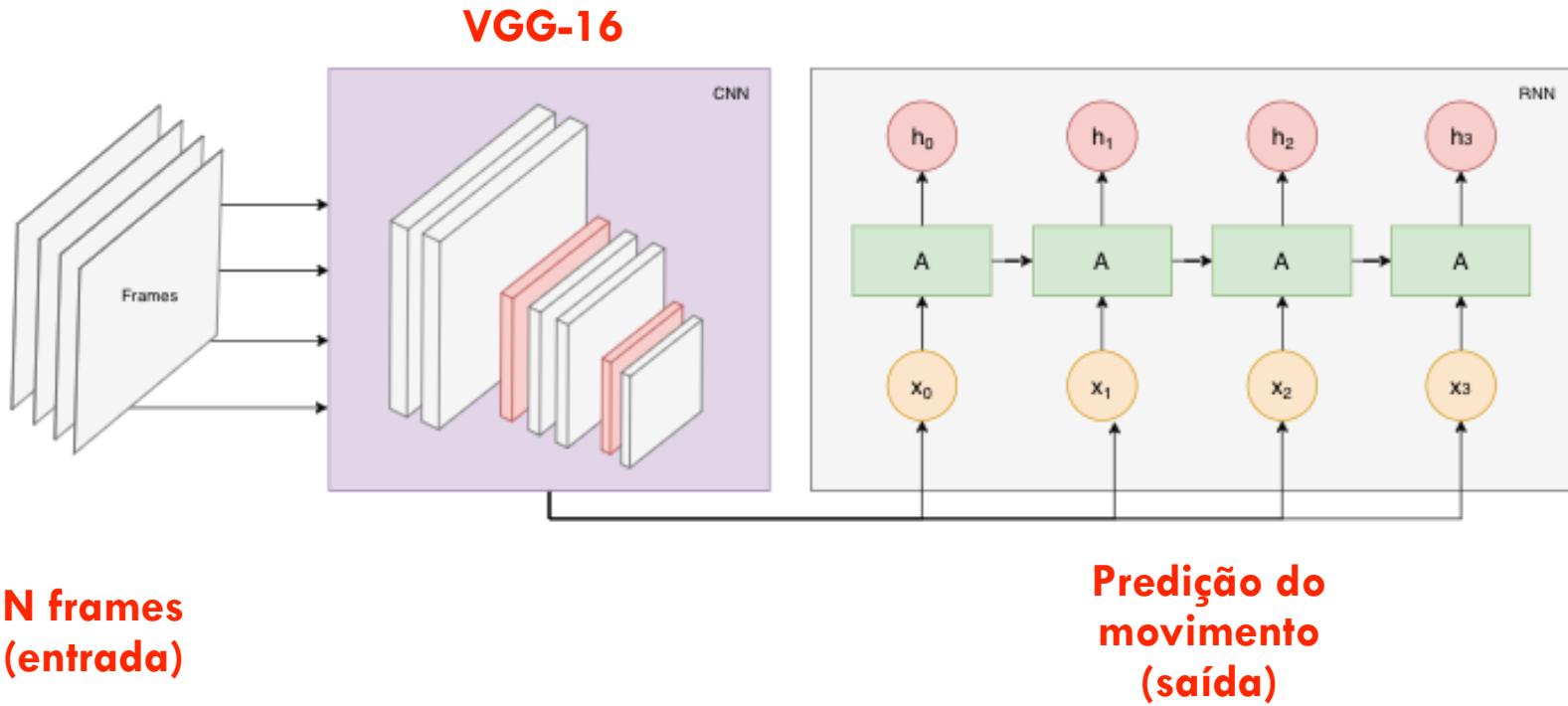
Exemplos de aplicações de DL



77

<https://blog.mgechev.com/2018/10/20/transfer-learning-tensorflow-js-data-augmentation-mobile-net/>

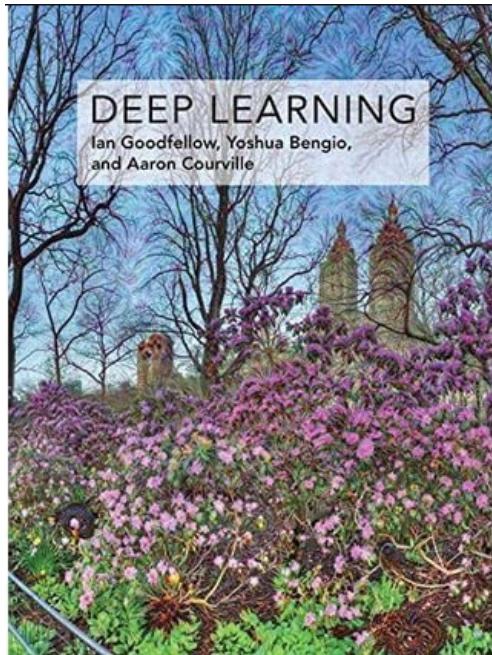
Exemplos de aplicações de DL



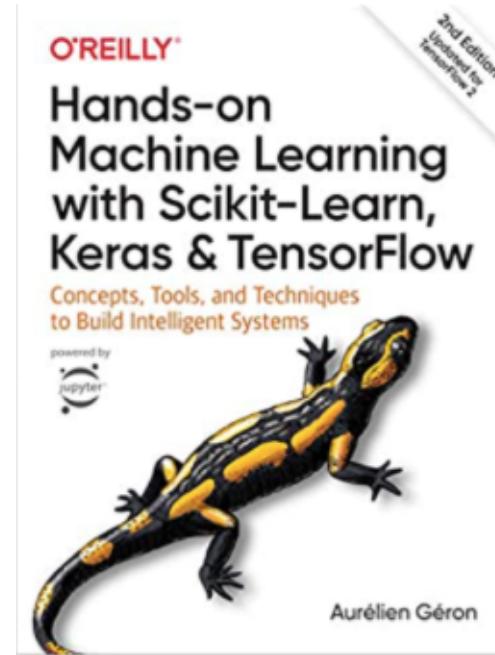
Roteiro

- 1 Introdução**
- 2 Deep Learning (DL)**
- 3 Treinamento de DL**
- 4 Exemplos**
- 5 Referências**

Literatura Sugerida

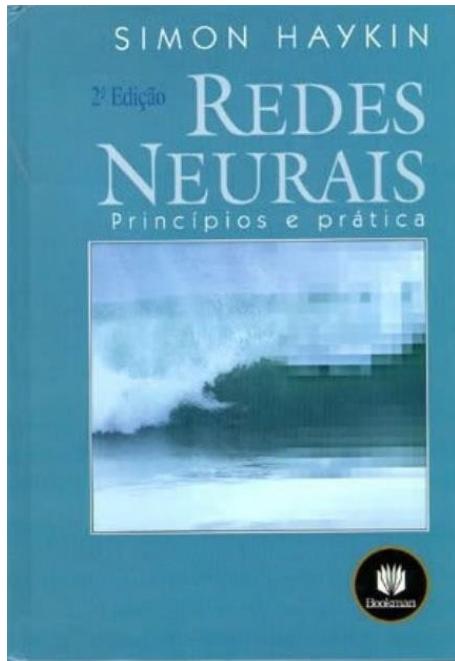


(Goodfellow, Bengio, Courville; 2015)

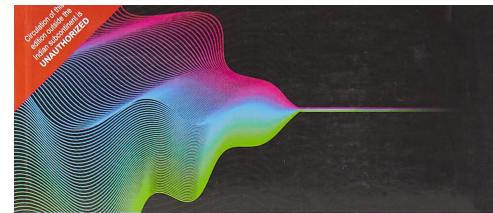


(Géron, 2019)

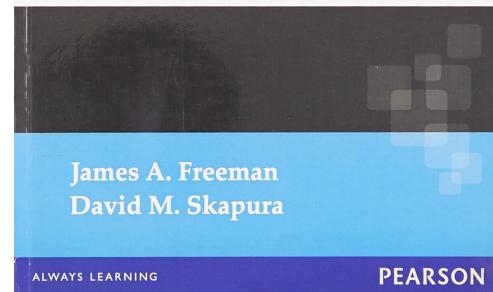
Literatura Sugerida



(Haykin, 1999)

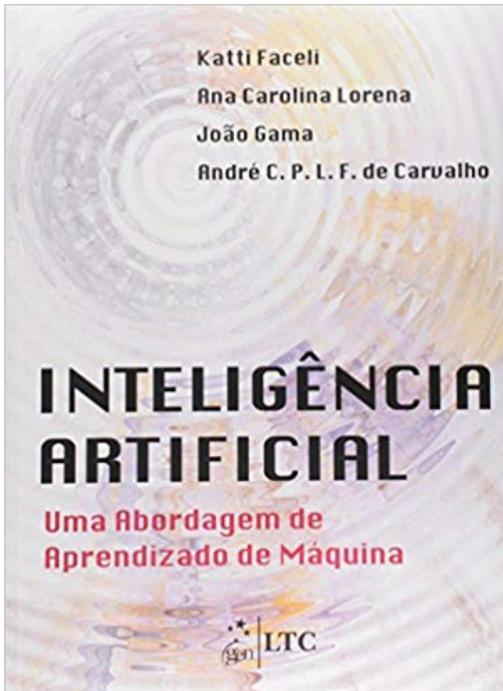


Neural Networks
Algorithms, Applications and
Programming Techniques

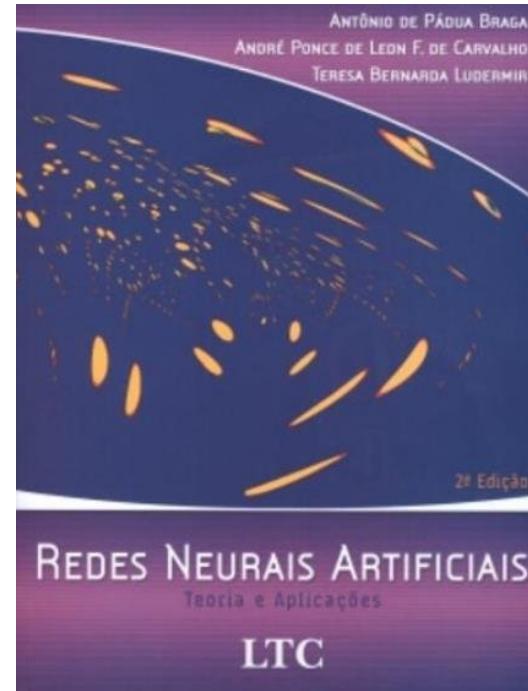


(Freeman & Skapura, 1991)

Literatura Sugerida



[Faceli et al, 2011]



[Braga et al, 2007]

Perguntas?

Prof. Rafael G. Mantovani

rgmantovani@gmail.com



Material complementar

Literatura Sugerida

- MIT book: <http://www.deeplearningbook.org>
- Deep Learning: <http://deeplearning.net>
- Andrew Ng: <https://www.deeplearning.ai>

Literatura Complementar

- Coursera: <https://www.coursera.org/specializations/deep-learning>
- Google AI: <https://ai.google/education/>
- Keras: <https://keras.io>
- Auto-Keras: <https://autokeras.com>
- h2o: <http://docs.h2o.ai/h2o/latest-stable/h2o-docs/index.html#>

Literatura Complementar

- Aulas de Hugo Larrochelle: <https://www.youtube.com/watch?v=vXMpKYRhpml>
- DL + MK: <https://blog.mgechev.com/2018/10/20/transfer-learning-tensorflow-js-data-augmentation-mobile-net/>
- Hide Screen: <https://github.com/Hironsan/BossSensor>