

FUNDAMENTOS DA INTELIGÊNCIA ARTIFICIAL

Aula 07 - Breve introdução
ao Deep Learning (DL)

Prof. Rafael G. Mantovani



Roteiro

- 1 Introdução**
- 2 Deep Learning (DL)**
- 3 Treinamento de DL**
- 4 Exemplos**
- 5 Síntese / Próximas Aulas**
- 6 Referências**

Roteiro

- 1 Introdução**
- 2 Deep Learning (DL)**
- 3 Treinamento de DL**
- 4 Exemplos**
- 5 Síntese / Próximas Aulas**
- 6 Referências**

Introdução

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

Introdução

□ Teorema da Aproximação Universal

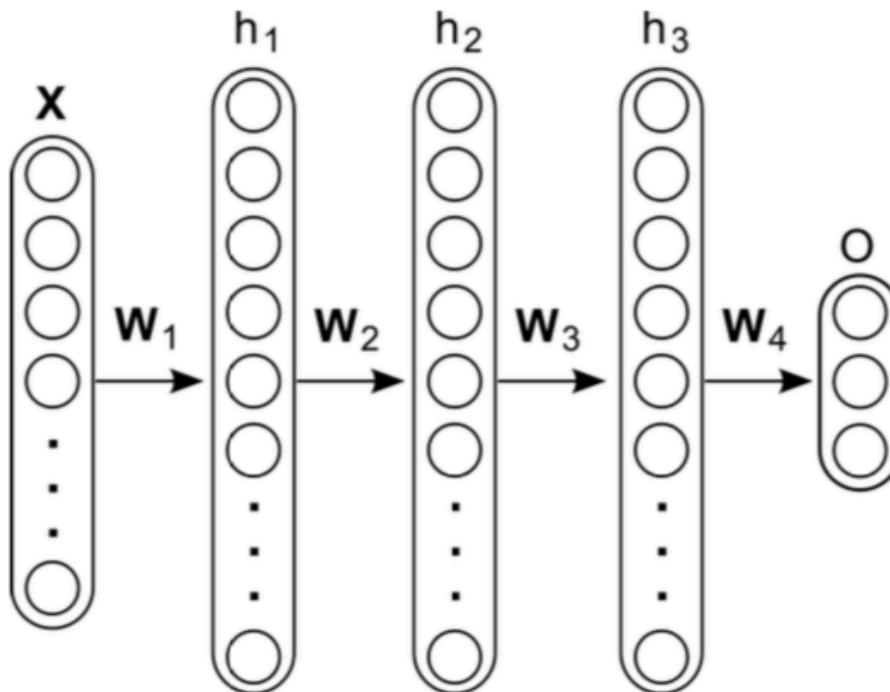
- "... uma RNA com única camada escondida e um número finito de neurônios pode aproximar qualquer função contínua".

- **Problemas - não diz nada sobre:**

- tempo de treinamento
 - facilidade de implementação
 - generalização

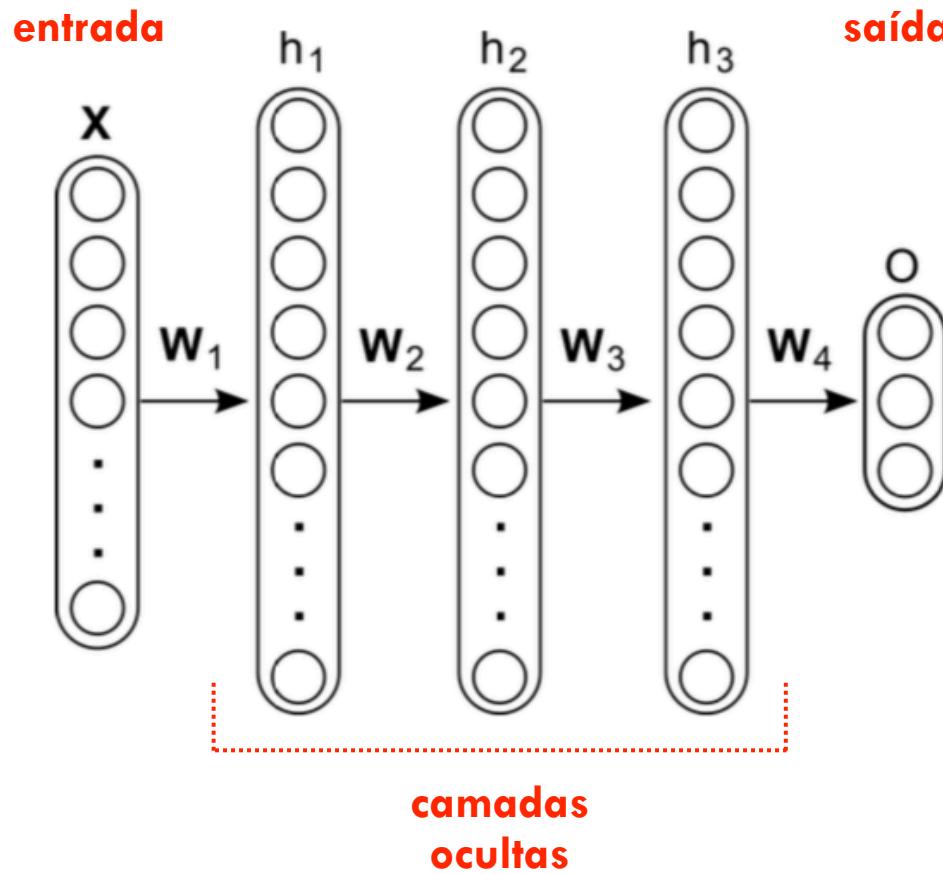
Introdução

- Em uma MLP podemos ter L camadas ocultas:



Introdução

- Em uma MLP podemos ter L camadas ocultas:

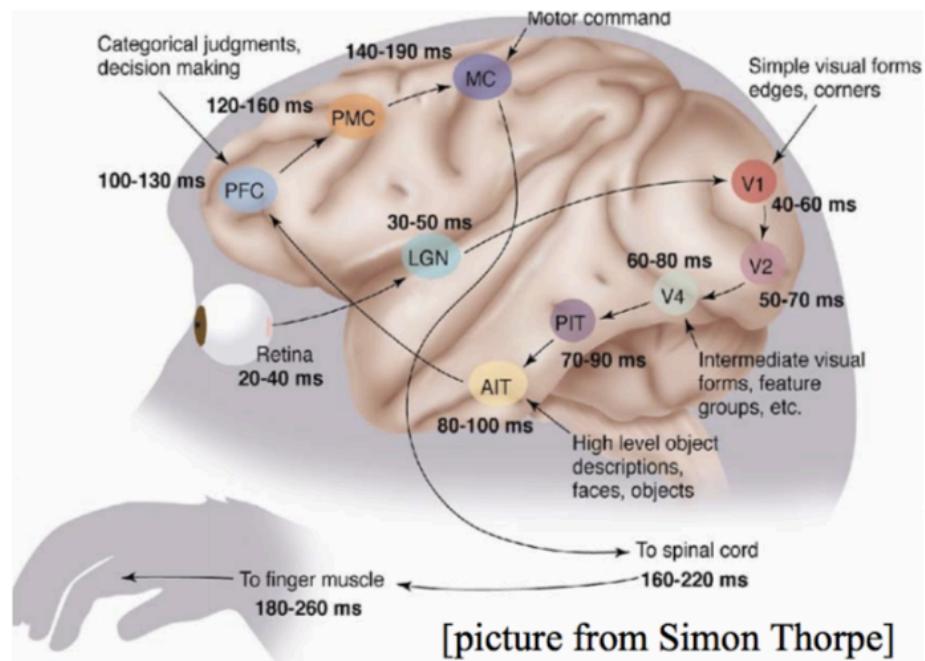


Introdução

- **Deep Learning**
 - **Objetivo:** aprender modelos com múltiplas camadas de representação. Exs:
 - MLPs, DNNs, DBNs, Deep Autoencoders,
 - CNNs, RBMs, LSMT, etc
 - Cada camada corresponde a uma representação
 - unidade → uma característica da entrada
 - unidades podem ser ativadas simultaneamente

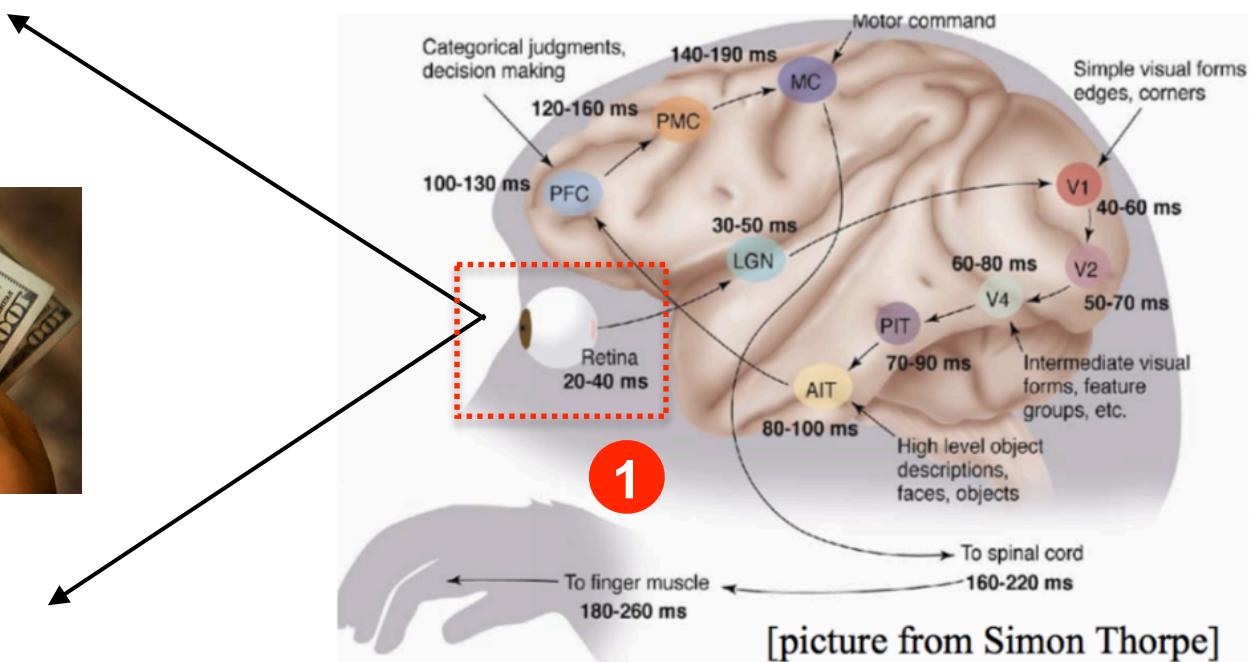
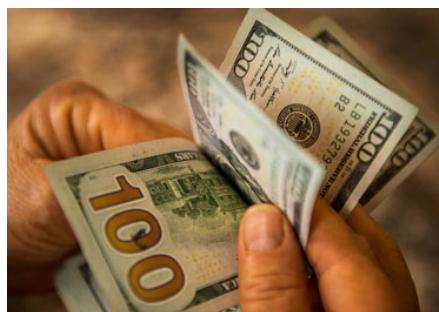
Motivação

□ CórTEX visual



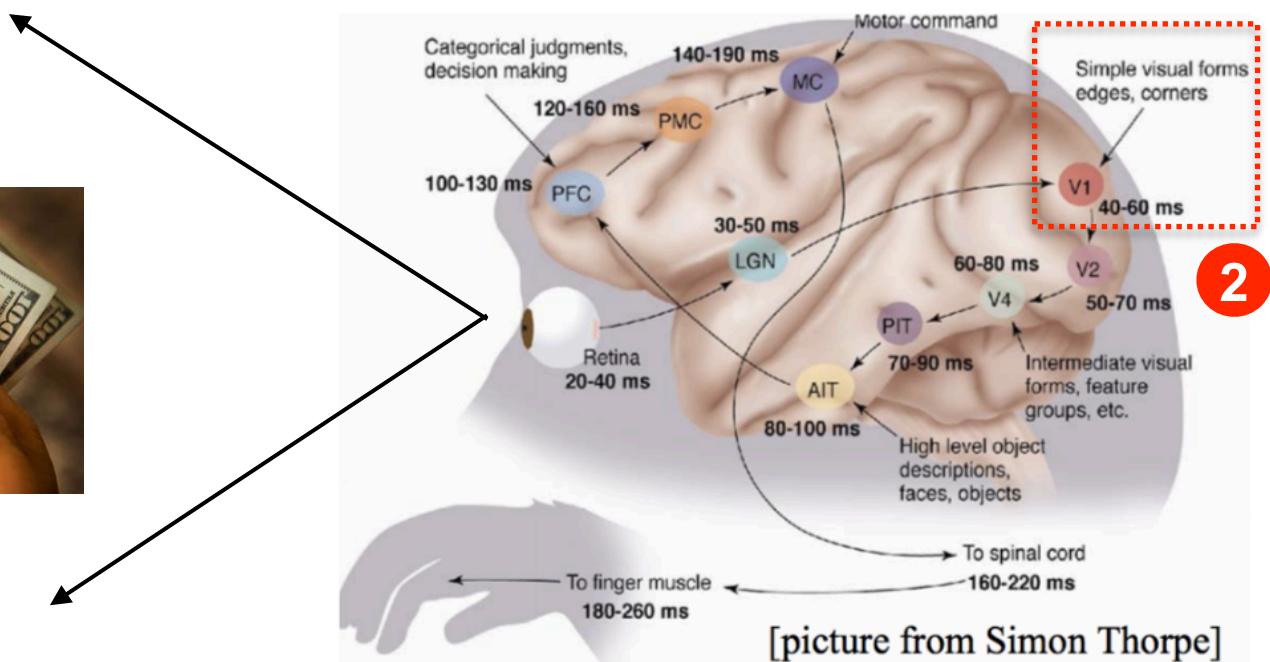
Motivação

□ Córtex visual



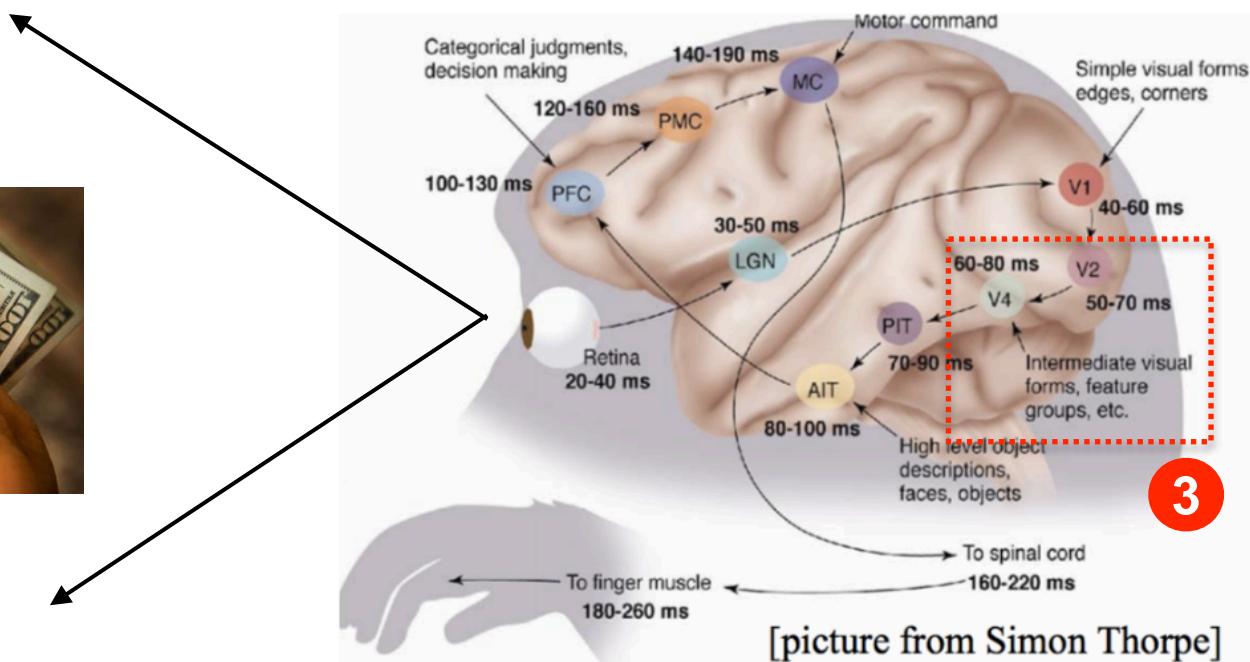
Motivação

□ Córtex visual



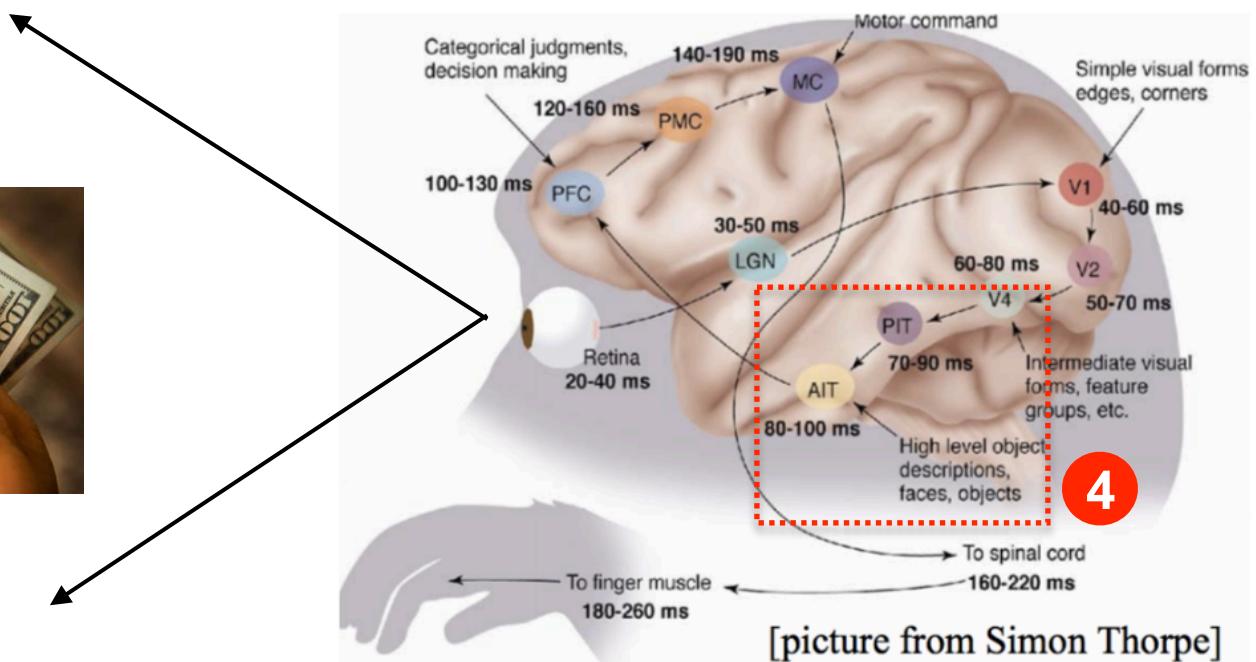
Motivação

□ Córtex visual



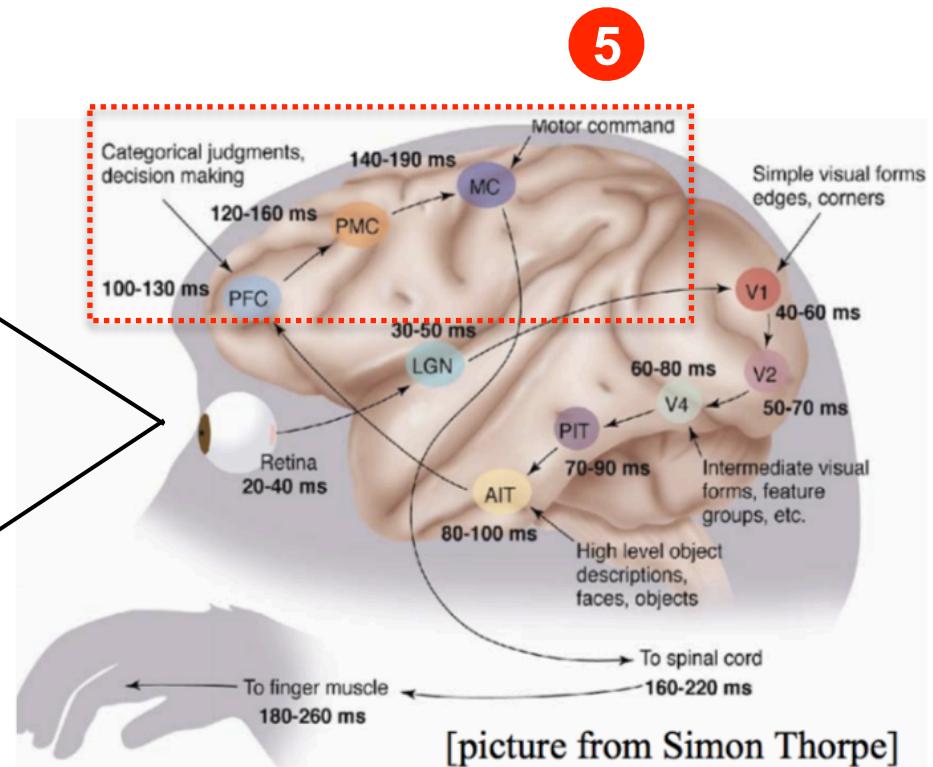
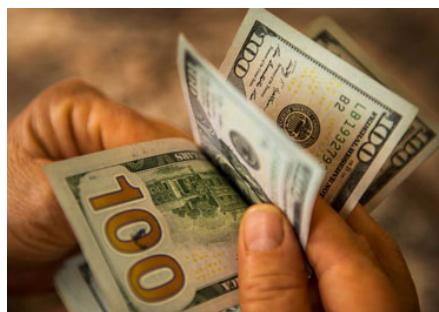
Motivação

□ Córtex visual



Motivação

□ Córtex visual

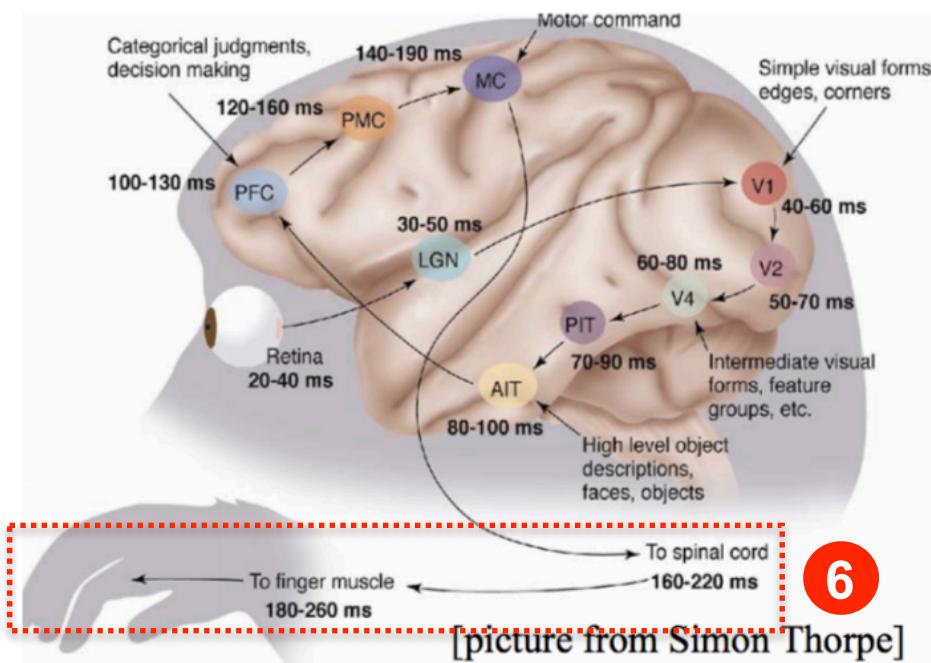


Motivação

□ Córtex visual



movimento



Motivação

- **Córtex visual**



Ação realizada

Motivação

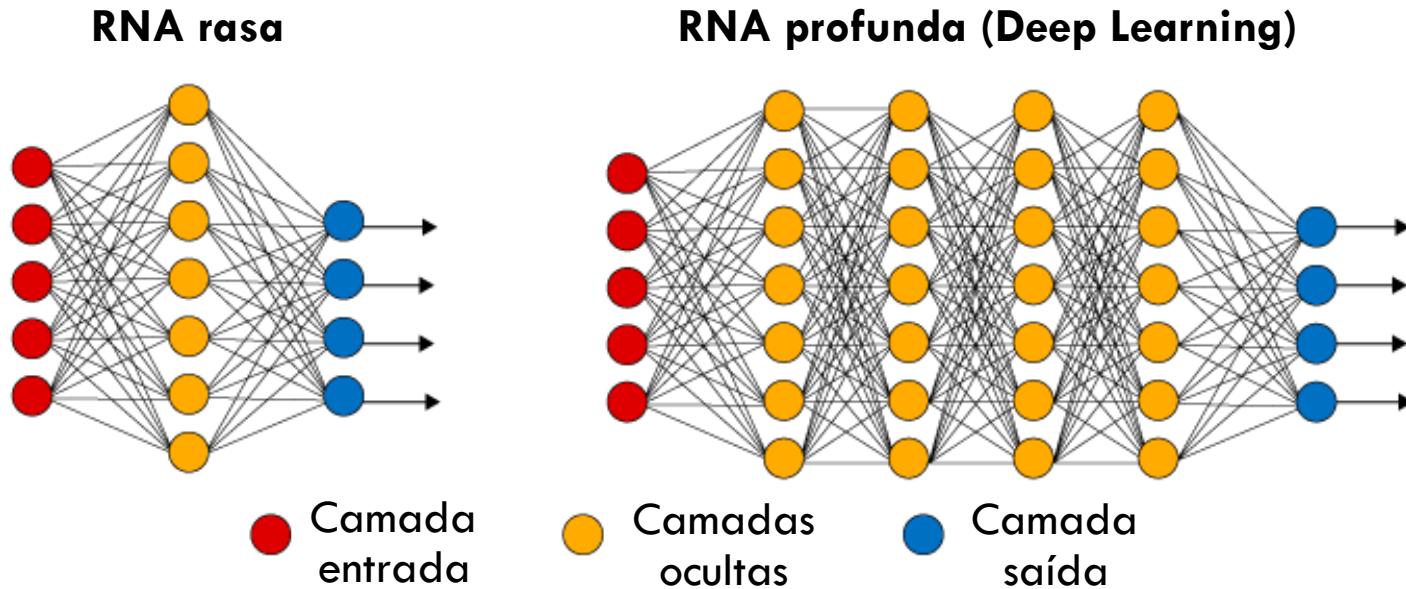
- Arquiteturas profundas (muitas camadas):
 - podem representar certas funções de maneira mais compacta
 - há funções que podem ser representadas com uma única camada, porém requerem um número exponencial de neurônios
 - Porém, pode ser necessário um número polinomial de neurônios, se pudermos aumentar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

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

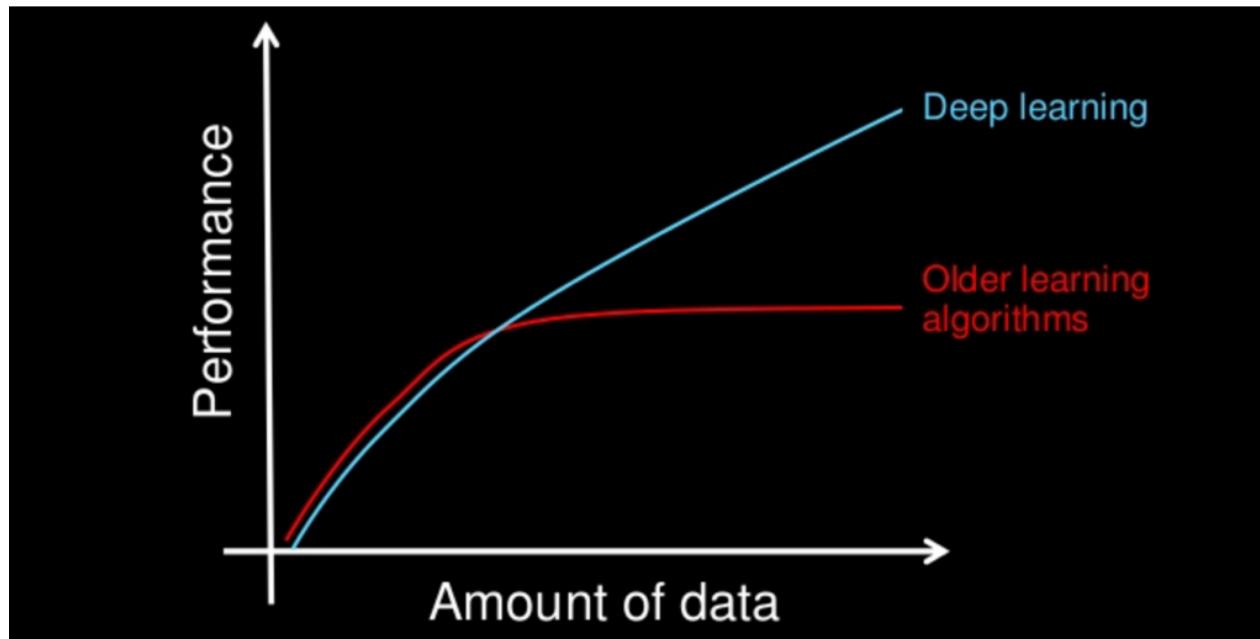
Roteiro

- 1 Introdução**
- 2 Deep Learning (DL)**
- 3 Treinamento de DL**
- 4 Exemplos**
- 5 Síntese / Próximas Aulas**
- 6 Referências**

Deep Learning

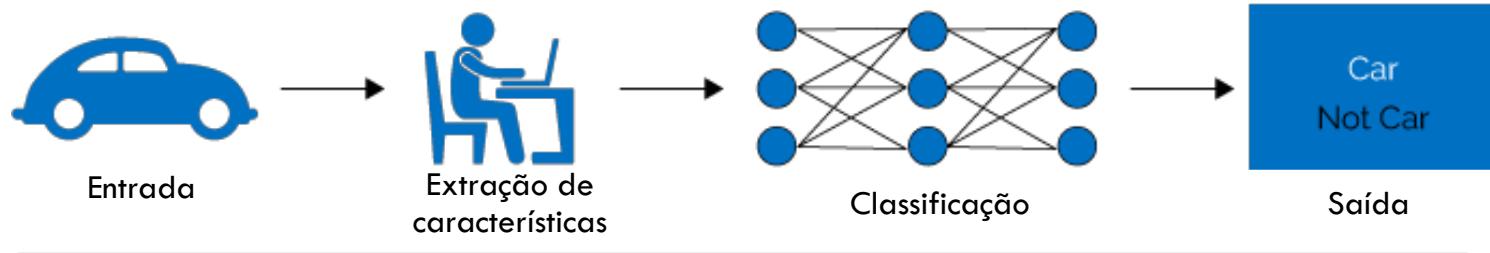


Deep Learning

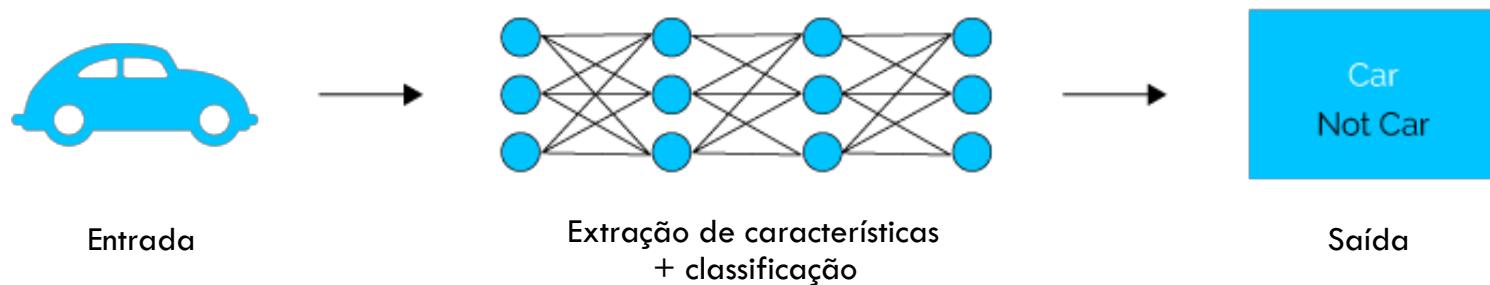


Deep Learning

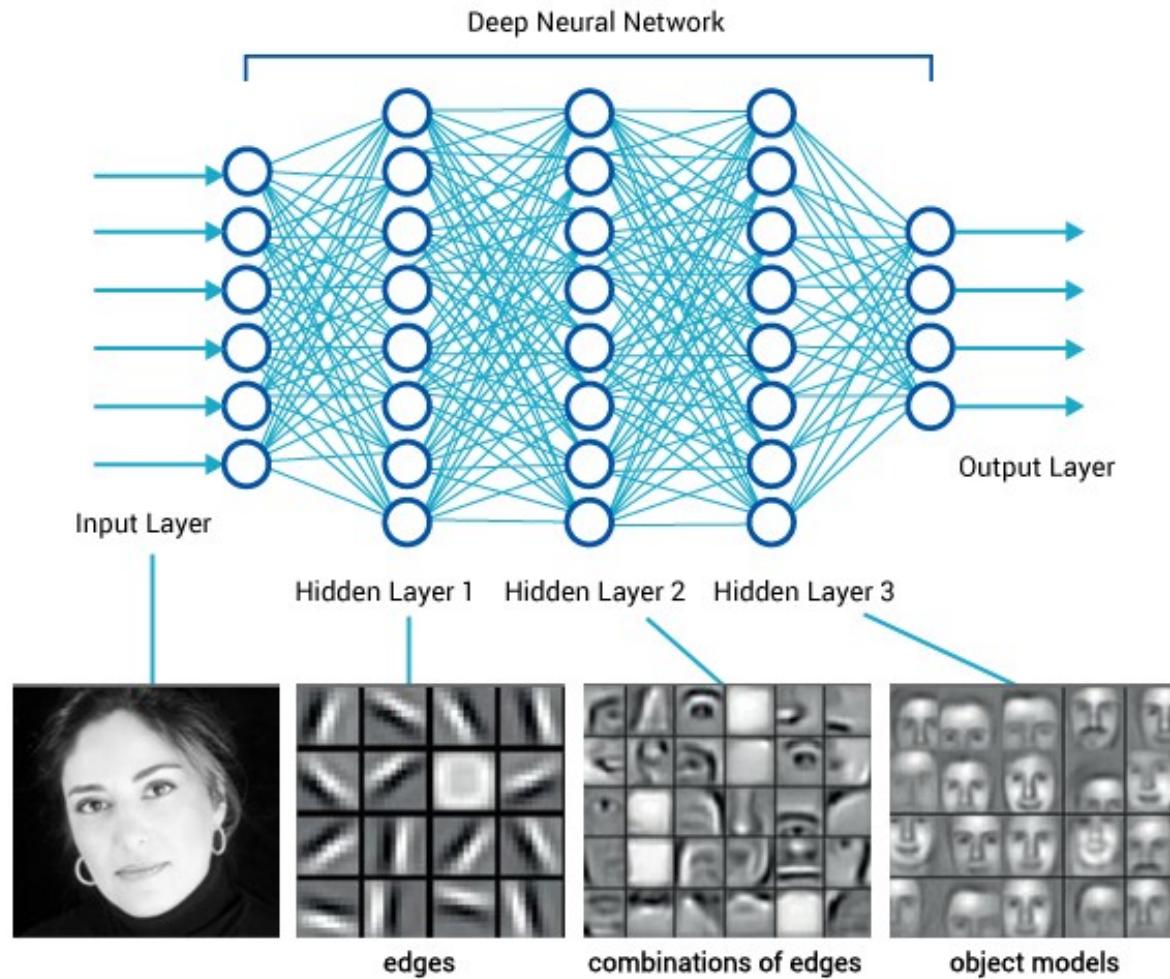
Machine Learning



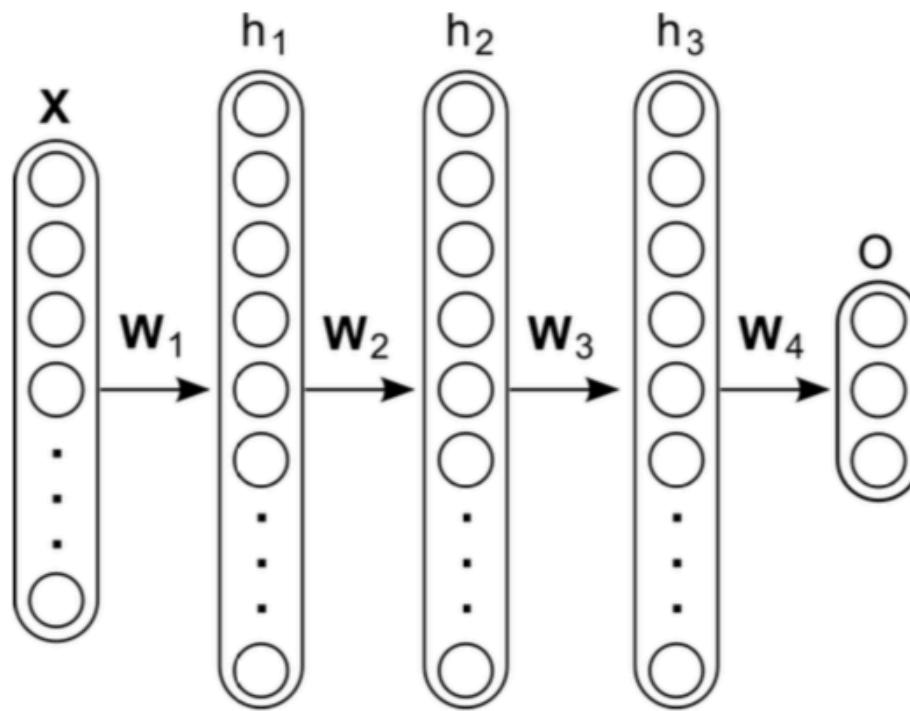
Deep Learning



Deep Learning

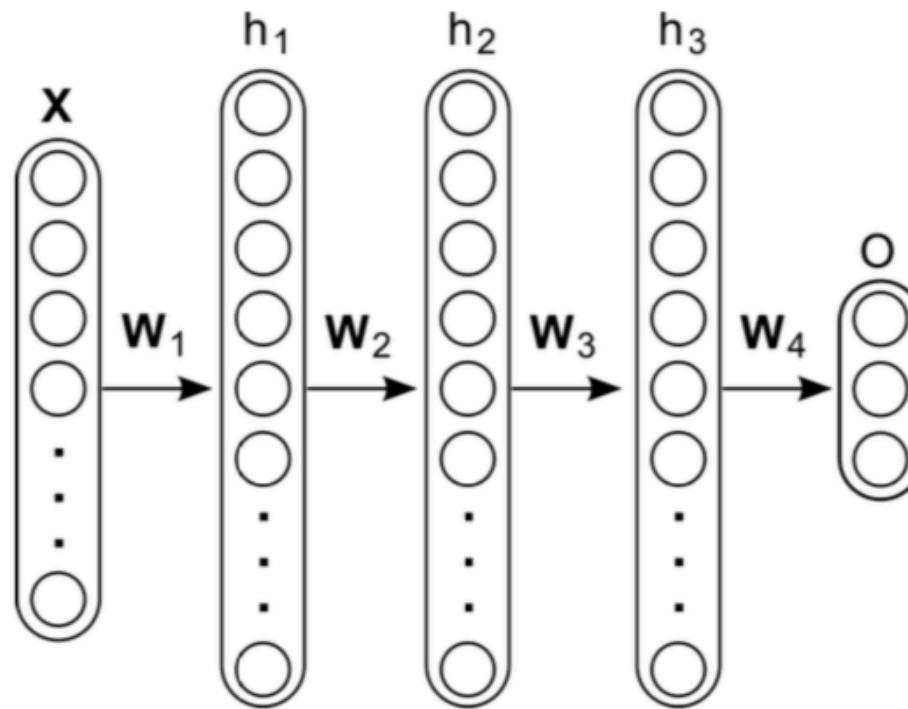


Problema



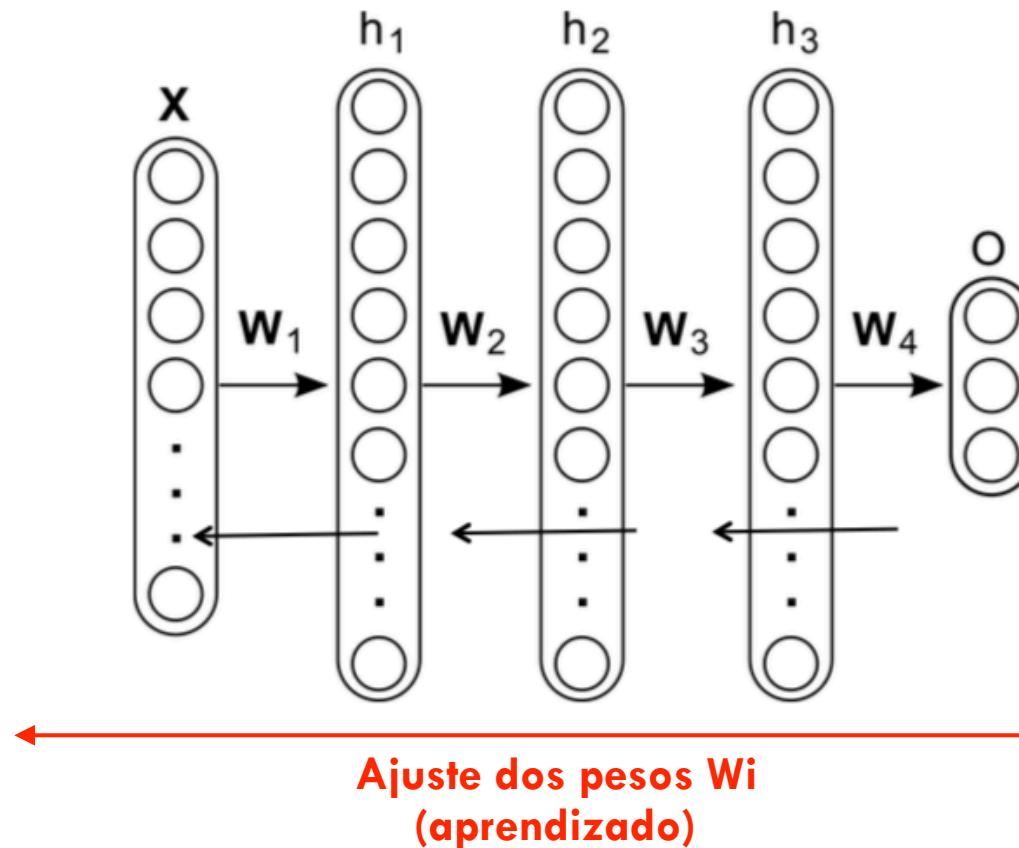
Problema

- Treinamento: porque é difícil treinar DL?



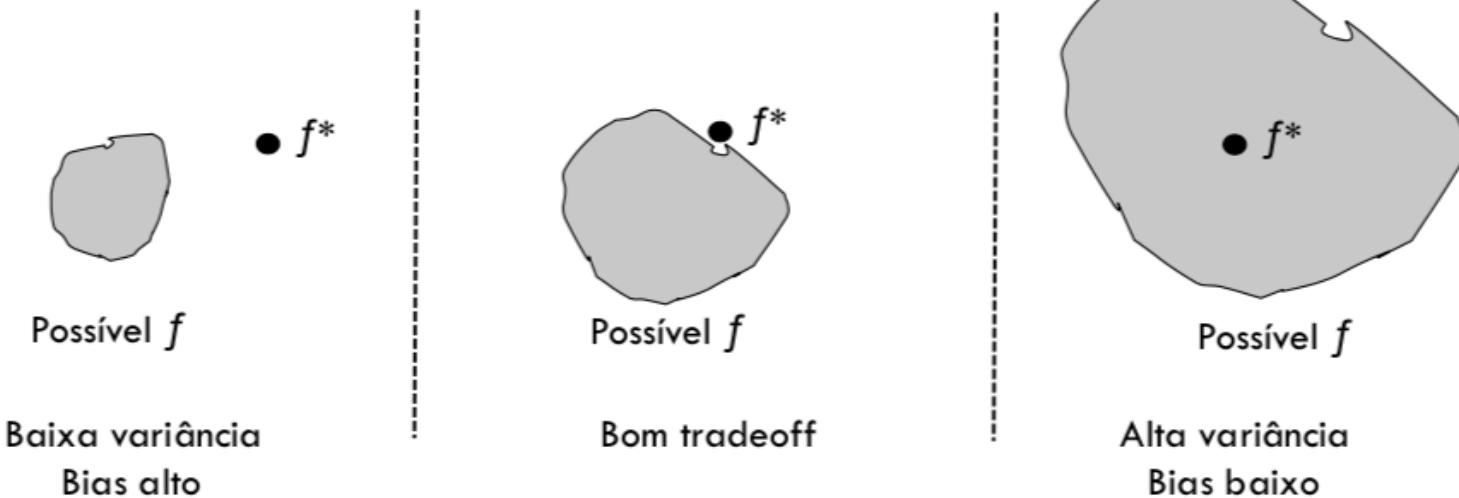
Problema

- *Underfitting* → Gradientes desaparecem



Problema

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



Roteiro

- 1 Introdução**
- 2 Deep Learning (DL)**
- 3 Treinamento de DL**
- 4 Exemplos**
- 5 Síntese / Próximas Aulas**
- 6 Referências**

Solução

- 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

1. Pré-treino

- **Estratégia:** procedimento ganancioso 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 ganancioso camada a camada

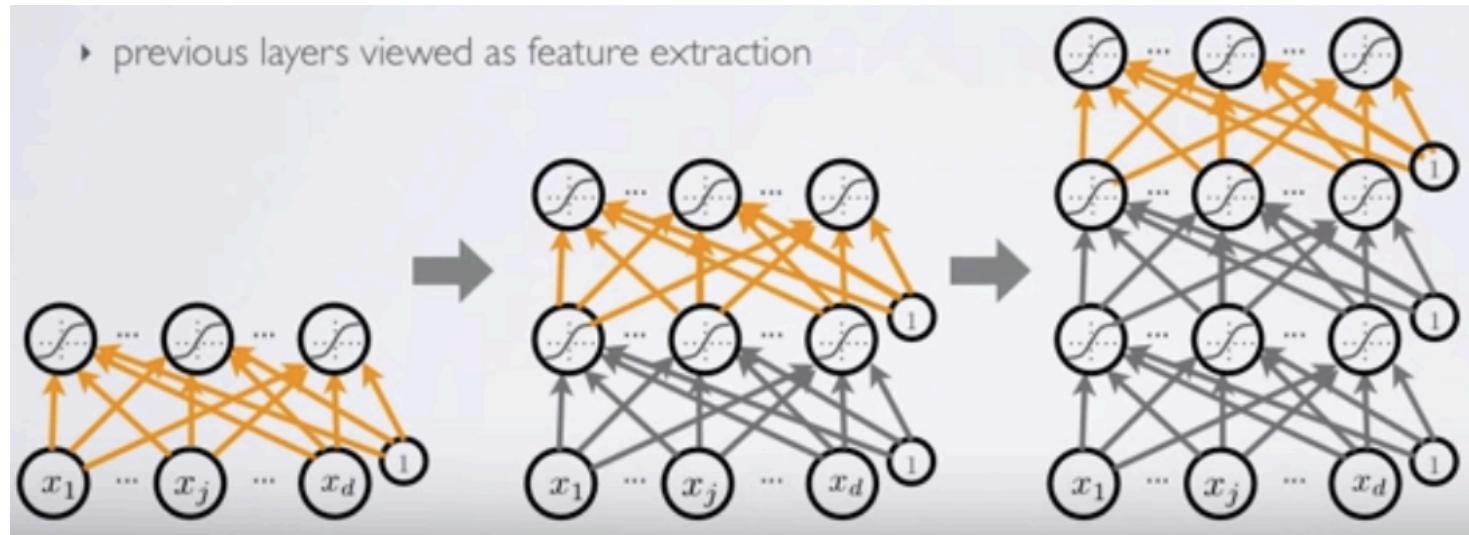


Figura de Larrochelle

1. Pré-treino

- **Camadas:**
 - 1 camada → encontra características que são mais comuns nos dados de treino do que em dados aleatórios
 - 2 camada → encontra combinações de características dos neurônios escondidos
 - 3 camada → combinações de combinações ...
 - 4 camada → ...

1. Pré-treino

- **Camadas:**

- 1 camada → encontra características que são mais comuns nos dados
 - **Pré-treino:** inicializar os parâmetros de uma região tal que haja menos overfitting
- 2 camadas → combinações de características dos neurônios escondidos
- 3 camada → combinações de combinações ...
- 4 camada → ...

Ajuste Fino

- Depois de treinadas todas as camadas:
 - adiciona a camada de saída
 - treina toda a rede com aprendizado supervisionado
- Aprendizado supervisionado é realizado de maneira convencional
- Todos os parâmetros são ajustados para a tarefa supervisionada
 - pesos sinápticos,
- O ajuste é feito para tornar a rede mais discriminativa

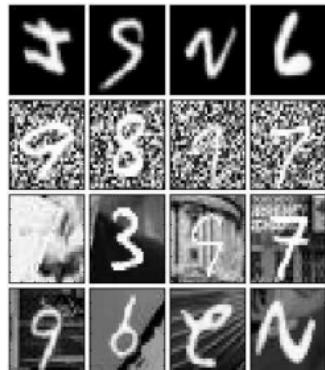
Exemplo

An Empirical Evaluation of Deep Architectures on Problems with Many Factors of Variation

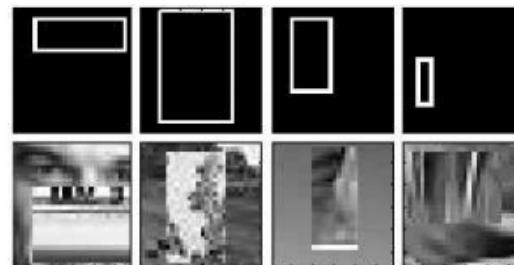
Hugo Larochelle
Dumitru Erhan
Aaron Courville
James Bergstra
Yoshua Bengio

Dept. IRO, Université de Montréal C.P. 6128, Montreal, Qc, H3C 3J7, Canada

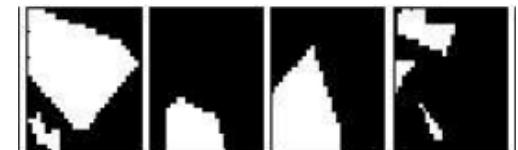
LAROCHEH@IRO.UMONTREAL.CA
ERHANDUM@IRO.UMONTREAL.CA
COURVILA@IRO.UMONTREAL.CA
BERGSTRJ@IRO.UMONTREAL.CA
BENGIOY@IRO.UMONTREAL.CA



Variações no MNIST



Variações de retângulos

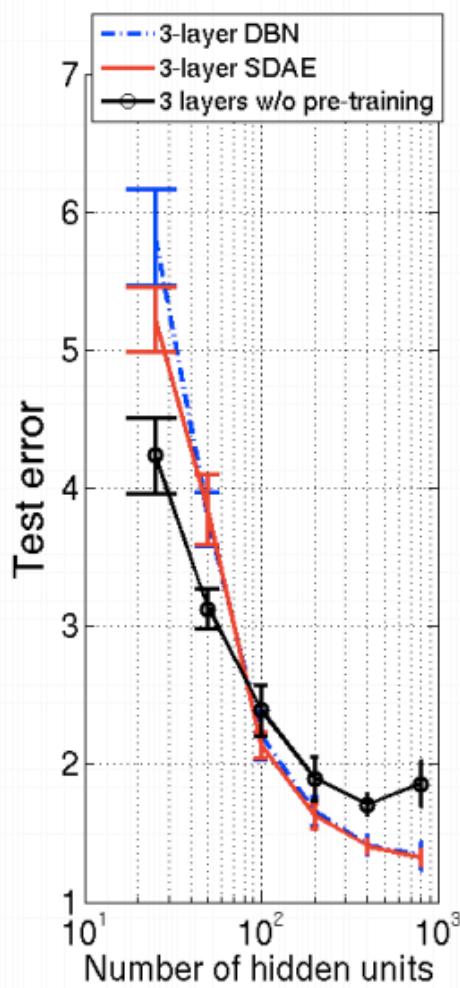
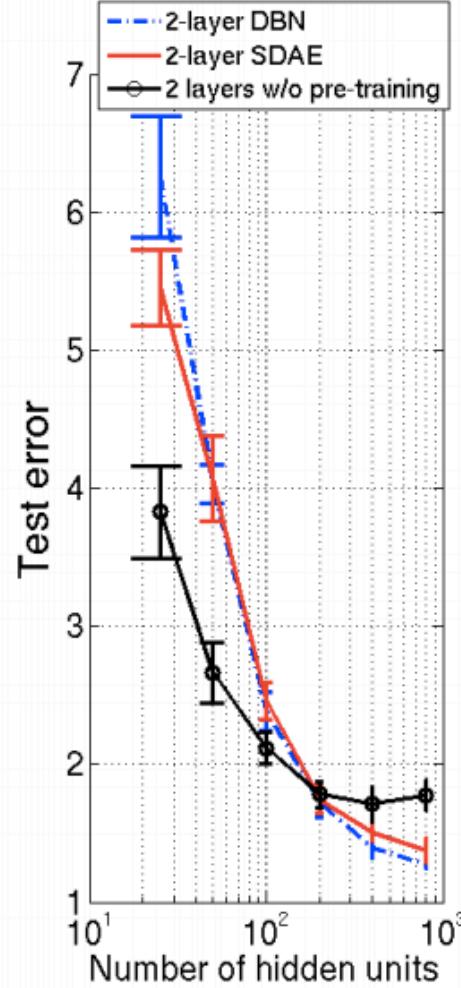
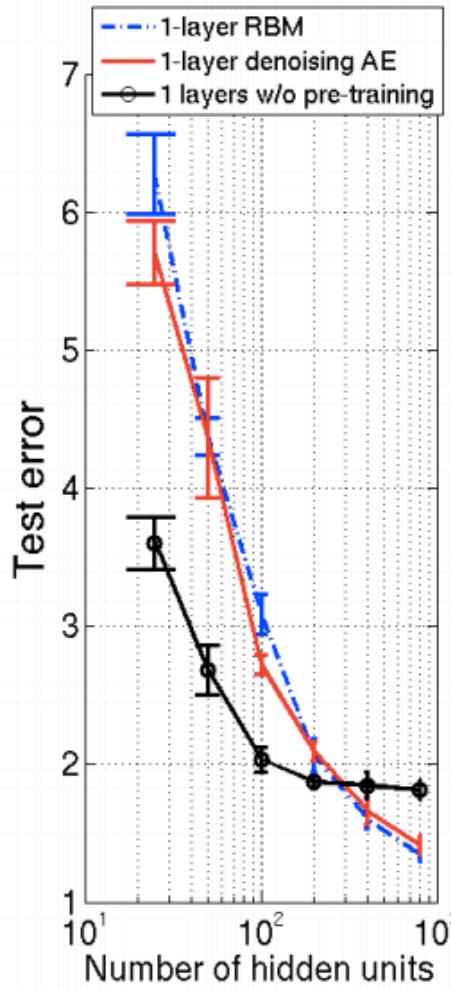


Convexa ou não

Exemplo

Network		MNIST-small classif. test error	MNIST-rotation classif. test error
Type	Depth		
Deep net	1	4.14 % \pm 0.17	15.22 % \pm 0.31
	2	4.03 % \pm 0.17	10.63 % \pm 0.27
	3	4.24 % \pm 0.18	11.98 % \pm 0.28
	4	4.47 % \pm 0.18	11.73 % \pm 0.29
Deep net + autoencoder	1	3.87 % \pm 0.17	11.43% \pm 0.28
	2	3.38 % \pm 0.16	9.88 % \pm 0.26
	3	3.37 % \pm 0.16	9.22 % \pm 0.25
	4	3.39 % \pm 0.16	9.20 % \pm 0.25
Deep net + RBM	1	3.17 % \pm 0.15	10.47 % \pm 0.27
	2	2.74 % \pm 0.14	9.54 % \pm 0.26
	3	2.71 % \pm 0.14	8.80 % \pm 0.25
	4	2.72 % \pm 0.14	8.83 % \pm 0.24

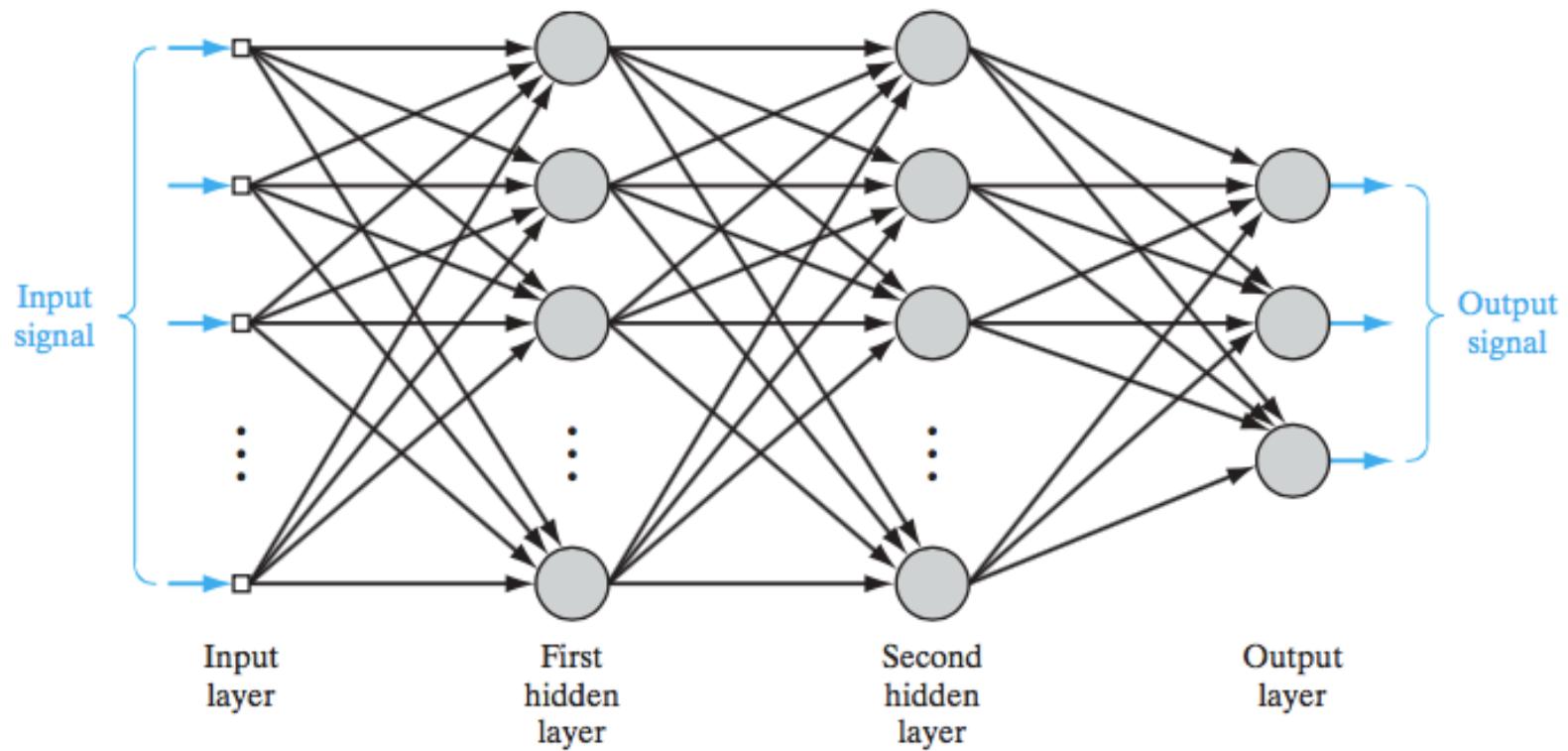
Exemplo



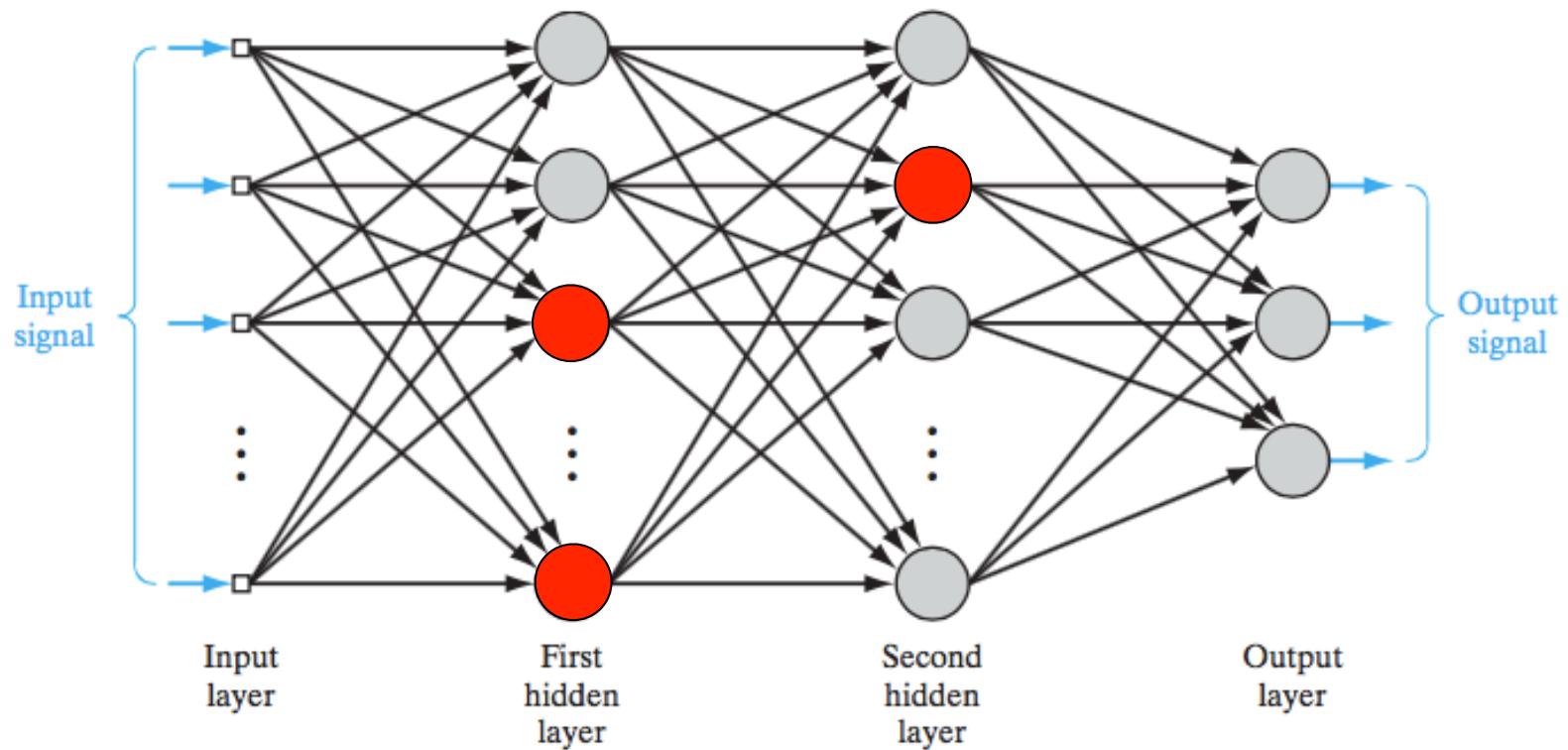
2. Dropout

- **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

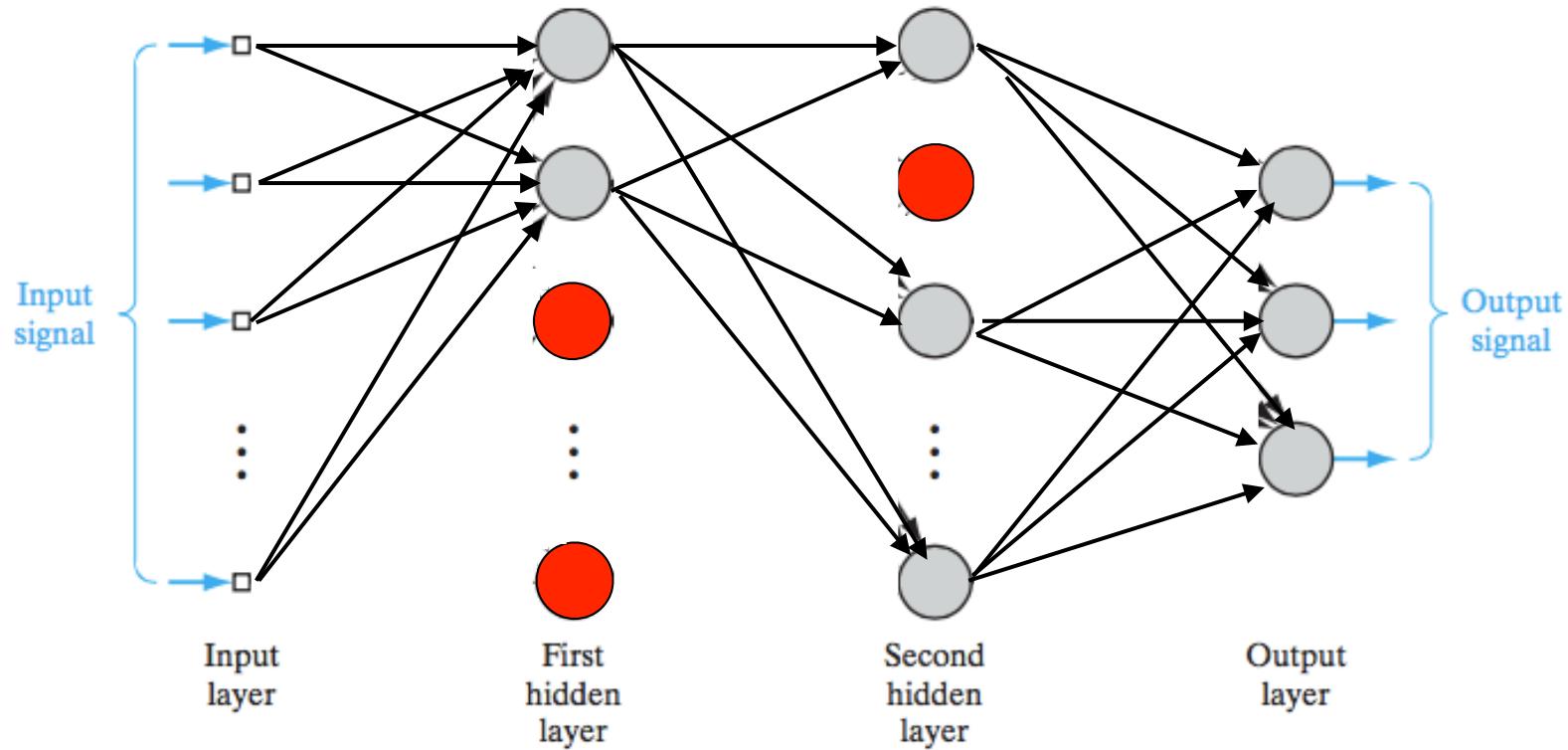
2. Dropout



2. Dropout



2. Dropout



3. Outros estimadores

▫ Adagrad:

- 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

- 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

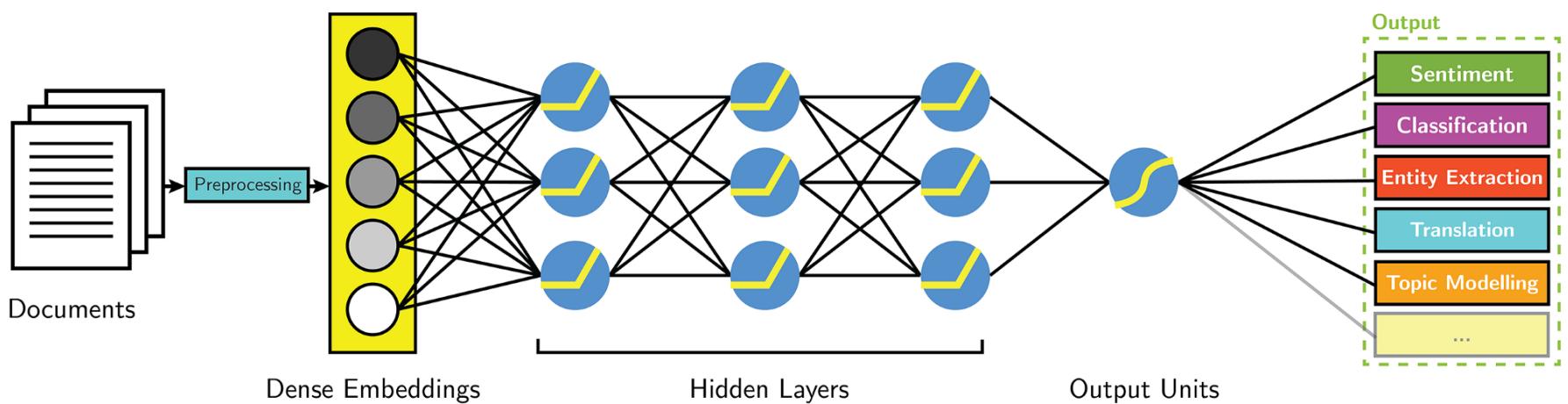
- Combinação do Adagrad e do RMSProp
- faz decremento dos **momentums** de primeira e segunda ordem

Roteiro

- 1 Introdução**
- 2 Deep Learning (DL)**
- 3 Treinamento de DL**
- 4 Exemplos**
- 5 Síntese / Próximas Aulas**
- 6 Referências**

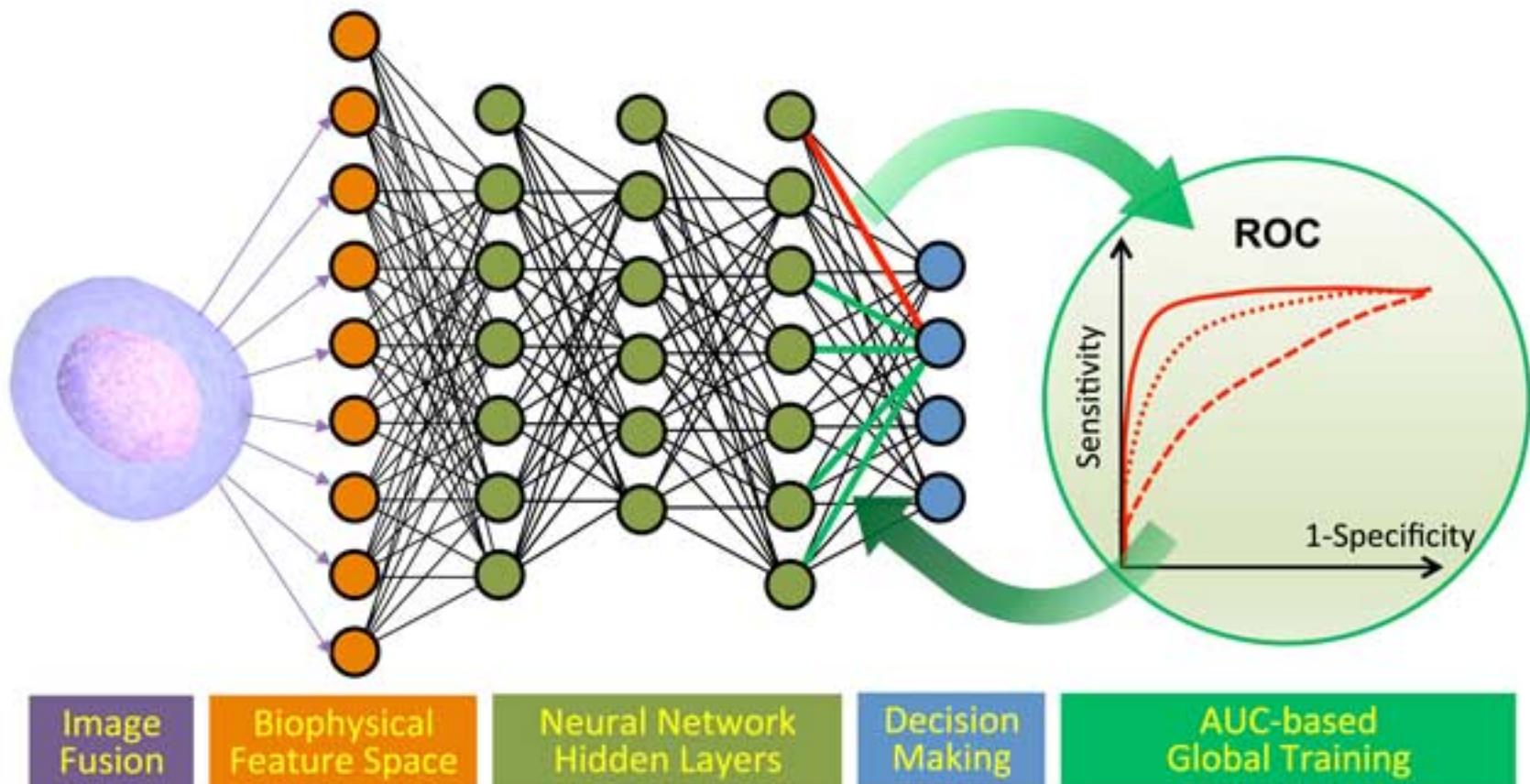
Exemplos de aplicações

- Detecção de sentimentos:



Exemplos de aplicações

- Diagnóstico de cancer:



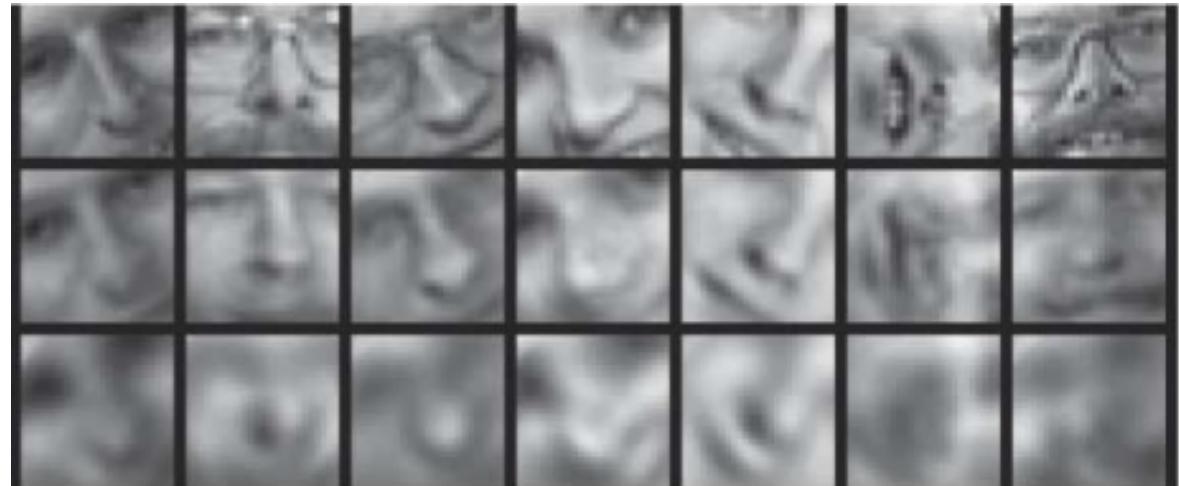
Exemplos de aplicações

- Redução de dimensionalidade (Autoencoders)

Reducing the Dimensionality of Data with Neural Networks

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

dados originais

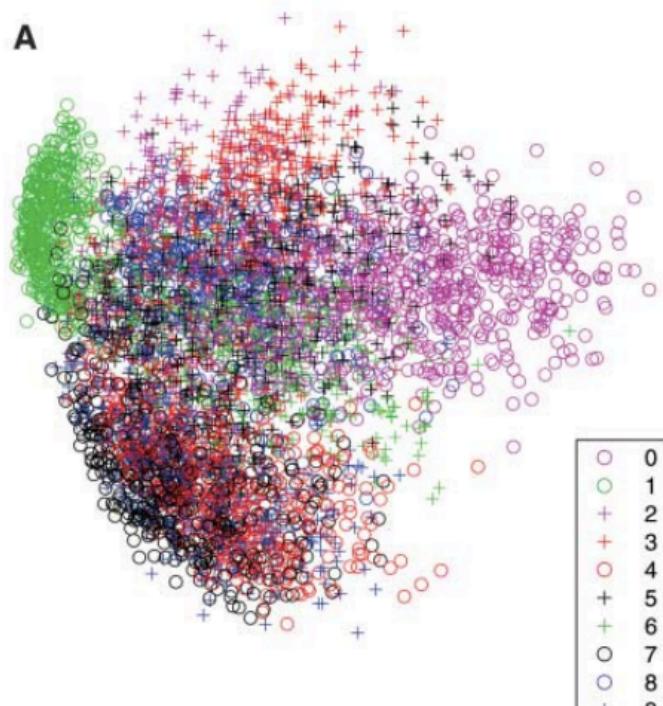


deep autoencoder

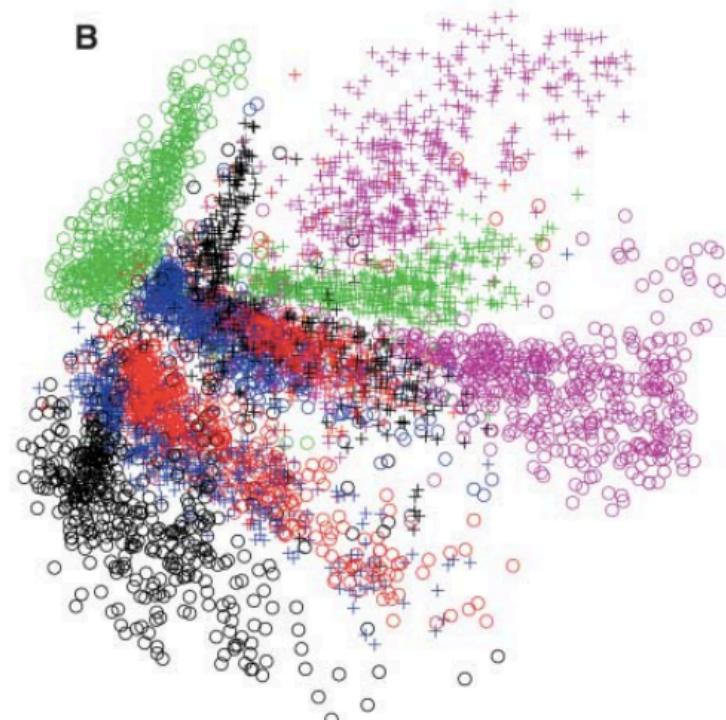
PCA

Exemplos de aplicações

- Reconhecimento de dígitos por imagens



PCA

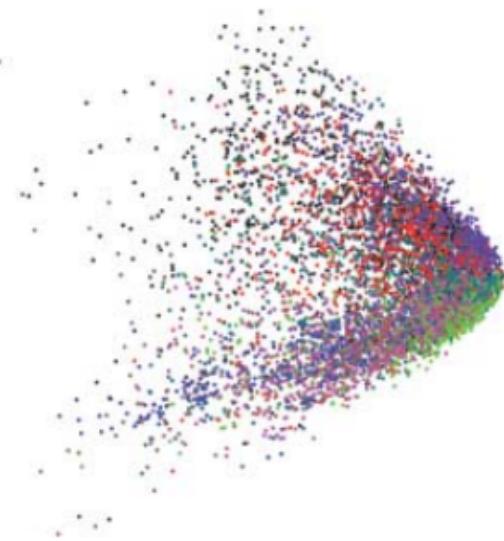


Autoencoder

Exemplos de aplicações

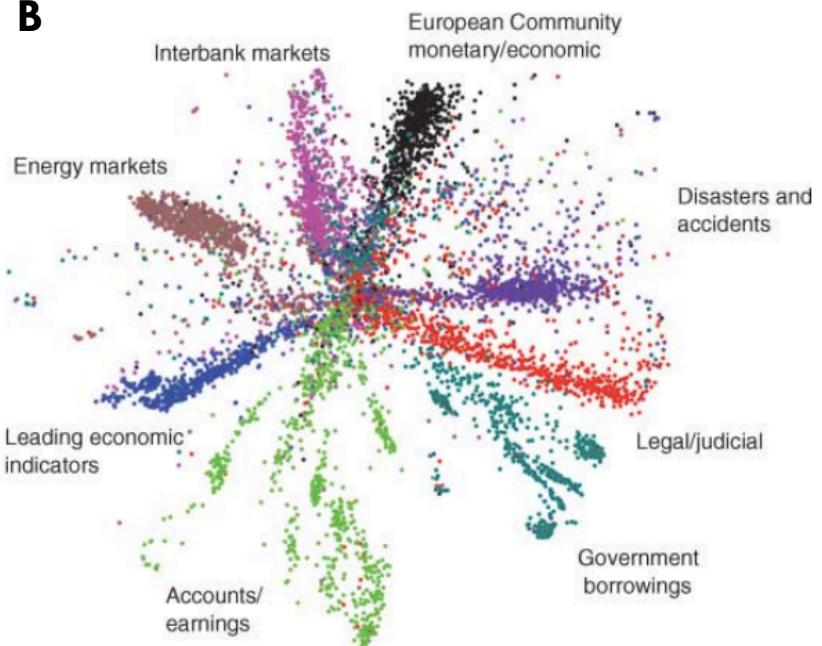
- Classificação de documentos por assunto (tipo)

A



Latent Semantic
Analysis (LSA)

B



Autoencoder

Exemplos de aplicações

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.

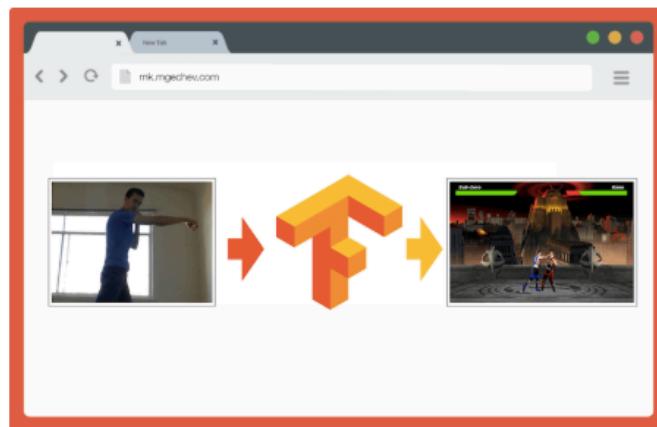
Exemplos de aplicações

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

51

[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.

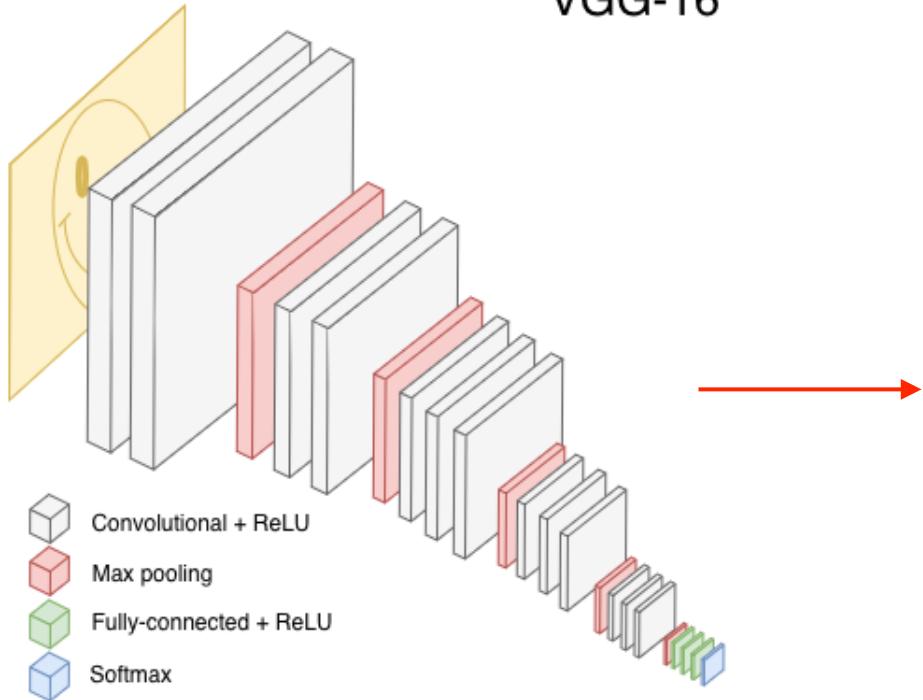


<https://blog.mgechev.com/2018/10/20/transfer-learning-tensorflow-js-data-augmentation-mobile-net/>

Exemplos de aplicações

52

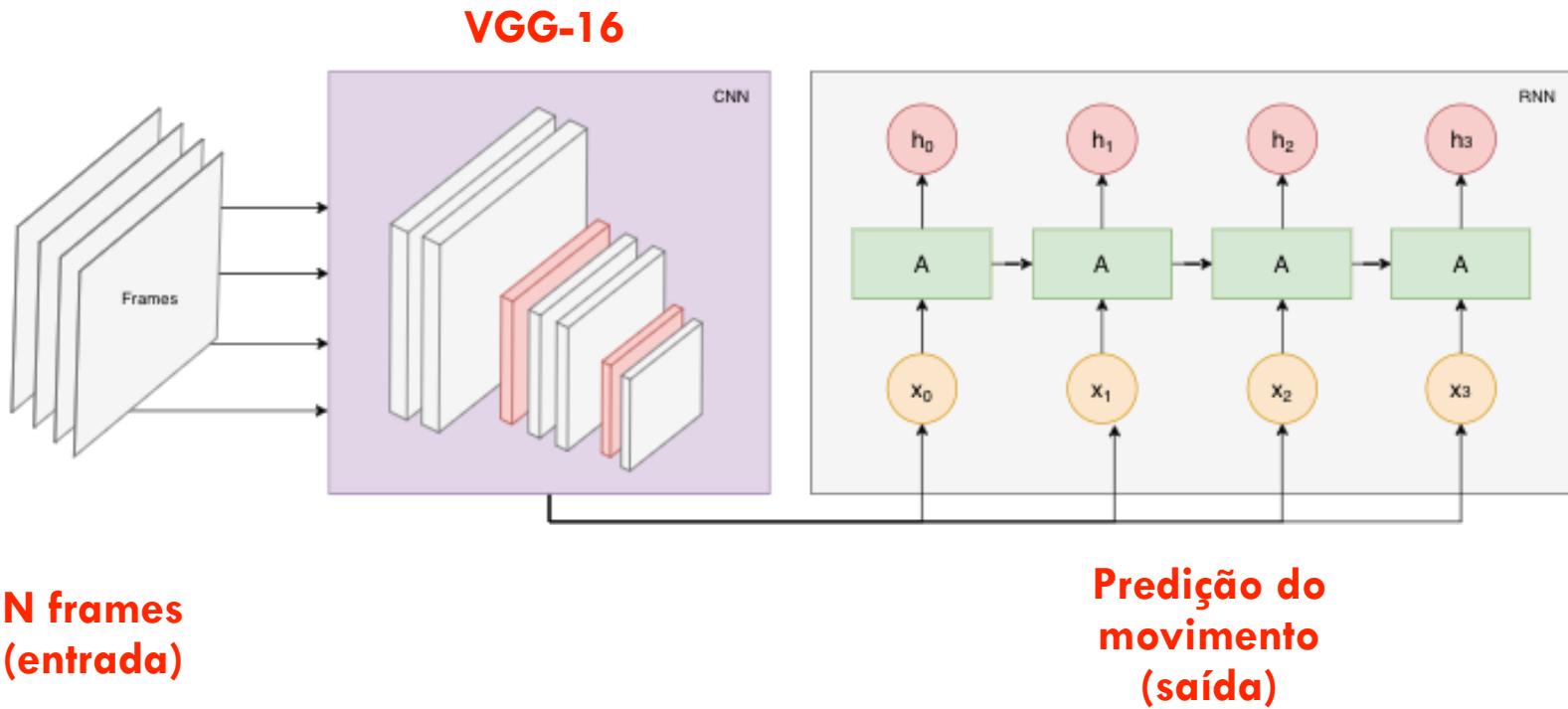
VGG-16



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

Exemplos de aplicações

53



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

Roteiro

- 1 Introdução**
- 2 Deep Learning (DL)**
- 3 Treinamento de DL**
- 4 Exemplos**
- 5 Síntese / Próximas Aulas**
- 6 Referências**

Síntese

- **Deep Learning**
 - modelos com muitas (muitas) camadas
 - processamento em vários níveis
 - Treinamento envolve:
 - pré-treino não supervisionado
 - dropouts
 - diferentes estimadores (algoritmos de aprendizado)
 - aplicações

Próxima Aula

- Visão geral sobre SVMs
- Paradigma evolucionista
 - Algoritmos Evolucionistas
 - Algoritmos Genéticos

Roteiro

- 1 Introdução**
- 2 Deep Learning (DL)**
- 3 Treinamento de DL**
- 4 Exemplos**
- 5 Síntese / Próximas Aulas**
- 6 Referências**

Literatura Sugerida

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

Material 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#>

Material 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>

Perguntas?

Prof. Rafael G. Mantovani

rgmantovani@uel.br