Introdução Inteligência Computacional

Profa. Dra. Ana Paula Abrantes de Castro e Shiguemori
anapaula.acs@ifsp.edu.br
Instituto Federal de Educação, Ciência e Tecnologia – IFSP

Dr. Elcio Hideiti Shiguemori
elcio@ieav.cta.br
Instituto de Estudos Avançados - IEAv

Curso: Análise e Desenvolvimento de Sistemas
4º. Semestre - IICI4 – 4 aulas Semanais

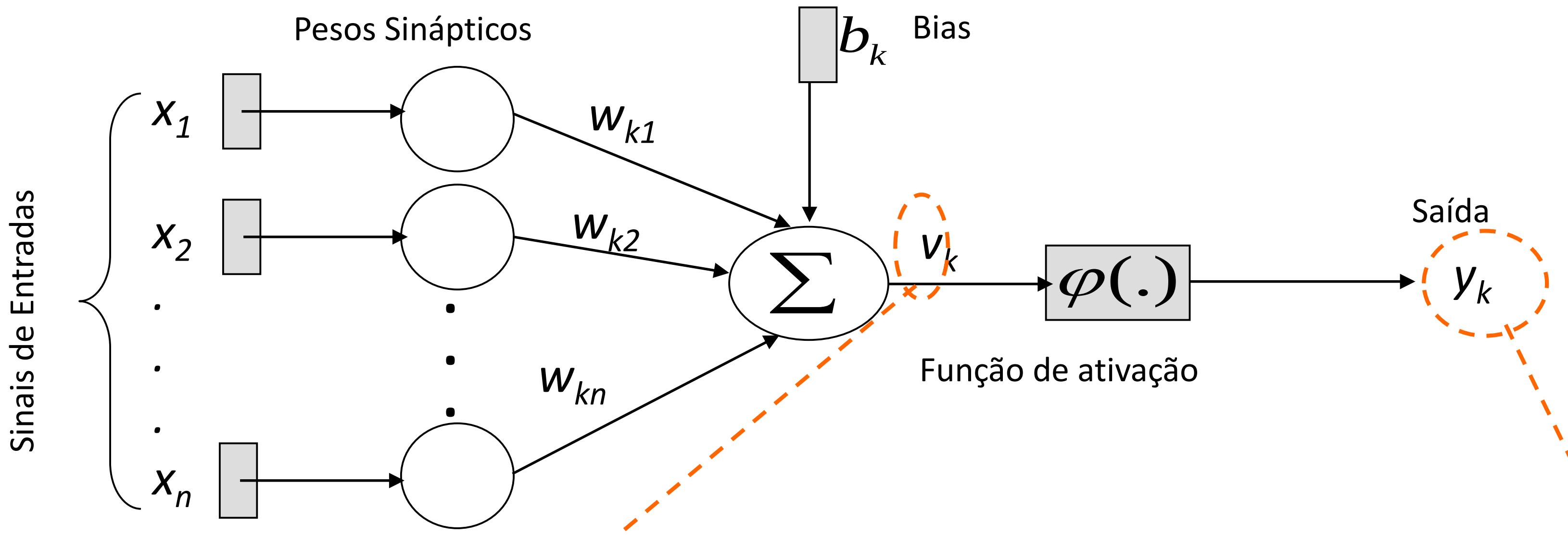
Perceptron

Roteiro

- Modelo Perceptron
- Treinamento
- Interpretação Geométrica
- Exemplos
- Limitações

Modelo do Neurônio Artificial

O Neurônio artificial é função matemática que associa pesos às entradas.



$$v_k = \sum_{j=1}^{n} w_{kj} x_j + b_k \quad \longrightarrow \quad y_k = \varphi(v_k)$$

(McCulloch e Pitts, 1943)

Histórico Perceptron

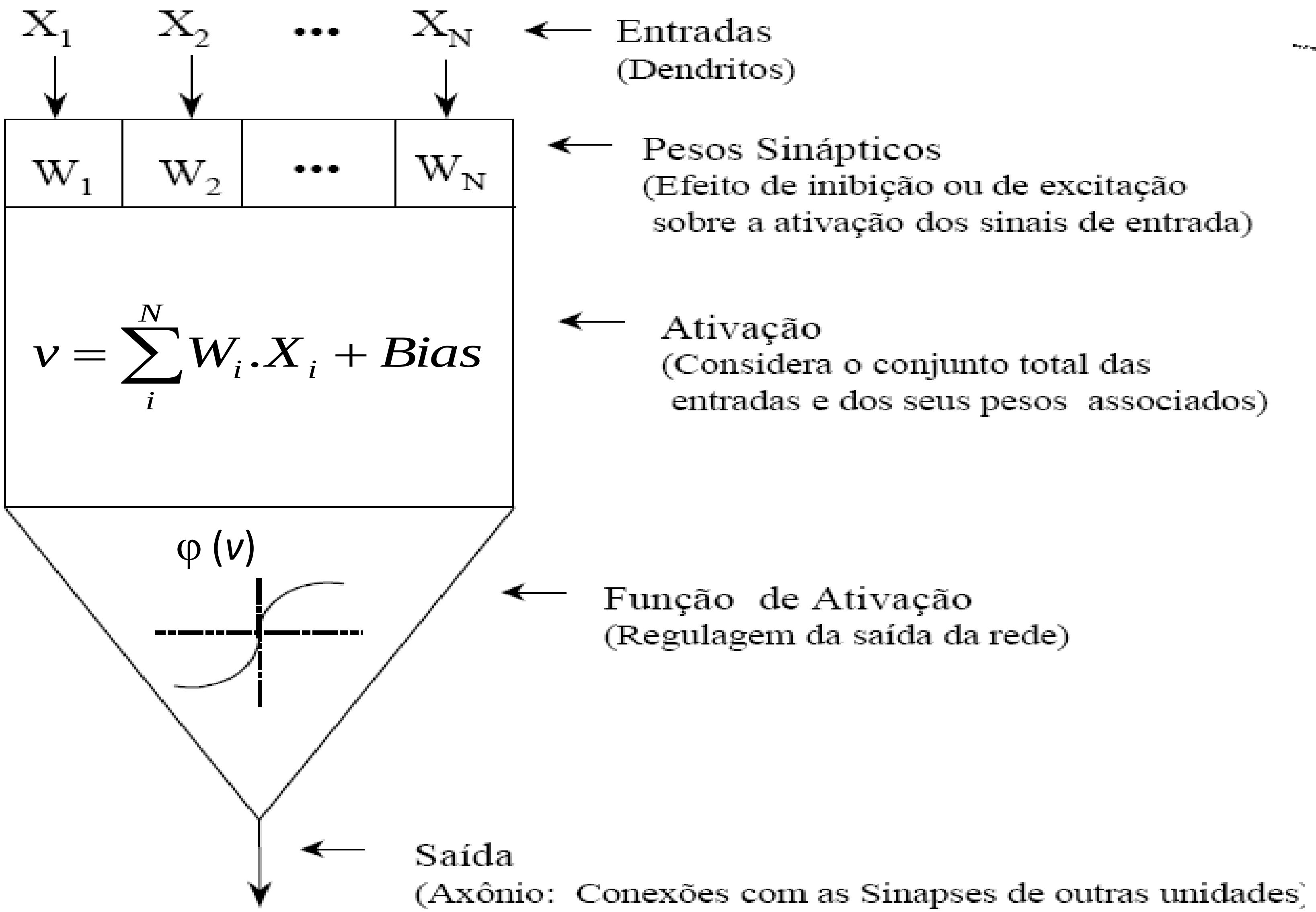
Rosemblat propôs o Perceptron (1958)

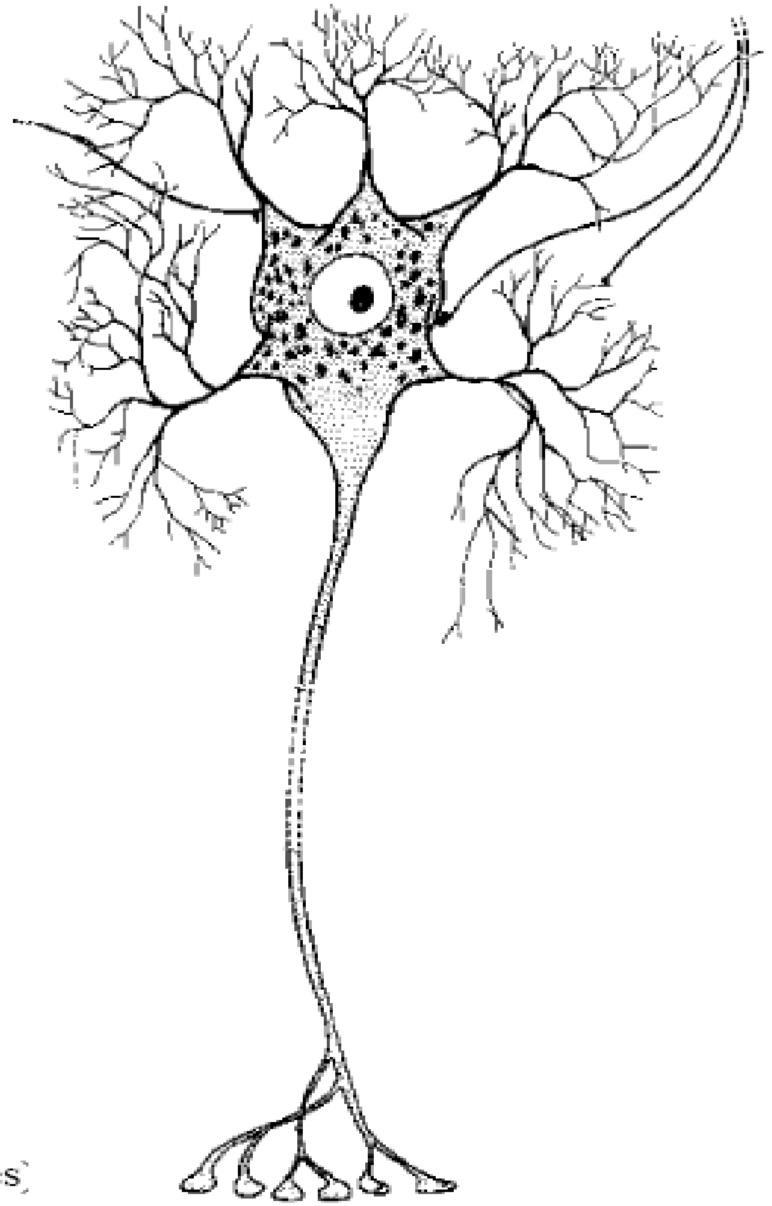
- Como um método inovador de aprendizagem supervisionada
- Demonstração do teorema da convergência
- A forma mais simples de uma rede neural usada para classificação de padrões linearmente separáveis

Características

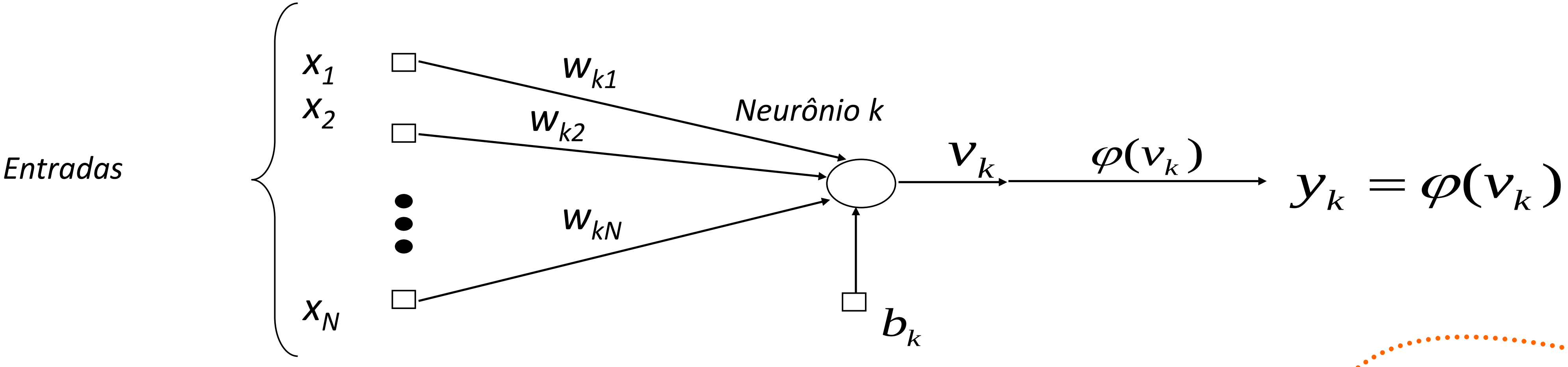
- Aprendizado supervisionado
- Representação binária
- Apenas uma camada de pesos ajustáveis

Modelo do Perceptron





Arquitetura Básica do Perceptron



$$y_k = \varphi\left(\sum_{i=1}^{N} w_{ki} x_i - b_k\right) \qquad \overline{W} = [w_{k0} \quad w_{k1} \quad \cdots \quad w_{kN}] \qquad \overline{X} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_N \end{bmatrix}$$

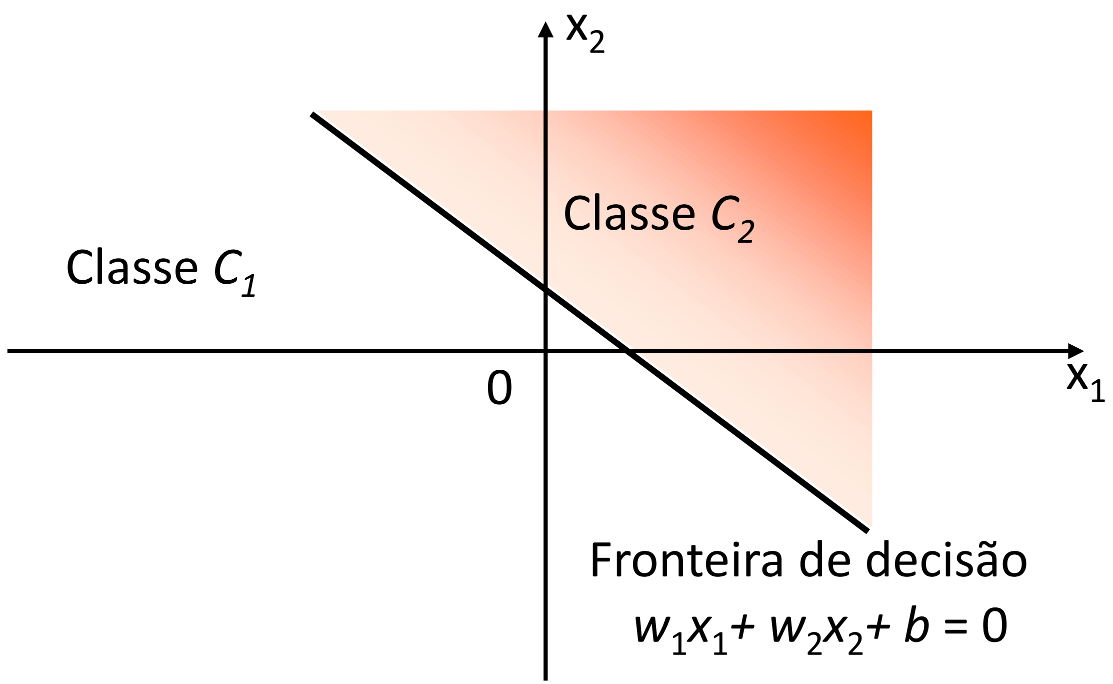
$$b_k = 1 \cdot w_{k0}$$

$$v_k = \varphi(w_{k1}x_1 + \ \cdots \ + w_{kN}x_N + (1)w_{k0})$$

$$= \varphi\left(\sum_{i=0}^{N} w_{ki} x_i\right) = \varphi(\overline{W}^T \overline{X})$$

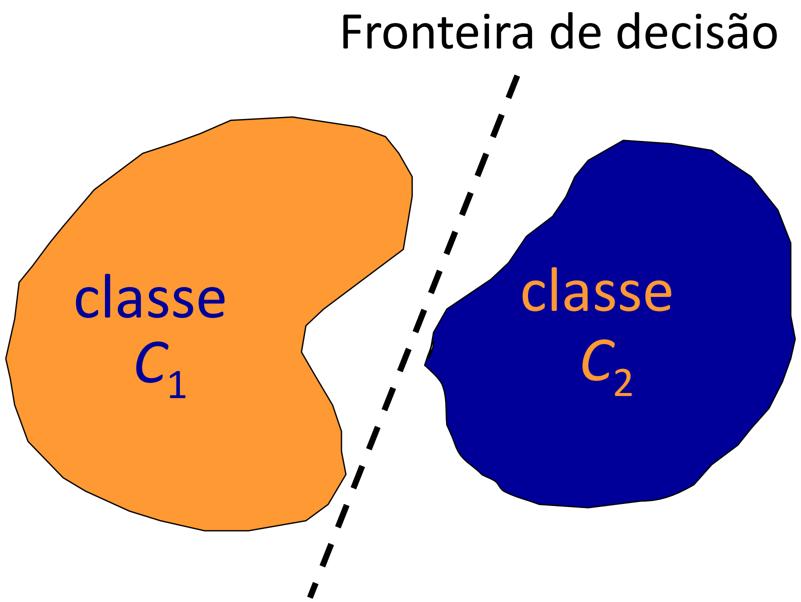
Fronteira de Decisão

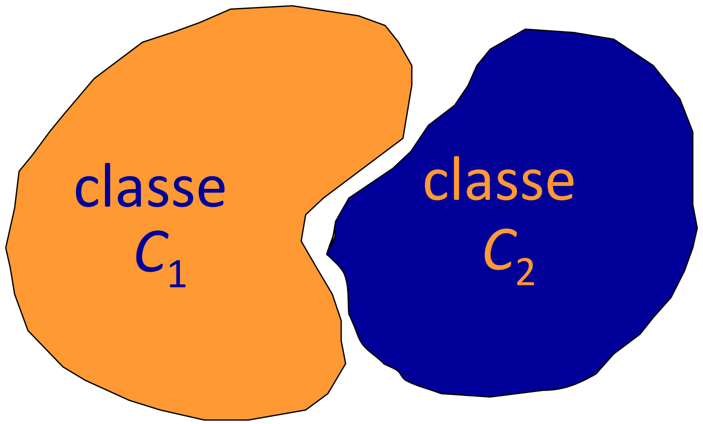
- Atribuir o ponto representado pelas entradas x_1, x_2, ..., x_n à classe C_1 se a saída y do Perceptron for igual a 1 ou à classe C_2 se ela for 0
- O efeito do bias "b" é de deslocar a fronteira de decisão em relação à origem



Limitações do Perceptron

- Não admite mais de uma camada de pesos ajustáveis
- Aprendizado nem sempre ocorre
- As duas classes C_1 e C_2 devem ser linearmente separáveis





Algoritmo de Treinamento

Inicialize w

Repita

Para cada padrão x_i faça

$y_i \leftarrow$ valor de saída do perceptron para o padrão x_i

$e \leftarrow d_i - y_i$

$w \leftarrow w + e * \eta * x_i$

fim

Até (o número de interações máximo ser atingido) ou (o perceptron acertar a saída para todos os padrões).

Fim

Algoritmo de Treinamento

Passo 1. Inicializa pesos: $\overline{W}(0) = \overline{0}$

Passo 2. Ativa a rede - vetor de entrada e a resposta desejada: $\overline{X}(n) \quad e \quad d(n)$

Passo 3. Calcula-se a saída: $y(n) = F(\overline{W}^T(n)\overline{X}(n))$ - *F*(.) é a <u>função sinal</u>

Passo 4. Atualiza os pesos e bias:

$$\overline{w}(n+1) = \overline{w}(n) + \eta[d(n) - y(n)]\overline{X}(n)$$

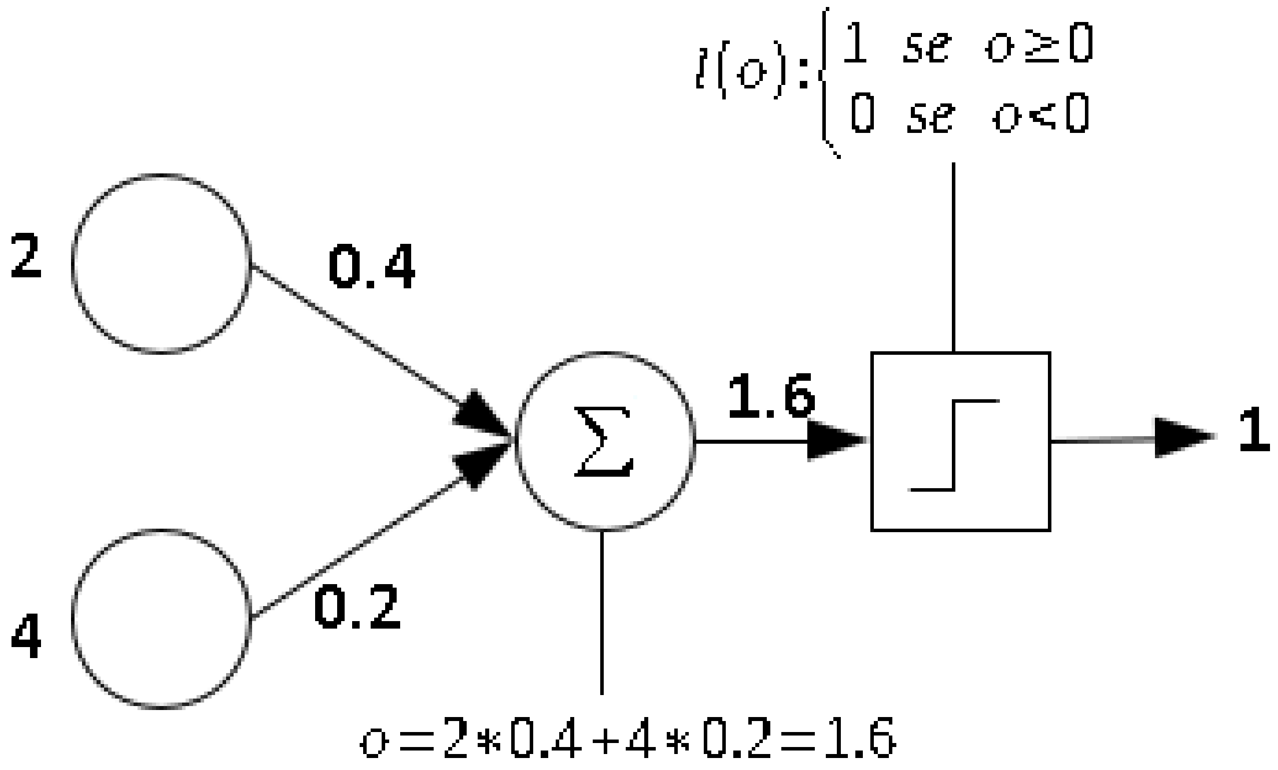
$$b(n+1) = b(n) + \eta[d(n) - y(n)] * 1$$

$$d(n) = \begin{cases} +1 & se \quad \overline{X}(n) \in C_1 \\ -1 & se \quad \overline{X}(n) \in C_2 \end{cases}$$

Passo 5. Incrementa *n* e volta para o Passo 2.

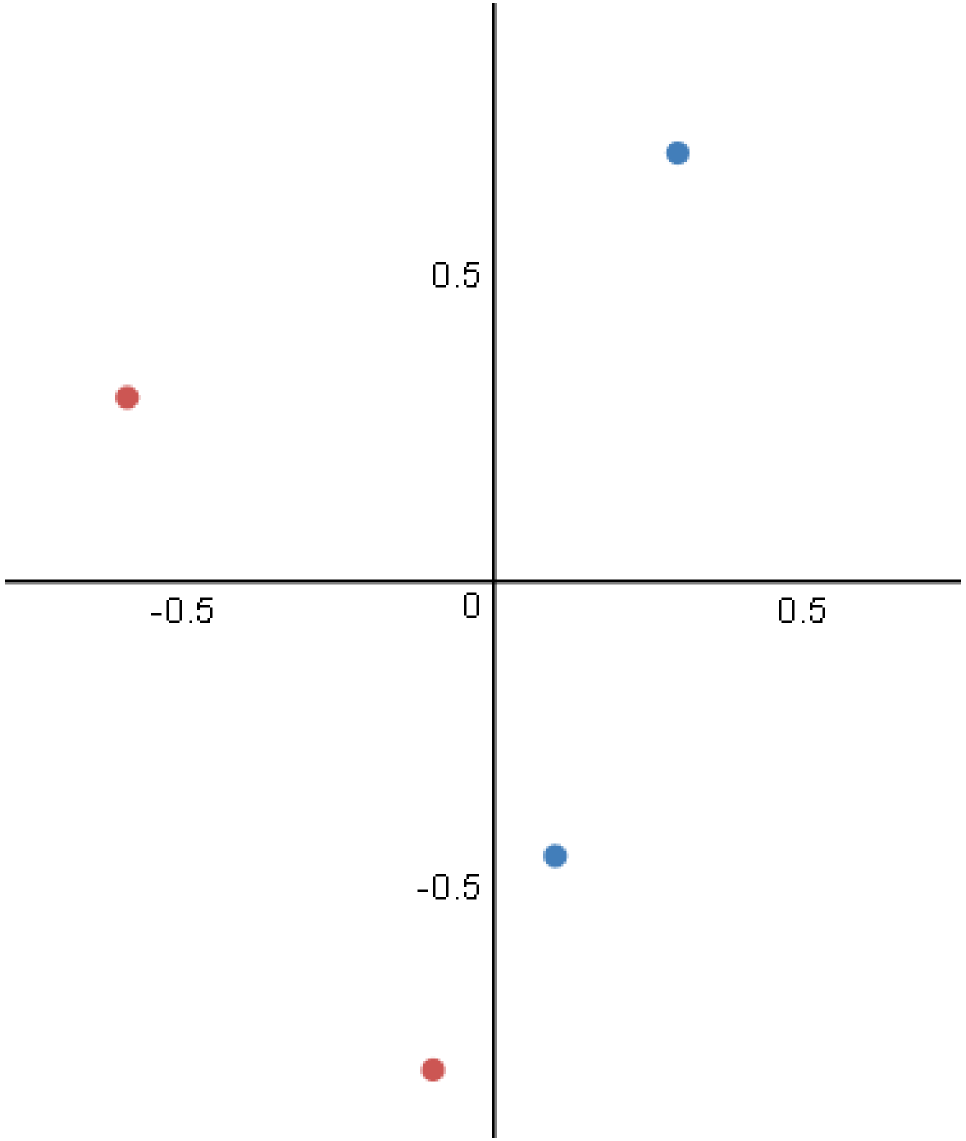
Exemplo

$x_1 = 2$
$x_2 = 4$
$w_1=0.1$
$w_2=0.2$



Exemplo – Classificação 2 Classes

- Treinar o Perceptron para achar um discriminante que divida em duas partes o conjunto de quatro pontos no plano que são Linearmente .



Iniciando os dados

x	y	Classe
0.3	0.7	1
-0.6	0.3	0
-0.1	-0.8	0
0.1	-0.45	1

Pesos

$w_1 = 0.8$
$w_2 = -0.5$

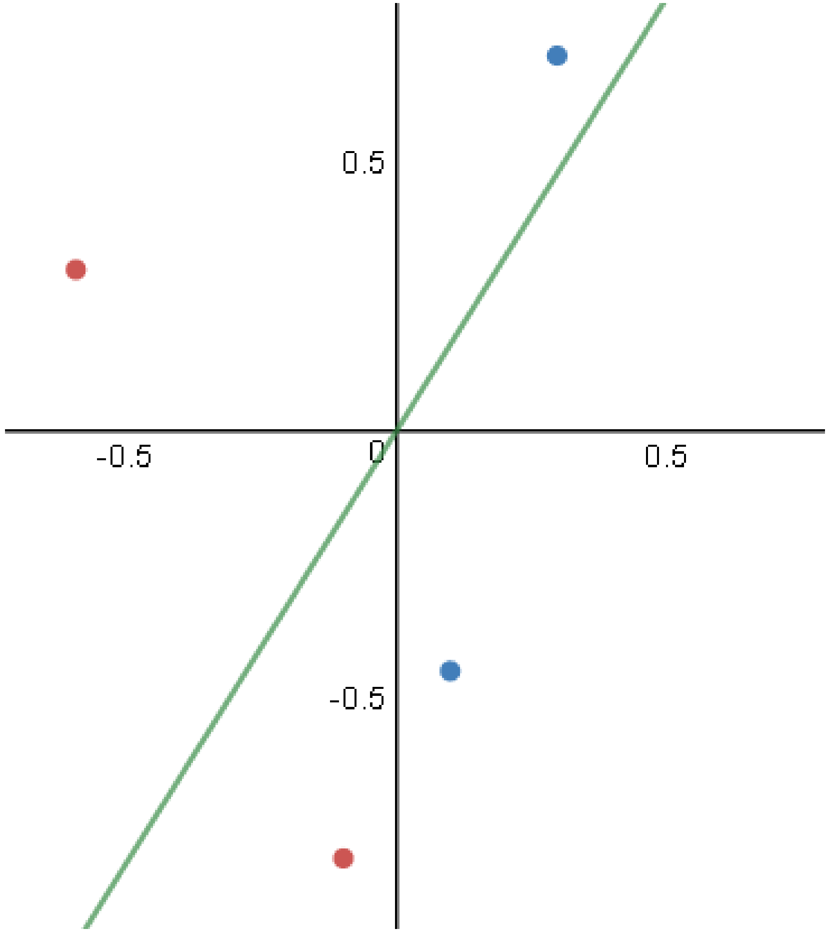
Taxa de aprendizagem

$n = 0.5$

Equações de Treinamento

Toda vez que gera uma saída compara com a classe esperada e computa o erro.

$$e = \left(saida_{desejada} - saida_{rede}\right)$$



Atualização dos pesos

$$\varphi(y) = \begin{cases} 1 \; se \; o \; \geq 0 \\ 0 \; se \; o \; < 0 \end{cases}$$

Atualização dos pesos

$$w_{new} = w_{old} + nex_i$$

Onde x_i é o elemento correspondente do vetor de input que gerou o erro.

Primeira Iteração

Entradas: $x_1 = 0.3 \;\; e \;\; x_2 = 0.7 \;\; \rightarrow Classe = 1$

Pesos: $w_1 = 0.8 \;\; e \;\; w_2 = -0.5$

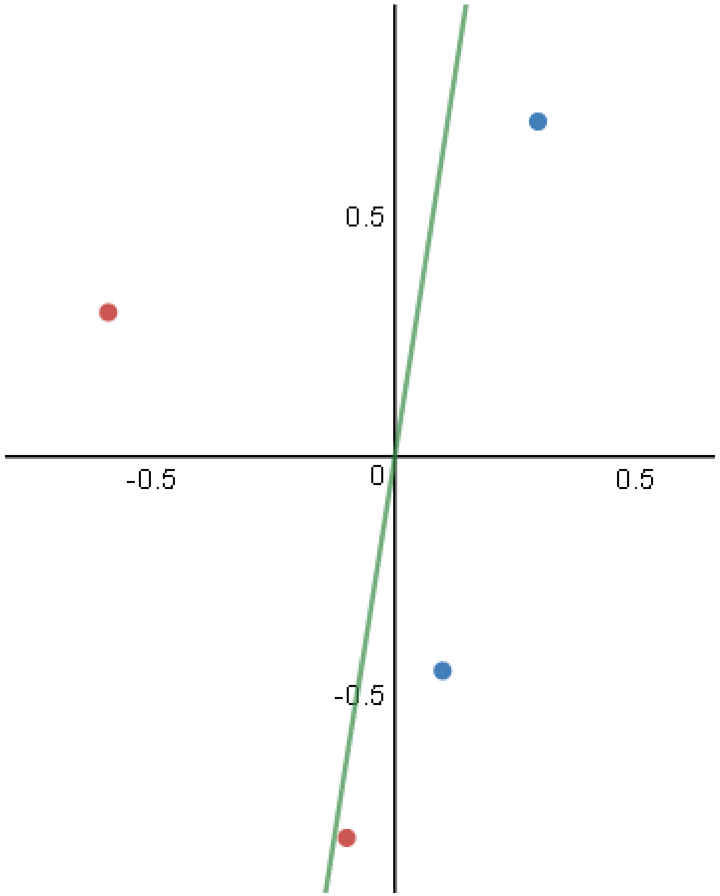
Saída: $y_1 = 0.3 * 0.8 + 0.7 * (-0.5) = -0.11$

Função de Ativação: $\varphi(-0.11) = \begin{cases} 1 \; se \; o \; \geq 0 \\ 0 \; se \; o \; < 0 \end{cases} \rightarrow 0$

Erro: $e_1 = (1 - 0) = 1$

Atualização dos Pesos: $w_1 = 0.8 + 0.5 * 1 * 0.3 = 0.95$

$w_2 = -0.5 + 0.5 * 1 * 0.7 = -0.15$



Segunda Iteração

Entradas: $x_1 = -0.6 \ \ e \ \ x_2 = 0.3 \ \rightarrow Classe = 0$

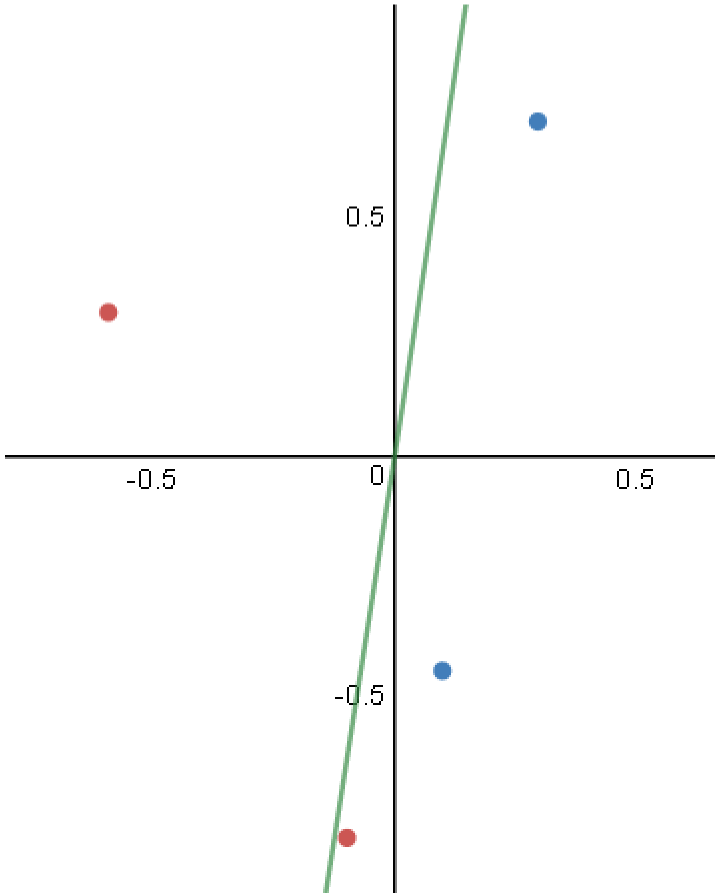
Pesos: $w_1 = 0.95 \ \ e \ \ w_2 = -0.15$

Saída: $y_1 = -0.6 * 0.95 + 0.3 * (-0.15) = -0.525$

Função de Ativação: $\varphi(-0.525) = \begin{cases} 1 \ se \ o \ \geq 0 \\ 0 \ se \ o \ < 0 \end{cases} \rightarrow 0$

Erro: $e_1 = (0 - 0) = 0$

Classificou de forma correta, não atualiza os pesos



Terceira Iteração

Entradas: $x_1 = -0.1 \; e \; x_2 = -0.8 \rightarrow Classe = 0$

Pesos: $w_1 = 0.95 \; e \; w_2 = -0.15$

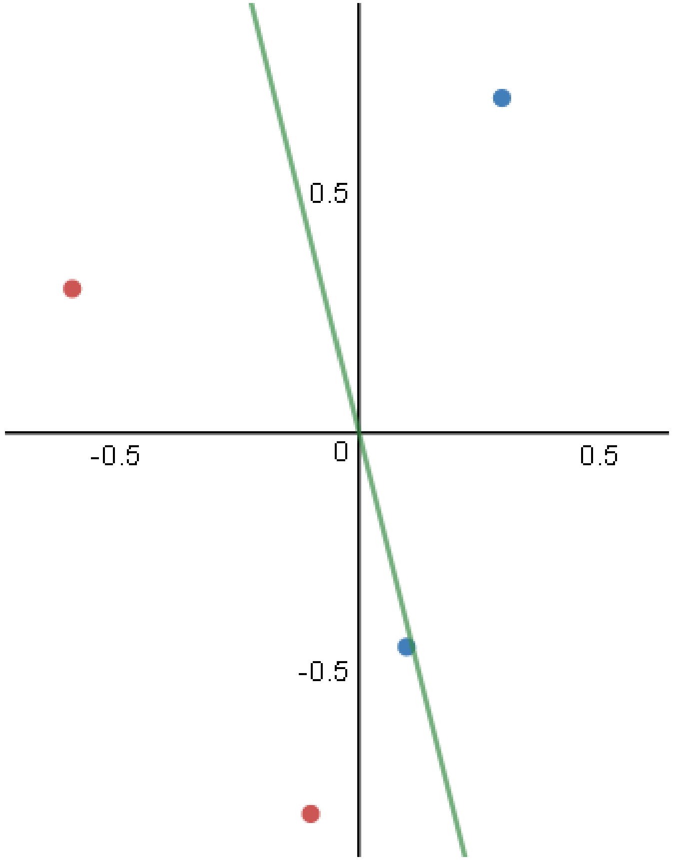
Saída: $y_1 = -0.1 * 0.95 + (-0.8) * (-0.15) = 0.025$

Função de Ativação: $\varphi(0.025) = \begin{cases} 1 \; se \; o \; \geq 0 \\ 0 \; se \; o \; < 0 \end{cases} \rightarrow 1$

Erro: $e_1 = (1 - 0) = 1$

Atualização dos Pesos: $w_1 = 0.95 + 0.5 *$1*(-0.1) = 1

$w_2 = -0.15 + 0.5 *$1*(-0.8) = 0.25



Quarta Iteração

Entradas: $x_1 = 0.1 \;\; e \;\; x_2 = -0.45 \rightarrow Classe = 1$

Pesos: $w_1 = 1 \;\; e \;\; w_2 = 0.25$

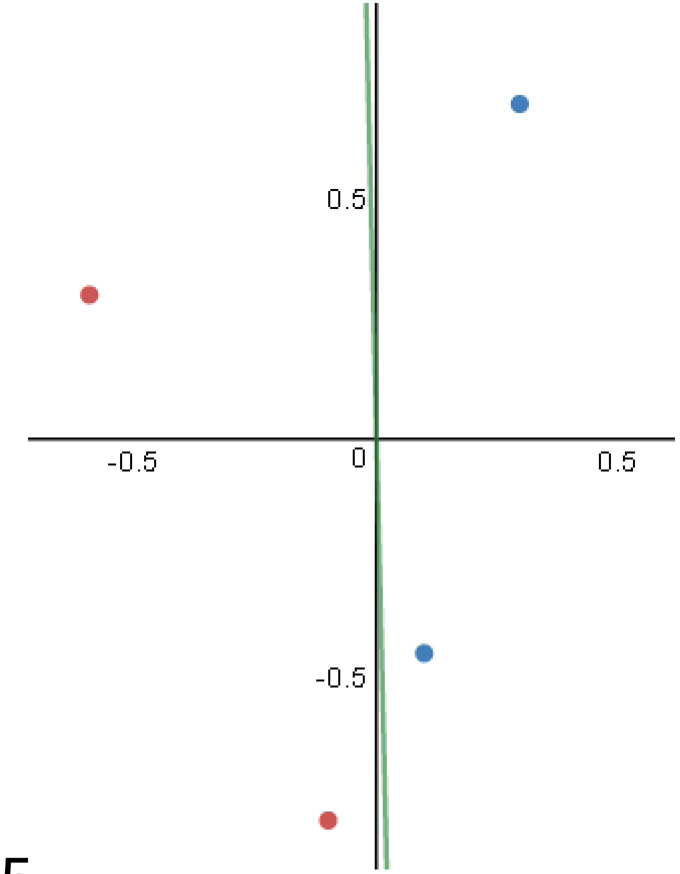
Saída: $y_1 = 0.1 * 1 + (-0.45) * 0.25 = -0.0125$

Função de Ativação: $\varphi(-0.11) = \begin{cases} 1 \; se \; o \; \geq 0 \\ 0 \; se \; o \; < 0 \end{cases} \rightarrow 0$

Erro: $e_1 = (1 - 0) = 1$

Atualização dos Pesos: $w_1 = 1 + 0.5 *1*(0.1) = 1.05$

$w_2 = 0.25 + 0.5 *1*(-0.45) = 0.025$



Quinta Iteração

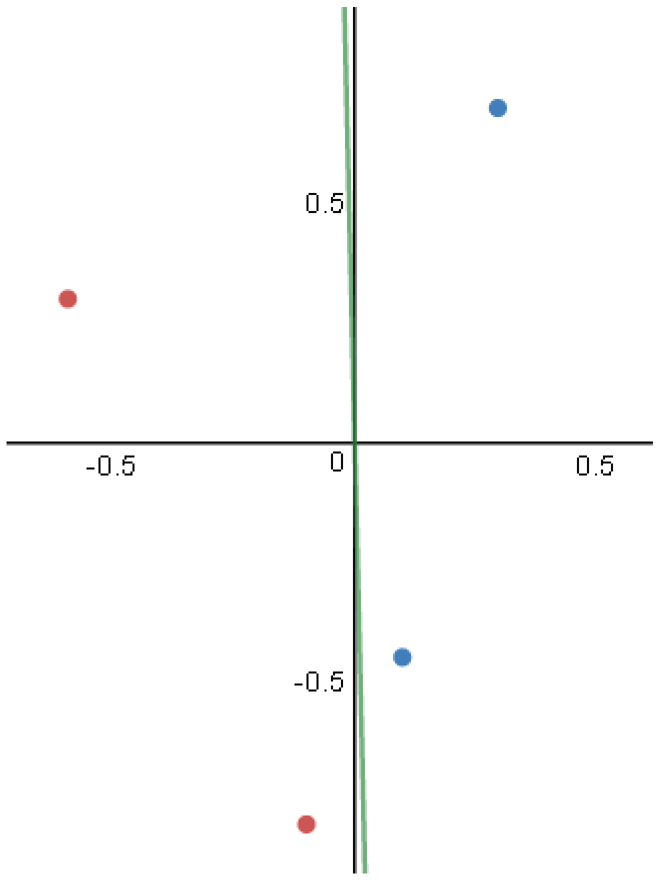
Entradas: $x_1 = 0.3\ e\ \ x_2 = 0.7 \rightarrow Classe = 1$

Pesos: $w_1 = 1.05\ \ e\ \ w_2 = 0.025$

Saída: $y_1 = 0.3 * 1.05 + 0.7 * 0.025 = 0.3325$

Função de Ativação: $\varphi(0.3325) = \begin{cases} 1\ se\ o\ \geq 0 \\ 0\ se\ o\ < 0 \end{cases} \rightarrow 1$

Erro: $e_1 = (1 - 1) = 0$



Classificou de forma correta, não atualiza os pesos

Sexta Iteração

Entradas: $x_1 = -0.6 \;\; e \;\; x_2 = 0.3 \;\; \rightarrow Classe = 0$

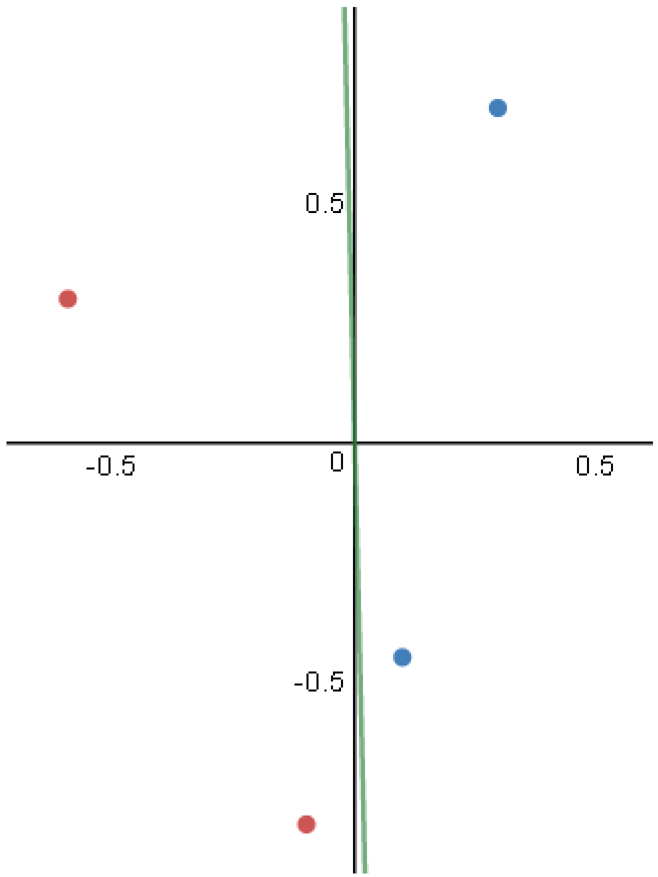
Pesos: $w_1 = 1.05 \;\; e \;\; w_2 = 0.025$

Saída: $y_1 = -0.6 * 1.05 + 0.3 * 0.025 = -0.6225$

Função de Ativação: $\varphi(-0.6225) = \begin{cases} 1 \; se \; o \; \geq 0 \\ 0 \; se \; o \; < 0 \end{cases} \rightarrow 0$

Erro: $e_1 = (0 - 0) = 0$

Classificou de forma correta, não atualiza os pesos



Sétima Iteração

Entradas: $x_1 = -0.1 \;\; e \;\; x_2 = -0.8 \rightarrow Classe = 0$

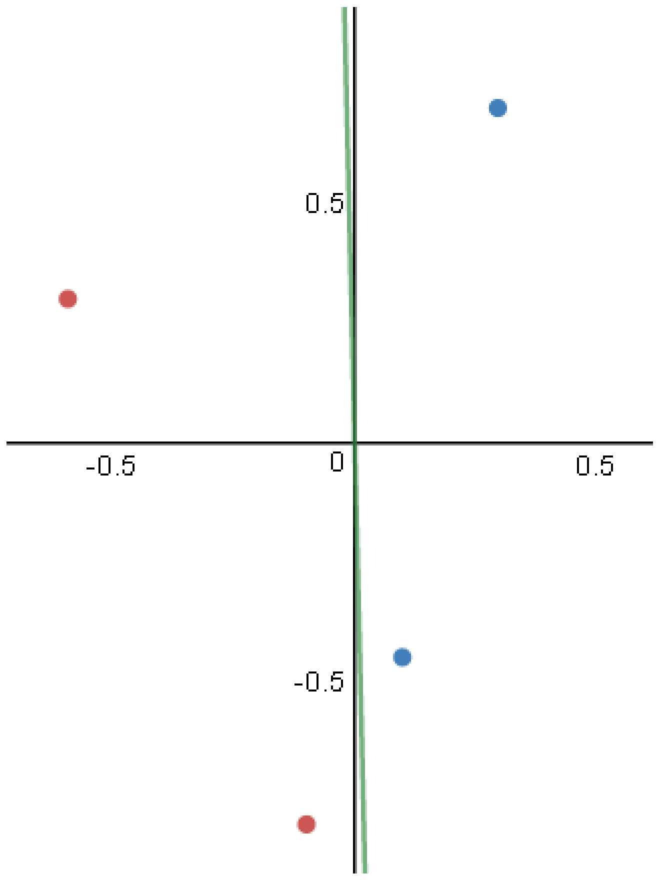
Pesos: $w_1 = 1.05 \;\; e \;\; w_2 = 0.025$

Saída: $y_1 = -0.1 * 1.05 + (-0.8) * 0.025 = -0.125$

Função de Ativação: $\varphi(-0.125) = \begin{cases} 1 \; se \; o \; \geq 0 \\ 0 \; se \; o \; < 0 \end{cases}$ → 0

Erro: $e_1 = (0 - 0) = 0$

Classificou de forma correta, não atualiza os pesos



Oitava Iteração

Entradas: $x_1 = 0.1 \;\; e \;\; x_2 = -0.45 \rightarrow Classe = 1$

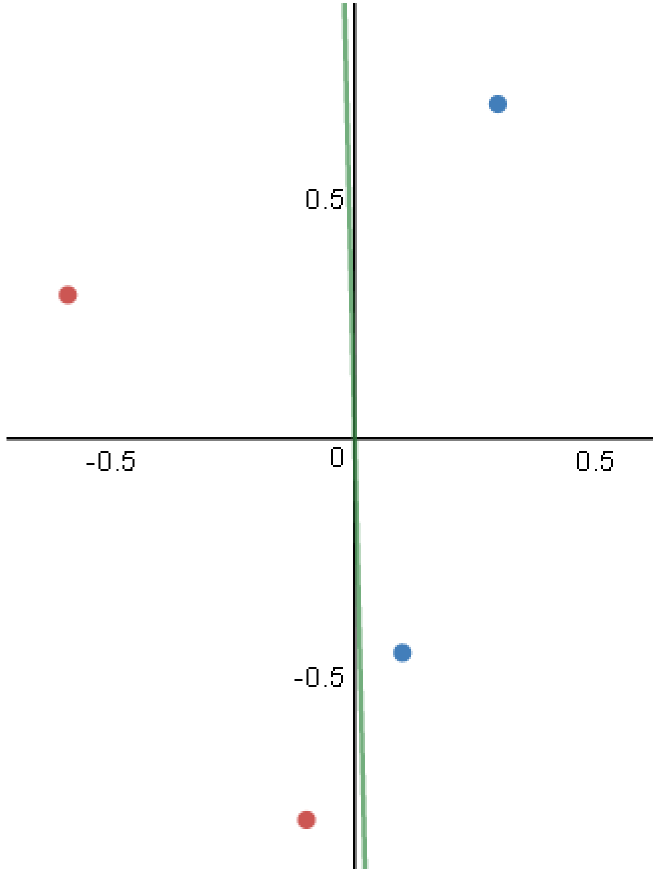
Pesos: $w_1 = 1.05 \;\; e \;\; w_2 = 0.025$

Saída: $y_1 = 0.1 * 1.05 + (-0.45) * 0.025 = 0.0937$

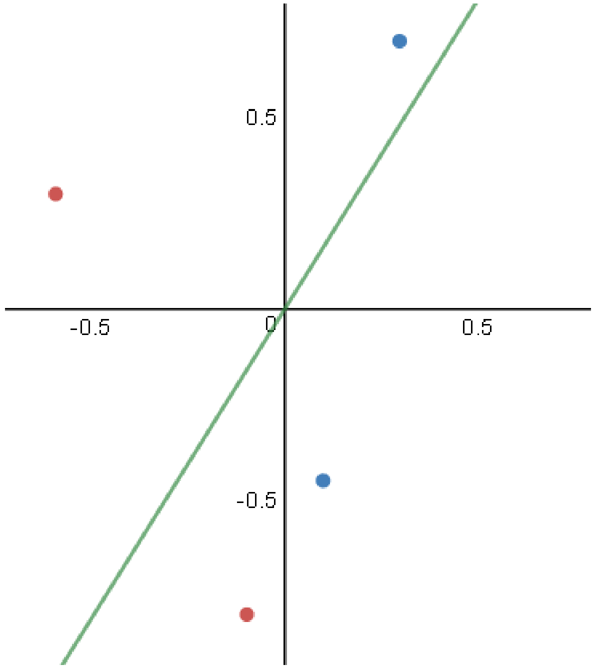
Função de Ativação: $\varphi(0.0937) = \begin{cases} 1 \; se \; o \;\; \geq 0 \\ 0 \; se \; o \;\; < 0 \end{cases} \rightarrow 1$

Erro: $e_1 = (1 - 1) = 0$

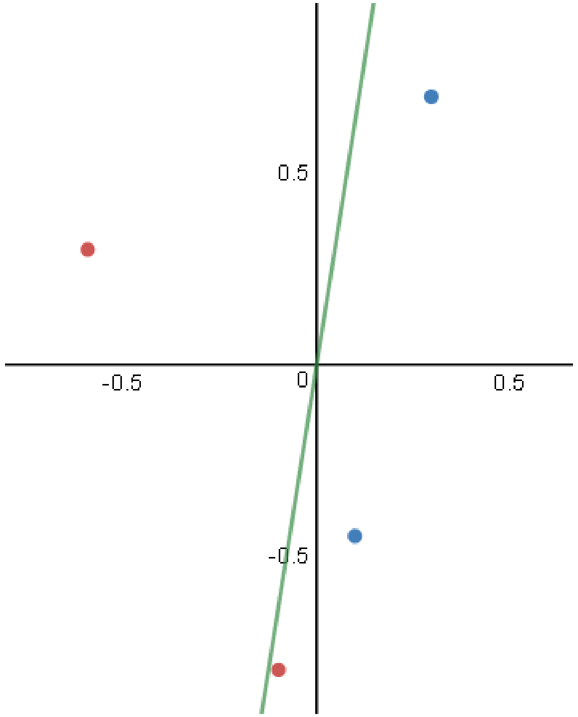
Classificou de forma correta, não atualiza os pesos



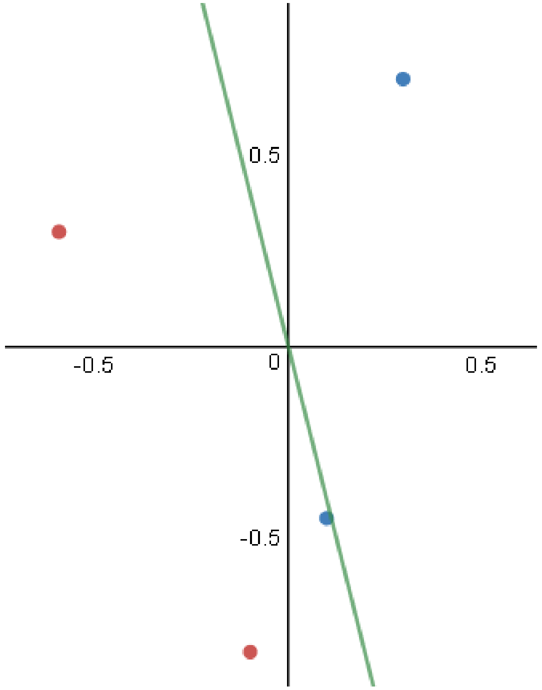
Gráfico



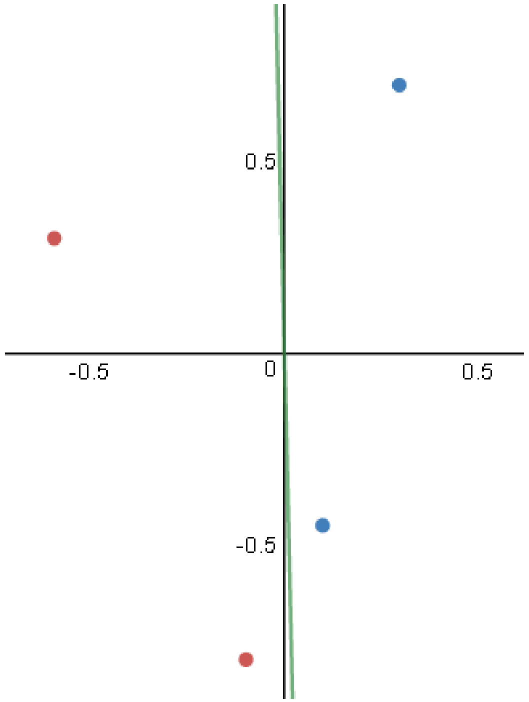
Início



1º. e 2º. Iteração



3º. Iteração



4º. Iteração

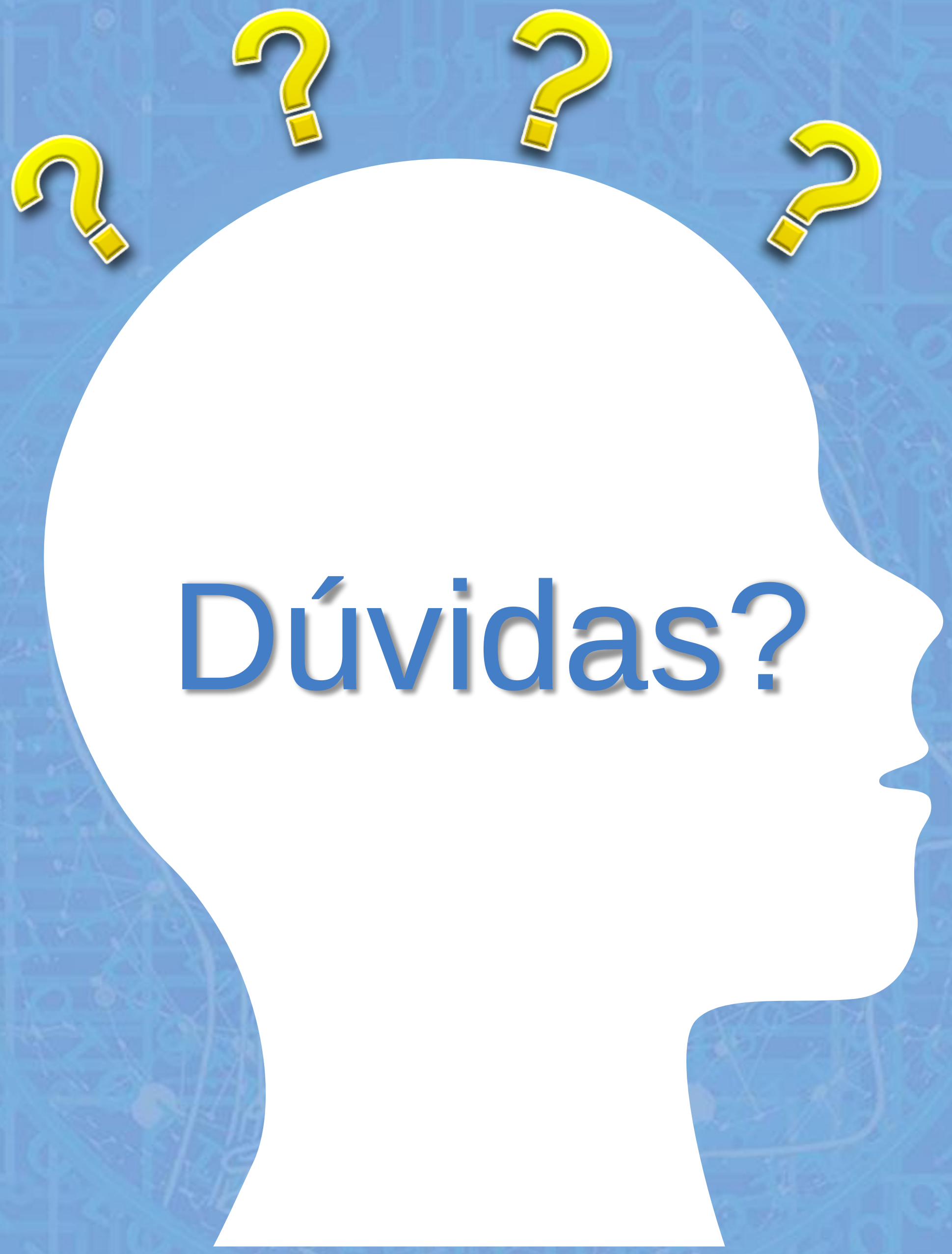
Considerações

- O perceptron só é capaz de separar classes através de funções lineares.
- As classes têm que ser linearmente separáveis, o que é uma condição não garantida para a maioria dos problemas de classificação
- Resolve o problema da porta AND e OR, por serem linearmente separáveis
- Não Resolve o problema da porta XOR, por não ser linearmente separável

Exercício

- Faça o treinamento da Rede Perceptron para o problema da porta lógica AND.

x	y	Classe
0	0	0
0	1	0
1	0	0
1	1	1


Dúvidas?

Referências Bibliográficas

- **José Demísio Simões da Silva – Notas de Aula**
- **Ana Paula A. C. Shiguemori – Notas de Aula**
- **Elcio Hideiti Shiguemori – Notas de Aula**