Deep Learning

Neurônio MP e Perceptron

Lucas Silveira Kupssinskü

Machine Learning Theory and Applications Laboratory
PUCRS

agosto de 2026

Overview

Visão Geral

Bioinspiração

- Quando não sabemos como fazer algo podemos copiar!
- Temos diversos exemplos de inteligência na natureza
- Computação Bioinspirada
  - Algoritmos Genéticos
  - Computação Evolutiva
  - Redes Neurais Artificiais...

Visão Geral

MP Neuron[1]

- Importância histórica por ser o primeiro modelo computacional de um neurônio
- Possibilitou mapear a lógica proposicional em neurônios artificiais
- Possível origem do termo *neurônio artificial* e *redes neurais artificiais*
- Modelo matemático, sem implementação concreta em um dispositivo computacional

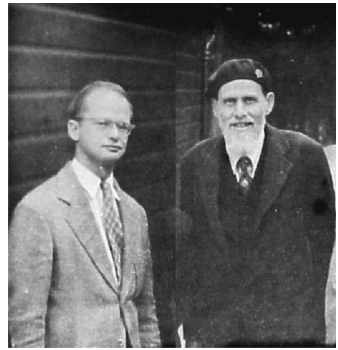
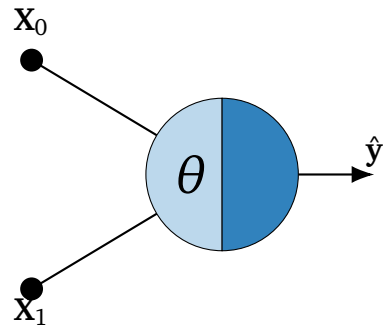


Figura: Warren S. McCulloch (à direita) e Walter Pitts (à esquerda).

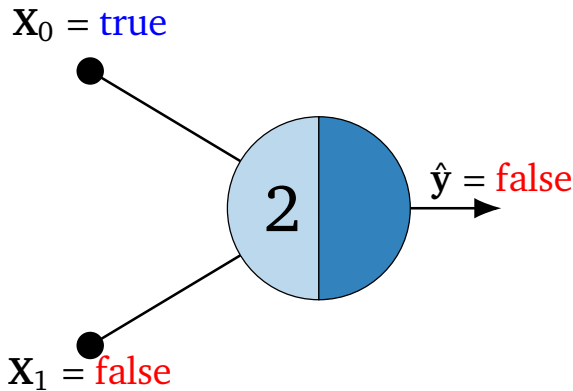
[1] Warren S McCulloch e Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". Em: The bulletin of mathematical biophysics 5 (1943), pp. 115–133.

MP Neuron – Funcionamento

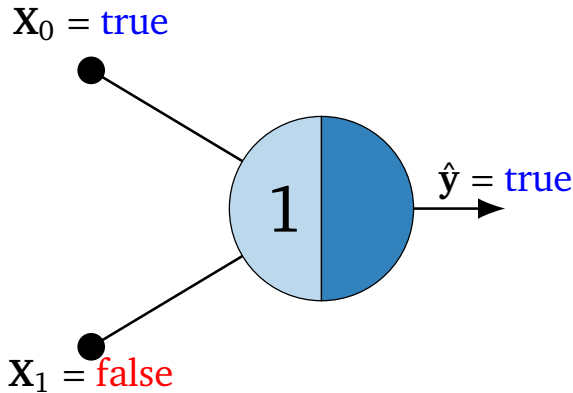
- Entradas binárias
- Estímulos podem ser:
  - Excitatórios: Ativos quando a entrada é Verdadeira
  - Inibitórios: Ativos quando a entrada é Falsa
- Neurônio dispara quando a quantidade de estímulos excitatórios ultrapassa um limiar θ



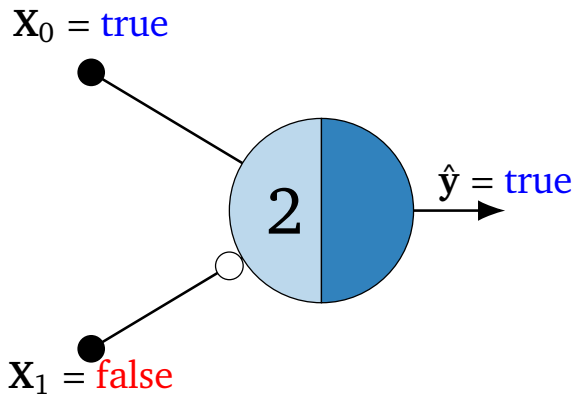
MP Neuron – Exemplo como *AND*



MP Neuron – Exemplo como *OR*



MP Neuron – Entrada inibitória X_0 *AND* NOT X_1



Visão Geral

Perceptron

Frank Rosenblatt[2] fez algumas modificações no Neurônio MP

$$\hat{\mathbf{y}} = \varphi(\mathbf{X}\boldsymbol{\theta}^\top)$$

- Entradas são números reais
- Cada entrada é ponderada por um parâmetro
- Existe um algoritmo de aprendizado para definir $\boldsymbol{\theta}$
- Função de ativação $\varphi(z) = \text{sinal}(z) = \begin{cases} +1, & z \geq 0 \\ -1, & z < 0 \end{cases}$

[2] F Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain". en. Em: Psychol. Rev. 65.6 (nov. de 1958), pp. 386–408.

Perceptron

"(...) identities of this sort must be *learned* (...) the number of functional units in the storage system or memory, should be less than the number of forms or memories to be retained."[3]

[3] F Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain". en. Em: Psychol. Rev. 65.6 (nov. de 1958), pp. 386–408.

Perceptron

- Classificador de imagens
- Resolução 20×20
- 400 fotocélulas capturavam a imagem
- Pesos eram controlados por potenciômetros

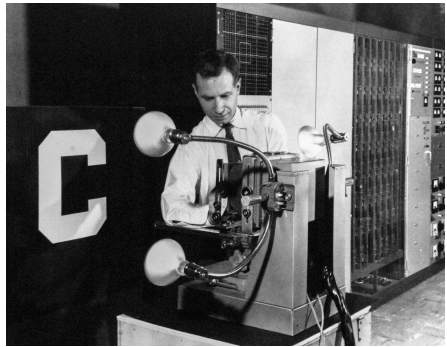


Figura: Frank Rosenblatt e o Perceptron

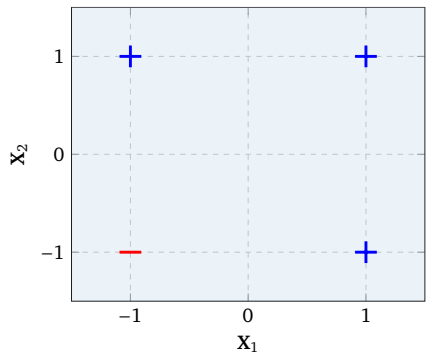
Algoritmo de Aprendizado do Perceptron (PLA)

1: Assuma $\mathbf{X}^{(i)}$ com primeira componente igual a 1 (truque do bias)
2: Inicialize os pesos $\boldsymbol{\theta}$ aleatoriamente
3: **while** existirem observações mal classificadas **do**
4: Selecione $\mathbf{X}^{(i)}$ tal que sinal$(\mathbf{X}^{(i)}\boldsymbol{\theta}^{\top}) \neq \mathbf{y}^{(i)}$
5: Atualize: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \mathbf{y}^{(i)}\mathbf{X}^{(i)}$
6: **end while**

Algoritmo de Aprendizado do Perceptron (PLA)

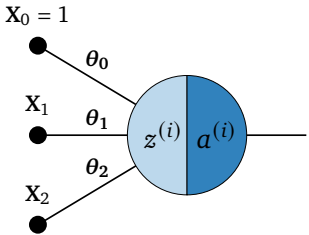
- Vamos ver um exemplo de treinamento com base no conjunto de dados abaixo
- À direita teremos uma representação do espaço de *features* junto da fronteira de decisão

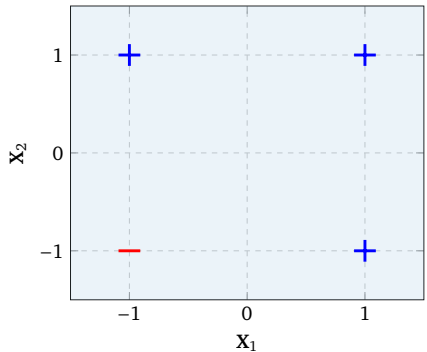
X_1	X_2	y
1	1	1
-1	1	1
1	-1	1
-1	-1	-1



Algoritmo de Aprendizado do Perceptron (PLA)

1: Assuma $\mathbf{X}^{(i)}$ com primeira componente igual a 1 (truque do bias)
2: Inicialize os pesos $\boldsymbol{\theta}$ aleatoriamente
3: **while** existirem observações mal classificadas **do**
4: Selecione $\mathbf{X}^{(i)}$ tal que sinal$(\mathbf{X}^{(i)}\boldsymbol{\theta}^{\top}) \neq \mathbf{y}^{(i)}$
5: Atualize: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \mathbf{y}^{(i)}\mathbf{X}^{(i)}$
6: **end while**

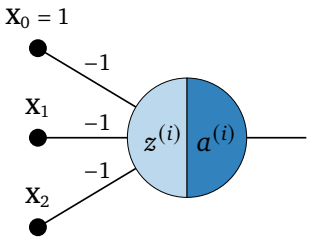


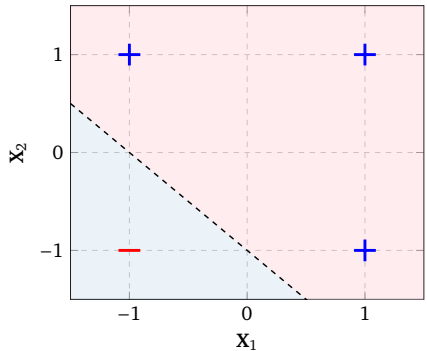


Algoritmo de Aprendizado do Perceptron (PLA)

1: Inicialize os pesos $\boldsymbol{\theta}$ aleatoriamente
2: **while** existirem observações mal classificadas **do**
3: Selecione $\mathbf{X}^{(i)}$ tal que sinal$(\mathbf{X}^{(i)}\boldsymbol{\theta}^{\top}) \neq \mathbf{y}^{(i)}$
4: Atualize: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \mathbf{y}^{(i)}\mathbf{X}^{(i)}$
5: **end while**

$\boldsymbol{\theta}^{\top} = [-1, -1, -1]$

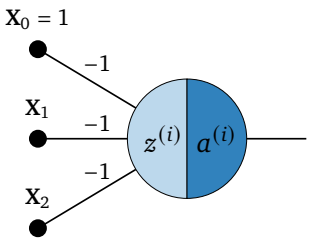


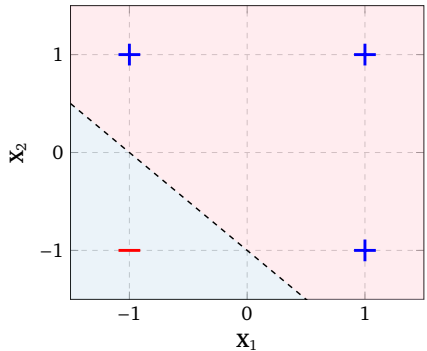


Algoritmo de Aprendizado do Perceptron (PLA)

1: Inicialize os pesos $\boldsymbol{\theta}$ aleatoriamente
2: **while** existirem observações mal classificadas **do**
3: Selecione $\mathbf{X}^{(i)}$ tal que sinal$(\mathbf{X}^{(i)}\boldsymbol{\theta}^{\top}) \neq \mathbf{y}^{(i)}$
4: Atualize: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \mathbf{y}^{(i)}\mathbf{X}^{(i)}$
5: **end while**

$\boldsymbol{\theta}^{\top} = [-1, -1, -1]$

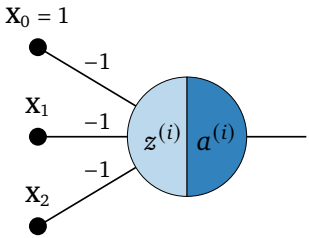


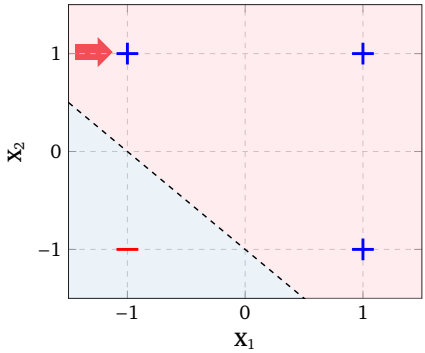


Algoritmo de Aprendizado do Perceptron (PLA)

1: Inicialize os pesos $\boldsymbol{\theta}$ aleatoriamente
2: **while** existirem observações mal classificadas **do**
3: Selecione $\mathbf{X}^{(i)}$ tal que sinal$(\mathbf{X}^{(i)}\boldsymbol{\theta}^\top) \neq \mathbf{y}^{(i)}$
4: Atualize: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \mathbf{y}^{(i)}\mathbf{X}^{(i)}$
5: **end while**

$\boldsymbol{\theta}^\top = [-1, -1, -1]$

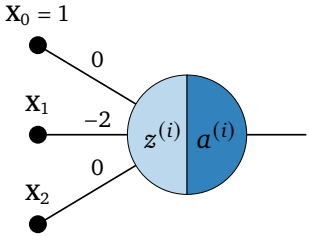


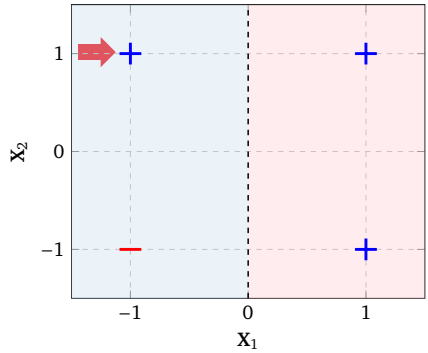


Algoritmo de Aprendizado do Perceptron (PLA)

1: Inicialize os pesos $\boldsymbol{\theta}$ aleatoriamente
2: **while** existirem observações mal classificadas **do**
3: Selecione $\mathbf{X}^{(i)}$ tal que sinal$(\mathbf{X}^{(i)}\boldsymbol{\theta}^{\top}) \neq \mathbf{y}^{(i)}$
4: Atualize: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \mathbf{y}^{(i)}\mathbf{X}^{(i)}$
5: **end while**

$$\boldsymbol{\theta}^{\top} = [0, -2, 0] = [-1, -1, -1] + [1, -1, 1]$$

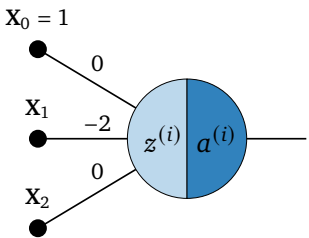


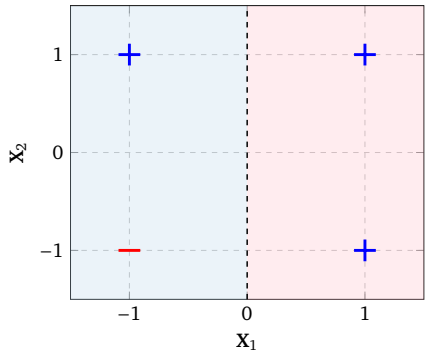


Algoritmo de Aprendizado do Perceptron (PLA)

1: Inicialize os pesos $\boldsymbol{\theta}$ aleatoriamente
2: **while** existirem observações mal classificadas **do**
3: Selecione $\mathbf{X}^{(i)}$ tal que sinal$(\mathbf{X}^{(i)}\boldsymbol{\theta}^\top) \neq \mathbf{y}^{(i)}$
4: Atualize: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \mathbf{y}^{(i)}\mathbf{X}^{(i)}$
5: **end while**

$\boldsymbol{\theta}^\top = [0, -2, 0]$

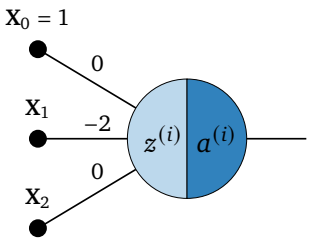


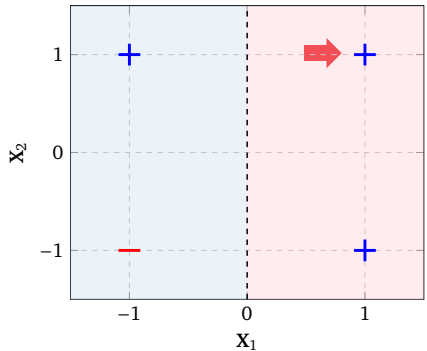


Algoritmo de Aprendizado do Perceptron (PLA)

1: Inicialize os pesos $\boldsymbol{\theta}$ aleatoriamente
2: **while** existirem observações mal classificadas **do**
3: Selecione $\mathbf{X}^{(i)}$ tal que sinal$(\mathbf{X}^{(i)}\boldsymbol{\theta}^\top) \neq \mathbf{y}^{(i)}$
4: Atualize: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \mathbf{y}^{(i)}\mathbf{X}^{(i)}$
5: **end while**

$\boldsymbol{\theta}^\top = [0, -2, 0]$

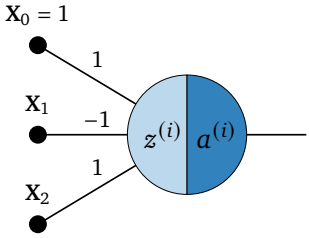


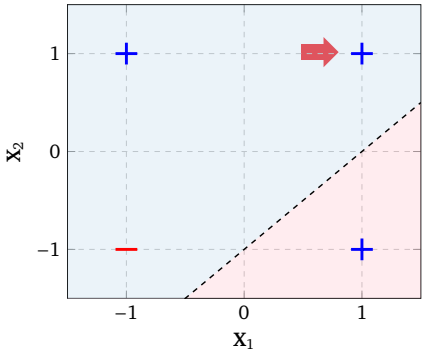


Algoritmo de Aprendizado do Perceptron (PLA)

1: Inicialize os pesos $\boldsymbol{\theta}$ aleatoriamente
2: **while** existirem observações mal classificadas **do**
3: Selecione $\mathbf{X}^{(i)}$ tal que sinal$(\mathbf{X}^{(i)}\boldsymbol{\theta}^\top) \neq \mathbf{y}^{(i)}$
4: Atualize: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \mathbf{y}^{(i)}\mathbf{X}^{(i)}$
5: **end while**

$$\boldsymbol{\theta}^\top = [1, -1, 1] = [0, -2, 0] + [1, 1, 1]$$

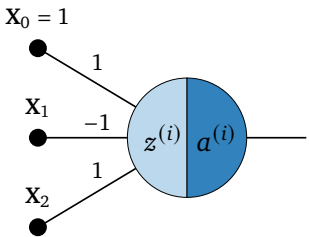


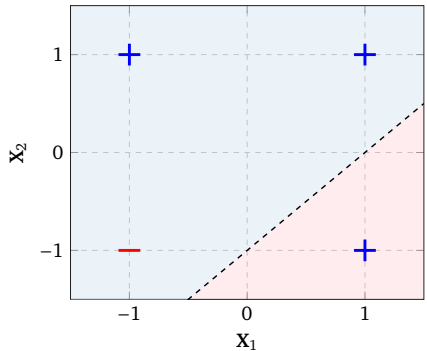


Algoritmo de Aprendizado do Perceptron (PLA)

1: Inicialize os pesos $\boldsymbol{\theta}$ aleatoriamente
2: **while** existirem observações mal classificadas **do**
3: Selecione $\mathbf{X}^{(i)}$ tal que sinal$(\mathbf{X}^{(i)}\boldsymbol{\theta}^\top) \neq \mathbf{y}^{(i)}$
4: Atualize: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \mathbf{y}^{(i)}\mathbf{X}^{(i)}$
5: **end while**

$\boldsymbol{\theta}^\top = [1, -1, 1]$

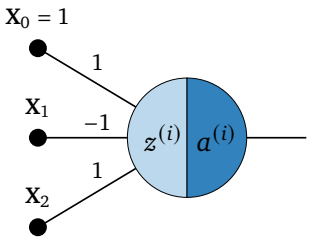


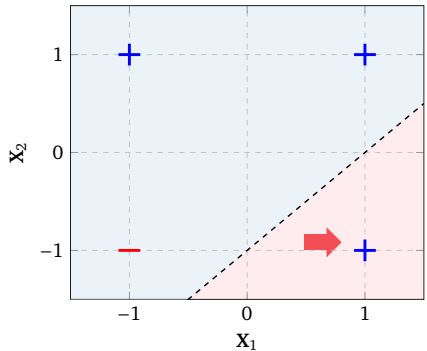


Algoritmo de Aprendizado do Perceptron (PLA)

1: Inicialize os pesos $\boldsymbol{\theta}$ aleatoriamente
2: **while** existirem observações mal classificadas **do**
3: Selecione $\mathbf{X}^{(i)}$ tal que sinal$(\mathbf{X}^{(i)}\boldsymbol{\theta}^\top) \neq \mathbf{y}^{(i)}$
4: Atualize: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \mathbf{y}^{(i)}\mathbf{X}^{(i)}$
5: **end while**

$\boldsymbol{\theta}^\top = [1, -1, 1]$

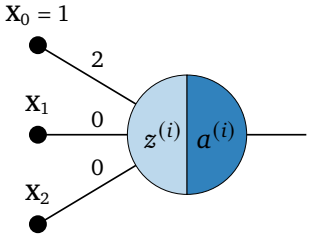


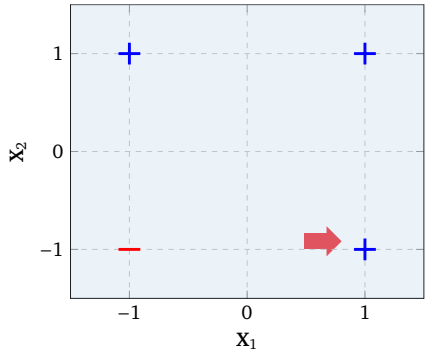


Algoritmo de Aprendizado do Perceptron (PLA)

1: Inicialize os pesos $\boldsymbol{\theta}$ aleatoriamente
2: **while** existirem observações mal classificadas **do**
3: Selecione $\mathbf{X}^{(i)}$ tal que sinal$(\mathbf{X}^{(i)}\boldsymbol{\theta}^{\top}) \neq \mathbf{y}^{(i)}$
4: Atualize: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \mathbf{y}^{(i)}\mathbf{X}^{(i)}$
5: **end while**

$\boldsymbol{\theta}^{\top} = [2, 0, 0] = [1, -1, 1] + [1, 1, -1]$

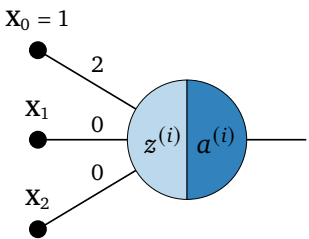


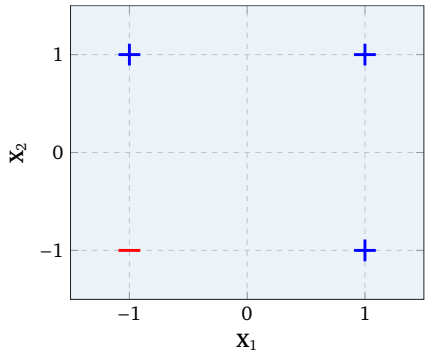


Algoritmo de Aprendizado do Perceptron (PLA)

1: Inicialize os pesos $\boldsymbol{\theta}$ aleatoriamente
2: **while** existirem observações mal classificadas **do**
3: Selecione $\mathbf{X}^{(i)}$ tal que sinal$(\mathbf{X}^{(i)}\boldsymbol{\theta}^\top) \neq \mathbf{y}^{(i)}$
4: Atualize: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \mathbf{y}^{(i)}\mathbf{X}^{(i)}$
5: **end while**

$\boldsymbol{\theta}^\top = [2, 0, 0]$

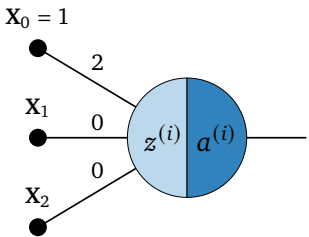


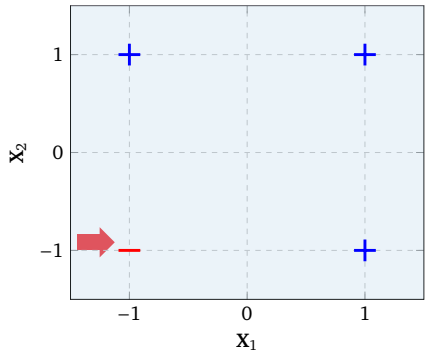


Algoritmo de Aprendizado do Perceptron (PLA)

1: Inicialize os pesos $\boldsymbol{\theta}$ aleatoriamente
2: **while** existirem observações mal classificadas **do**
3: Selecione $\mathbf{X}^{(i)}$ tal que sinal$(\mathbf{X}^{(i)}\boldsymbol{\theta}^\top) \neq \mathbf{y}^{(i)}$
4: Atualize: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \mathbf{y}^{(i)}\mathbf{X}^{(i)}$
5: **end while**

$\boldsymbol{\theta}^\top = [2, 0, 0]$

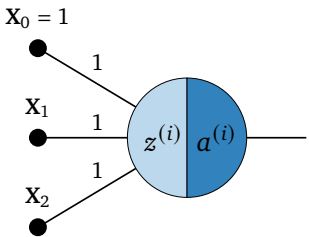


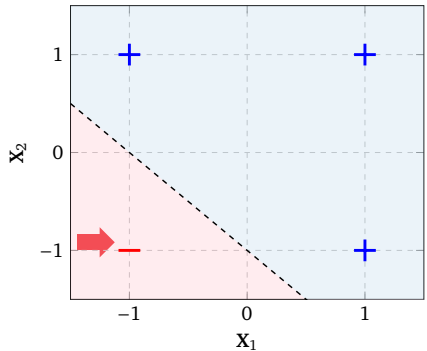


Algoritmo de Aprendizado do Perceptron (PLA)

1: Inicialize os pesos $\boldsymbol{\theta}$ aleatoriamente
2: **while** existirem observações mal classificadas **do**
3: Selecione $\mathbf{X}^{(i)}$ tal que sinal$(\mathbf{X}^{(i)}\boldsymbol{\theta}^{\top}) \neq \mathbf{y}^{(i)}$
4: Atualize: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \mathbf{y}^{(i)}\mathbf{X}^{(i)}$
5: **end while**

$$\boldsymbol{\theta}^{\top} = [1, 1, 1] = [2, 0, 0] + (-1)[1, -1, -1]$$

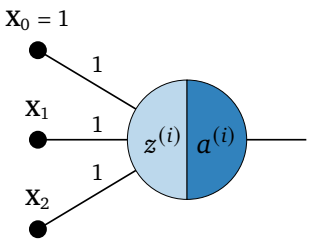


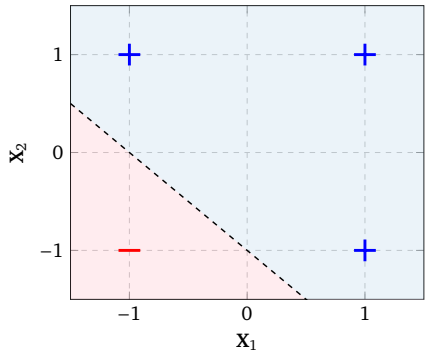


Algoritmo de Aprendizado do Perceptron (PLA)

1: Inicialize os pesos $\boldsymbol{\theta}$ aleatoriamente
2: **while** existirem observações mal classificadas **do**
3: Selecione $\mathbf{X}^{(i)}$ tal que sinal$(\mathbf{X}^{(i)}\boldsymbol{\theta}^\top) \neq \mathbf{y}^{(i)}$
4: Atualize: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \mathbf{y}^{(i)}\mathbf{X}^{(i)}$
5: **end while**

$\boldsymbol{\theta}^\top = [1, 1, 1]$

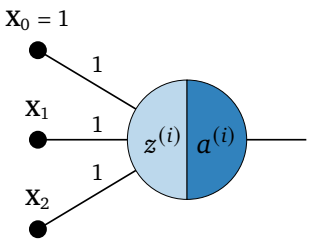


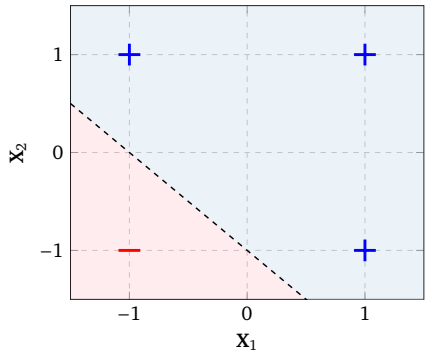


Algoritmo de Aprendizado do Perceptron (PLA)

1: Inicialize os pesos $\boldsymbol{\theta}$ aleatoriamente
2: **while** existirem observações mal classificadas **do**
3: Selecione $\mathbf{X}^{(i)}$ tal que sinal$(\mathbf{X}^{(i)}\boldsymbol{\theta}^\top) \neq \mathbf{y}^{(i)}$
4: Atualize: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \mathbf{y}^{(i)}\mathbf{X}^{(i)}$
5: **end while**

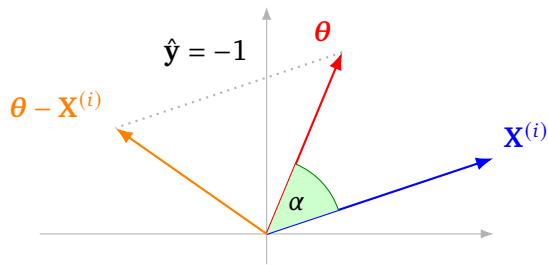
$\boldsymbol{\theta}^\top = [1, 1, 1]$





Considerações sobre o PLA

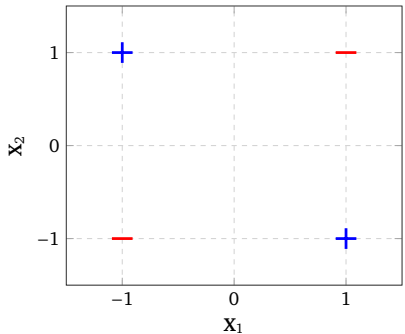
- Descoberto empiricamente
- Intuição geométrica: produto escalar
  - positivo quando $\alpha < 90°$
  - negativo quando $\alpha > 90°$
- A regra de atualização de $\boldsymbol{\theta}$ ajusta o ângulo do vetor de pesos em relação às instâncias
- Sempre converge para problemas linearmente separáveis



Visão Geral

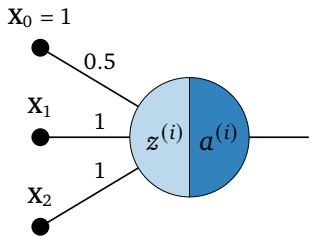
Limitação do Perceptron

- O Perceptron é um modelo linear (traça um hiperplano como fronteira de decisão)
- Marvin Minsky e Seymour Papert salientaram essa limitação com o problema XOR[4]

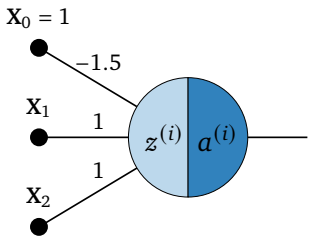


[4] Marvin Minsky e Seymour Papert. "Perceptron: an introduction to computational geometry". Em: The MIT Press, Cambridge, expanded edition 19.88 (1969), p. 2.

Vamos observar esses dois Perceptrons

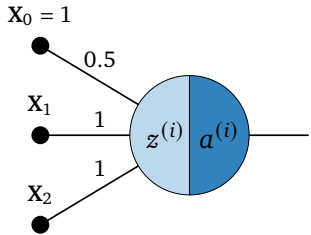


x_1	x_2	y
-1	-1	
-1	1	
1	-1	
1	1	



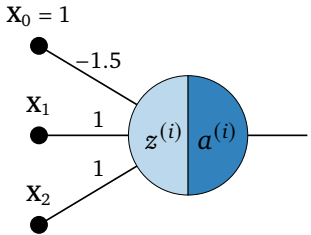
x_1	x_2	y
-1	-1	
-1	1	
1	-1	
1	1	

Vamos observar esses dois Perceptrons



OR

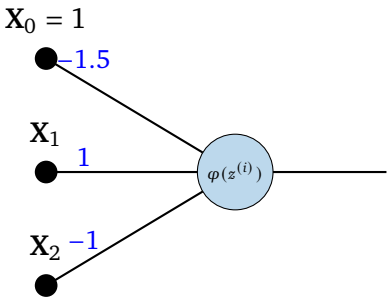
x_1	x_2	y
-1	-1	-1
-1	1	1
1	-1	1
1	1	1

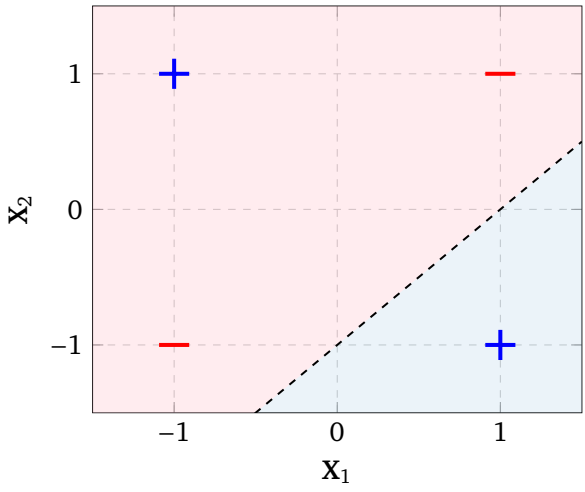


AND:

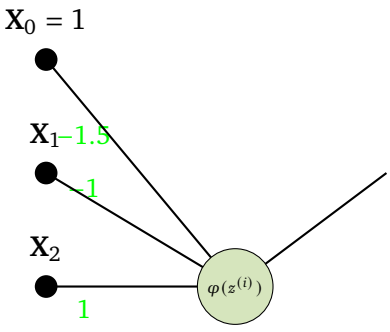
x_1	x_2	y
-1	-1	-1
-1	1	-1
1	-1	-1
1	1	1

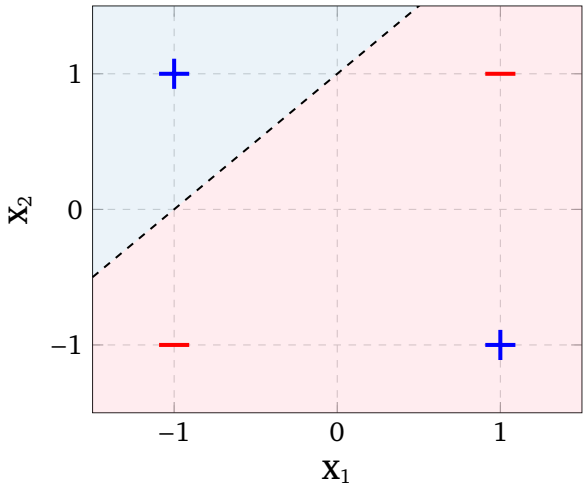
Resolvendo XOR



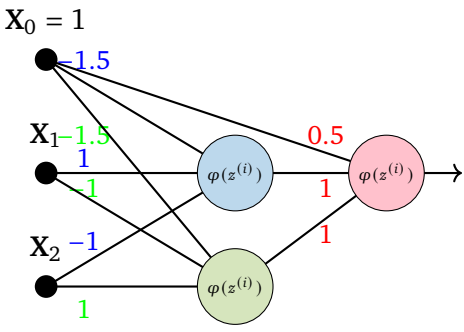


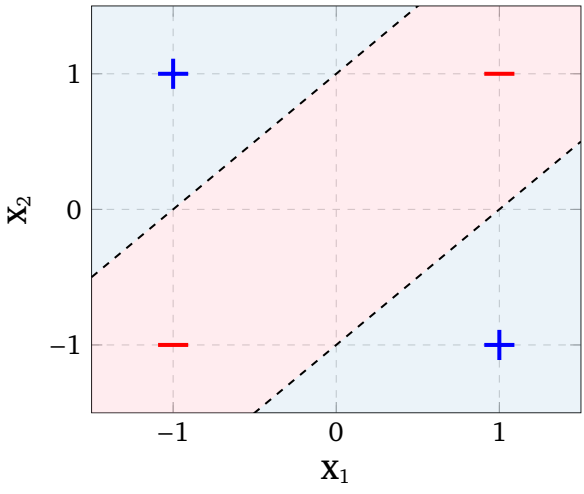
Resolvendo XOR





Resolvendo XOR





Resolvendo XOR

- Quando organizamos vários Perceptrons em camadas chamamos de *MultiLayer Perceptron* (MLP)
- PLA não funciona para MLP
- Essa limitação minou o interesse por redes neurais durante anos (Inverno da IA)
- Para treinar um MLP vamos precisar usar retropropagação de erros (*backpropagation*) e descida de gradiente (*gradient descent*)

Leituras Recomendadas

- Capítulo 1: Y.S. Abu-Mostafa, M. Magdon-Ismail e H.T. Lin. Learning from Data: A Short Course. AMLBook.com, 2012. ISBN: 978-1-60049-006-4
- Capítulo 1: Simon Haykin. Neural networks and learning machines, 3/E. Pearson Education India, 2009

Referências

[1] Y.S. Abu-Mostafa, M. Magdon-Ismail e H.T. Lin. Learning from Data: A Short Course. AMLBook.com, 2012. ISBN: 978-1-60049-006-4.

[2] Simon Haykin. Neural networks and learning machines, 3/E. Pearson Education India, 2009.

[3] Warren S McCulloch e Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". Em: The bulletin of mathematical biophysics 5 (1943), pp. 115–133.

[4] Marvin Minsky e Seymour Papert. "Perceptron: an introduction to computational geometry". Em: The MIT Press, Cambridge, expanded edition 19.88 (1969), p. 2.

[5] F Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain". en. Em: Psychol. Rev. 65.6 (nov. de 1958), pp. 386–408.