Deep Learning

Semana 01 - Aula 01

Renato Assunção - DCC - UFMG

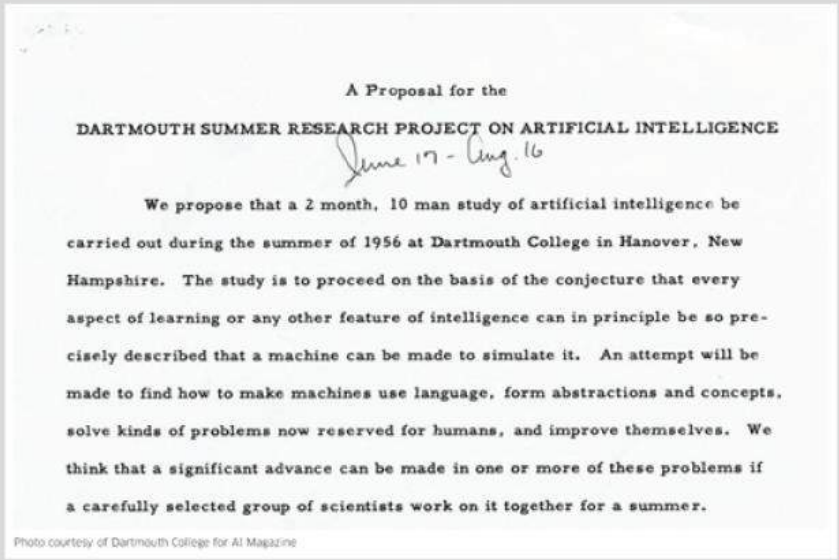
Inteligência Artificial (IA)

- 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

O nascimento de AI

1956

The term "artificial intelligence" is coined at the Dartmouth summer seminar



A Proposal for the

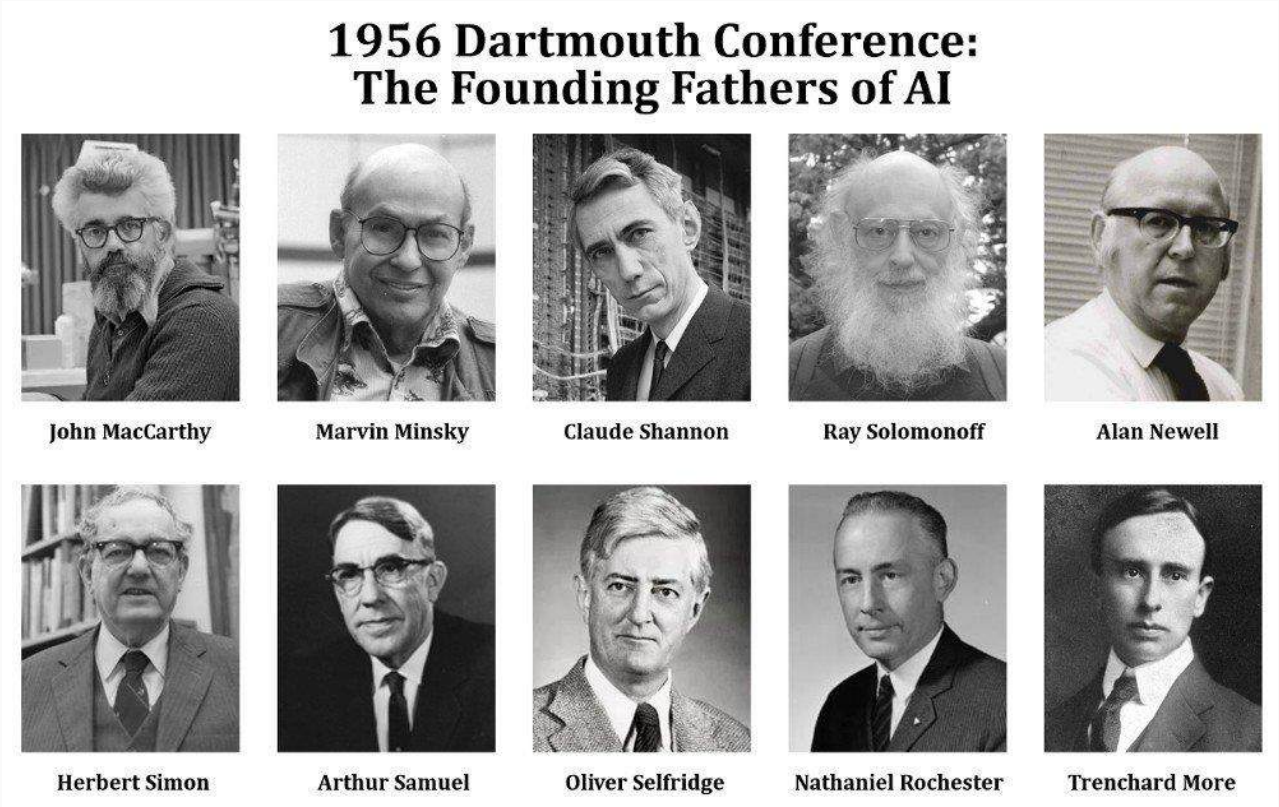
DARTMOUTH SUMMER RESEARCH PROJECT ON ARTIFICIAL INTELLIGENCE

June 17 - Aug. 16

We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.

Photo courtesy of Dartmouth College for AI Magazine

Os pioneiros de Dartmouth



Pioneiros vivos de Dartmouth em 2006

Cinco dos participantes do Dartmouth Summer Research Project on Artificial Intelligence em 1956.

Da esquerda: Trenchard More, John McCarthy, Marvin Minsky, Oliver Selfridge e Ray Solomonoff.

(Foto de Joseph Mehling '69)

IA cresce

- Em 1960, IA viveu um boom, com muito financiamento e muito hype.
- Herbert Simon:
  - "as máquinas serão capazes, dentro de vinte anos, de fazer qualquer trabalho que um homem possa fazer".
- Marvin Minsky:
  - "dentro de uma geração ... o problema de criar 'inteligência artificial' será substancialmente resolvido".
- IA deveria realizar com sucesso qualquer tarefa intelectual que um ser humano pudesse realizar.
- McCarthy, 1955:
  - "todos os aspectos da aprendizagem ou qualquer outra característica da inteligência podem, em princípio, ser descritos tão precisamente que uma máquina pode ser feita para simulá-la.”

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: elicitar 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...

ML - generalidades

- 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 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 de usuários da internet
    - 1 = imagens com pelo menos um gato presente
    - 0 = imagens sem gato

  - 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

Tarefa de classificação supervisionada

- Duas classes de objetos: 0 e 1
- Exemplos:
  - Concessão de crédito para clientes de uma instituição financeira
    - 1 = clientes que não pagarão o crédito no prazo
    - 0 = caso contrário

  - Fragmentos de crânios humanos em escavações arqueológicas
    - 1 = crânio feminino
    - 0 = crânio masculino

  - Adultos de uma população
    - 1 = pessoas com úlcera
    - 0 = sem úlcera

Tarefa de classificação supervisionada

- Mais Exemplos:
  - Duas espécies de flor
    - 1 = Iris Versicolor
    - 0 = Iris Setosa

  - Usuários de um site
    - 1 = clicam num anúncio
    - 0 = não clicam

  - Alunos de uma escola ou curso a distância
    - 1 = evadem sem completar o curso
    - 0 = completam o curso

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*

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

- Com base nas medições em **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*

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

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.

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

- Possuímos m_0 exemplos do grupo 0
  e mais m_1 exemplos do grupo 1

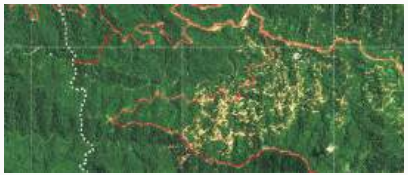
- Estes dados são usados na fase de treinamento (aprendizagem da regra de classificação)

Item	Classe Y	Variáveis/Features X_1	X_2	...	X_n	Classificador $g(X_1,...,X_n) = \mathbb{P}(Y=1\|\mathbf{X})$
1	0	X_{11}	X_{12}	...	X_{1n}	0.07
2	0	X_{21}	X_{22}	...	X_{2n}	0.15
3	0		⋮			⋮
⋮	⋮		⋮			⋮
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
⋮	⋮		⋮			⋮
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^*	$g(X_1^*,...,X_n^*) = 0.09$

Dados para os exemplos anteriores

- Imagens-fotos de usuários da internet
  - Y ⟶ 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 ⟶ 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 ⟶ 1 = CTI urgente, 0 = sem urgência
  - X = p medições clínicas rápidas feitas no momento de entrada no PS

Dados para os exemplos anteriores

- Concessão de crédito para clientes de uma instituição financeira
  - Y ⟶ 1 = clientes que não pagarão o crédito no prazo, 0 = caso contrário
  - X = empréstimo/faturamento, tempo como cliente, saldo mensal, ...

- Fragmentos de crânios humanos em escavações arqueológicas
  - Y ⟶ 1 = crânio feminino, 0 = crânio masculino
  - X = circunferência, largura, altura (estimadas)

- Adultos de uma população
  - Y ⟶ 1 = pessoas com úlcera, 0 = sem úlcera
  - X = medidas de grau de ansiedade, de perfeccionismo, de sentimento de culpa

Dados para os exemplos anteriores

- Duas espécies de flor
  - Y ⟶ 1 = Iris Versicolor, 0 = Iris Setosa
  - X = largura e comprimento de pétala e de sépala

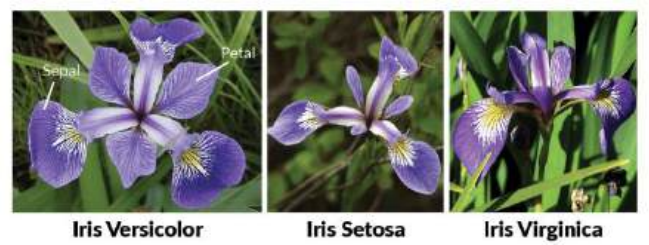
- Usuários de um site
  - Y ⟶ 1 = clicam num anúncio, 0 = não clicam
  - X = posição do anúncio na página, seu tamanho, tem imagem?

- Alunos de uma escola ou curso a distância
  - Y ⟶ 1 = evadem, 0 = completam o curso
  - X = notas de entrada, medidas de motivação à entrada, renda familiar,

Por que precisamos predizer a classe de um item novo?

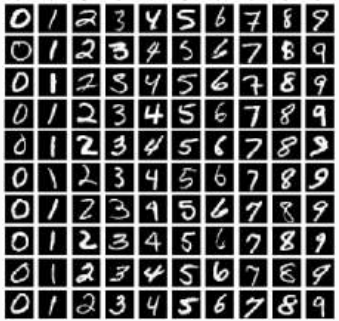
- Classe pode ser conhecida apenas no futuro
  - risco de crédito: no momento em que o crédito é solicitado, não sabemos se o crédito do indivíduo é bom ou ruim
- Informação sobre classe não é conhecida com certeza:
  - crânios arqueológicos danificados
- Obter a classe implica em destruir o item
  - classificar o paciente chegando ao pronto-socorro com lesão na cabeça: UTI ou não-UTI?
- Custo de conhecer a classe de cada instância nova com certeza seria proibitivo
  - conhecer a cobertura vegetal numa grande extensão territorial
- Várias razões para ter interesse em predizer classe

Várias classes, e não apenas duas classes



- Três classes: três espécies de flor
  - Y → 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

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



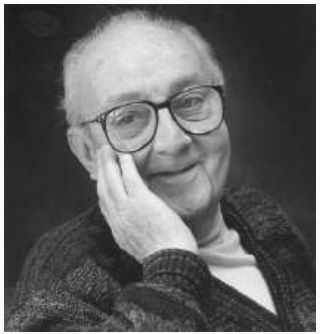
Modelos ML para classificação supervisionada

- 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)

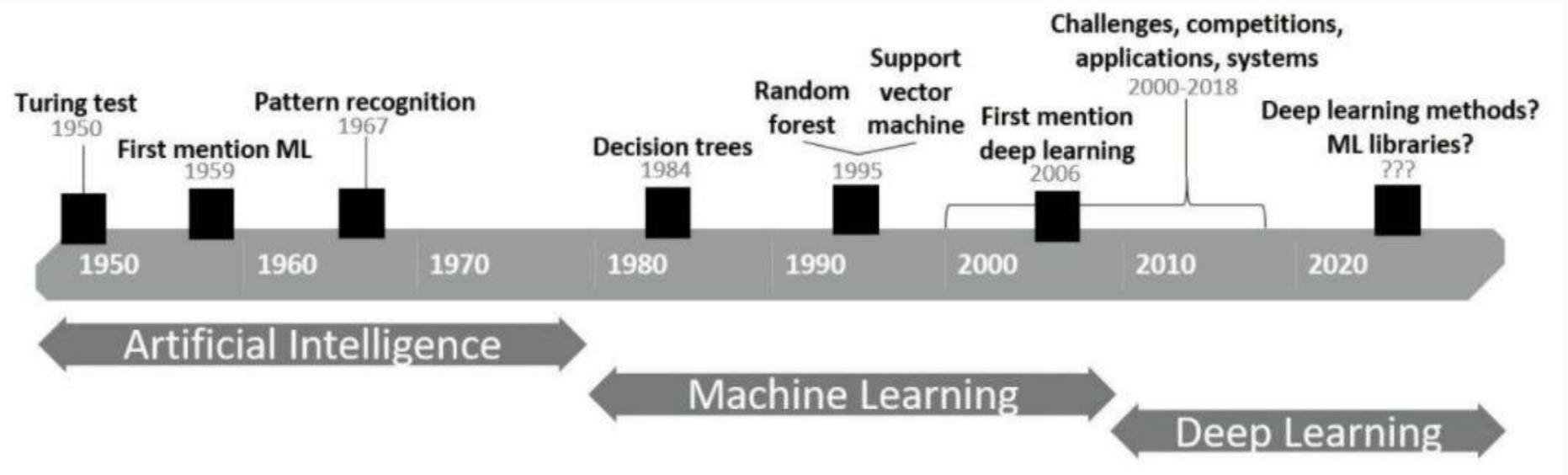
- George Box:
  - Todos os modelos são falsos; alguns são úteis.



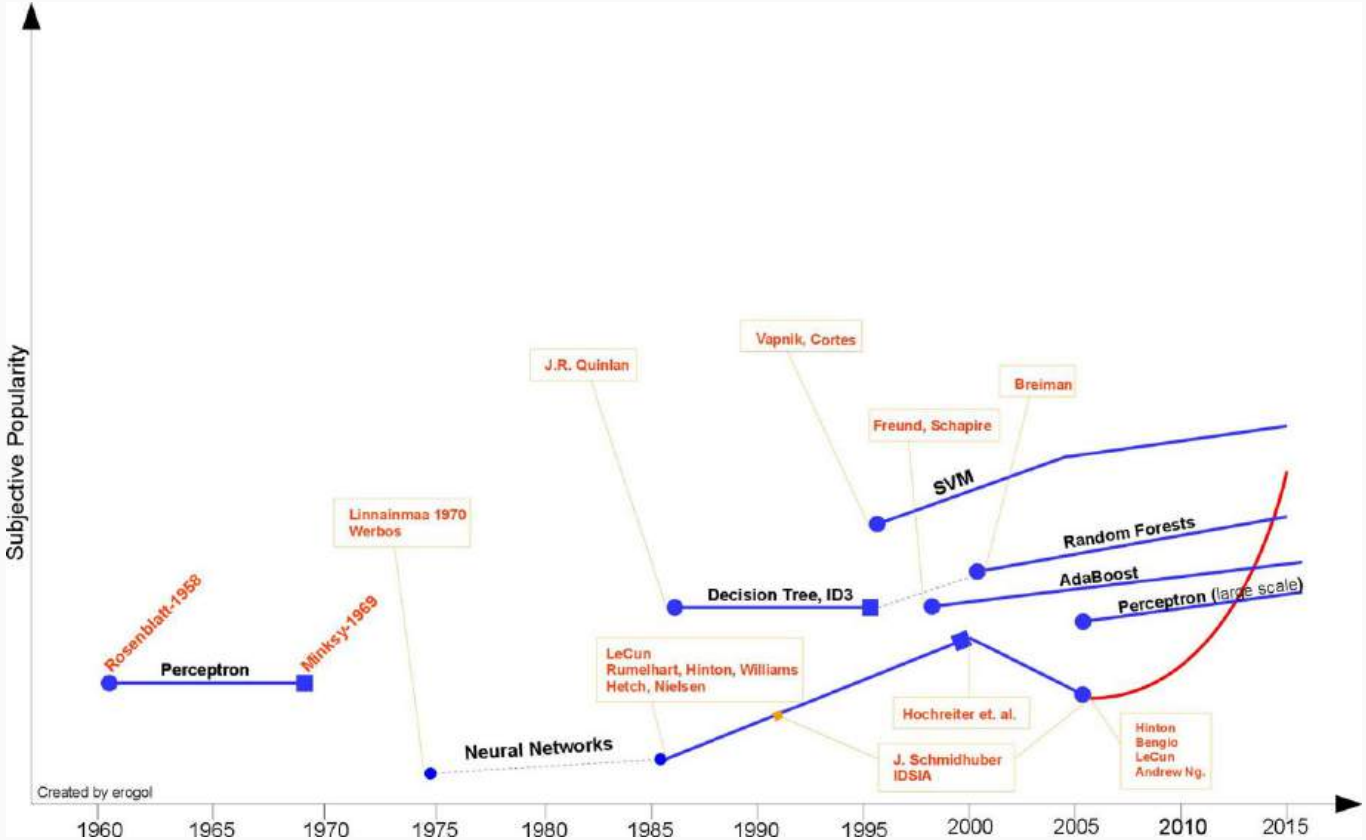
Os algoritmos de ML

- Todos os algoritmos 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



Timeline de ML, com popularidade dos algoritmos



Redes Neurais - Deep Learning: Por que o sucesso?

Duas razões:

- Primeira: Na maioria (todos exceto RN-DL) 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 ...

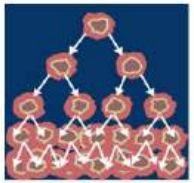
UCI Machine Learning dataset repository

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

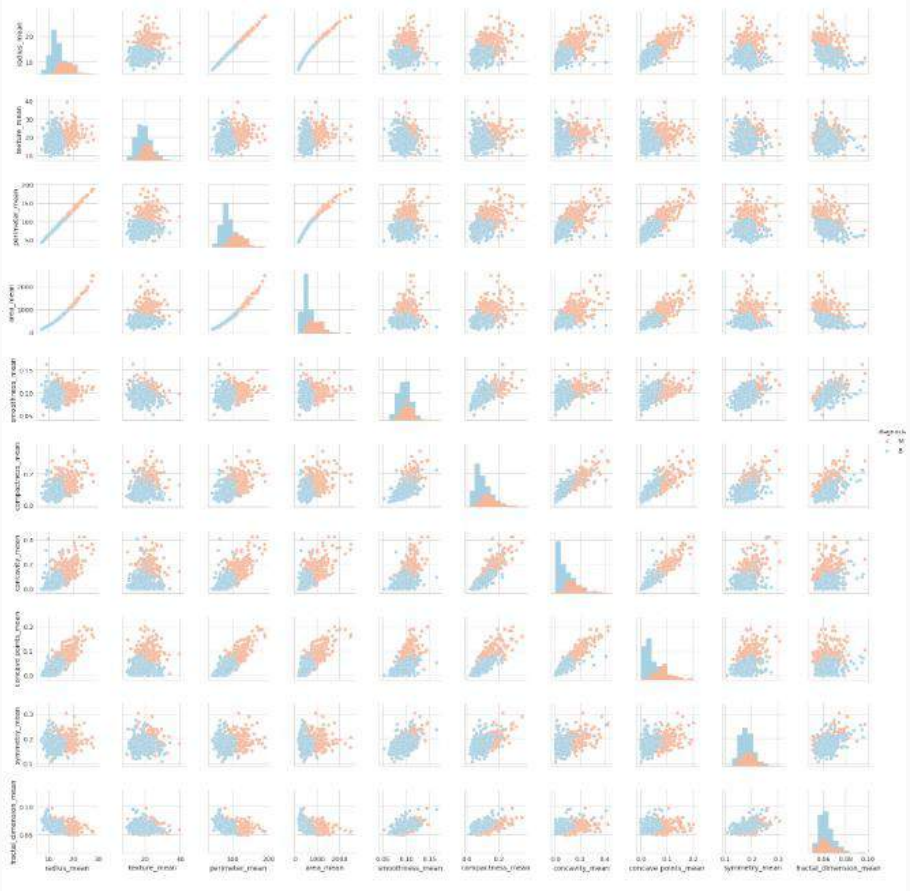
https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

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 (perimeter^2 / 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 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?

- Esta especificação ***prévia não é necessária*** com RN-DL.
- RN-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 saturado de parâmetros.
  - Mais dados, mais features, mais parâmetros
  - O número de parâmetros aumenta com o crescimento do número de exemplos.
  - Diferente de outros modelos de ML, existe um controle interno-automático de overfitting.
  - A mágica acontece.
  - Slide de Andrew Ng

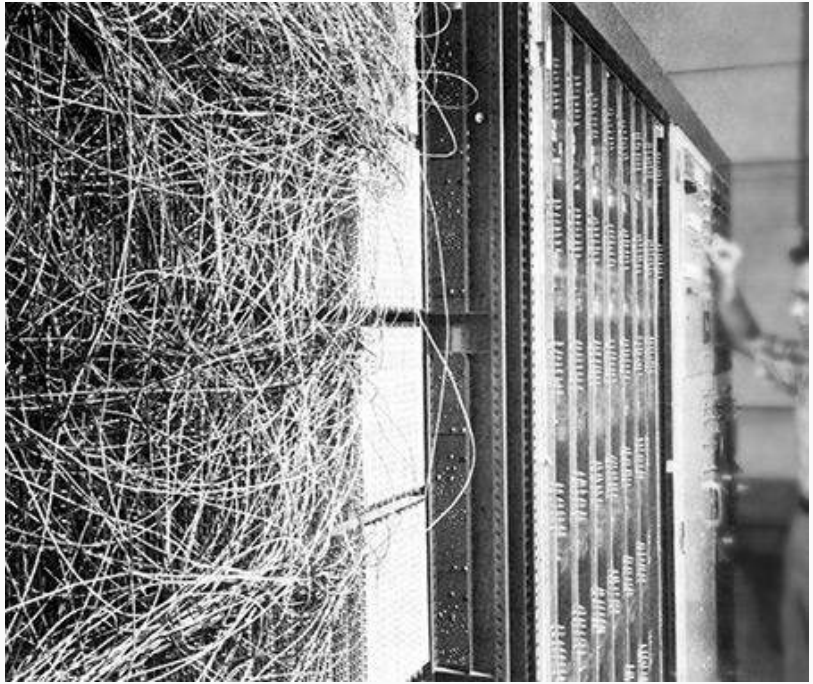
Como viemos parar aqui?

- Primeira rede neural: perceptron
- Frank Rosenblatt (1928-1971, falece aos 43 anos)
- Laboratório Aeronáutico - Cornell University
- Perceptron: um dispositivo eletrônico construído de acordo com princípios biológicos e com capacidade de aprender (discriminar entre duas classes de objetos).
- Inicialmente simulado em um computador IBM 704 em 1957.

Frank Rosenblatt,
aos 21 anos

Mark I perceptron, atualmente no Smithsonian Museum, Washington DC

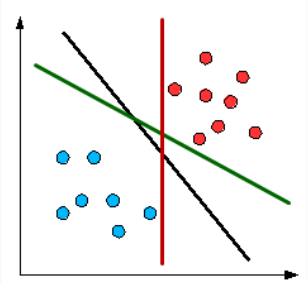
Frank Rosenblatt e Mark I Perceptron



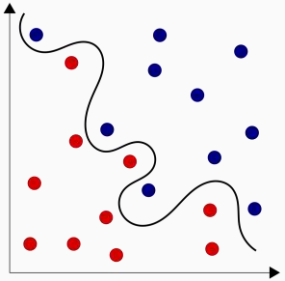
Abordagem geral

- Comece com um problema simples, bem simples
- Resolva o problema simples
- Como a solução do problema simples pode ser generalizada para problemas mais complicados?
- Problema mais complicado = problema mais realista, mais próximo da realidade
- Problema simples:
  - Classificar objetos em DUAS classes: 0 ou 1
  - **Problema linearmente separável**
    - **Existe pelo menos uma linha reta que separa os dados das duas classes**

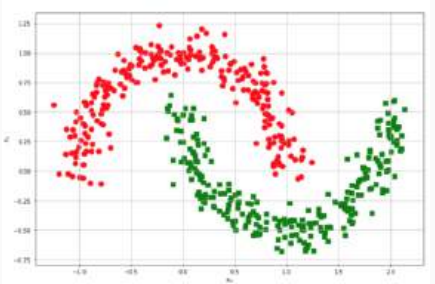
SIM



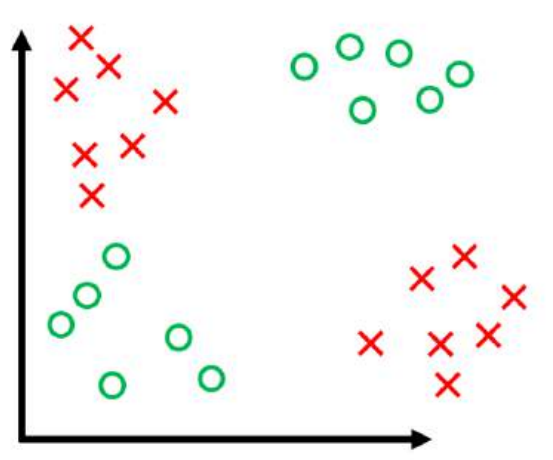
NÃO



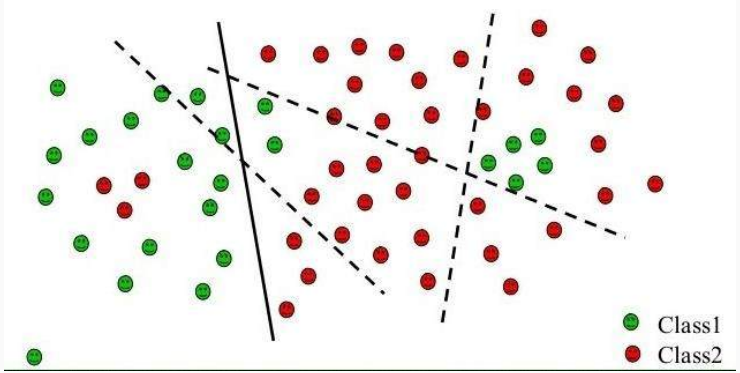
NÃO



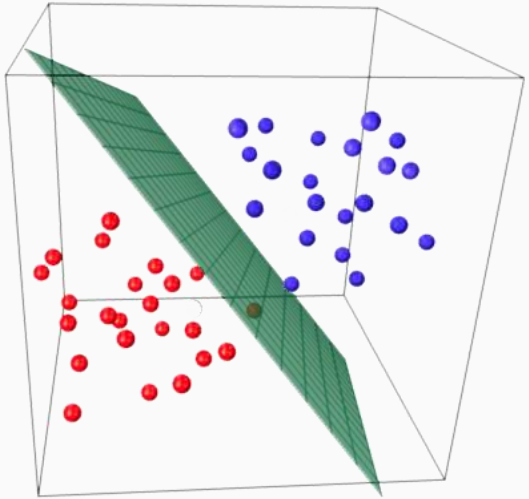
Linearmente separável



NÃO



NÃO



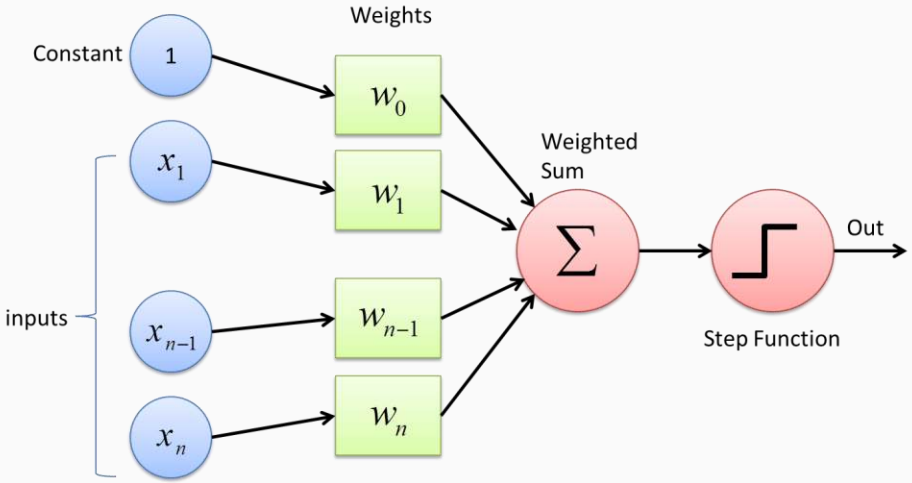
SIM

com 3 atributos ou features

- Dados de entrada:
  - Features $\mathbf{x} = (x_1, \ldots, x_n)$
- Processamento no "neurônio"
- Output:
  - 0 ou 1
- Como é feito o processamento?
- Modelo generativo da saída:
  - combine as features linearmente:

$$s = w_n x_n + \ldots + w_2 x_2 + w_1 x_1$$

  - O escore s é uma soma ponderada dos inputs-features
  - Cada feature tem um peso w_j que multiplica o valor do input
  - Se um input x_j não for influente, se puder ser ignorado para classificar, ele tem peso $w_j = 0$
  - Se for muito influente, seu peso será muito positivo ou muito negativo.



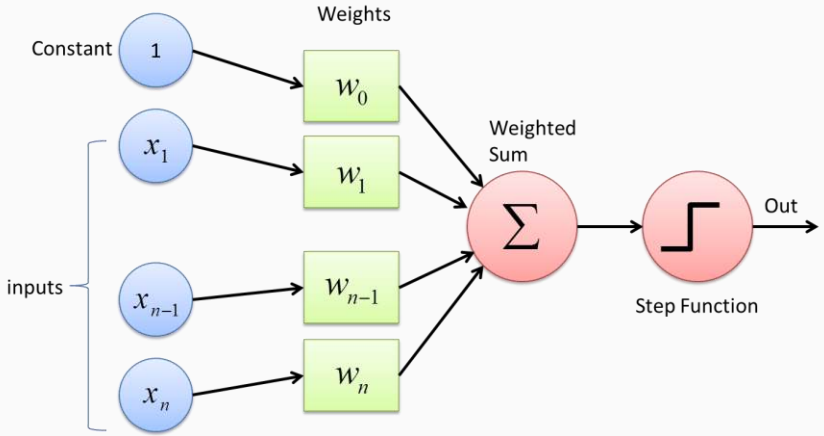
Perceptron, em 1958

- Como o escore é convertido na saída?

$$s = w_n x_n + \ldots + w_2 x_2 + w_1 x_1$$

- Aplique um limiar ℓ ao escore s para classificar
  - se $s \geq \ell$ então classifique na categoria 1
  - caso contrário, ponha na classe 0

- Como encontrar os pesos $\mathbf{w} = (w_1, \ldots, w_n)$ e o limiar ℓ?
- Pelo algoritmo do perceptron



Antes do algoritmo, alguma manipulação...

- Temos o escore
  - $s = w_n x_n + \ldots + w_2 x_2 + w_1 x_1$
- e classificamos o item na classe 1 se $s \geq \ell$
- Isto significa que classificamos na classe 1 se

$$w_n x_n + \ldots + w_2 x_2 + w_1 x_1 \geq \ell$$
$$w_n x_n + \ldots + w_2 x_2 + w_1 x_1 - \ell \geq 0$$
$$w_n x_n + \ldots + w_2 x_2 + w_1 x_1 + w_0 \geq 0$$
$$\mathbf{w}'\mathbf{x} + w_0 \geq 0$$

- onde $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ é um vetor-COLUNA e $\mathbf{w}'$ é o vetor-COLUNA transposto
- *Para nós, um vetor será sempre um vetor-COLUNA nas operações matriciais,* ***MESMO QUE ESCREVAMOS COMO UMA LINHA NO TEXTO, como fizemos com o vetor w acima***

Interpretação geométrica

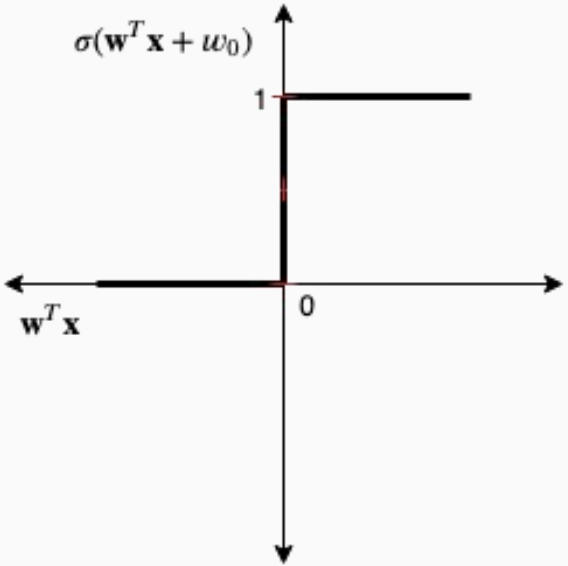
- Resumo:
  - classificamos na classe 1 se $\mathbf{w}'\mathbf{x} + w_0 \geq 0$
  - caso contrário, classificamos na classe 0

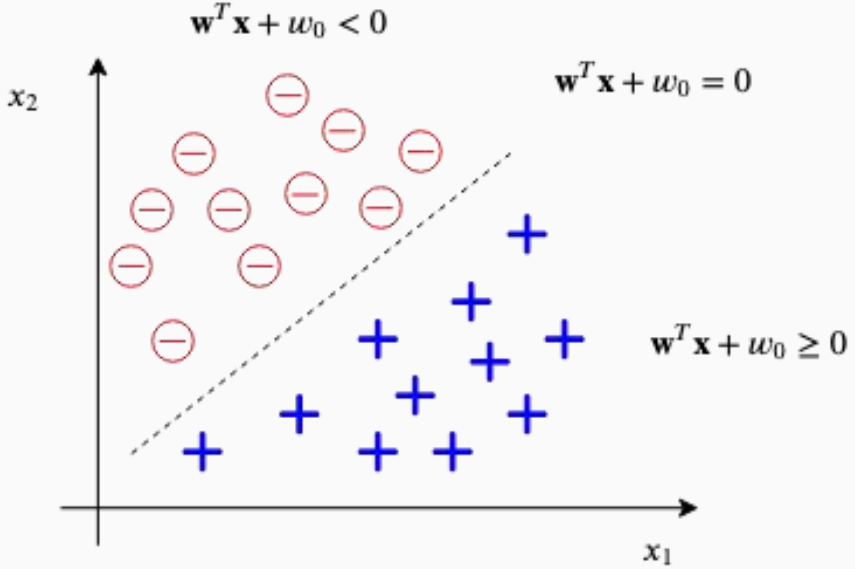
- Vamos criar mais uma notação: a função de ativação, que fornece a saída (0 ou 1)
- Defina a função $\sigma(\mathbf{x})$ da seguinte forma

$$\sigma(\mathbf{x}) = \begin{cases} 1, & \text{se } w_0 + \mathbf{w}'\mathbf{x} \geq 0 \\ 0, & \text{caso contrário} \end{cases}$$

Exemplo com duas features

Desenhar no quadro e explicar com o próximo slide





Geometria da regra de decisão com duas features

- Reta no plano (x_1, x_2) é equivalente a

$$w_0 + w_1 x_1 + w_2 x_2 = 0$$

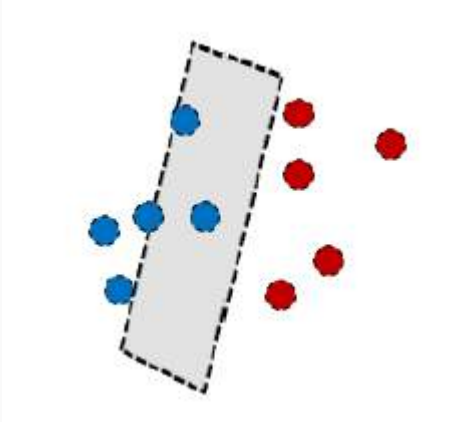
- para certa escolha dos pesos (ver exercícios teóricos)
- Por exemplo, se $w_2 \neq 0$ então podemos isolar x_2 e escrever

$$x_2 = -\frac{w_0}{w_2} - \frac{w_1}{w_2} x_1 = \alpha + \beta x_1$$

- Isto é, uma reta com intercepto $-w_0/w_2$ e inclinação $-w_1/w_2$
- Mostra-se que o vetor **normal** (ortogonal) a esta reta é o vetor (w_1, w_2)
- Além disso, (w_1, w_2) aponta na direção em que $\mathbf{w}'\mathbf{x} + w_0 \geq 0$
- Exemplo numérico

Caso geral

- Generalizando para n features
  - classificamos na classe 1 se $\mathbf{w}'\mathbf{x} + w_0 \geq 0$
- O conjunto de pontos x tais que $\mathbf{w}'\mathbf{x} + w_0 = 0$ forma um hiper-plano (um plano que passa fora da origem se w_0 é diferente de zero) no espaço das features
- O vetor **w** é normal ao hiperplano
- w_0 é chamado de termo de vício ou viés (bias, pronúncia: BAI' AS)
- O vetor **w** é chamado de vetor de coeficientes



Expandindo a notação

- É útil incorporar o termo de bias w_0 no vetor de coeficientes.
- Escrevemos:

$$\mathbf{w}_* = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{bmatrix} \quad \text{e} \quad \mathbf{x}_* = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$$

- Note que

$$\mathbf{w}_*^t \mathbf{x}_* = w_0\ 1 + w_1 x_1 + \ldots + w_n x_n = \sum_{j=0}^{n} w_j x_j$$

- **Note que criamos uma "feature"** $x_0 = 1$
- **Esta expressão vai aparecer inúmeras vezes.** Devemos memorizá-la. Vemos que uma soma das features x's ponderadas pelos pesos w's é o mesmo que multiplicar um vetor deitado (transposto) pelo outro em pé (vetor-coluna)

Como encontrar os pesos?

- Procuramos um bom conjunto de pesos $w_0, w_1, \ldots, w_n$

- Um bom conjunto de pesos seria aquele que classificasse corretamente **todos** os exemplos (se isto for possível, como é, no caso linearmente separável).
- Assim, se os exemplos forem classificados corretamente, não mexemos nos pesos.
- Se algum deles estiver incorreto, mexemos nos pesos para ajustá-los melhor.
- Como fazer isto?
- Podemos tentar fazer sequencialmente, passando pelos exemplos, um de cada vez.
- Começamos com pesos arbitrários e vamos ajustando ao passar por cada exemplo:
    - se o exemplo está bem classificado, não mexa
    - se estiver classificado errado, ajuste um pouco.
- Repita até que todos os exemplos estejam classficados corretamente

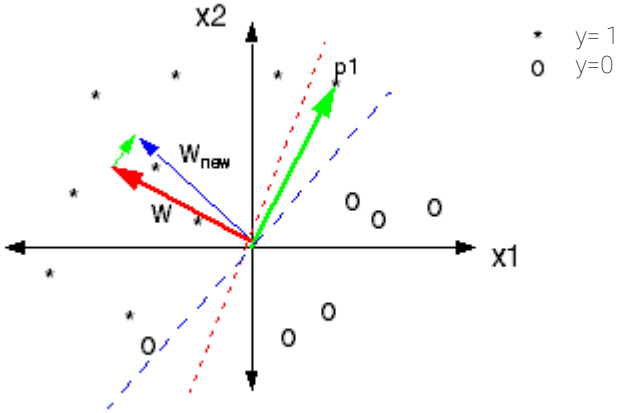
Algoritmo do perceptron

- Inicialize os pesos $w_0, w_1, \ldots, w_n$
- Escolha uma taxa de aprendizagem m em (0, 1] (intervalo inclui 1)
- Até que a condição de parada seja satisfeita (por exemplo, os pesos não mudam muito):
  - Para cada exemplo (y, **x**) do conjunto de treinamento:
    - calcule a predição $\hat{y} = \sigma(\mathbf{w}'_* \mathbf{x}_*)$
      - Se $y = \hat{y}$, não faça nada
      - Se $y \neq \hat{y}$, atualize os pesos:

$$\mathbf{w}_*^{(new)} = \mathbf{w}_*^{(old)} + m(y - \hat{y})\mathbf{x}_*$$

- Note que TODO o vetor de pesos $w_0, w_1, \ldots, w_n$ é atualizado

Atualizando os coeficientes

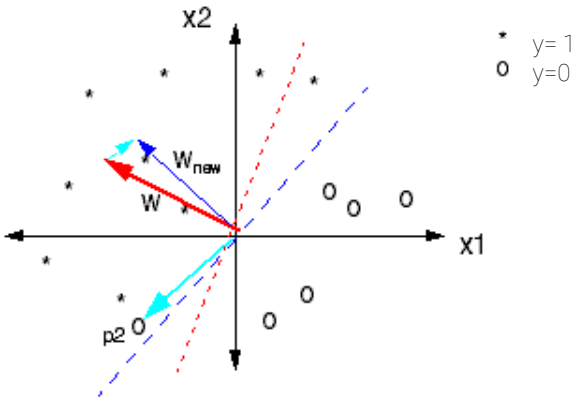


- Decision boundary atual = linha tracejada vermelha
- Vetor de coeficientes atual = **vetor W** (vermelho)
- W é normal à fronteira de decisão

- No loop, chegou a vez de usar o ponto P_1
- Ele está incorretamente classificado e vai alterar o vetor w.

- P_1 tem y = 1
- W é movido uma pequena quantidade na direção de P_1, passando a ser vetor $\mathbf{W}_{new}$ (azul)
- Novo decision boundary = linha tracejada azul

Atualizando os coeficientes



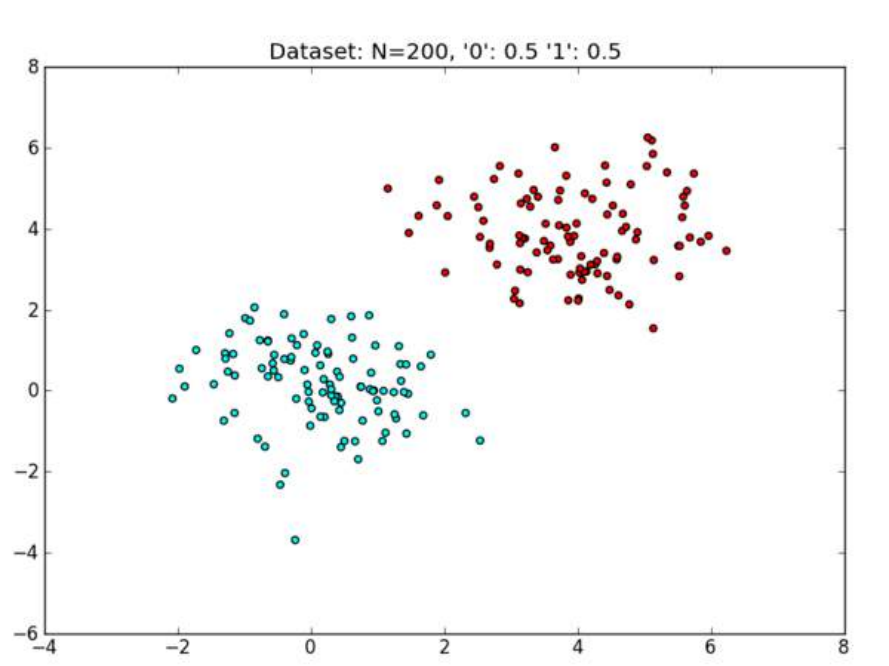
- Decision boundary atual = linha tracejada vermelha
- Vetor de coeficientes atual = **vetor W** (vermelho)
- W é normal à fronteira de decisão

- No loop, chegou a vez de usar o ponto P_2
- Ele está incorretamente classificado e vai alterar o vetor w.

- P_2 tem y = 0
- W é movido uma pequena quantidade na direção OPOSTA a P_2, passando a ser vetor $\mathbf{W}_{new}$ (azul)
- Novo decision boundary = linha tracejada azul

Visualização do perceptron em ação

https://www.youtube.com/watch?time_continue=1&v=vGwemZhPlsA



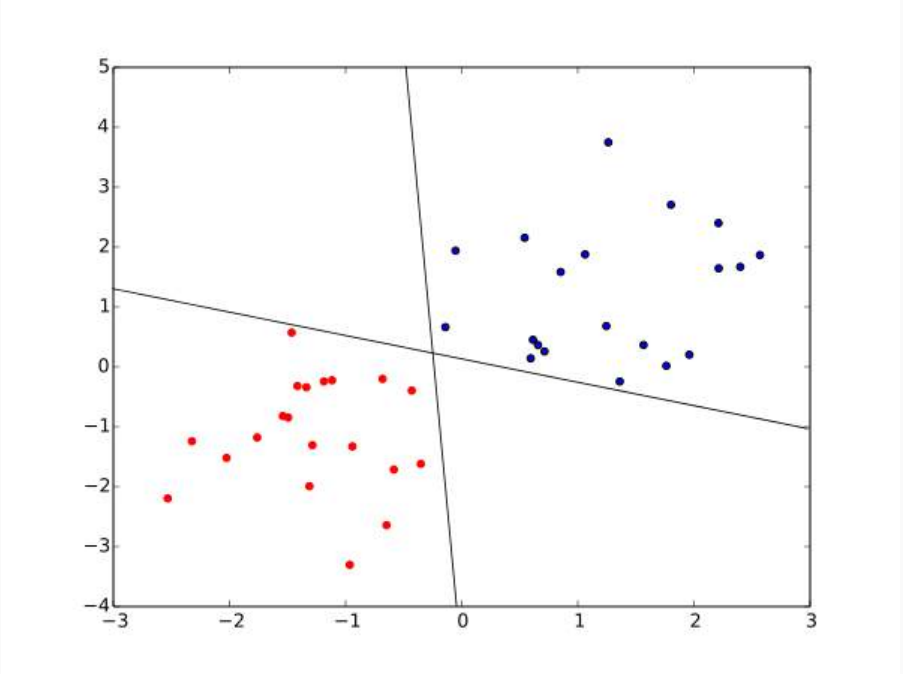
Convergência

- Garantia teórica:
  - Se o conjunto de dados for linearmente separável, o algoritmo do perceptron sempre encontrará uma função que separa os dados.
    - Isto é, o algoritmo converge para pesos que classificam todos os exemplos corretamente
  - Se não for linearmente separável, o perceptron nunca chegará ao estado com todos os vetores de entrada classificados corretamente.
    - Ou seja, se os exemplos positivos não puderem ser separados dos exemplos negativos por um hiperplano, nenhuma solução será gradualmente "aproximada"
    - Ao invés disso, o aprendizado falhará completamente (o algoritmo não converge).

- Prova: Duda, Hart e Stork, Pattern Classification, 2nd Edition, page 229
- Se a separabilidade linear não puder ser garantida, precisamos de uma alternativa (mais a frente)

Muitas soluções são possíveis

- É garantido que o algoritmo do perceptron converge para ALGUMA solução no caso de um conjunto de treinamento linearmente separável.
- Mas podem existir infinitas soluções (veja desenho)
- Perceptron escolhe uma delas em função da condição inicial e da sequência que percorre os exemplos.
- As possíveis soluções são de qualidade variável.
- O perceptron de estabilidade ótima é o SVM (Krauth e Mezard, 1987)



O que se almejava

- Colete muitos exemplos e TODAS as features que potencialmente podem afetar a classificação
- Talvez várias dessas features sejam inúteis no final
- Sem problema: seu peso deve ficar igual a zero no perceptron.
- FIGURA com x_1 sendo inútil (caso bi-dimensional e caso tri-dimensional)

O que se almejava - 2

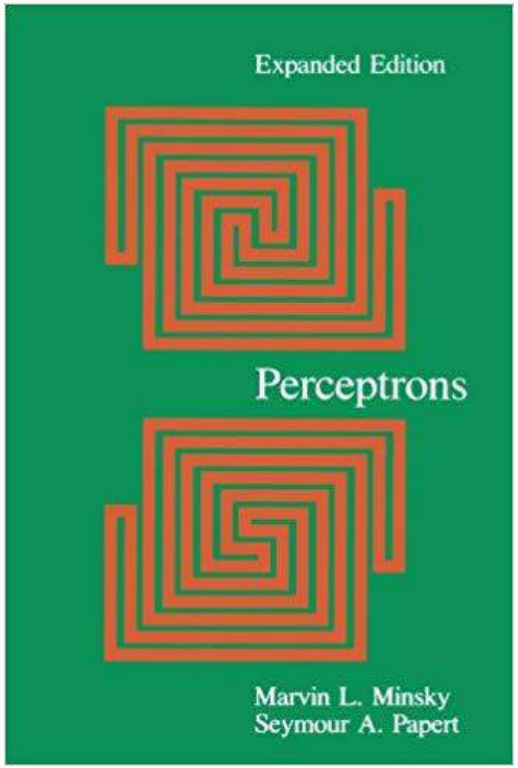
- Com poucas features, é possível que não tenhamos separabilidade linear
- Mas, talvez ...
    - com uma coleção GRANDE de features talvez seja possível obter separabilidade linear

- Boa notícia: se isto for verdade (se for possível obter **l.s.**), o perceptron acha a regra de classificação
- Não tem de pensar, a máquina descobre como separar o joio do trigo

- Grande excitação em alguns círculos
- Mas...

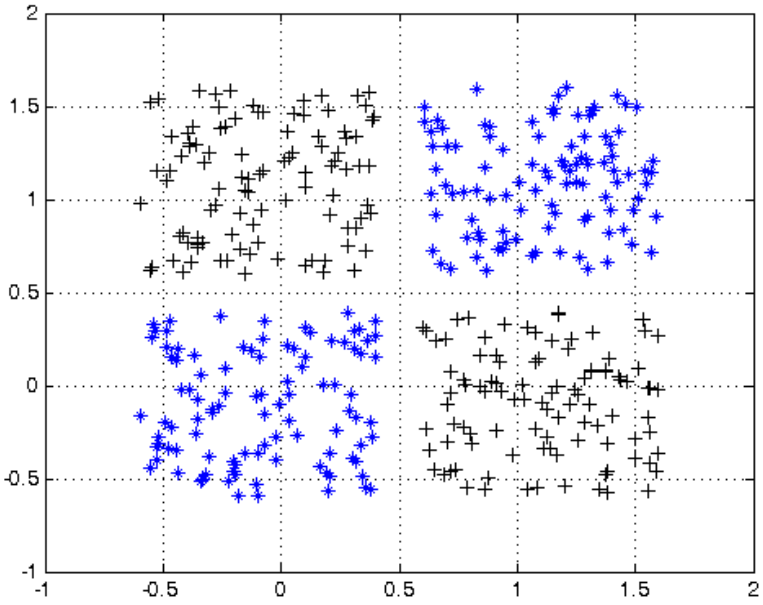
Ducha de água fria

- Em 1969 aparece o livro Perceptrons de Marvin Minsky e Seymour Pappert (MIT)
- Marvin Minsky é um dos pioneiros em Dartmouth
- Eles mostram que um perceptron não seria capaz de classificar dados XOR



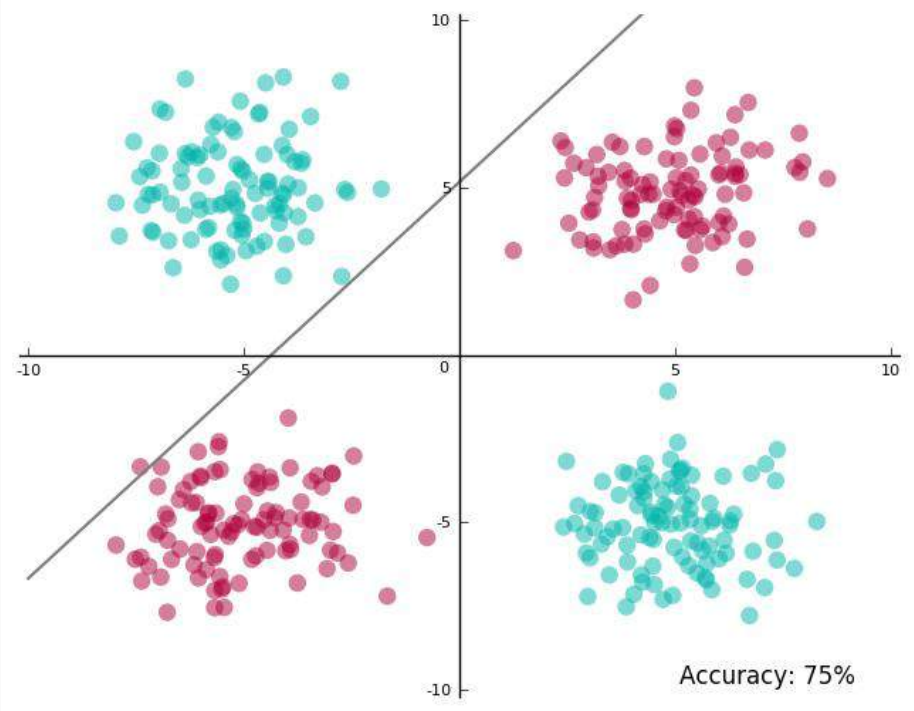
O XOR matador

- Situação muito simples
- Os dois grupos perfeitamente separáveis
- Mas não LINEARMENTE separáveis



O XOR matador

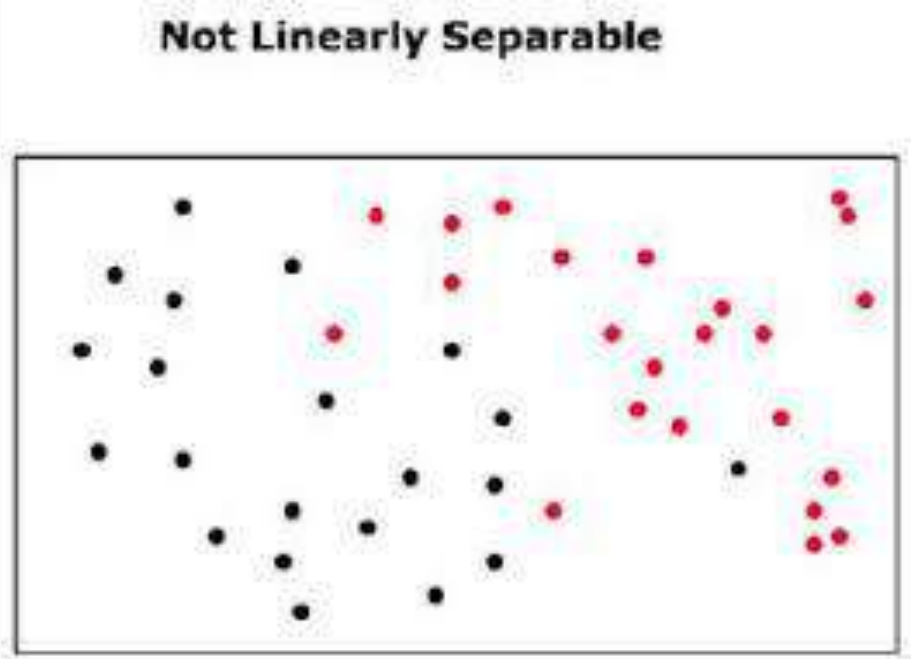
- Não interessa quantos exemplos você colete
- O perceptron nunca será capaz de aprender a separar os dois grupos com bases nas features x_1 e x_2
- Parece incrível que isto tenha estancado o desenvolvimento.



Caso QUASE linearmente separável

- O que queremos dizer com caso QUASE linearmente separável?

- É uma situação em que uma combinação linear das features classifica bem uma porção substancial dos casos mas não TODOS eles.



Quo vadis?

- Dois problemas para resolver:
  - caso QUASE linearmente separável
  - caso XOR

- O problema de situações QUASE linearmente separáveis foi resolvido com a regressão logística

- O problema do XOR foi resolvido com várias camadas de perceptron junto com regressão logística

- Vamos ver cada um desses problemas e sua solução

Logística - motivação

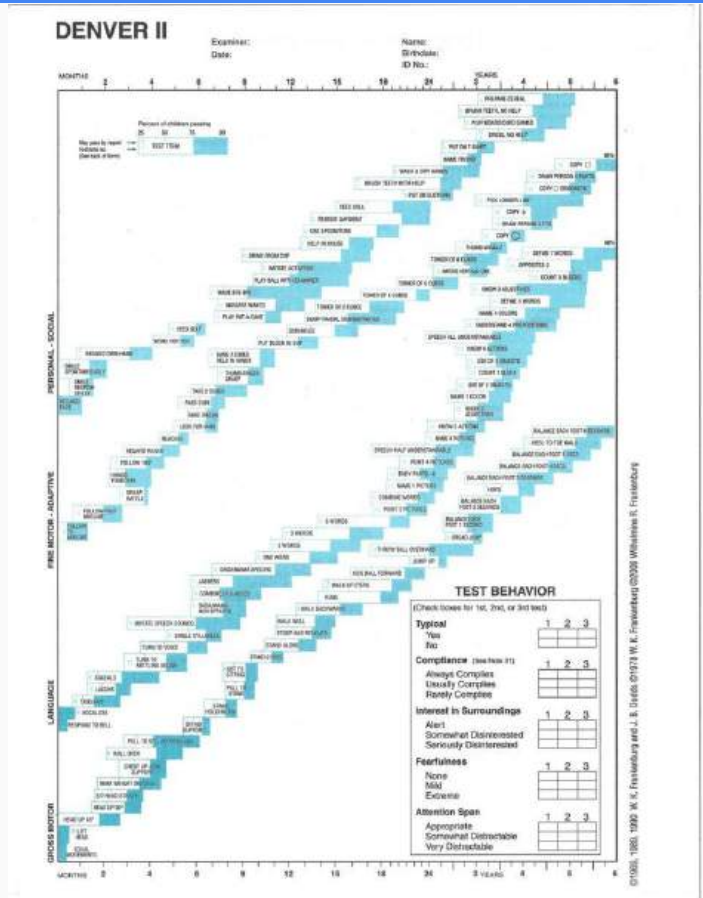
- Detecção precoce de problemas e o Teste de Triagem de Desenvolvimento de Denver.
- Aplicado em crianças entre o nascimento e os seis anos de idade
  - Para confirmação de suspeitas na avaliação subjetiva do desenvolvimento
  - Para monitorar crianças com risco de apresentar alterações.

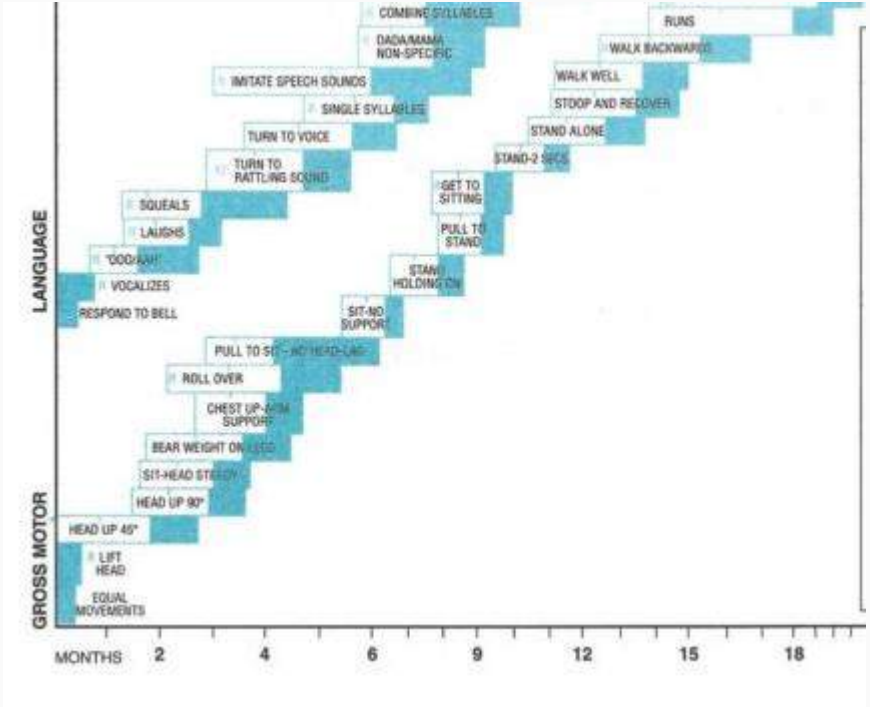
- O teste é composto por 125 itens (ou tarefas), subdivididos em quatro domínios de funções:
  - pessoal-social,
  - motor-adaptativo,
  - linguagem,
  - e motor grosseiro.

Exemplos de itens-tarefas

- coordenação do olho e da mão
- manipulação de pequenos objetos,
- produção de som,
- capacidade de reconhecer, entender e usar a linguagem,
- controle do corpo para sentar, caminhar, pular
- Usa gestos simples, como balançar a cabeça "não" ou acenar "tchau-tchau"
- Diz "mamã" e "papá" e exclamações como "uh-oh!"
- Explora as coisas de maneiras diferentes, como agitar, bater, jogar
- Aponta para mostrar aos outros algo interessante
- ETC..

Teste de Desenvolvimento Infantil de Denver





Caso uni-dimensional

- Cada um dos 125 itens é representado por uma barra que contém as idades em que 25%, 50%, 75% e 90% das crianças estudadas apresentaram as habilidades sugeridas.
- Uma criança de idade x realiza alguns poucos itens, aqueles indicados para sua faixa etária.
- O pediatra sabe que, dentre crianças com aquela idade x, uma certa porcentagem p(x) executa corretamente a tarefa do item.
- Suponha que a criança sob exame não executou a tarefa. Se p(x) = 0.25, não há motivos para preocupação.
- Mas se p(x) = 0.90, pode haver motivo de preocupação e exames mais minuciosos são então indicados.
- Como estas idades críticas são determinadas para uso rotineiro nos consultórios? Com regressão logística, uma para tarefa.

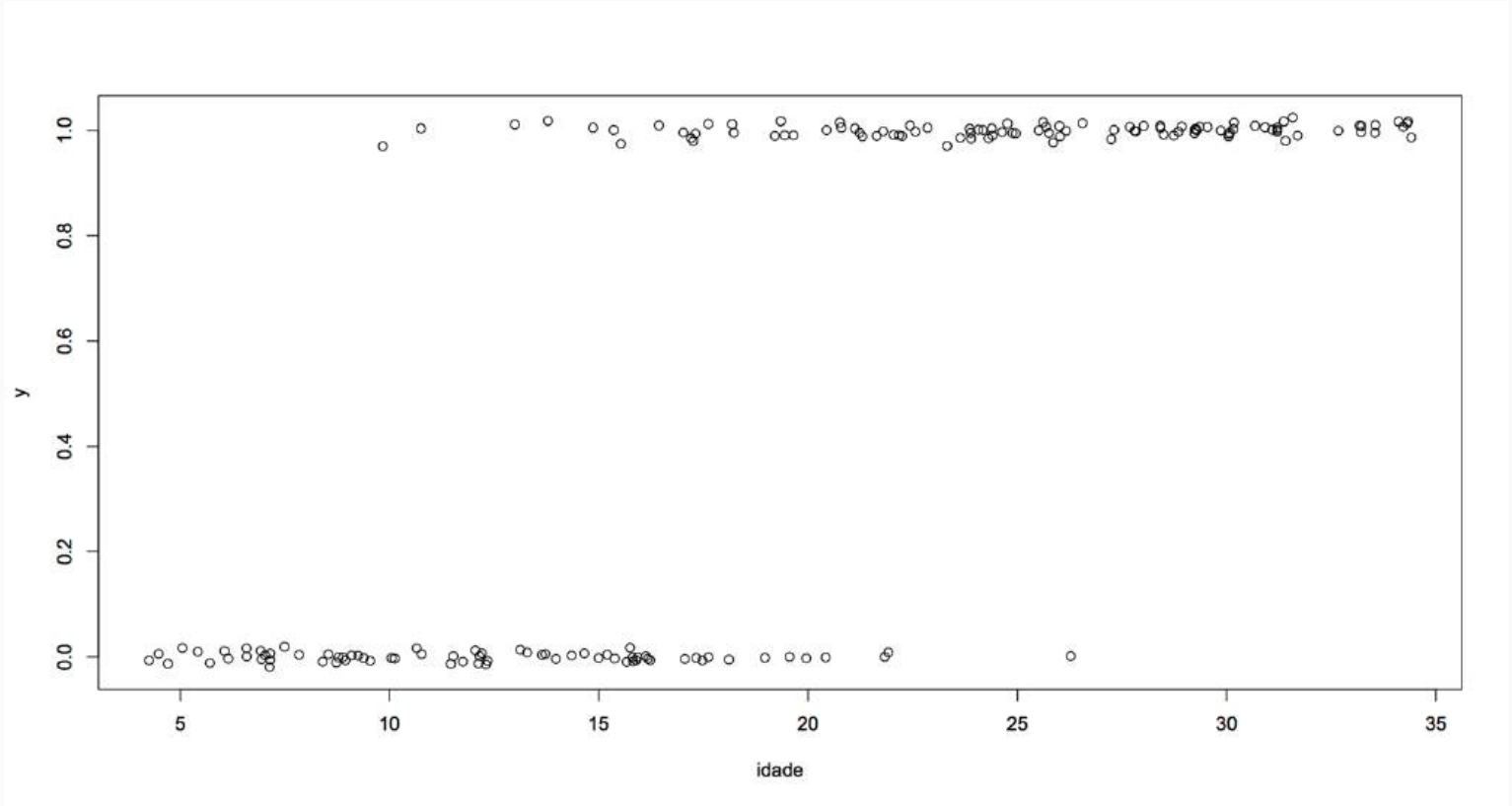
Modelo para uma única tarefa

- Amostra de m crianças de diversas idades e sem problemas de desenvolvimento procuram executar a tarefa de um dos itens.

- $Y_i = 1$ denota o sucesso da i -ésima criança na execução da tarefa.
- $Y_i = 0$ denota o fracasso

- No caso de crianças sem problemas de desenvolvimento, mais cedo ou mais tarde, todas acabam executando a tarefa sem erros.
- O sucesso ou fracasso depende principalmente da idade x da criança.
- Sendo velha o suficiente, a criança vai executar a tarefa.
- Por outro lado, se for muito nova ainda, e quase impossível executar a tarefa.
- As faixas etárias apropriadas variam com o tipo de tarefa.

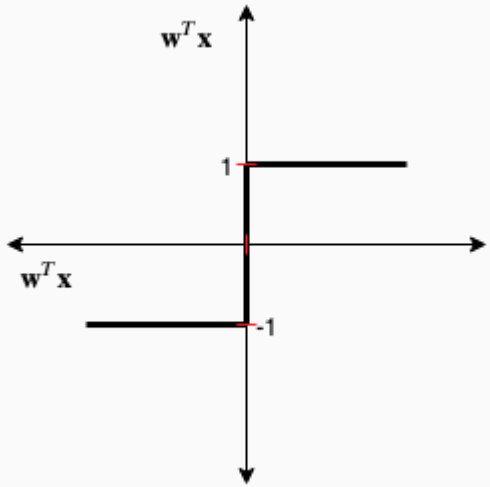
173 crianças



Perceptron não é uma boa solução neste problema

- Veja que os dados não são perfeitamente separáveis nas duas classes (0 e 1)
- Modelo do perceptron tem um limiar (threshold) rígido.

$$\sigma(\mathbf{x}) = \begin{cases} 1, & \text{se } w_0 + \mathbf{w}'\mathbf{x} \geq 0 \\ 0, & \text{caso contrário} \end{cases}$$



- Caso de uma única variável: se x >= $-w_0/w_1$ → sigma(x)=1
- Isto é, y=1 se, e somente se, x >= limiar $-w_0/w_1$
- Muito rígido: deveríamos ter uma idade limiar $-w_0/w_1$: antes dela, ninguém executa a tarefa. Depois dela, todos executam
- Modelo do perceptron não é bom para este problema
- Talvez um soft thershold resolva?

Modelo estatístico

- Resultados das crianças são independentes pois o sucesso ou fracasso de uma das crianças não afeta o desempenho das demais.

- $Y_1, Y_2, \ldots, Y_m$ ensaios de Bernoulli independentes

$$Y_i = \begin{cases} 1, & \text{com probabilidade } p_i \\ 0, & \text{com probabilidade } 1 - p_i \end{cases}$$

- A probabilidade p_i vai variar de criança para criança: por isto é indexada por i

Como queremos que esta probabilidade de sucesso varie?

- p_i é a probabilidade da criança i ter sucesso
- Esta probabilidade não é a mesma para todas as crianças

- Queremos que a probabilidade aumente com a idade x.
- Quando x for muito grande, a probabilidade deve ser praticamente 1

- Queremos que a probabilidade seja praticamente zero se a idade x for muito pequena.

- Queremos $p = \sigma(x)$ onde $\sigma(x)$ seja:
  - suave, crescente com x, que vá para 1 quando x crescer e vá para 0 quando x diminuir

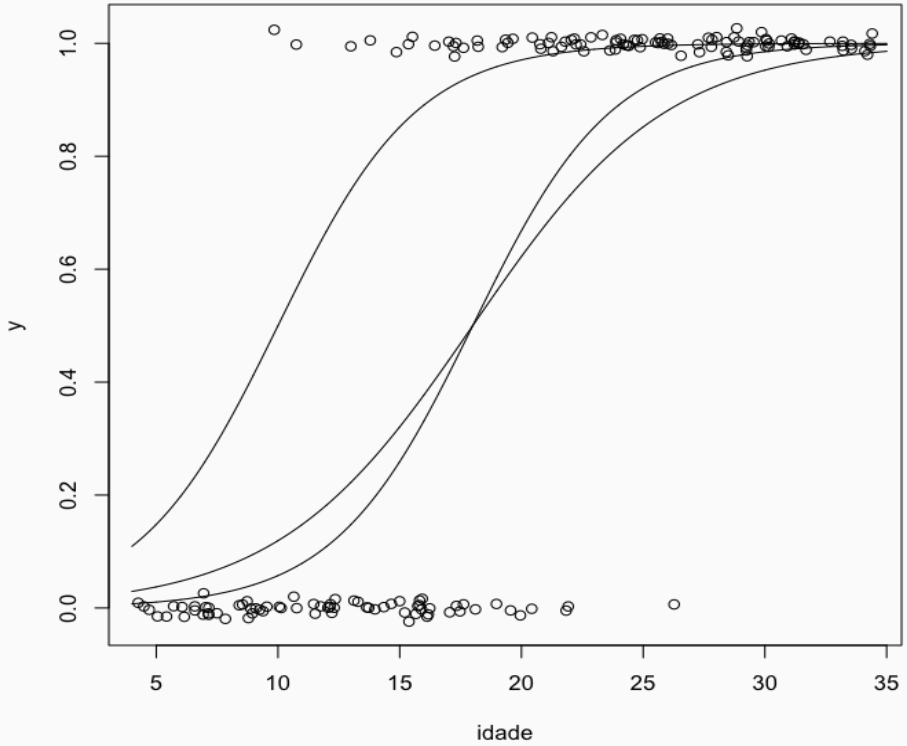
Função logística

- Usamos a curva logística

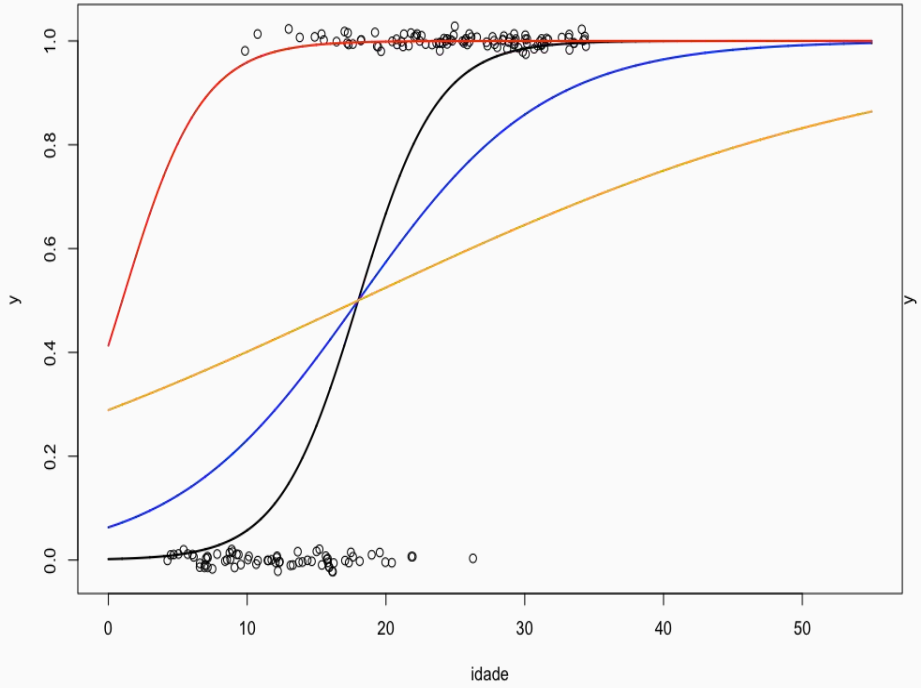
$$\sigma(x) = \frac{1}{1+e^{-(w_0+w_1 x)}}$$

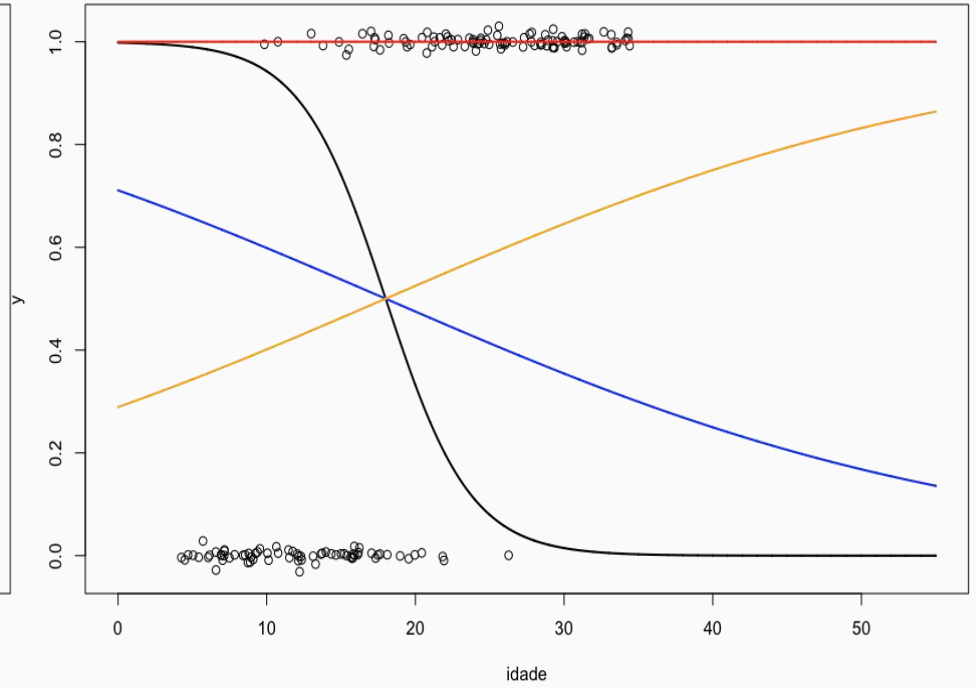
- Diferentes coeficientes w0 e w1 geram diferentes curvas em forma de S

- Efeito de w1: controlar quão íngreme é a subida e a direção do S
- Efeito de w0: localizar a curva no eixo horizontal



Diferentes parâmetros, diferentes curvas. Qual a melhor? Como decidir?





Várias perguntas. Alguma resposta?

- Várias perguntas:
  - Modelo logístico é bom?
  - Como generalizar para quando tivermos várias features?
    - E se a probabilidade também da escolaridade da mãe, do sexo da criança, ...
  - Como escolher a "melhor" curva logística? "Melhor" em que sentido?
- Roteiro da estrada à frente (aula 02):
  - Método de máxima verossimilhança (ML) e função de custo
  - Otimização: Newton e gradiente descendente
  - Heurística e otimalidade da ML
  - Stochastic gradient descent