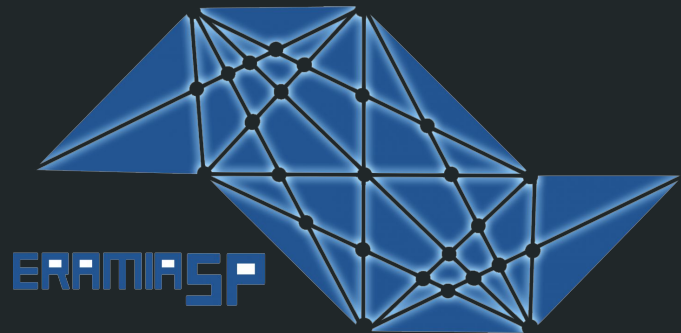
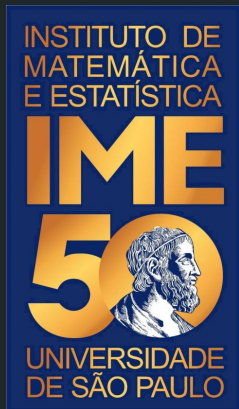


Sejam bem-vindas, sejam bem vindos!



Visão Computacional e Aprendizado Profundo



Oito anos da AlexNet

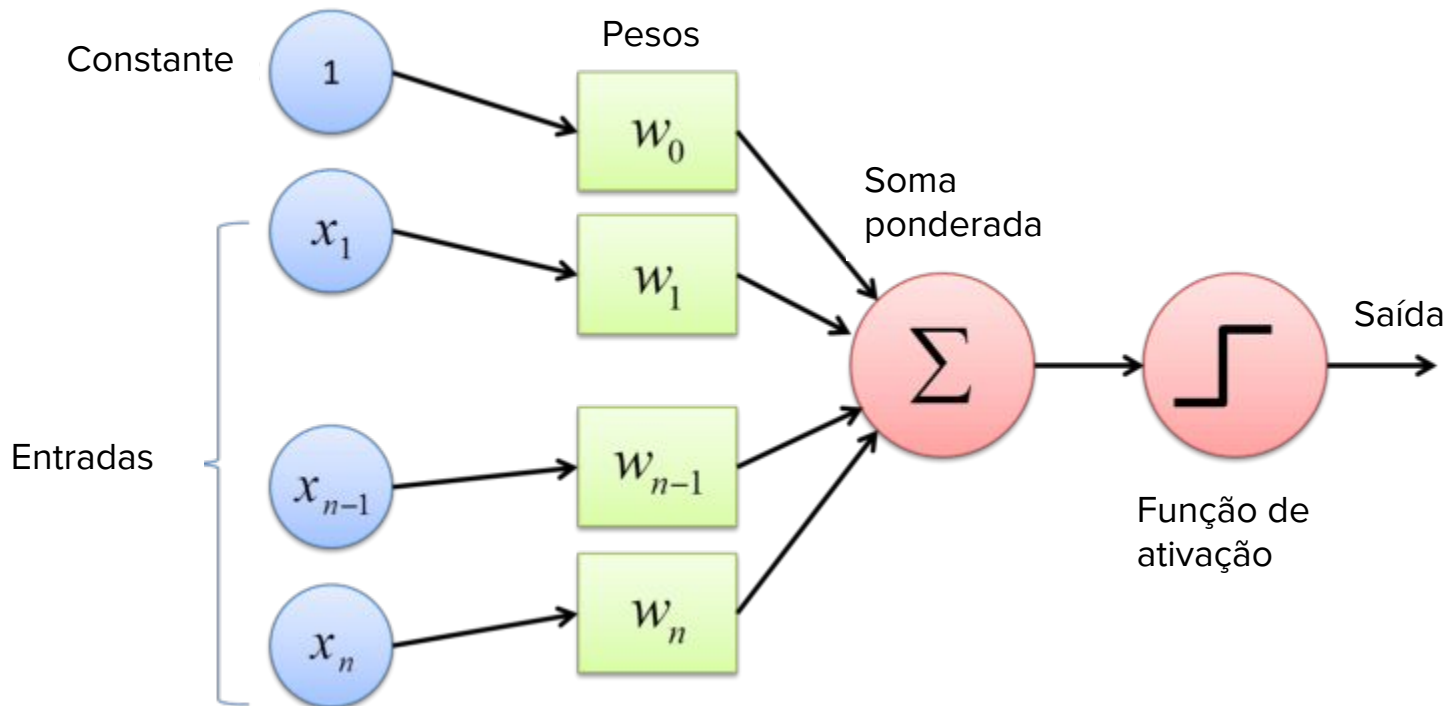
R. Hirata Jr.
IME - USP

Redes neurais

Redes neurais

- Conjunto de métodos de aprendizado supervisionado, ou não supervisionado, que teve grande interesse da década de 60 a 90.
- A inspiração e parte da popularidade das redes neurais vêm das redes neuronais - redes de neurônios, modelo de um neurônio e a relação entre eles.
- Prova matemática que qualquer classificador (função de um conjunto multidimensional real no conjunto dos números reais) pode ser expresso por uma rede neural de duas camadas.
- O modelo mais simples de um neurônio é o **Perceptron**

Redes neurais - Perceptron



Algoritmo do perceptron

Entrada: Vetor de pesos iniciais W , vetor multidimensional de exemplos X , vetor de rótulos Y
Inteiro maxIter (número máximo de iterações)
Número real η (razão de aprendizado)

Saída: Vetor de pesos W
Inteiro $nIter$ (número de iterações realizado)

Repita até que todos os elementos de X estejam corretamente classificados, ou maxIter :

Repita para todos os pares de elementos (x_i, y_i) de $X \times Y$:

/* Verifica se o exemplo está bem classificado */

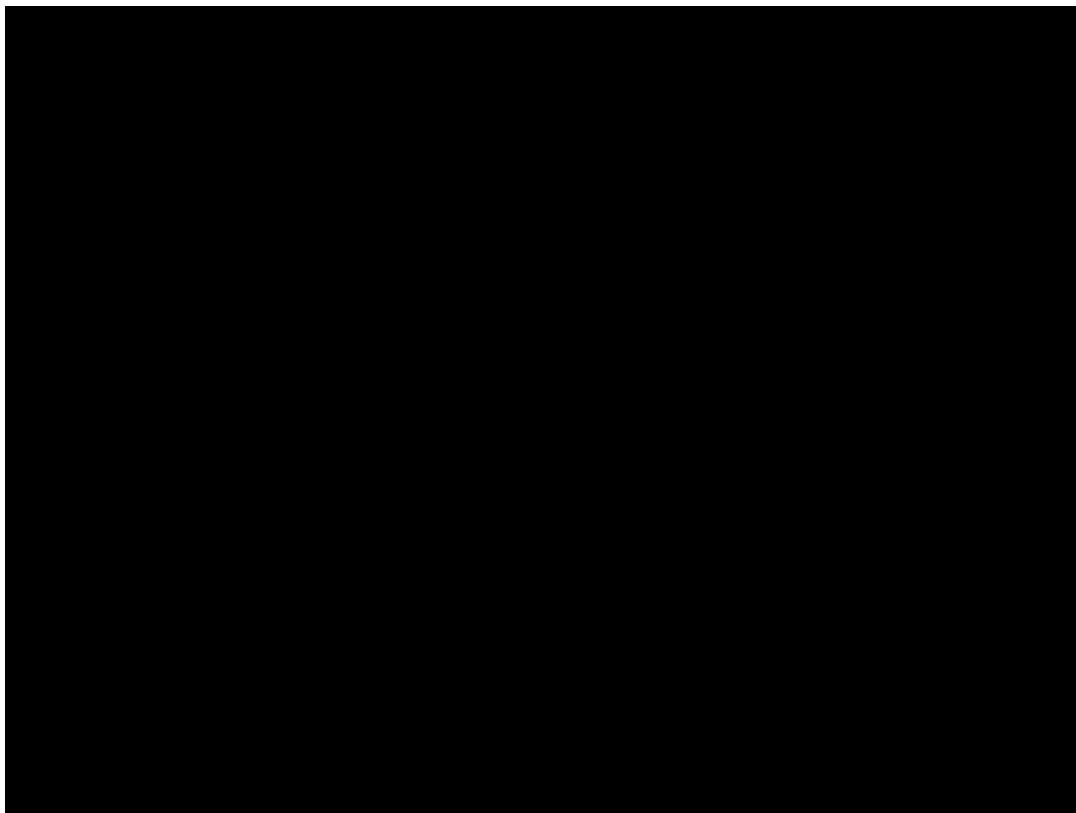
se $y_i \neq \text{sign}(x_i^T W)$:

/* Atualiza o hiperplano na direção do vetor x_i */

$$W = W + \eta (y_i * x_i)$$

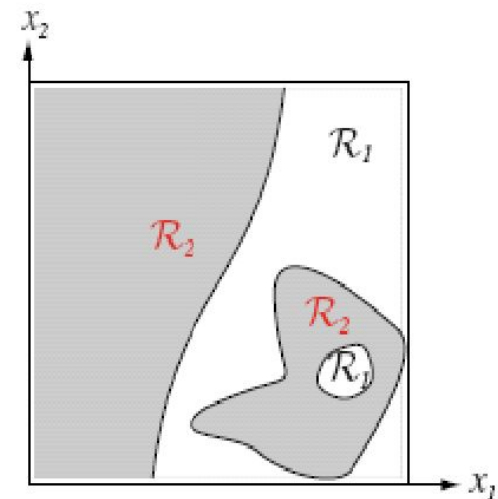
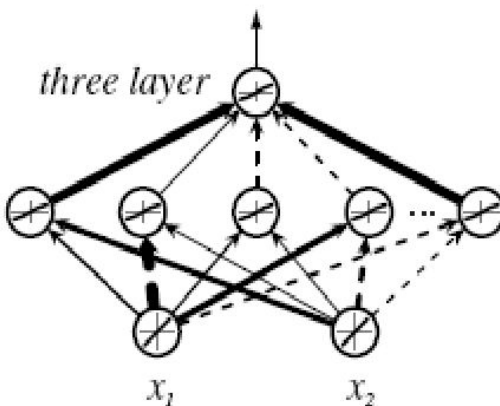
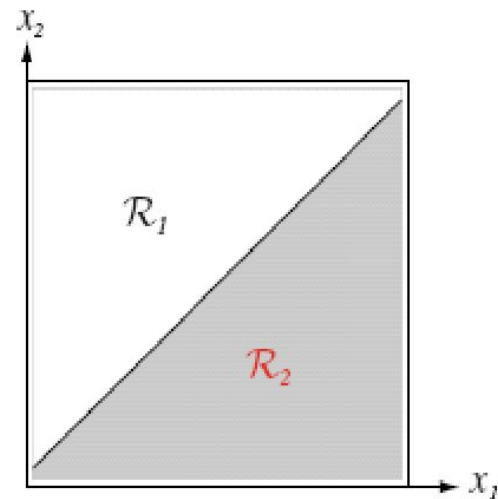
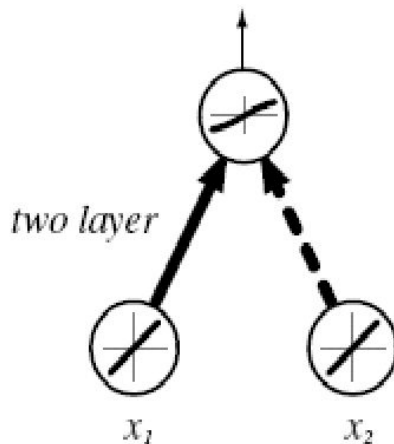
Nota: Nos algoritmos reais, η vai decaindo com o número de épocas.

Algoritmo do perceptron - exemplo



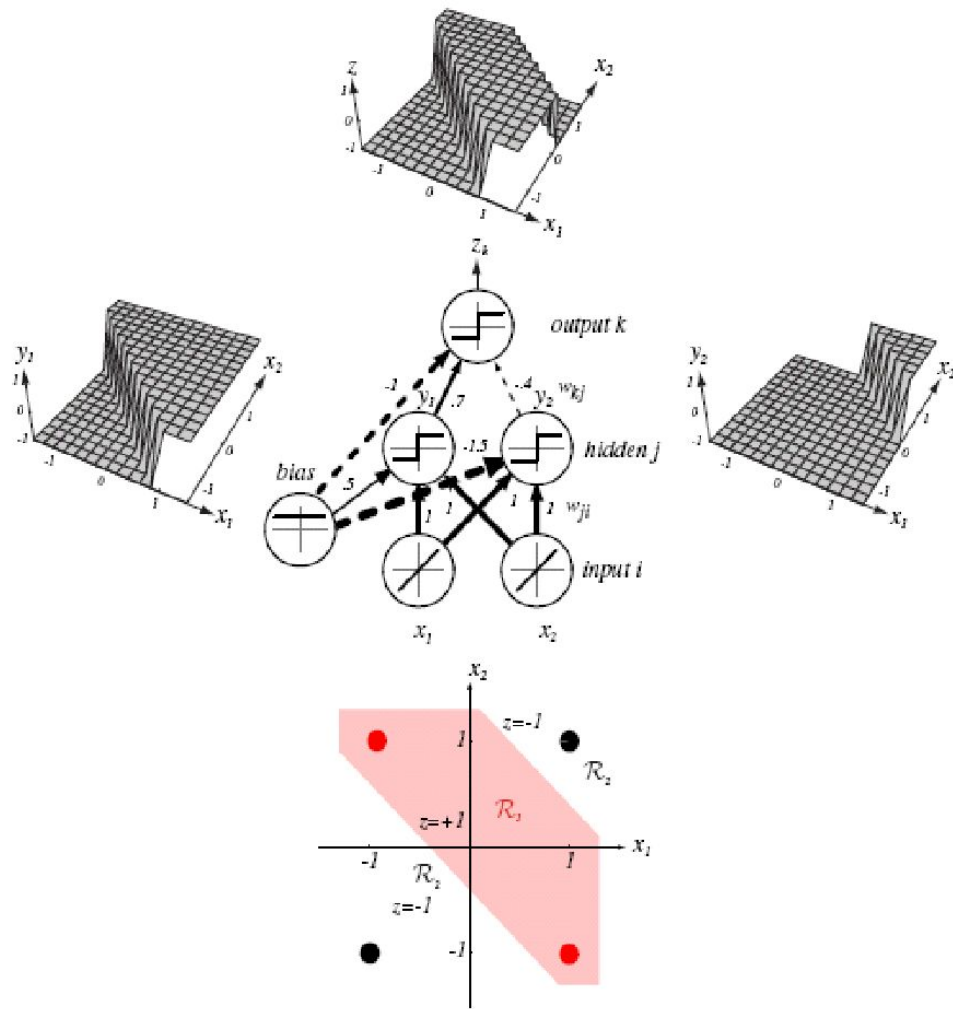
Perceptron

- A soma ponderada feita pelo perceptron nada mais é que um plano num espaço multidimensional, ou hiperplano.
- A função de ativação é usada, normalmente, para atribuir rótulos para os exemplos. Nos dois casos ao lado, \mathcal{R}_1 e \mathcal{R}_2 são regiões onde todos os pontos têm rótulos do mesmo valor em cada região.
- Quando combinamos dois, ou mais perceptrons, podemos criar regiões mais complexas.

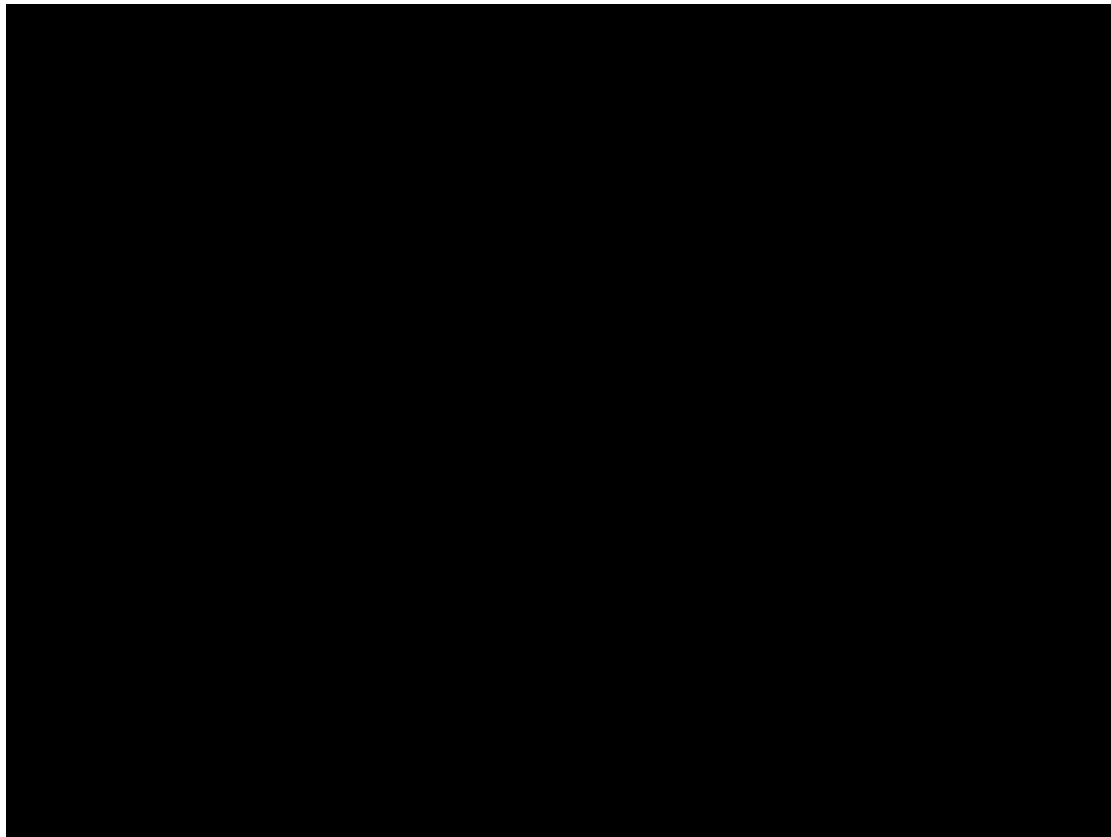


XOR - OU exclusivo

- Um problema clássico que necessita de mais de um perceptron é o XOR, ou OU exclusivo.



Algoritmo do pocket perceptron - exemplo



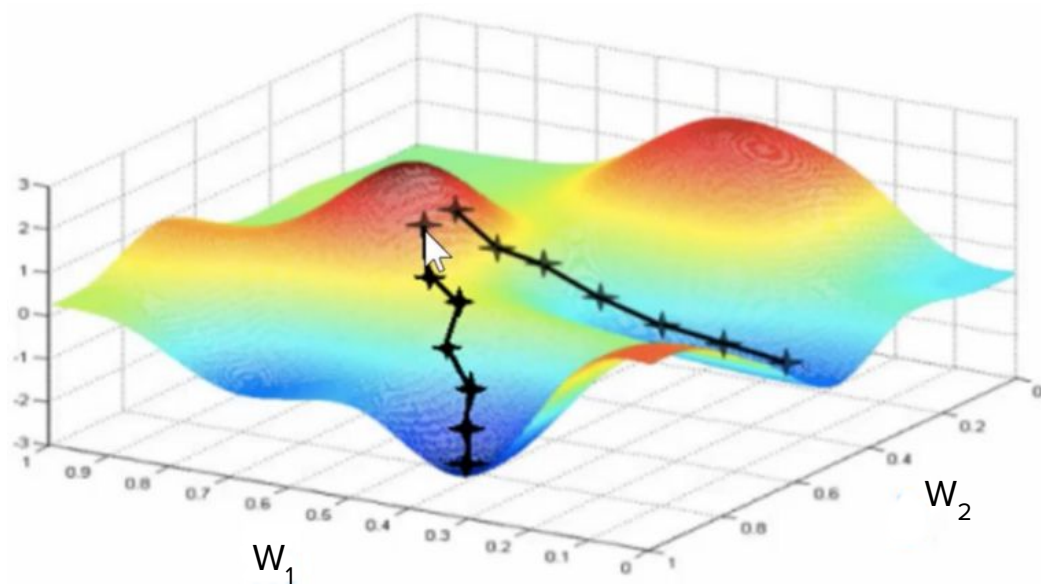
Otimização/treinamento de redes neurais

- O treinamento de redes neurais de várias camadas é feito por um algoritmo conhecido como **backpropagation** que tenta minimizar uma **função de perda**:

$$E(\mathbf{w}) = \sum_{j=1}^N (y_j - f_{\mathbf{w}}(\mathbf{x}_j))^2$$

- Os pesos são atualizados segundo a fórmula:

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha \frac{\partial E}{\partial \mathbf{w}}$$

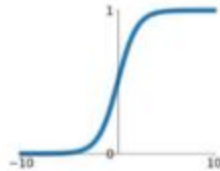


Funções de ativação

Activation Functions

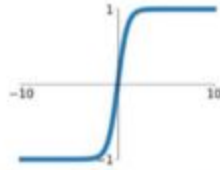
Sigmoid

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



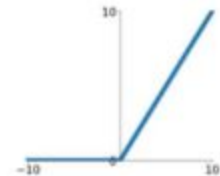
tanh

$$\tanh(x)$$



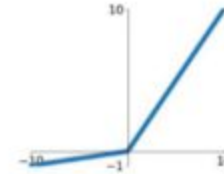
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$



Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

