

Deep Learning

Mini curso - SINAPE 2022, Gramado RS

Renato Assunção - ESRI Inc. e DCC/UFMG

Daniel Falbel - RStudio

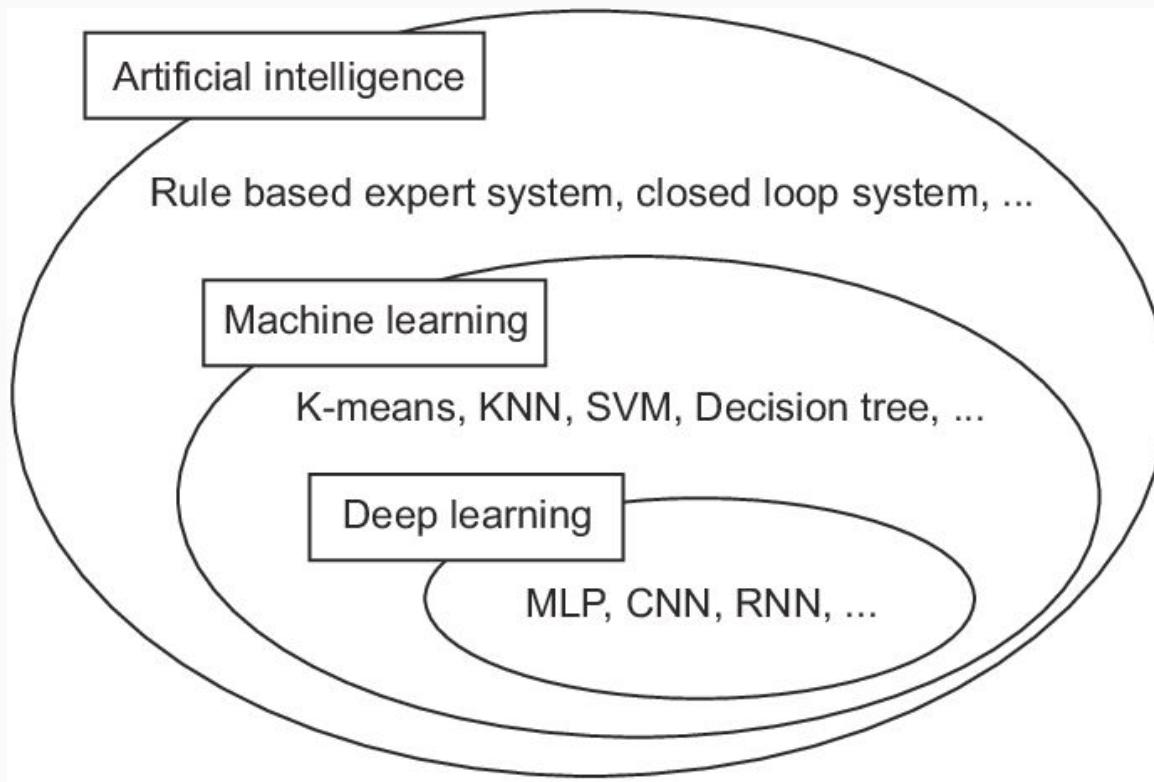
Overview do minicurso

- Dia 01:
 - Apresentação de Deep Learning
 - Histórico e posição atual
 - Principais aplicações de sucesso:
 - Tarefas com imagens
 - Tarefas com textos
 - Tarefas com grafos complexos
 - O que é deep learning e como difere de estatística tradicional?
 - Exemplos usando R

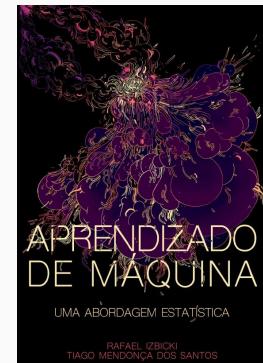
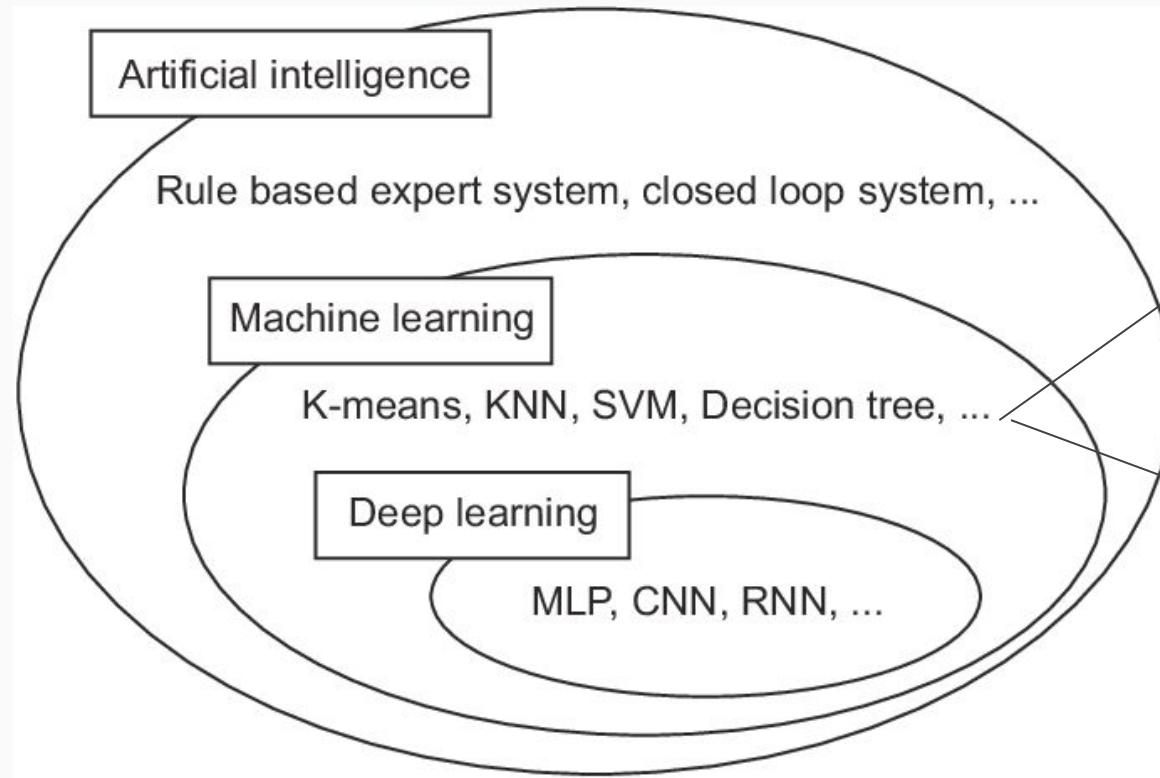
Overview do minicurso (02)

- Dia 02:
 - Apresentação de GAN: Generative Adversarial Networks
 - Histórico e posição atual
 - Principais aplicações de sucesso
 - Simulação de imagens
 - Simulação de textos
 - Outras tarefas
 - Teoria de GAN
 - Exemplos usando R

Diagrama de Veen



Literatura em português



Inteligência Artificial (IA)

- Começou com um workshop em Dartmouth College em 1956.
- Organizado por Allen Newell (CMU), Herbert Simon (CMU), John McCarthy (MIT), Marvin Minsky (MIT) e Arthur Samuel (IBM)
- Feitos surpreendentes apareceram logo:
 - Computadores jogando damas (1954) (melhor que humanos em 1959),
 - resolvendo problemas simples de álgebra,
 - provando teoremas lógicos (Logic Theorist, em 1956)
 - falando em inglês.
- IA cresce; é criada uma grande expectativa de imensos sucessos

Mas o progresso não vem tão rápido

- Financiamento é cortado na década de 70.
- AI Winter
- AI muda de foco: weak AI
- Passa a se concentrar em tarefas específicas ao invés de soluções globais inteligentes.
 - Sistemas especialistas para diagnóstico médico: eliciar regras lógicas de especialistas e criar sistema que, de posse de dados, possa deduzir as decisões corretas
 - Tradução automática
 - Reconhecimento de padrões em imagens
 - Visão computacional...

- Aprendizado de máquina (ML) e AI têm uma relação íntima
- ML = algoritmos e modelos estatísticos usados em AI pelo computador para realizar uma tarefa sem usar instruções explícitas, confiando em padrões e inferência aprendidos a partir dos dados estatísticos.
- Ideia é deixar a máquina aprender as regras necessárias para realizar uma tarefa a partir dos dados estatísticos.
- Como aprender dos dados?
- Depende da tarefa.
- AI escolheu algumas tarefas simples iniciais, tentou resolvê-las. Em seguida, generalizar e escalar
- Grande sucesso

Tarefa de classificação supervisionada

- Duas classes de objetos: 0 e 1
- Exemplos:
 - Imagens/fotos da internet
 - 1 = imagens com pelo menos um gato presente
 - 0 = imagens sem gatos
 - Imagens de sensoriamento remoto (LANDSAT)
 - 1 = pixel dominado por cobertura florestal
 - 0 = outro tipo de cobertura dominante
 - Pacientes chegando no pronto socorro com ferimento na cabeça
 - 1 = com necessidade de internação imediata no CTI
 - 0 = sem esta necessidade



Dados de onde aprendemos

- Coletamos amostras estatísticas de instâncias dos objetos das duas populações
- Sinônimos: instâncias, itens, exemplos, casos, indivíduos, observações
- Conjunto de exemplos = amostra
- Em cada exemplo, medimos um conjunto de n **variáveis** ou **features** em cada um deles

$$\mathbf{X} = (X_1, X_2, \dots, X_n)$$

- Com base nas medições em \mathbf{X} queremos aprender a distinguir os objetos dos dois grupos
- Anotamos também o verdadeiro rótulo associado a cada instância: classe 0 ou 1
- Este **rótulo (label)** é denotado por Y

$$(Y, \mathbf{X}) = (Y, X_1, X_2, \dots, X_n)$$

Objetivo e visualização da tarefa

- Novos itens chegam COM as n variáveis X's mas SEM o rótulo Y
- Objetivo: construir uma regra de classificação para esses novos itens
- Com base nas n variáveis X's, obter uma função matemática que prediga a classe do item.

$$\mathbf{X} = (X_1, X_2, \dots, X_n) \longrightarrow \text{Classificador} \longrightarrow \sigma(\mathbf{X}) = \mathbb{P}(Y = 1|\mathbf{X})$$

- Possuímos m_0 exemplos do grupo 0
e mais m_1 exemplos do grupo 1
- Estes dados são usados na fase de treinamento (aprendizagem da regra de classificação)

Dataset e tarefa

Item	Classe Y	Variáveis/Features				$g(X_1, \dots, X_n) = \mathbb{P}(Y = 1 \mathbf{X})$
		X_1	X_2	...	X_n	
1	0	X_{11}	X_{12}	...	X_{1n}	0.07
2	0	X_{21}	X_{22}	...	X_{2n}	0.15
3	0		\vdots			\vdots
\vdots	\vdots		\vdots			\vdots
m_0	0	$X_{m_0,1}$	$X_{m_0,2}$...	$X_{m_0,n}$	0.11
1	1	$X_{m_0+1,1}$	$X_{m_0+2,2}$...	$X_{m_0+1,n}$	0.85
2	1	$X_{m_0+2,1}$	$X_{m_0+2,2}$...	$X_{m_0+2,n}$	0.79
\vdots	\vdots		\vdots			\vdots
m_1	1	$X_{m_0+m_1,1}$	$X_{m_0+m_1,2}$...	$X_{m_0+m_1,n}$	0.93
Novo Item	?	X_1^*	X_2^*	...	X_n^*	$g(X_1^*, \dots, X_n^*) = 0.09$

Dados para os exemplos anteriores

- Imagens-fotos da internet
 - $Y \rightarrow 1$ = imagens com gatos, 0 = imagens sem gato
 - X = intensidade (R,G,B) em cada pixel de cada imagem
- Imagens de sensoriamento remoto (LANDSAT)
 - $Y \rightarrow 1$ = pixel com floresta, 0 = sem floresta
 - X = espectro de intensidade de frequência em cada pixel
- Pacientes chegando no pronto socorro (PS) com ferimento na cabeça
 - $Y \rightarrow 1$ = CTI urgente, 0 = sem urgência
 - X = p medições clínicas rápidas feitas no momento de entrada no PS

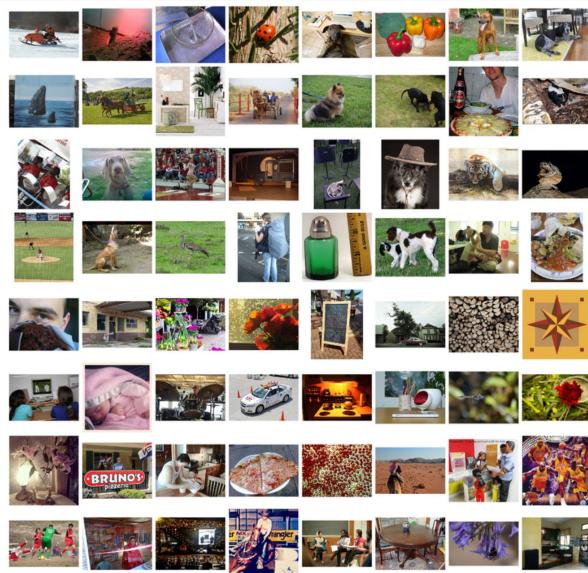


Várias classes, e não apenas duas classes

- Três classes: três espécies de flor
 - $Y \rightarrow 0$ = Iris Versicolor, 1 = Iris Setosa, 2 = Iris Virginica
 - X = largura e comprimento de pétala e de sépala



- Centenas de classes em bancos de imagens, não apenas gatos
Imagenet: 14M images with 20K classes



- Dez classes: reconhecer os dígitos 0, 1, 2, ..., 9

Modelos ML para classificação supervisionada

- Qual a melhor regra de classificação possível e imaginável? Existe? Sabemos qual é?
- Ótima em que sentido?
- No sentido de minimizar erros de classificação (muito mais detalhes à frente)
- Resposta:
 - Sim, existe regra ótima, imbatível
 - Ela é a Regra de Bayes
 - Sabemos qual é esta regra
 - Temos até mesmo a fórmula matemática da regra!
 - Infelizmente...regra de Bayes é incalculável na prática
- ML algoritmos:
 - diferentes modelos para obter uma boa aproximação para a Regra de Bayes

Modelos ML para classificação supervisionada

- Abordagem geral de ML:
 - colete muitos exemplos do que se deseja classificar
 - Em cada exemplo, obtenha a sua verdadeira classe (label Y)
 - Em cada exemplo, obtenha longa lista de features (variáveis em X) que potencialmente afetam ou determinam a classe Y
 - Use criatividade, matemática e capacidade de processamento para rodar algoritmo que seja uma boa aproximação da regra ótima (regra de Bayes)

- George Box:

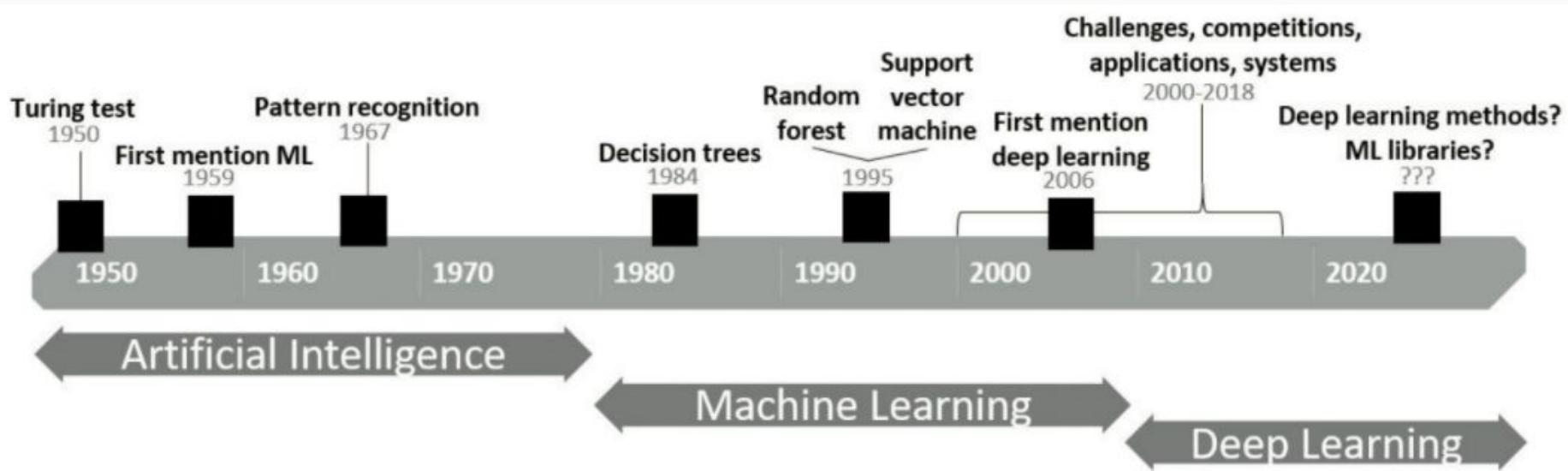
- Todos os modelos são falsos; alguns são úteis.



Os algoritmos de ML

- Todos os algoritmos de ML são tentativas de aproximar-se do ótimo (regra de Bayes)
 - SVM,
 - Regressão Logística,
 - Redes Neurais,
 - Modelos Gráficos Probabilísticos (redes Bayesianas)
 - Árvores de Classificação,
 - Florestas Aleatórias,
 - Boosting,
 - Gradient Boosting, etc...
- Anos 2010:
 - sucesso muito grande de Deep Learning: redes neurais com muitas camadas
 - Muito sucesso em algumas tarefas: Classificação de imagens e NLP (e outras chegando)

Timeline de ML



Redes Neurais - Deep Learning: Por que o sucesso?

Duas razões:

- Primeira:
 - Na maioria dos algoritmos de ML, existe a necessidade de se especificar ou pré-construir as features, as variáveis no vetor X
 - Muito do desempenho do algoritmo depende de sermos capazes de especificar bons preditores para predizer a classe
 - Isto é difícil em muitos problemas, um pesadelo em vários casos.
 - Exemplo: câncer ...

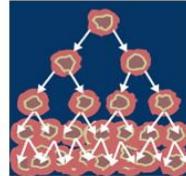
UCI Machine Learning dataset repository

UCI 

Machine Learning Repository
Center for Machine Learning and Intelligent Systems

Breast Cancer Wisconsin (Diagnostic) Data Set
Download: [Data Folder](#), [Data Set Description](#)

Abstract: Diagnostic Wisconsin Breast Cancer Database



Data Set Characteristics:	Multivariate	Number of Instances:	569	Area:	Life
Attribute Characteristics:	Real	Number of Attributes:	32	Date Donated	1995-11-01
Associated Tasks:	Classification	Missing Values?	No	Number of Web Hits:	939316

Source:

Creators:

1. Dr. William H. Wolberg, General Surgery Dept.
University of Wisconsin, Clinical Sciences Center
Madison, WI 53792
wolberg '@' eagle.surgery.wisc.edu
2. W. Nick Street, Computer Sciences Dept.
University of Wisconsin, 1210 West Dayton St., Madison, WI 53706
street '@' cs.wisc.edu 608-262-6619
3. Olvi L. Mangasarian, Computer Sciences Dept.
University of Wisconsin, 1210 West Dayton St., Madison, WI 53706
olvi '@' cs.wisc.edu

Donor:

Nick Street

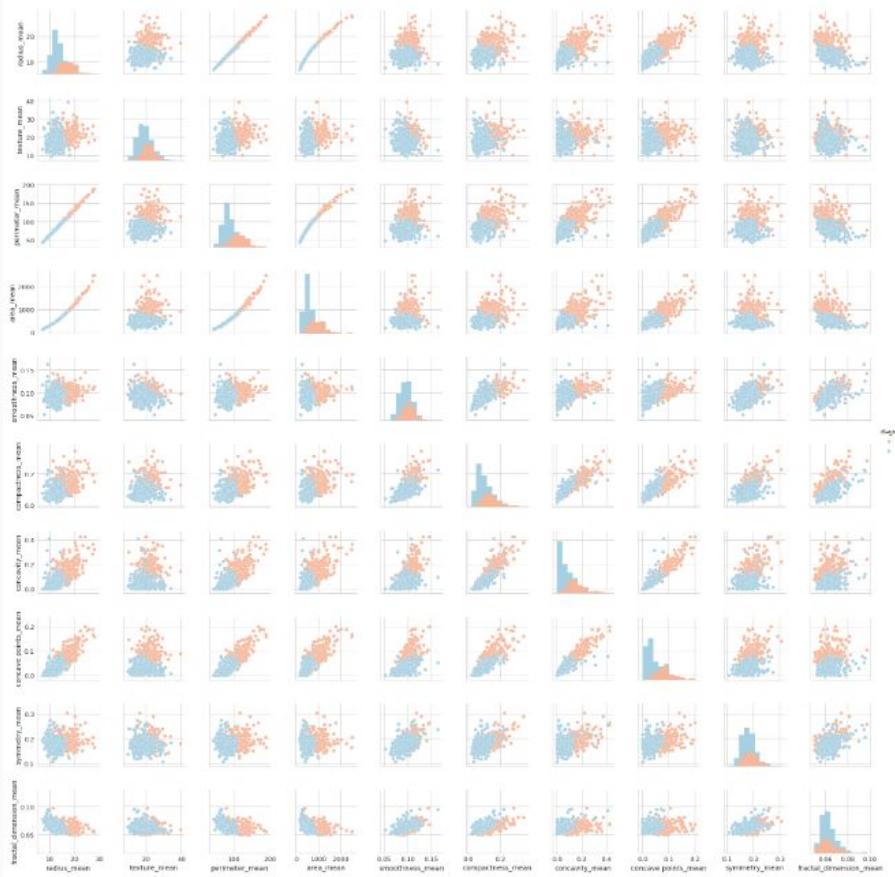
[https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Diagnostic\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic))

UCI: Breast Cancer Wisconsin (Diagnostic) Data Set

- 569 exemplos-pacientes
 - classe 1 = câncer de mama presente
 - ou classe 0 = sem câncer de mama
- Em cada imagem, o núcleo de algumas células foram observados.
- Foram medidas 10 variáveis em cada núcleo:
 - a) radius (mean of distances from center to points on the perimeter)
 - b) texture (standard deviation of gray-scale values)
 - c) perimeter
 - d) area
 - e) smoothness (local variation in radius lengths)
 - f) compactness ($\text{perimeter}^2 / \text{area} - 1.0$)
 - g) concavity (severity of concave portions of the contour)
 - h) concave points (number of concave portions of the contour)
 - i) symmetry
 - j) fractal dimension ("coastline approximation" - 1)

- 10 variáveis em cada núcleo/célula.
- Em cada célula, alguns núcleos.
- No final, 30 features em cada imagem = 10 variáveis * 3 resumos
- Exemplo:
 - uma das variáveis é a raio do núcleo (aprox esférico)
 - vários núcleos em cada imagem → vários raios
 - Deriva-se então 3 features:
 - o raio médio dos núcleos
 - o DP dos raios dos núcleos
 - a médias dos 3 "piores" (maiores) raios
- No final, 30 features em cada imagem.

Matriz de scatterplots com as 10 features de médias por imagem



- Veja como raio, área, e perímetro são "redundantes" como fonte de informação
- Duas features altamente correlacionadas entre si:
 - g) concavity
 - h) number of concave portions points
- Precisa das duas?

Fonte:

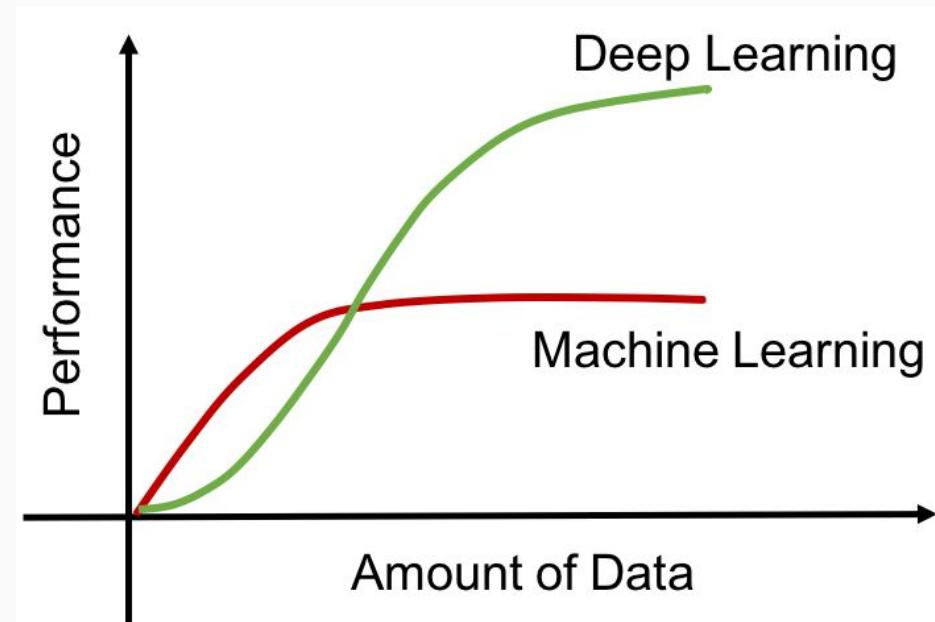
<https://www.kaggle.com/leemun1/predicting-breast-cancer-logistic-regression>

Engenharia de features

- Quais as features devem ser colocadas no modelo de classificação?
- Área? Raio? Perímetro? Todas as 3? Outra coisa?
- Deve-se transformar uma variável?
 - Que tal usar $\text{SQRT}(\text{RAIO MÉDIO})$?
 - ou $\text{LOG}(\text{RAIO MÉDIO}) * \text{DP}(\text{ÁREA})^{**2}$?
 - Transformações não-lineares das features iniciais.
- Precisa listar todas essas variáveis, incluindo as transformações
 - Na verdade, alguns modelos (árvores, SVM tentam obter as não-linearidades automaticamente)
- Esta especificação ***prévia de variáveis não é necessária*** com DL.
- **DL constrói features automaticamente a partir de uma coleção inicial de potenciais preditores.**

Redes Neurais - Deep Learning: Por que o sucesso?

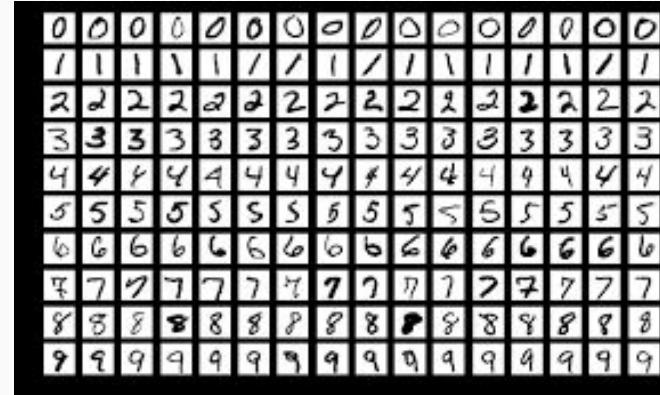
- Segunda razão para o sucesso de DL:
 - usamos um modelo com um número enorme de parâmetros e ele aumenta com os dados.
 - Regressão logística: há efeito de platô (platôs)
 - depois de certa quantidade de dados, não melhoramos substancialmente o modelo para predizer novos casos.
 - Diferente de outros modelos de ML, existe um controle interno-automático de overfitting.



Classificação multi-classes

Classificação com k categorias

- Exemplo canônico: MNIST
- Dados são 70 mil imagens de dígitos manuscritos: cada imagem, um único dígito.
- $Y = 0, 1, 2, \dots, 9$ (resposta é o dígito exibido na imagem)
- Input: o tom de cinza (0-255) em cada pixel da imagem, empilhados como vetor



Estrutura estocástica para o caso multi-classe

- Em cada exemplo, a resposta Y é um rótulo indicando sua classe

$$Y_i = \begin{cases} 1, & \text{com probab } \sigma_1 \\ 2, & \text{com probab } \sigma_2 \\ \vdots & \\ K, & \text{com probab } \sigma_K \end{cases}$$

Especificando as probabilidades das classes: softmax

- Para o caso multi-classes, especificamos um vetor de pesos para cada uma das K classes:

$$\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_K$$

- As probabilidades

$$\sigma_k(\mathbf{x}_i) = \mathbb{P}(Y_i = k | \mathbf{x}_i) \propto e^{\mathbf{w}'_k \mathbf{x}_i}$$

- Elas devem somar 1. Basta normalizarmos agora (modelo softmax):

$$\sigma_k(\mathbf{x}_i) = \mathbb{P}(Y_i = k | \mathbf{x}_i) = \frac{e^{\mathbf{w}'_k \mathbf{x}_i}}{\sum_{j=1}^K e^{\mathbf{w}'_j \mathbf{x}_i}}$$

Resultado final: log-verossimilhança para multi-classe

- Combinando a log-verossimilhança de antes com esta expressão para as probabilidades das classes, temos

$$\begin{aligned}\ell &= \ell(\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_K) \\ &= \sum_{i=1}^m \sum_{k=1}^K [I[y_i = k] \log \sigma_k(\mathbf{x}_i)] \\ &= \sum_{i=1}^m \sum_{k=1}^K \left[I[y_i = k] \log \left(\frac{e^{\mathbf{w}'_k \mathbf{x}_i}}{\sum_{j=1}^K e^{\mathbf{w}'_j \mathbf{x}_i}} \right) \right]\end{aligned}$$

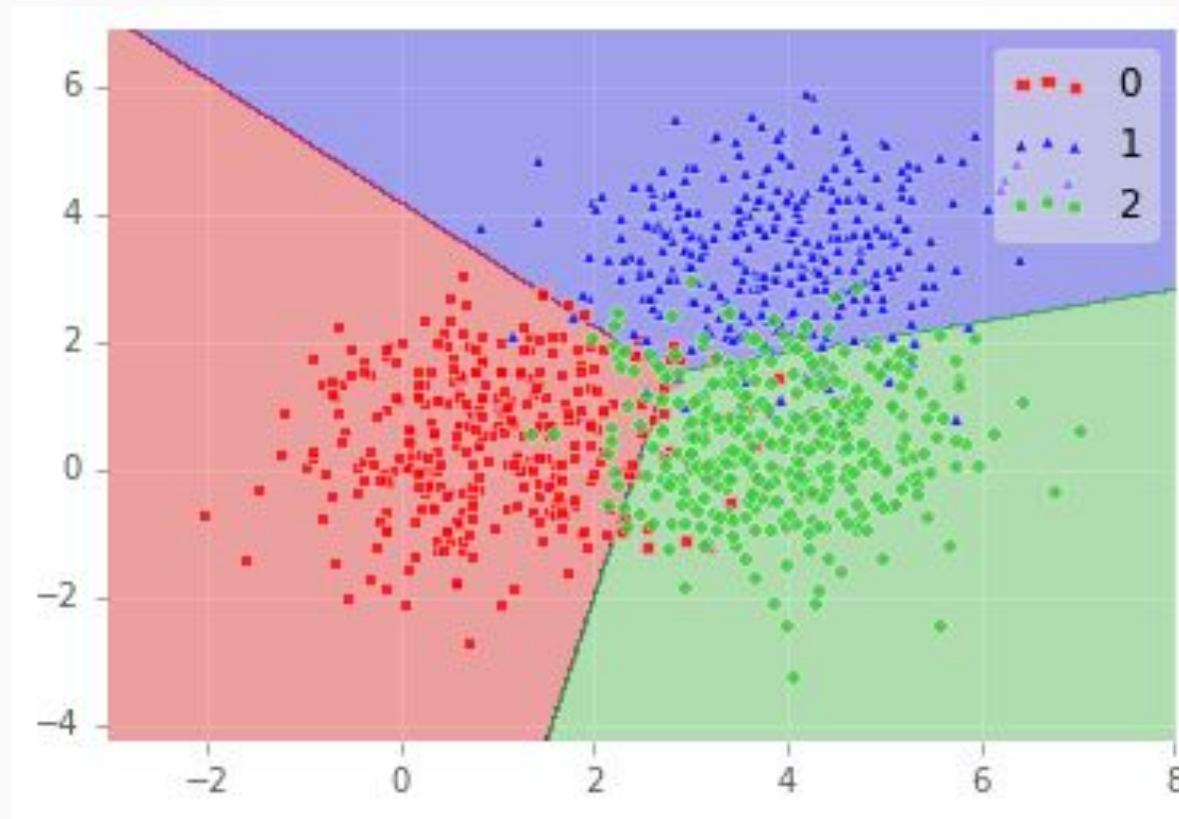
Como obter o estimador de máxima verossimilhança?

- Método numérico de Newton

$$\mathbf{w}^{t+1} = \begin{bmatrix} w_0^{t+1} \\ w_1^{t+1} \\ \vdots \\ w_n^{t+1} \end{bmatrix} = \begin{bmatrix} w_0^t \\ w_1^t \\ \vdots \\ w_n^t \end{bmatrix} - \begin{bmatrix} & J^2(\mathbf{w}^t) \\ & \text{matriz der. parciais 2a ordem de } J \end{bmatrix}^{-1} \underbrace{\nabla J(\mathbf{w}^t)}_{\text{vetor gradiente de } J}$$

- Vetor gradiente da log-verossimilhança.
- Se temos K classes e p inputs, teremos vetor gradiente de dimensão $K^*(p+1)$ -dim
- Matriz hessiana de derivadas parciais de segunda ordem: $K^*(p+1) \times K^*(p+1)$

A regra de decisão e os decision boundaries



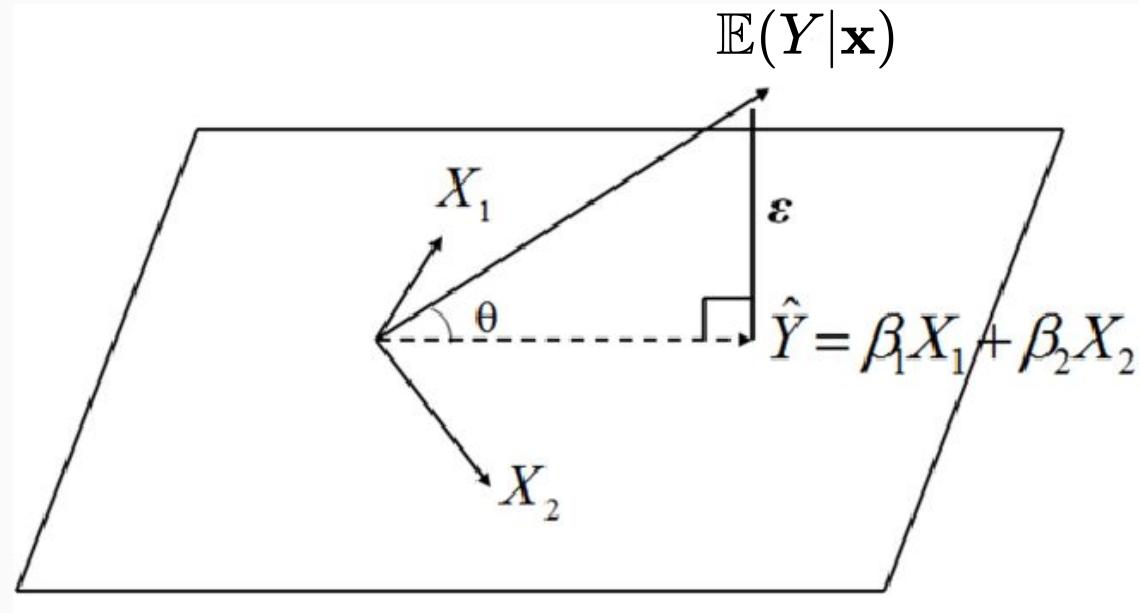
Resumo do problema supervisionado

- Amostra de m exemplos de dados:
 - Uma resposta Y
 - Inputs-features num vetor x
- Objetivo: representar o valor esperado de Y através de uma função matemática dos inputs
 - Dado x, quanto é $\mathbb{E}(Y|x)$?
 - No caso binário: $\mathbb{E}(Y|x) = \mathbb{P}(Y = 1|x)$
 - No caso multi-classe: $\mathbb{P}(Y = k|x)$
 - No caso contínuo (regressão): $\mathbb{E}(Y|x) = \mu(x)$
- Esta função de x depende de parâmetros-pesos $\mathbf{w} = (w_0, w_1, \dots, w_n)$
- Obtenha a log-verossimilhança dos pesos: a probabilidade de gerar os dados da amostra para cada possível valor dos pesos.
- Função de custo: $J(w_0, w_1, \dots, w_n) = -\text{Log-verossimilhança} \rightarrow \text{Minimize a função com Newton /GD}$

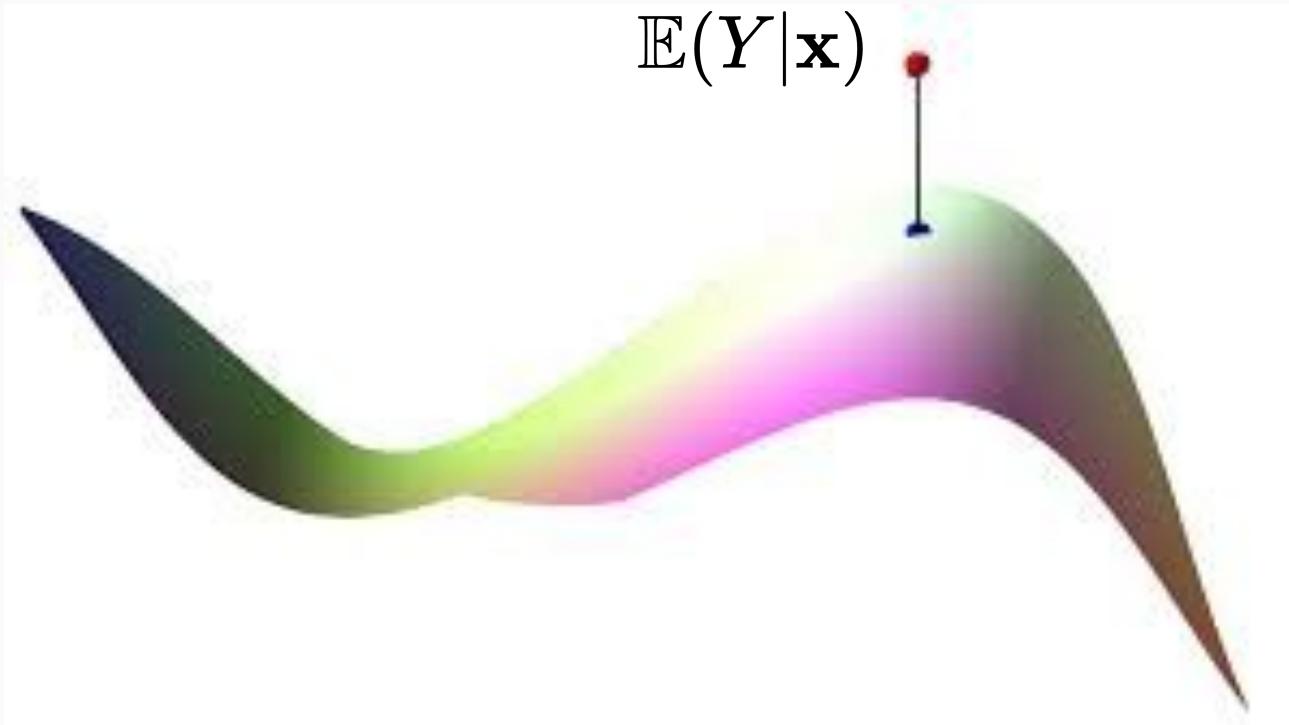
Redes neurais e aproximações em espaços de funções

- O problema de classificação ou de regressão é obter uma boa aproximação para uma função matemática das features:
- $g(\mathbf{x}) = \mathbb{E}(Y|\mathbf{x})$
- Seja $\mathcal{G} = \{g(\mathbf{x})\}$ a classe de todas as possíveis funções
- Cada modelo propõe uma classe de funções:
 - linear em x
 - SVM
 - árvore de classificação/regressão

Regressão linear



Outros modelos

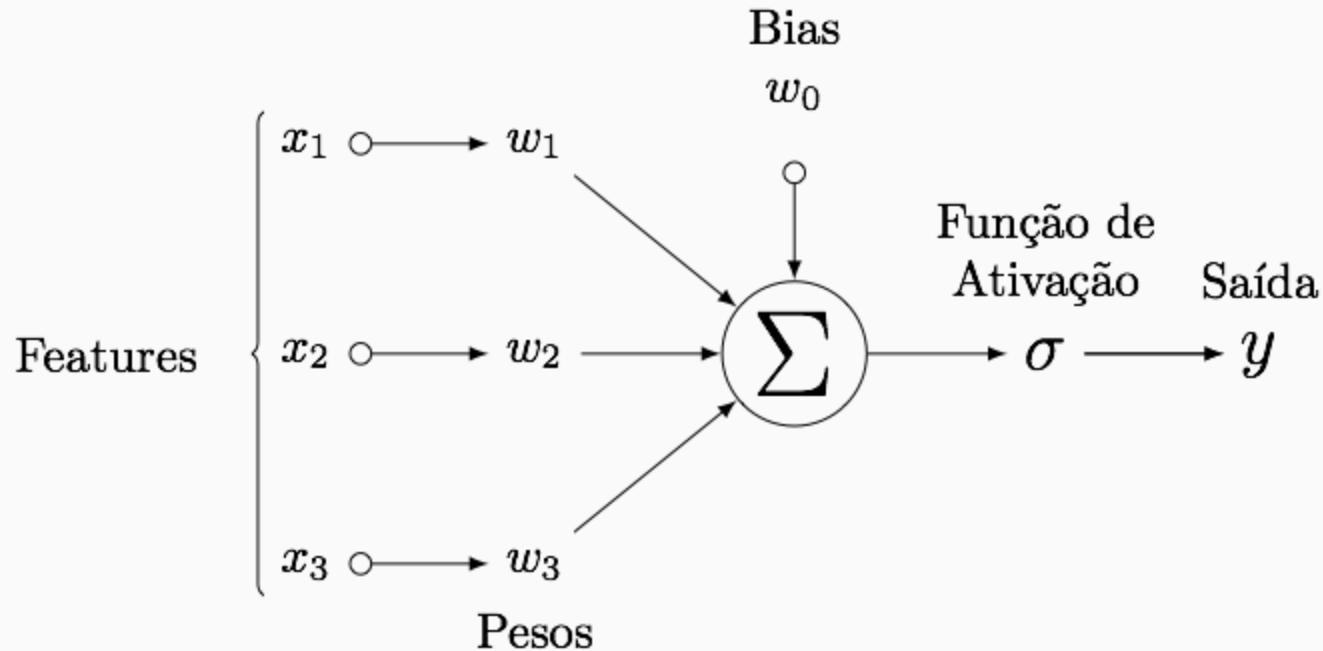


Redes neurais

- Modelos de redes neurais definem um espaço funcional para $E(Y | x)$
- É um espaço parametrizado por coeficientes w (como regressão)
- $\mathcal{G} = \{g_w(\mathbf{x})\}$
- É um espaço muito rico
 - Prova-se que qualquer função $\mathbb{E}(Y|\mathbf{x})$ pode ser arbitrariamente aproximada por elementos $g_w(\mathbf{x})$ de \mathcal{G}
 - Teorema da aproximação universal
- Mas então, o que é uma rede neural?

Redes Neurais

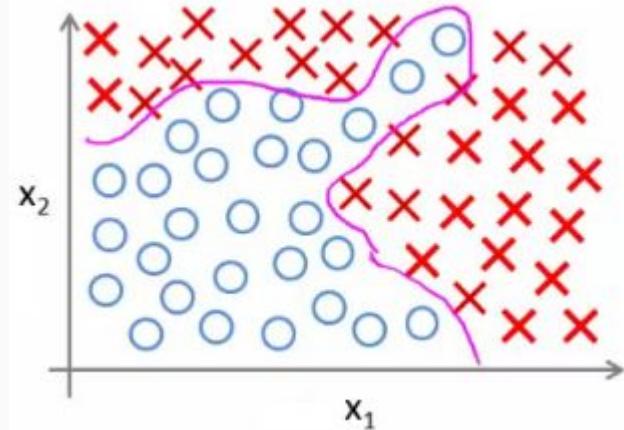
De volta a reg logística como rede neural



Por que usar redes neurais?

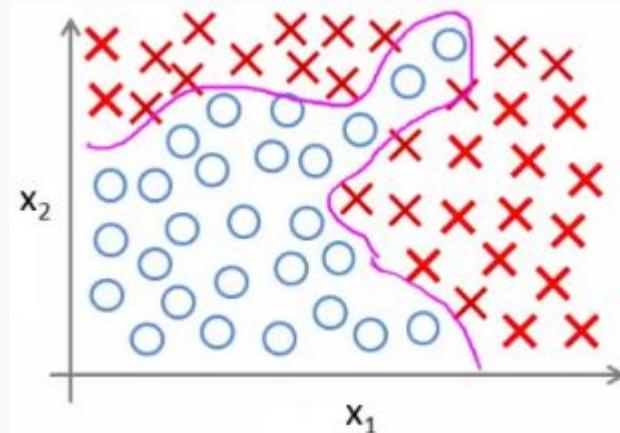
- Para aprender uma decision boundary não-linear com regressão logística → precisamos de muitos termos não lineares das features "básicas"
- Por exemplo, com duas features x_1 e x_2 , podemos buscar os pesos w com

$$\mathbb{P}(Y = 1 | x_1, x_2) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2 + w_5 x_1 x_2 + w_6 x_1^3 + w_7 x_1^2 x_2 + w_8 x_1 x_2^2)}}$$

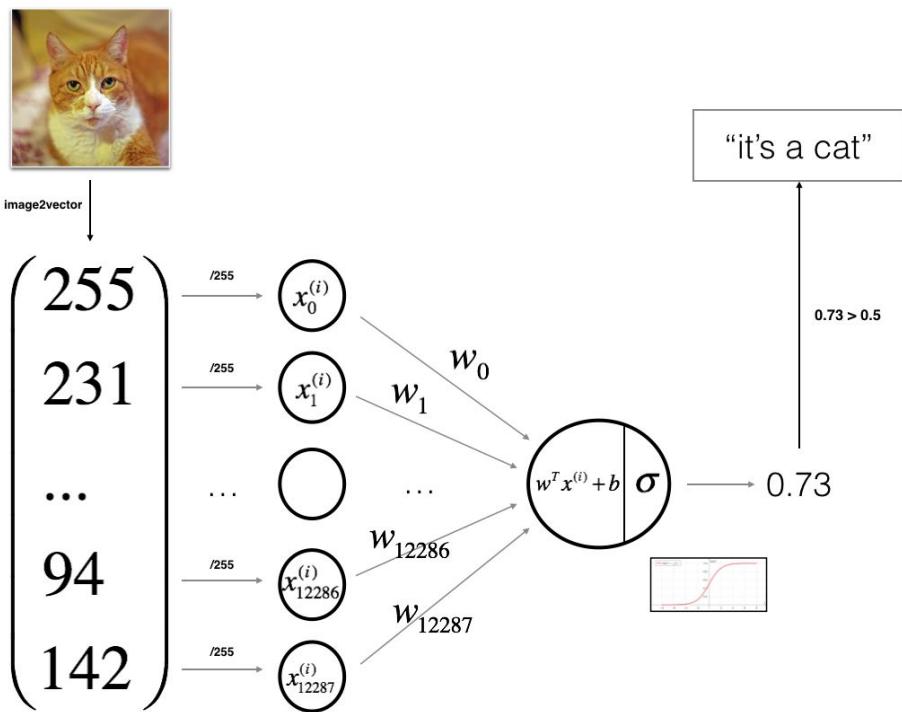


Por que usar redes neurais?

- Mas se temos muitas features no vetor x ...
- Para ter uma função flexível, vamos usar termos polinomiais de x
- Não vai funcionar bem se temos muitas features
- Exemplo: 10 features básicas + 10 features $**2$ + 10 features $**3$ + 45 pares de features (produtos) + 120 triplas = 195 features
- Vai precisar de amostra com $>> 195$ dados
- Além disso, multicolinearidade extrema

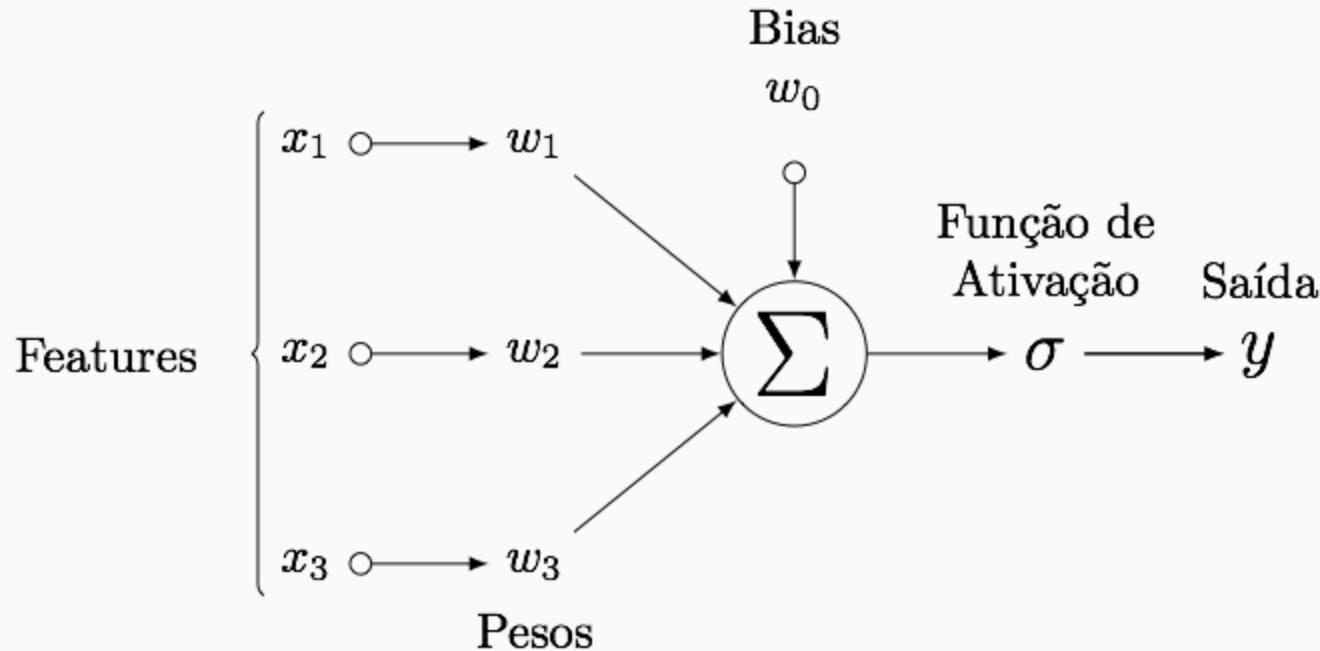


Logística para imagem?



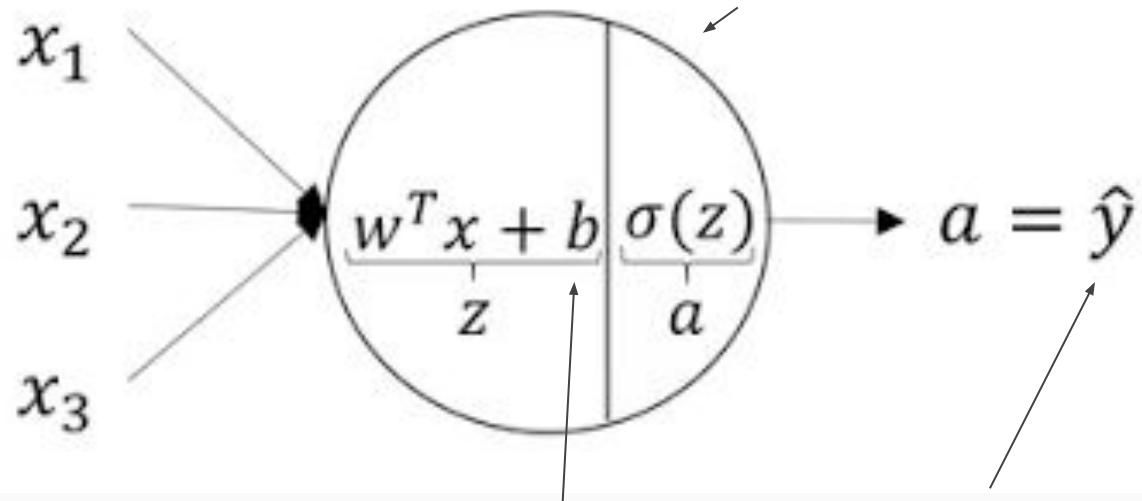
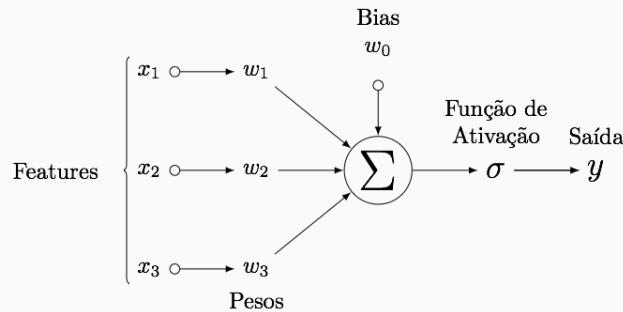
- Espaço de features
- 50×50 pixels em tons de cinza → 2500 pixels (ou features)
- Se RGB, então 7500 features
- Acrescentando os termos quadráticos → 3 milhões de features
- Regressão logística simples não é adequada para modelos complexos
- Redes neurais são muito melhores para modelos não-lineares, mesmo quando o espaço de features é enorme

Introdução à terminologia e notação

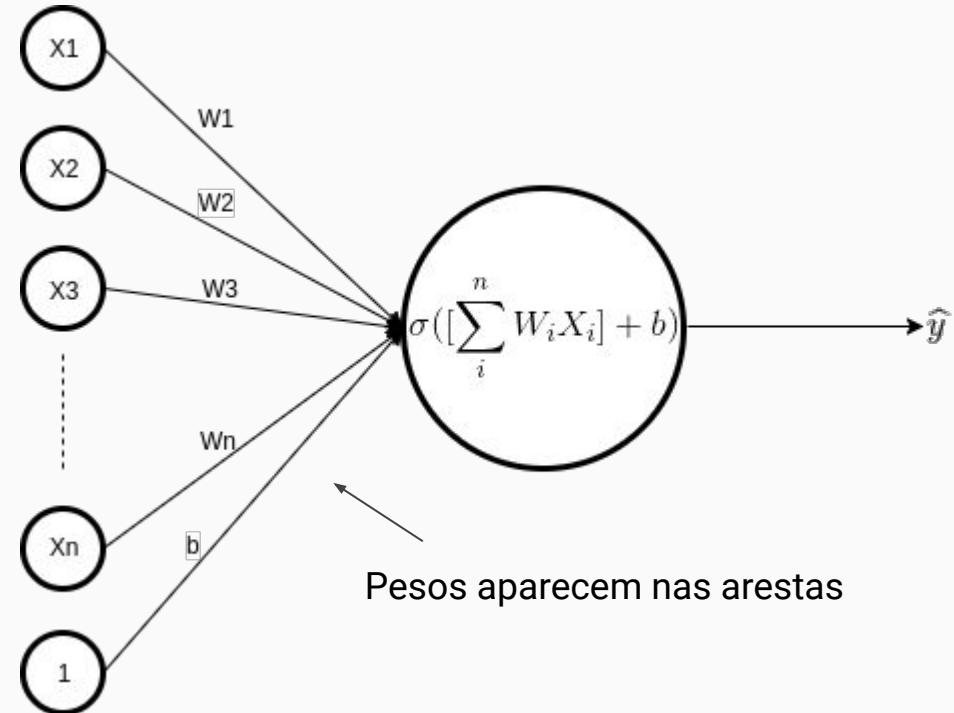
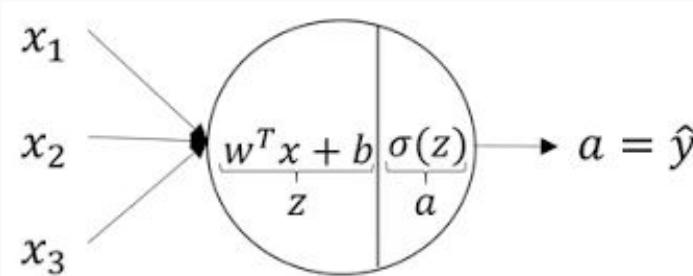
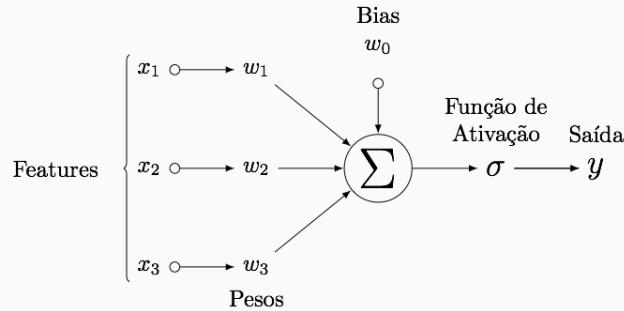


Notação não é uniforme

Representação dos dois passos no neurônio: a combinação linear das features e a ativação com sigma



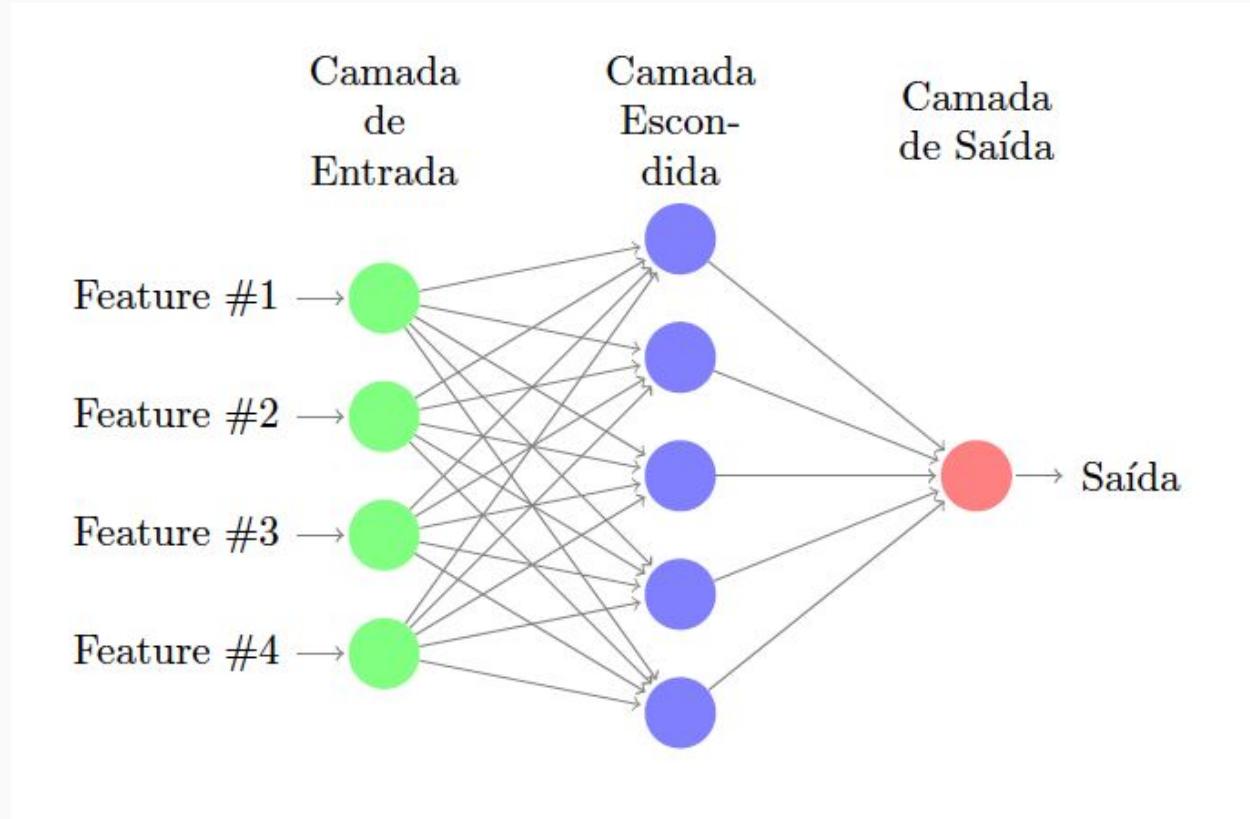
Notação varia entre autores



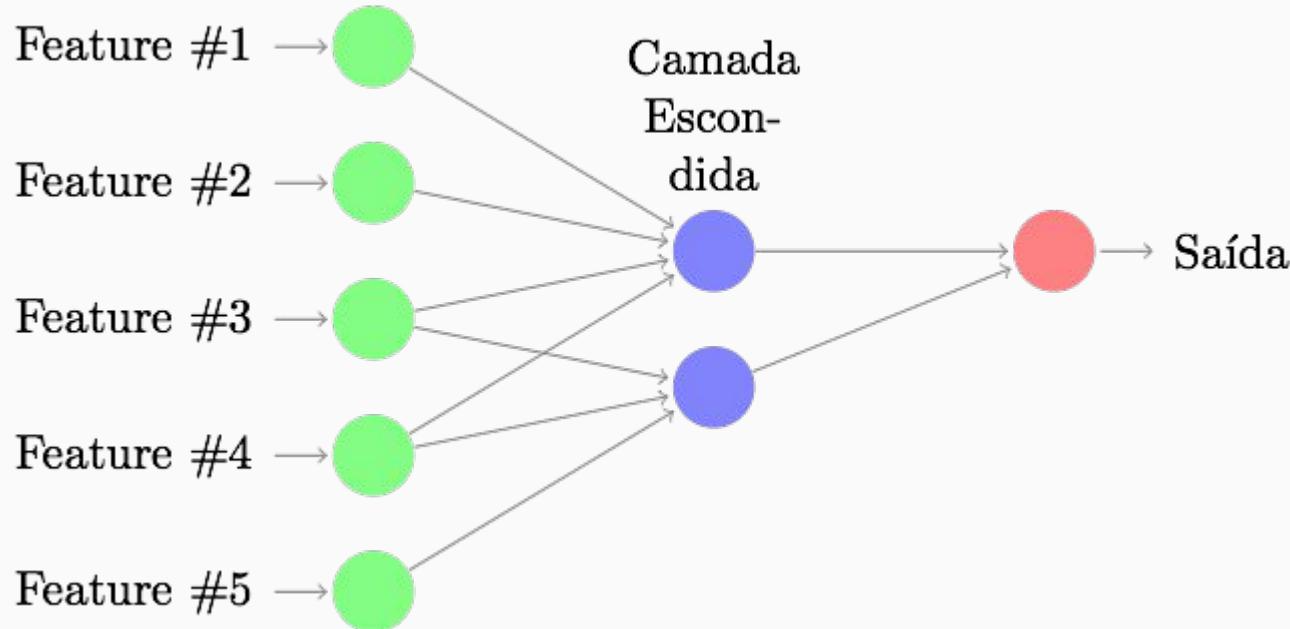
Redes neurais: interpondo camadas

- Este modelo é muito simples.
- Não vamos usar os inputs diretamente para modelar a probabilidade de sucesso
- Os inputs são agregados em fatores/dimensões (as unidades escondidas).
- Cada fator/dimensão mistura os inputs de forma linear num primeiro passo
 - Esta mistura usa pesos para cada input
 - Diferentes fatores escondidos usam pesos diferentes
- Num segundo passo, os fatores latentes são submetidos a uma função de ativação, tal como a logística para gerar NOVOS inputs (inputs transformados).
- Estes inputs transformados são então combinados em mais uma função de ativação (pode ser a logística de novo) para predizer a resposta: $\mathbb{E}(Y|\mathbf{x})$

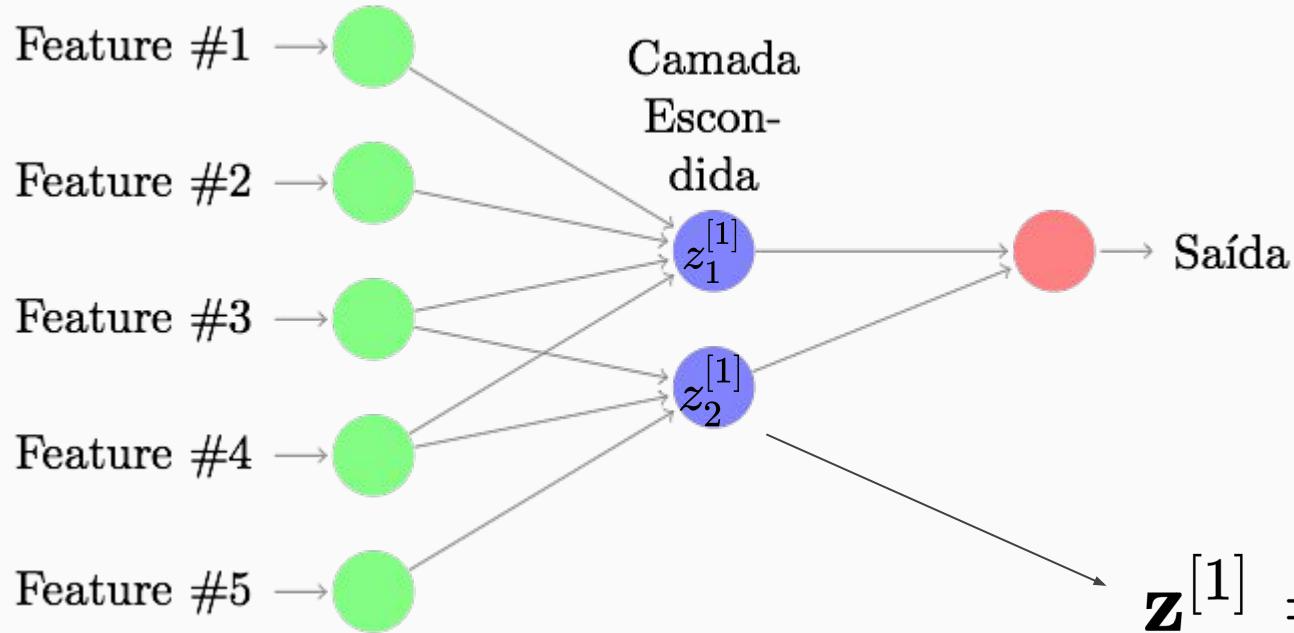
Sobrepondo uma camada escondida



Uma rede com menos neurônios (mais esparsa)

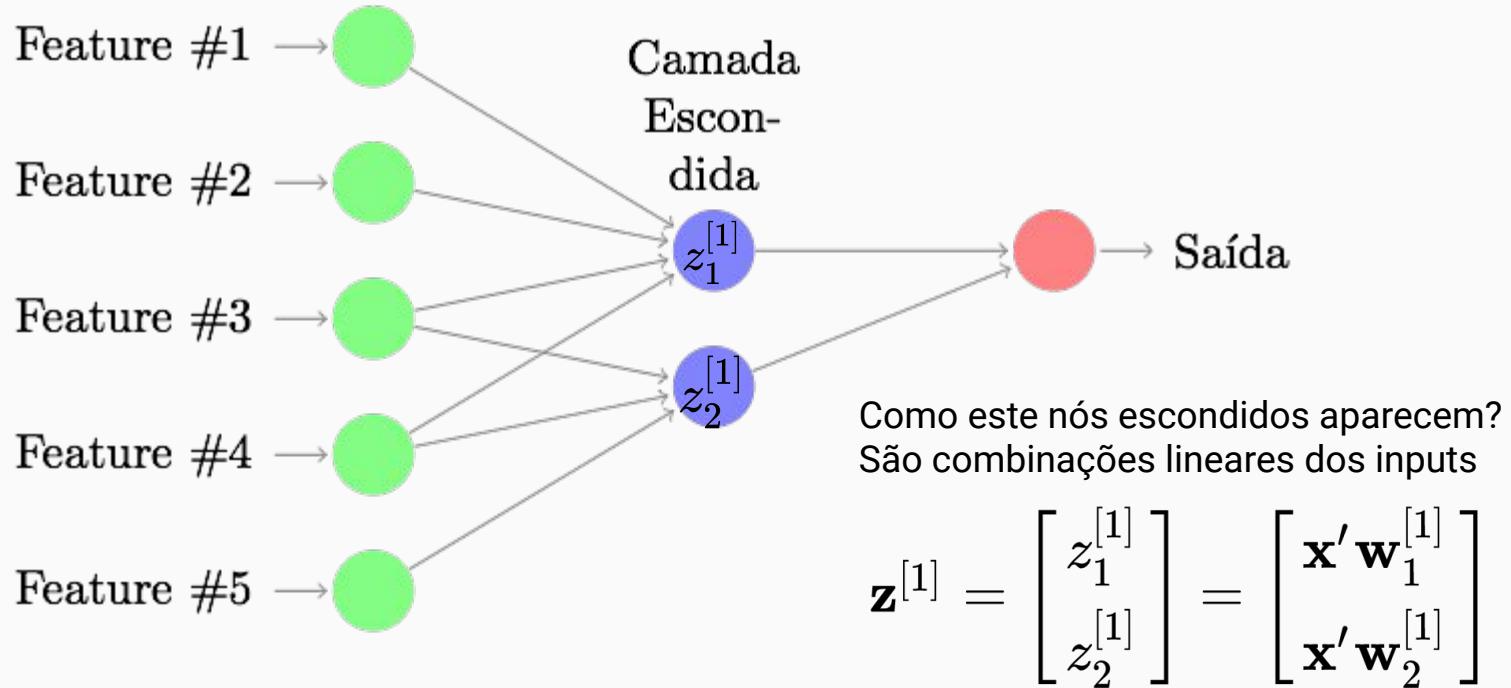


Uma rede com menos neurônios (mais esparsa)

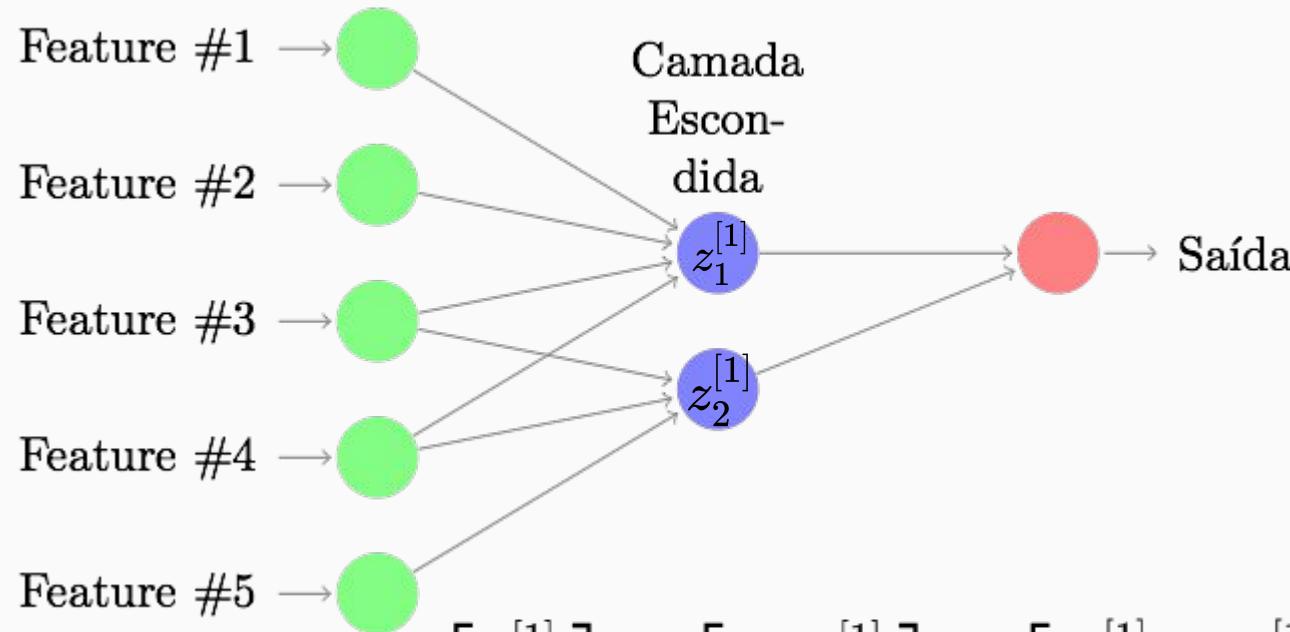


$$\mathbf{z}^{[1]} = \begin{bmatrix} z_1^{[1]} \\ z_2^{[1]} \end{bmatrix}$$

Uma rede com menos neurônios (mais esparsa)

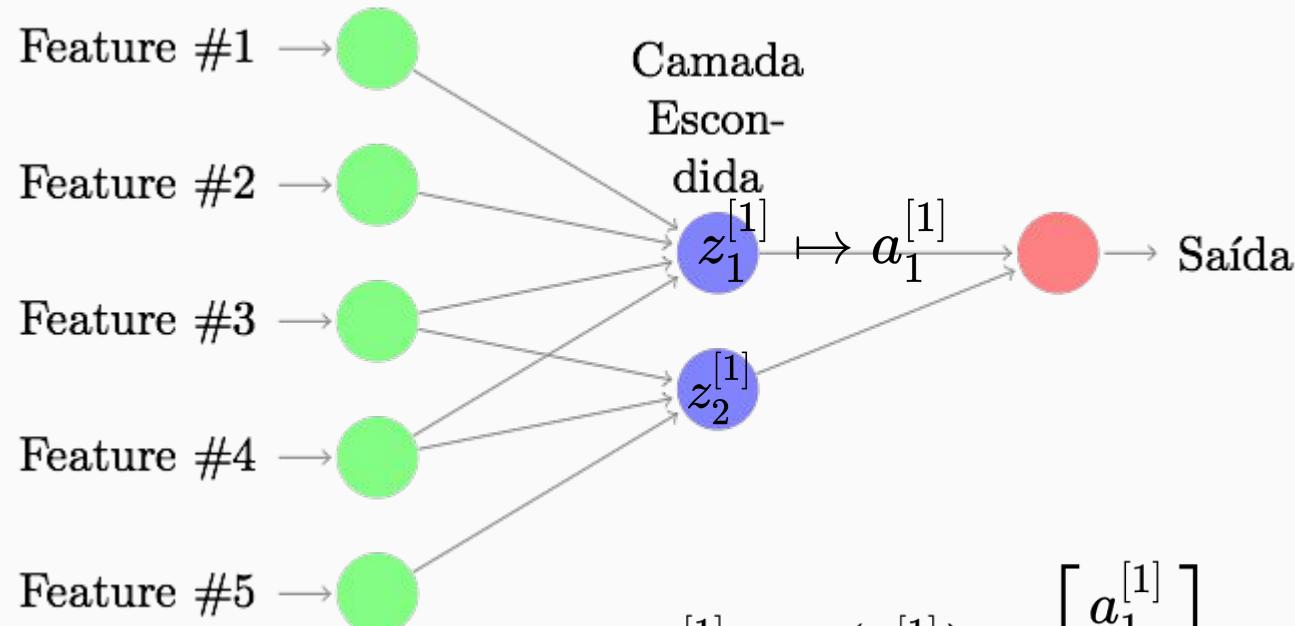


Uma rede com menos neurônios (mais esparsa)



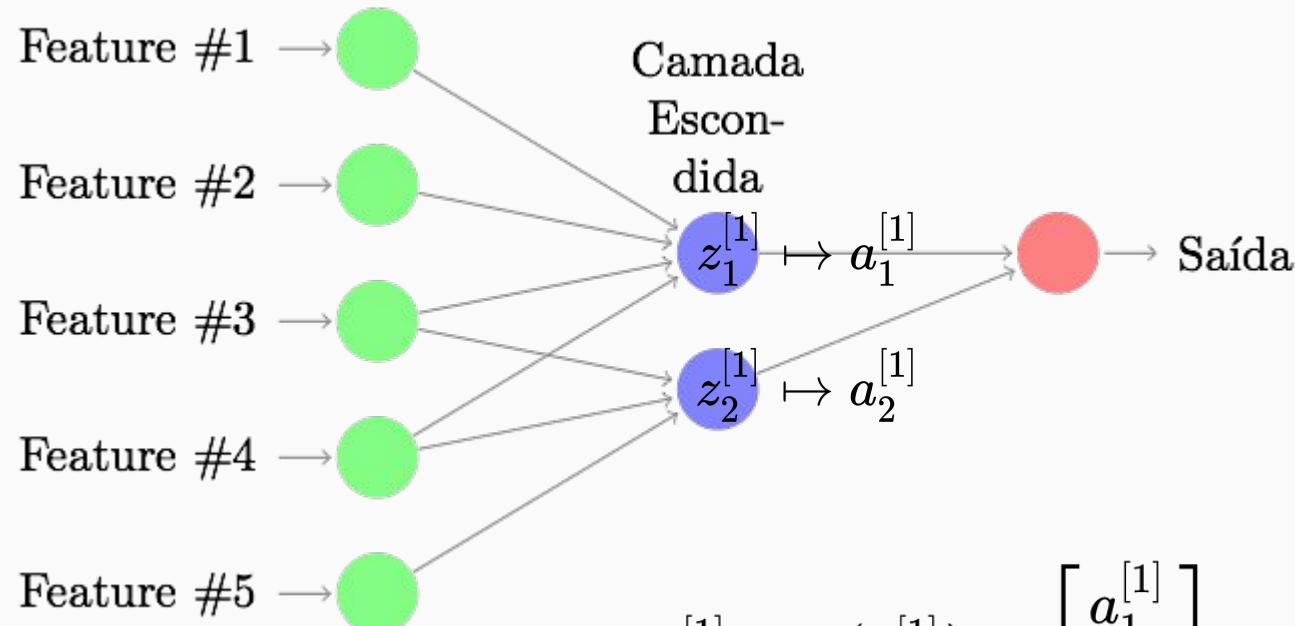
$$\mathbf{z}^{[1]} = \begin{bmatrix} z_1^{[1]} \\ z_2^{[1]} \end{bmatrix} = \begin{bmatrix} \mathbf{x}' \mathbf{w}_1^{[1]} \\ \mathbf{x}' \mathbf{w}_2^{[1]} \end{bmatrix} = \begin{bmatrix} w_{01}^{[1]} + w_{11}^{[1]} x_1 + \dots + w_{n1}^{[1]} x_n \\ w_{02}^{[1]} + w_{12}^{[1]} x_1 + \dots + w_{n2}^{[1]} x_n \end{bmatrix}$$

Uma rede com menos neurônios (mais esparsa)



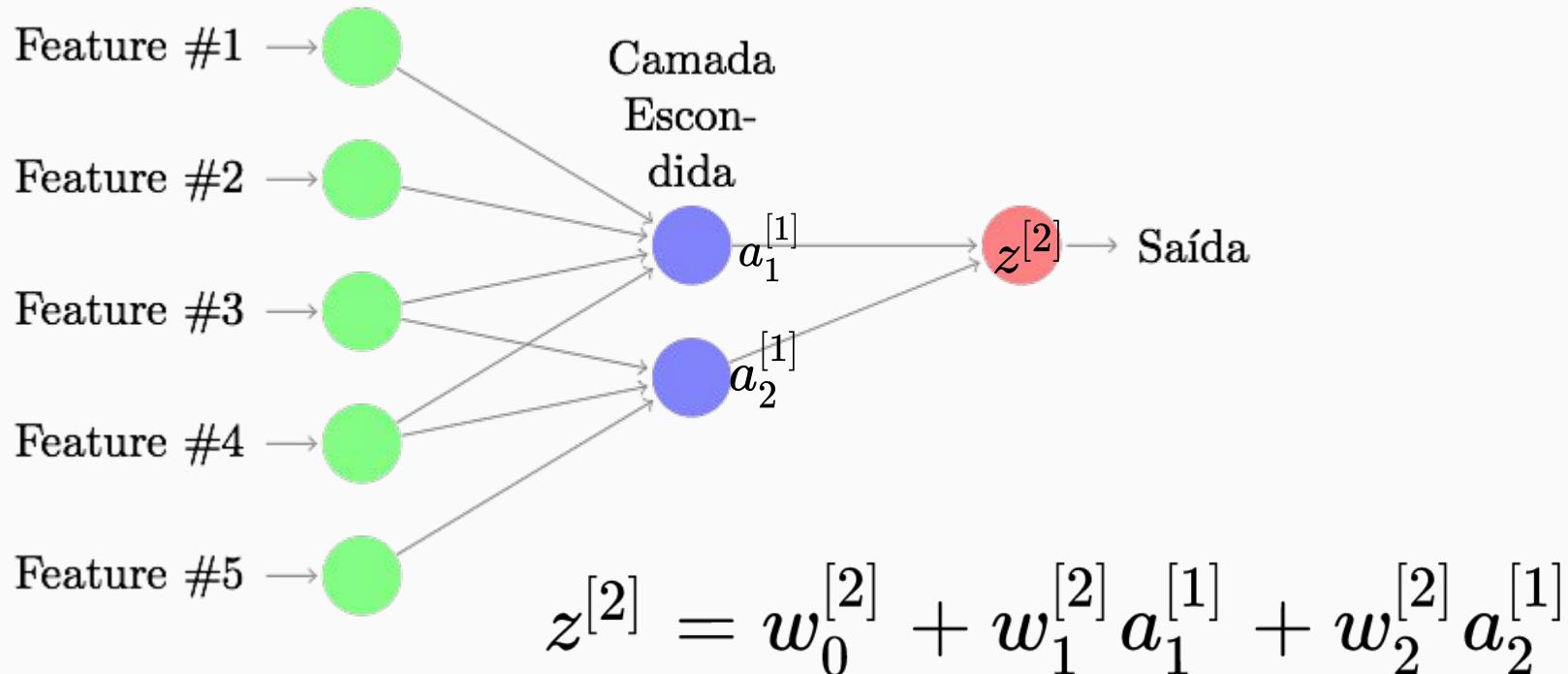
$$\mathbf{a}^{[1]} = \sigma(\mathbf{z}^{[1]}) = \begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \end{bmatrix} = \begin{bmatrix} \sigma(z_1^{[1]}) \\ \sigma(z_2^{[1]}) \end{bmatrix}$$

Uma rede com menos neurônios (mais esparsa)

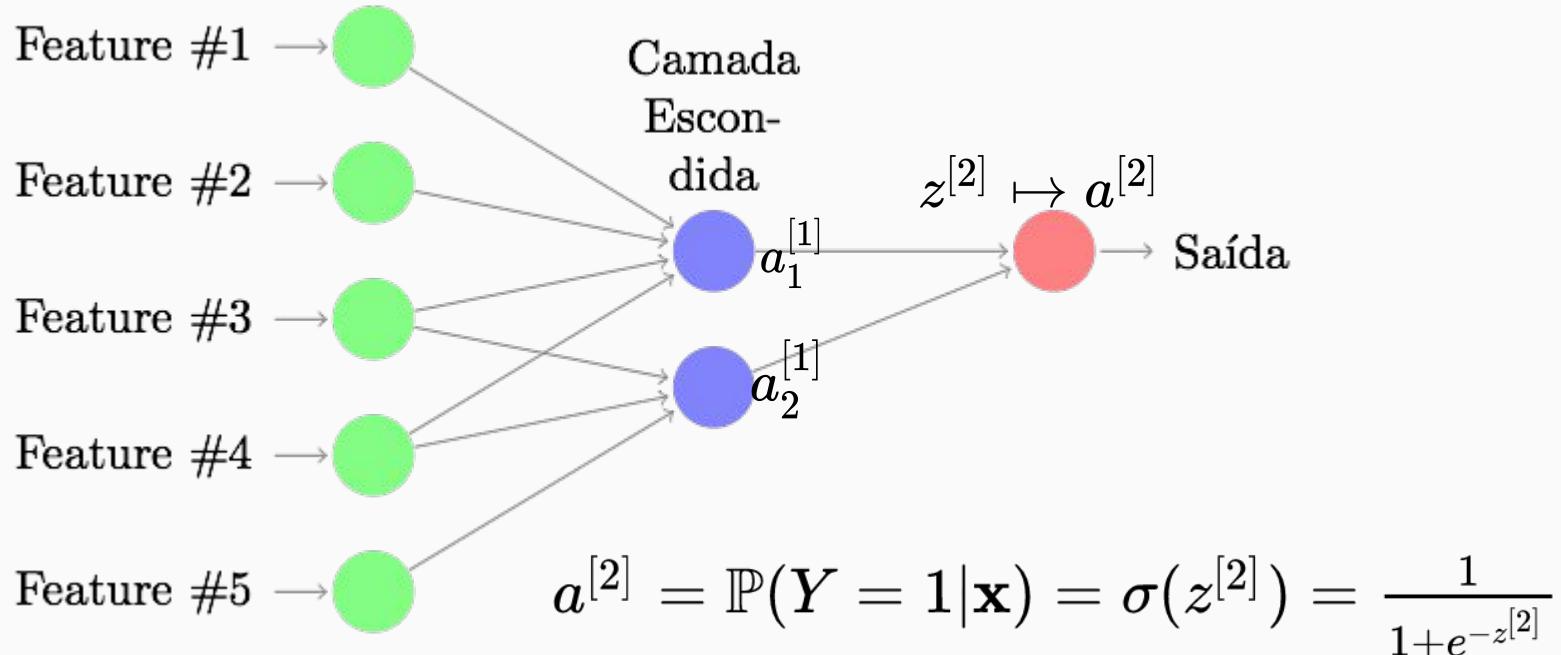


$$\mathbf{a}^{[1]} = \sigma(\mathbf{z}^{[1]}) = \begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \end{bmatrix} = \begin{bmatrix} \sigma(z_1^{[1]}) \\ \sigma(z_2^{[1]}) \end{bmatrix}$$

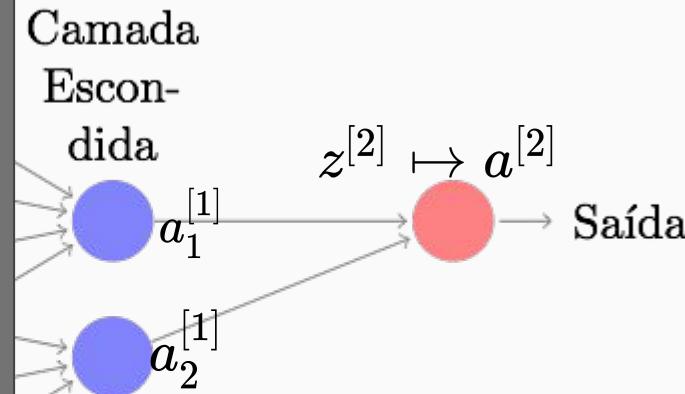
Crie um preditor linear baseado nos a's



A saída binária é uma logística com os a's como features



A saída binária é uma logística com os a's como features



$$z^{[2]} = w_0^{[2]} + w_1^{[2]} a_1^{[1]} + w_2^{[2]} a_2^{[1]}$$

$$a^{[2]} = \mathbb{P}(Y = 1 | \mathbf{x}) = \sigma(z^{[2]}) = \frac{1}{1+e^{-z^{[2]}}}$$

Exemplos

- Preços de imóveis
- Imagine que todos os fatores relevantes podem ser agrupados em três índices:
 - Um deles agraga as informações sobre as características físicas do imóvel:
 - Tamanho, ano de construção, número de banheiros, vagas de garagem
 - Outro agraga informações sobre a vizinhança onde o imóvel está localizado
 - Um terceiro captura a presença ou não de uma vista exterior agradável
- Ao invés de predizer o preço com base nas muitas features individuais, use os 3 índices para fazer isto.
- É como se os os três índices fossem as features de um modelo de regressão

Outra possibilidade para preços de imóveis

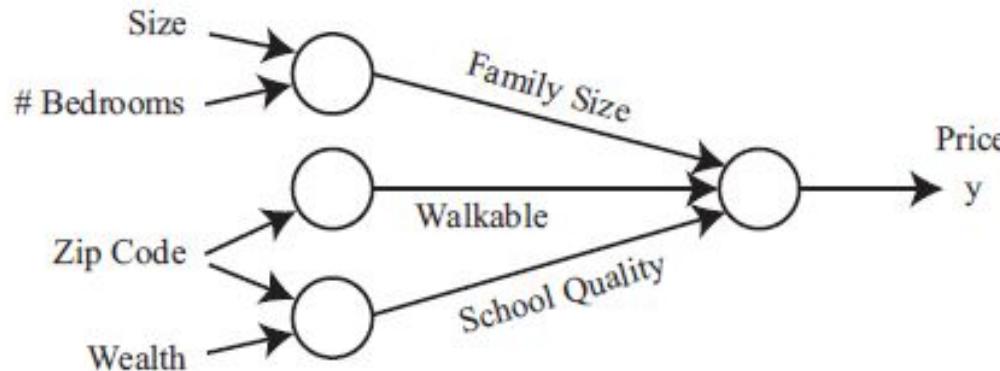
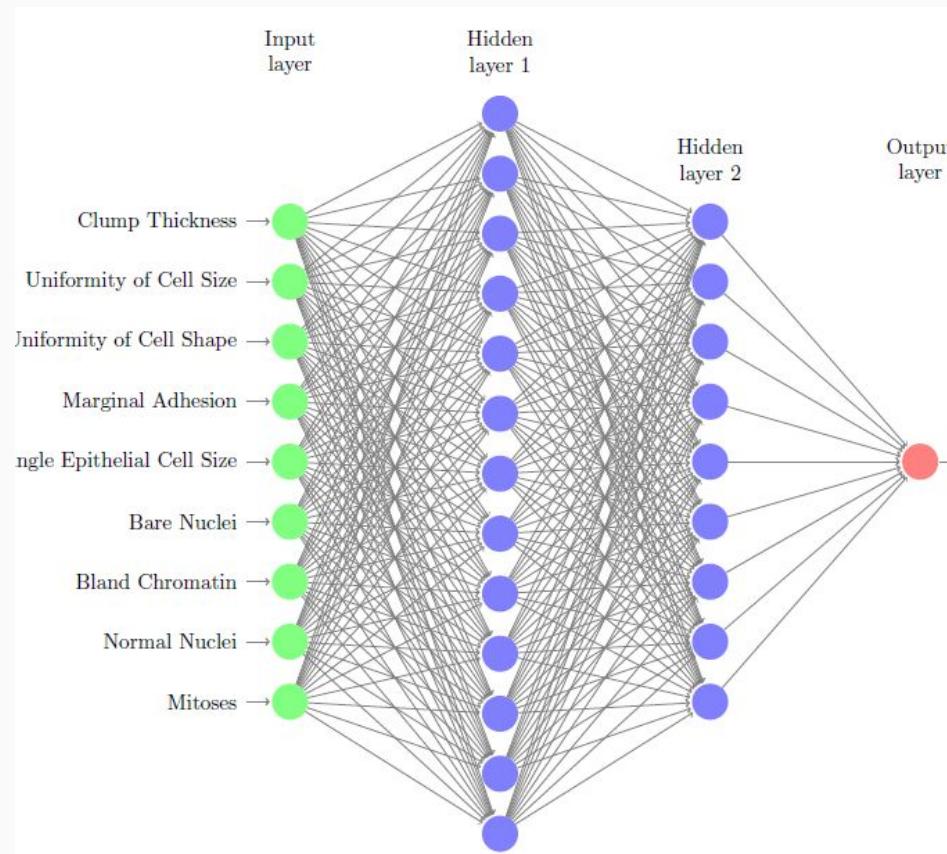


Figure 2: Diagram of a small neural network for predicting housing prices.

De curso de Andrew Ng

Redes com mais de uma camada escondida



DL - um pouco mais formal, mas não muito

X

- INPUT: features ou covariáveis ou variáveis independentes:
- Primeira camada escondida com k nós:
 - k combinações lineares das variáveis da camada anterior

$$\mathbf{z}^{[1]} = \begin{bmatrix} z_1^{[1]} \\ z_2^{[1]} \end{bmatrix} = \begin{bmatrix} \mathbf{x}' \mathbf{w}_1^{[1]} \\ \mathbf{x}' \mathbf{w}_2^{[1]} \end{bmatrix} = \begin{bmatrix} w_{01}^{[1]} + w_{11}^{[1]} x_1 + \dots + w_{n1}^{[1]} x_n \\ w_{02}^{[1]} + w_{12}^{[1]} x_1 + \dots + w_{n2}^{[1]} x_n \end{bmatrix}$$

- ativação com função não-linear (tipicamente ReLU)

Este vetor são as
variáveis de entrada da
próxima camada

$$\longrightarrow \mathbf{a}^{[1]} = \sigma(\mathbf{z}^{[1]}) = \begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \end{bmatrix} = \begin{bmatrix} \sigma(z_1^{[1]}) \\ \sigma(z_2^{[1]}) \end{bmatrix}$$

UCI Machine Learning dataset repository

UCI 

Machine Learning Repository
Center for Machine Learning and Intelligent Systems

Breast Cancer Wisconsin (Diagnostic) Data Set
Download: [Data Folder](#), [Data Set Description](#)

Abstract: Diagnostic Wisconsin Breast Cancer Database



Data Set Characteristics:	Multivariate	Number of Instances:	569	Area:	Life
Attribute Characteristics:	Real	Number of Attributes:	32	Date Donated	1995-11-01
Associated Tasks:	Classification	Missing Values?	No	Number of Web Hits:	939316

Source:

Creators:

1. Dr. William H. Wolberg, General Surgery Dept.
University of Wisconsin, Clinical Sciences Center
Madison, WI 53792
wolberg '@' eagle.surgery.wisc.edu
2. W. Nick Street, Computer Sciences Dept.
University of Wisconsin, 1210 West Dayton St., Madison, WI 53706
street '@' cs.wisc.edu 608-262-6619
3. Olvi L. Mangasarian, Computer Sciences Dept.
University of Wisconsin, 1210 West Dayton St., Madison, WI 53706
olvi '@' cs.wisc.edu

Donor:

Nick Street

[https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Diagnostic\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic))

Wisconsin Breast Cancer Dataset

- Não temos certeza de qual é o melhor número de camadas
- E nem qual é o melhor número de unidades ocultas em cada camada.
- Vamos tentar algumas combinações diferentes:
 - Rede de uma camada escondida com 5 unidades, saída logística
 - Rede de uma camada escondida com 10 unidades, saída logística
 - Rede de uma camada escondida com 15 unidades, saída logística
 - Rede com duas camada escondidas, 5 unidades em cada uma, saída logística
 - Duas camadas com 10 unidades cada, saída logística
 - Duas camadas com 15 unidades cada, saída logística

Comparação

- Usei 80% dos dados (= 455) para treinamento das redes
- Deixei 20% dos dados (= 114) para teste
- Fiz uma análise 5-fold (média sobre 5 partições aleatórias dos dados)

Acurácia = % dos casos de teste corretamente classificada

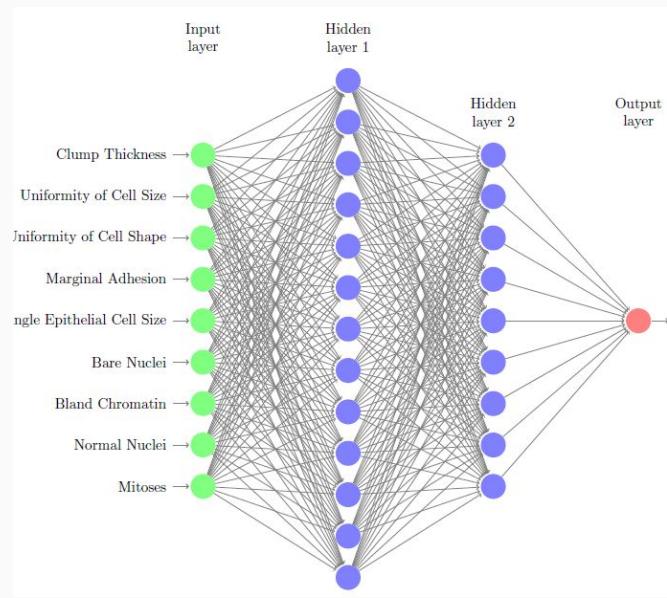
```
## 1 Hidden Layer, 5 Hidden Units: 0.965
## 1 Hidden Layer, 10 Hidden Units: 0.949
## 1 Hidden Layer, 15 Hidden Units: 0.947
## 2 Hidden Layers, 5 Hidden Units Each: 0.956
## 2 Hidden Layers, 10 Hidden Units Each: 0.895
## 2 Hidden Layers, 15 Hidden Units Each: 0.912
```

Ajustando e visualizando uma rede neural

<https://playground.tensorflow.org/>

Arquiteturas de DL

- Arquitetura da rede neural é a estrutura composta por:
 - quantas camadas escondidas
 - quantos nós em cada camada
 - qual a função de ativação a ser usada em cada camada
- Vimos até agora as redes Multi-Layer perceptron (MLP)
- Nó na saída se liga a todos os nós da entrada (precisa de um coeficiente para cada nó)



Arquiteturas especializadas

- Existem algumas arquiteturas especializadas que são mais poderosas ou eficientes:
 - CNN: convolutional neural networks
 - ideal para tratar imagens
 - RNN: recurrent neural networks
 - ideal para tratar sequências
 - Não somente as séries temporais tradicionais mas TEXTOS
 - GNN: graph neural network
 - ideal para dados organizados como grafos (redes sociais, por exemplo)

CNN - Visão geral



What We See

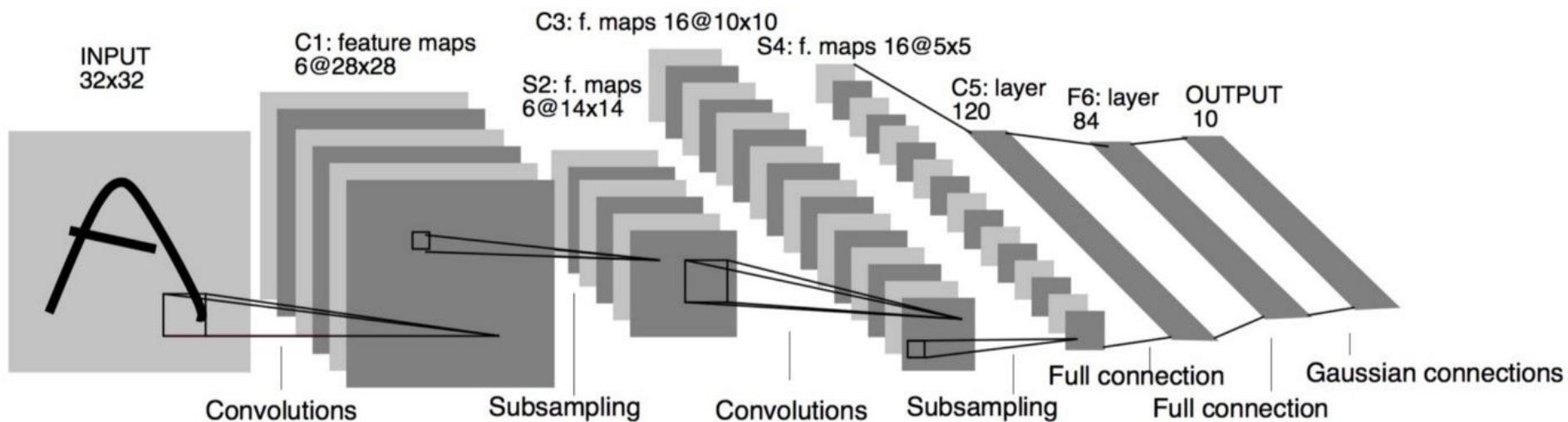
08 02 22 97 38 15 00 40 00 75 04 05 07 78 52 12 50 77 91 08
49 49 99 40 17 81 18 57 60 87 17 40 98 43 69 48 04 56 62 00
81 49 31 73 55 79 14 29 93 71 40 67 53 88 30 03 49 13 36 65
52 70 95 23 04 60 11 42 69 24 68 56 01 32 56 71 37 02 36 91
22 31 16 71 51 67 63 89 41 92 36 54 22 40 40 28 66 93 13 80
24 47 32 60 98 03 45 02 44 75 33 53 78 36 84 20 35 17 12 50
32 98 81 28 66 23 67 10 26 38 40 67 99 54 70 66 18 38 64 70
67 26 20 68 02 62 12 20 95 63 94 39 63 08 40 91 66 49 94 21
24 55 58 05 66 73 99 26 97 17 78 78 96 83 14 88 34 89 63 72
21 36 23 09 75 00 76 44 20 45 35 14 00 61 33 97 34 31 33 95
78 17 53 28 22 75 31 67 15 94 03 80 04 62 16 14 09 53 56 92
16 39 05 42 96 35 31 47 55 58 88 24 00 17 54 24 36 29 85 57
86 56 00 48 35 71 89 07 05 44 44 37 49 60 21 58 51 54 17 55
19 80 81 68 05 94 47 69 28 75 92 13 86 52 17 77 04 89 55 40
04 52 08 83 97 35 99 16 07 97 57 32 14 26 26 79 33 27 98 66
88 36 68 87 57 62 20 72 03 46 33 67 46 55 12 32 63 93 53 69
04 42 16 73 38 25 39 11 24 94 72 18 08 46 29 32 40 62 76 36
20 69 36 41 72 30 23 88 34 62 99 69 82 67 59 85 74 04 36 16
20 73 35 29 78 31 90 01 74 31 49 71 48 86 81 16 23 57 05 54
01 70 54 71 83 51 54 69 16 92 33 48 61 43 52 01 89 19 67 48

What Computers See

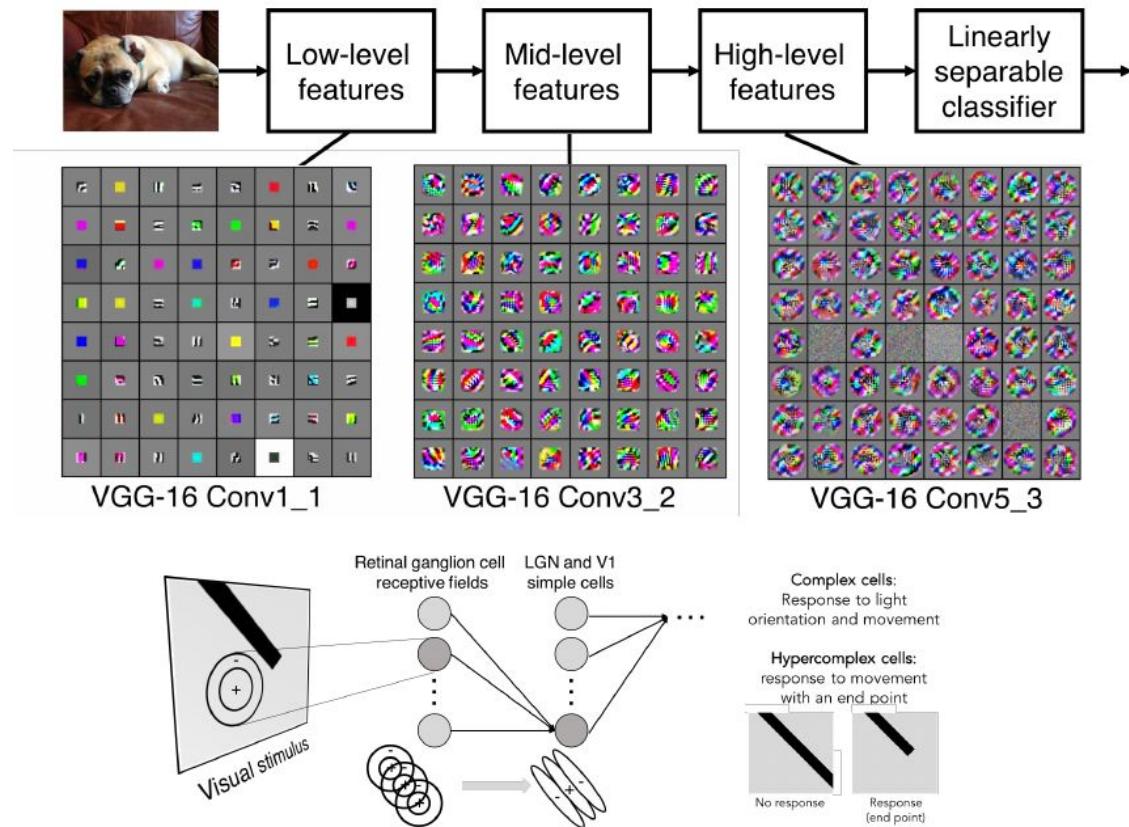
Imagen em tons de cinza: uma única matriz

Imagen colorida: Três matrizes (ou canais), para as cores RGB

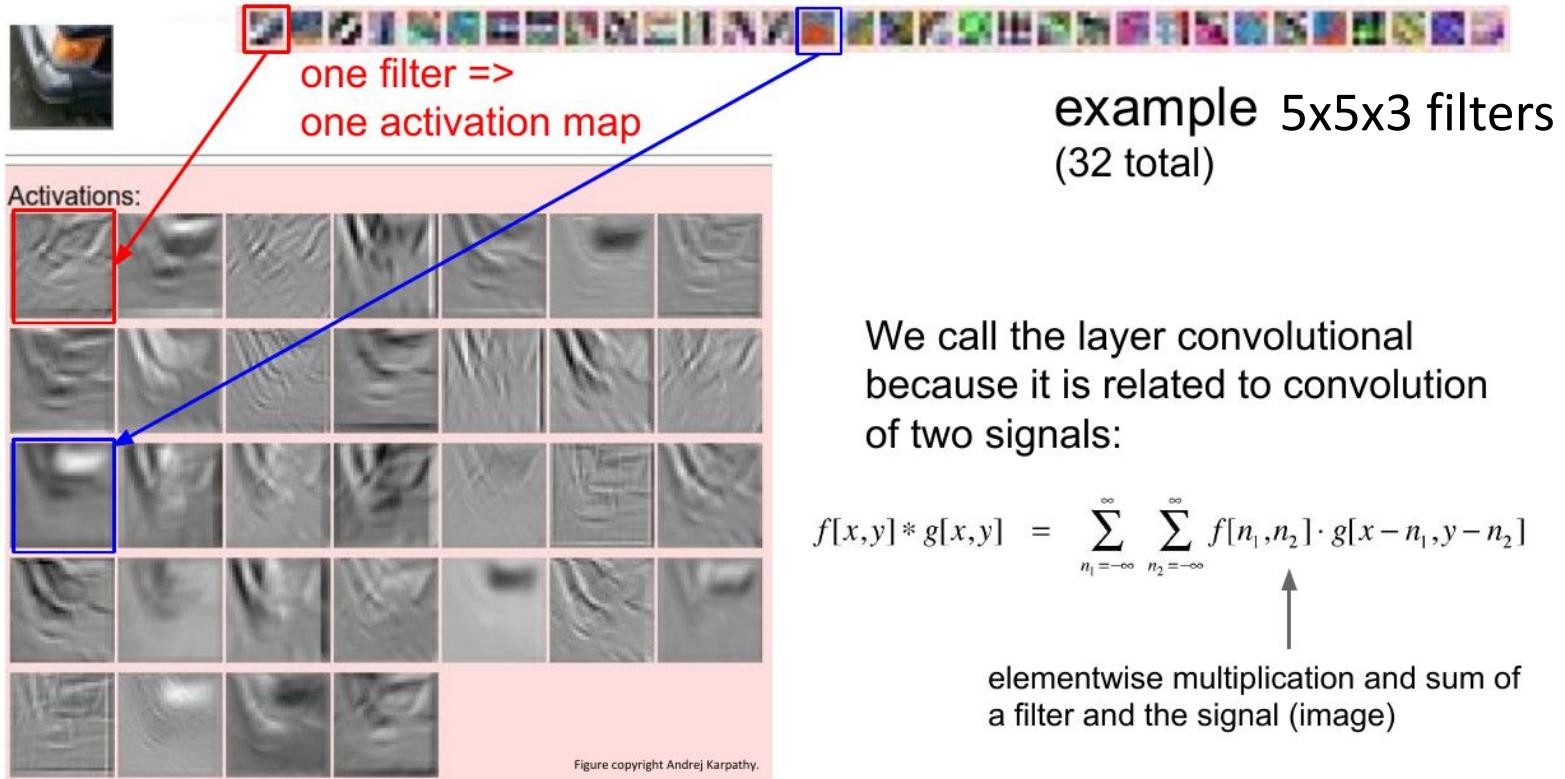
Arquitetura CNN



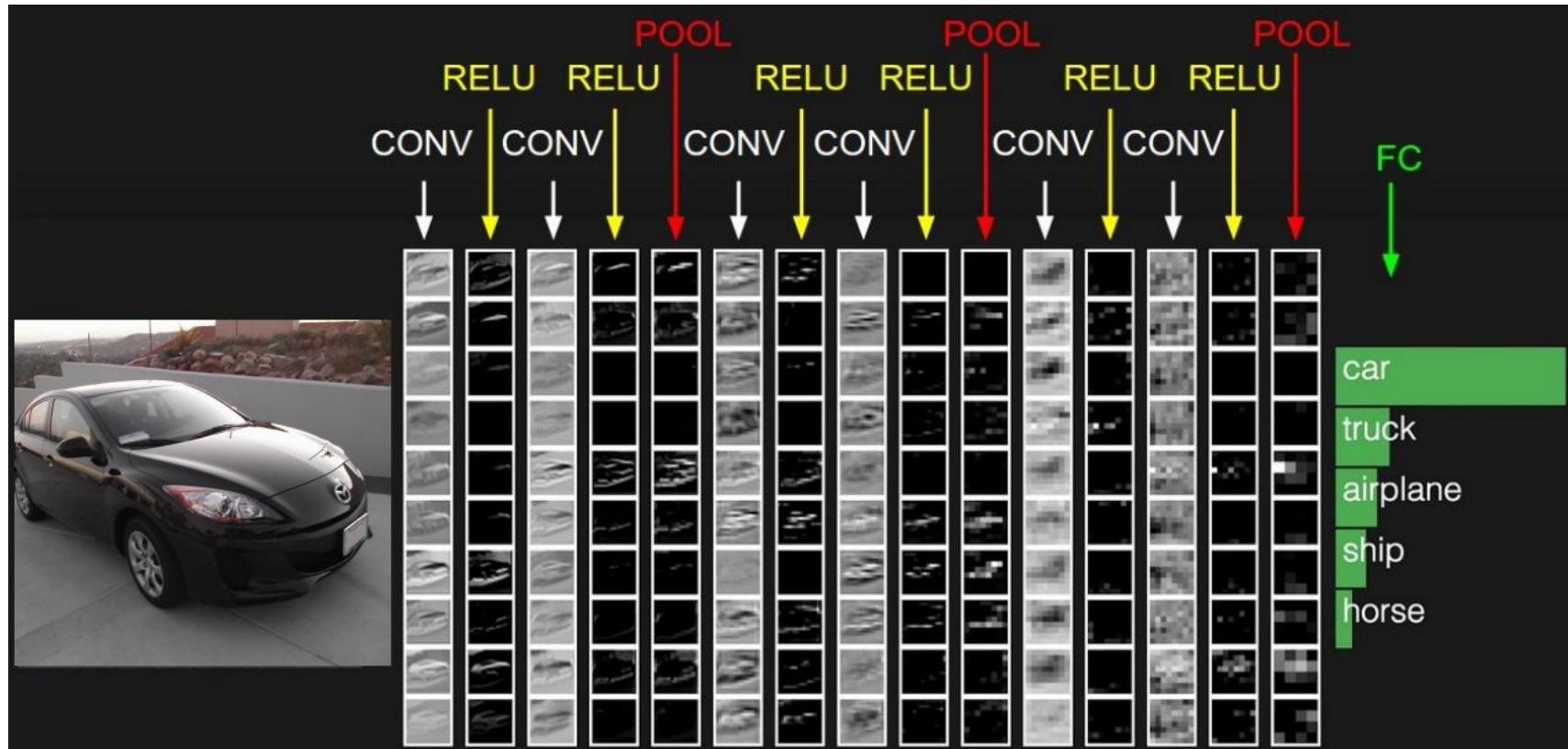
Hierarquia de filtros



Prévia: mapas de ativação



Prévia



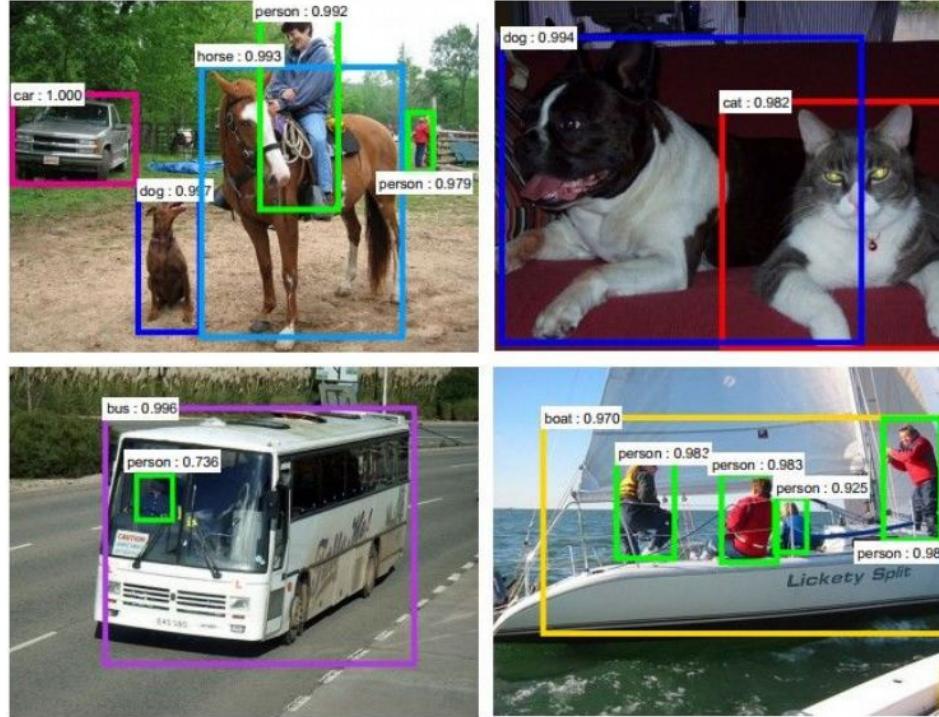
Classificação de imagens



Recuperação de imagens



Detecção de objetos



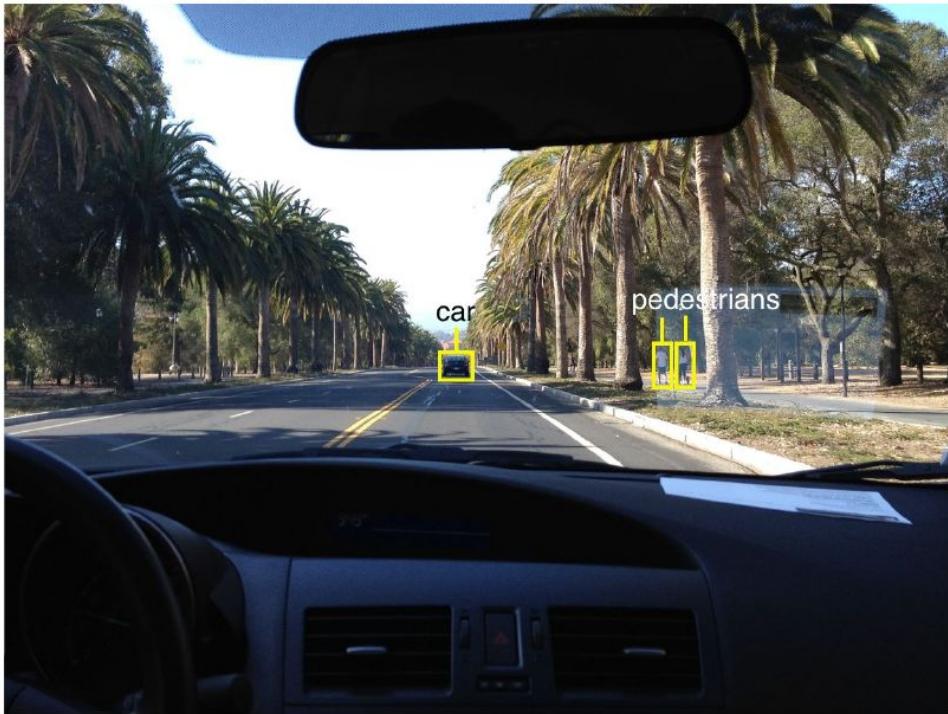
[Faster R-CNN: Ren, He, Girshick, Sun 2015]

Segmentação



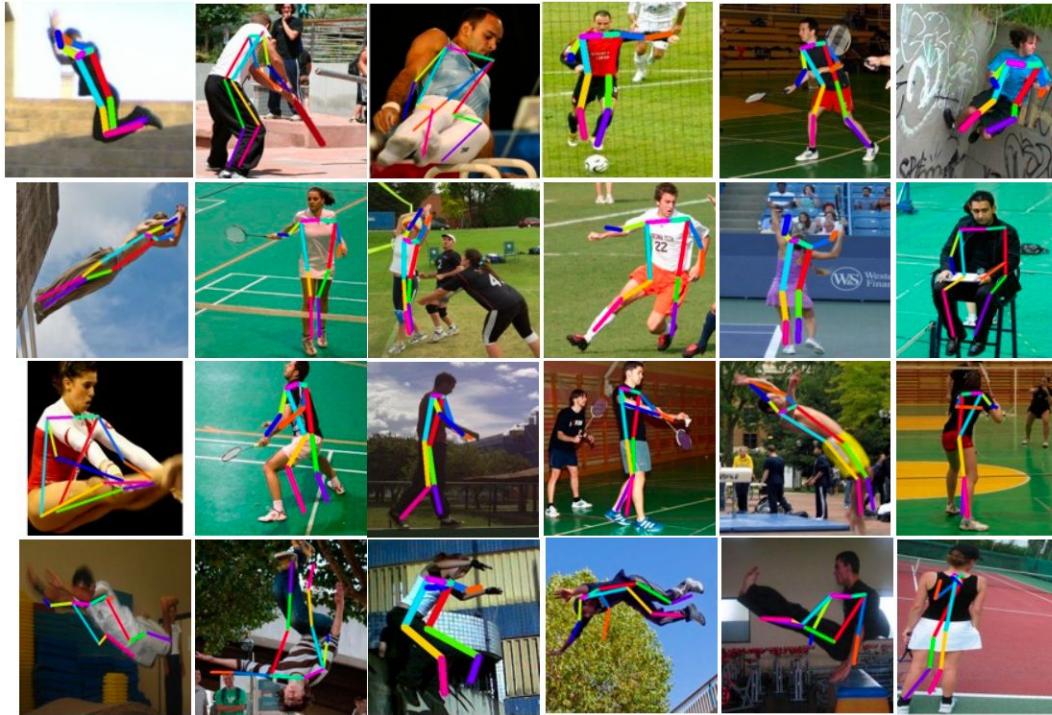
[Farabet et al., 2012]

Self-driving cars

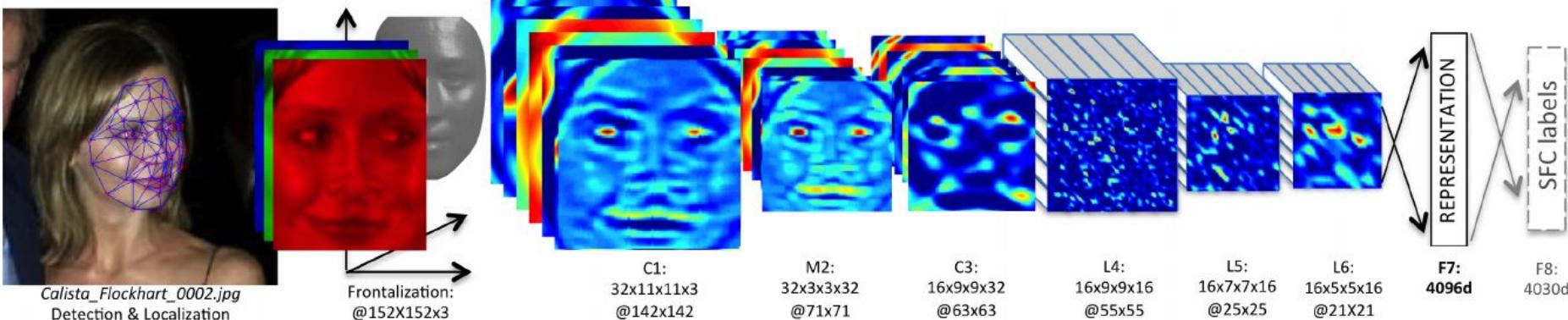
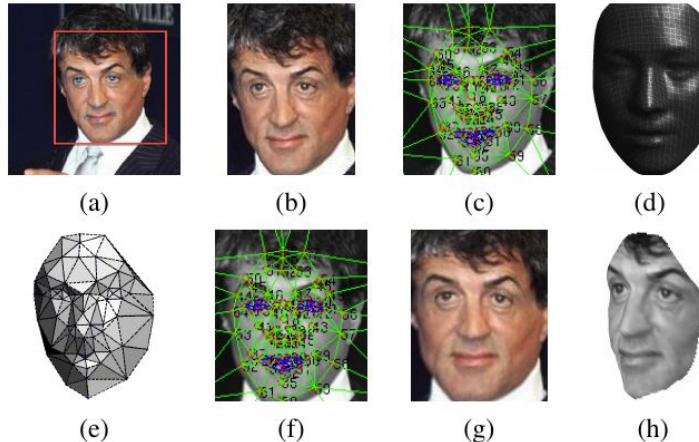


NVIDIA Tesla line

Detecção de pose

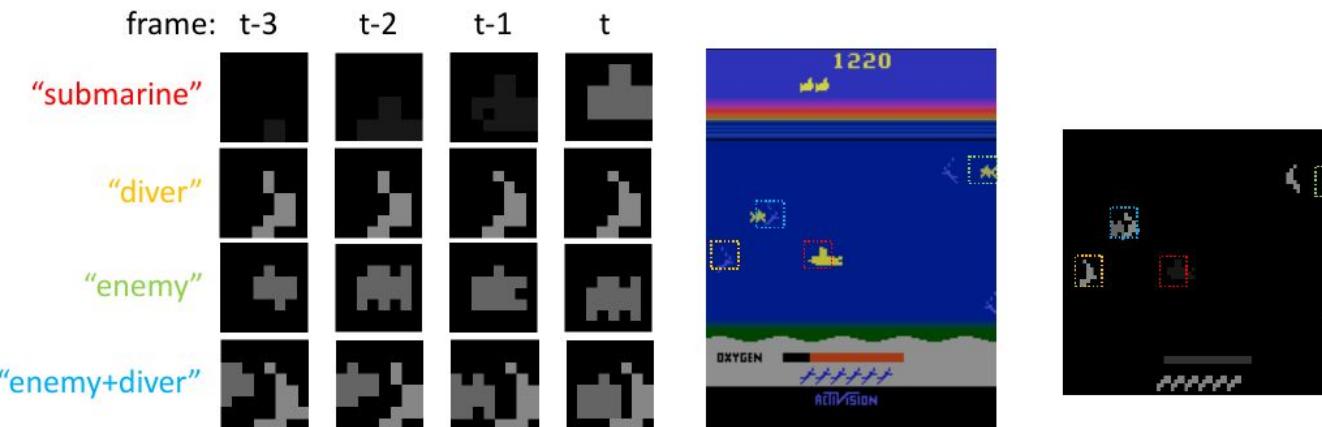
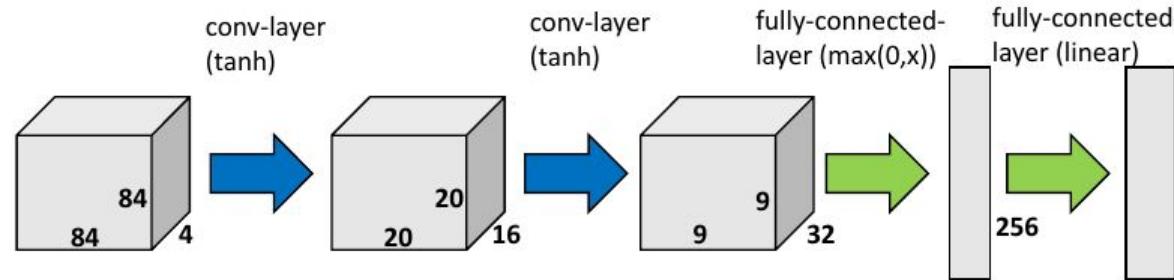


Detecção de face

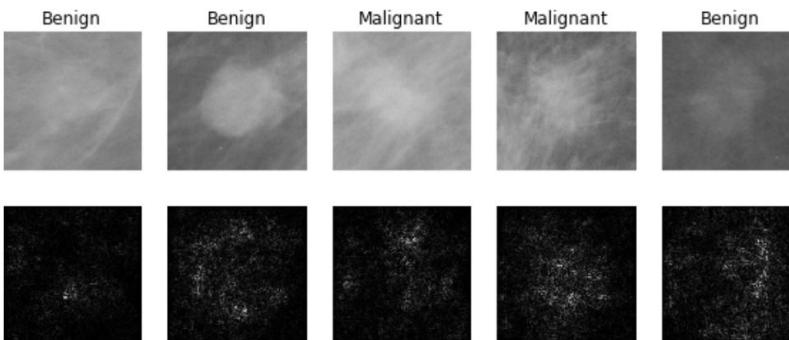


Taigman, Yaniv, et al. Deepface: Closing the gap to human-level performance in face verification, 2014.

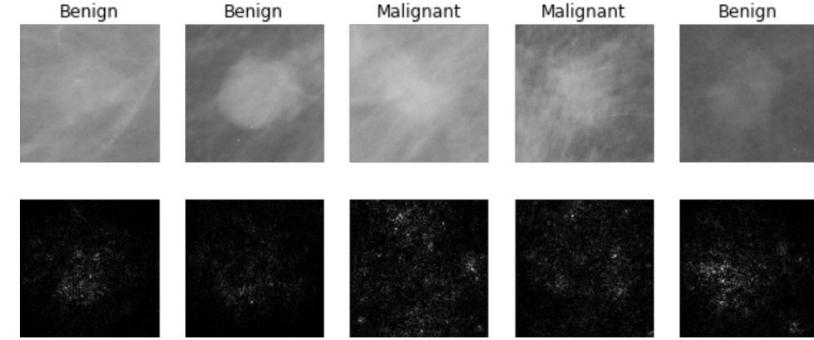
IA para jogos



Classificação de imagens médicas



(a) AlexNet



(b) GoogLeNet

Model	Accuracy	Precision	Recall	# Epochs
Baseline (Aug-Large Context)	0.604	0.587	0.703	35
AlexNet (Aug - Large Context)	0.890	0.908	0.868	30
GoogLeNet (Aug - Large Context)	0.929	0.924	0.934	30

Geração de *captions* de imagens

No errors



*A white teddy bear sitting in
the grass*



*A man riding a wave on
top of a surfboard*

Transferência de estilo



Mais recursos

- Cursos gratuitos no Coursera:
 - Andrew Ng é um professor excepcional
- Cursos em universidades, alguns online, fruto da pandemia
 - nós oferecemos um curso anual na UFMG:
 - <https://github.com/deep-ufmg>
- Iniciativas privadas mas com muito material gratuito
 - Curso-R: empresa de ex-alunos da USP: <https://curso-r.com/>
 - Didática Tech: ex-alunos da UFRGS: <https://didatica.tech/>