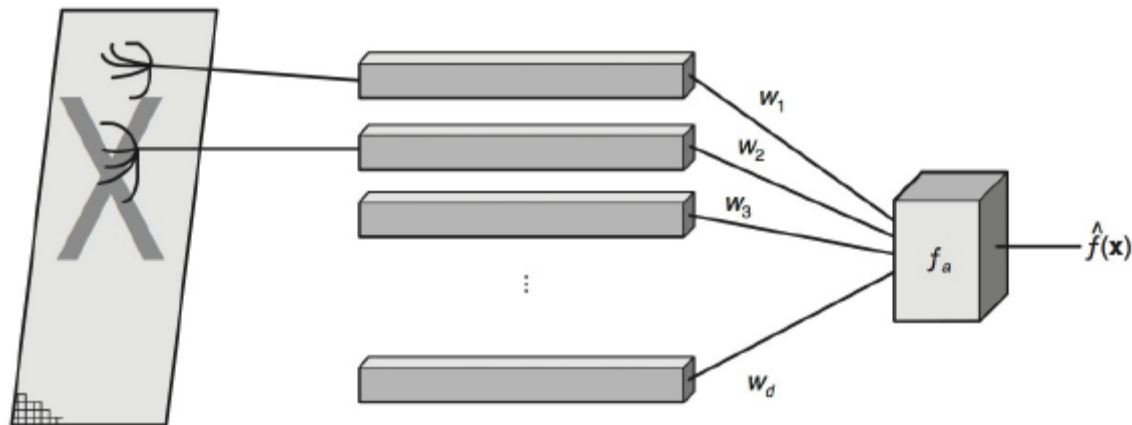


REDES NEURAIS PERCEPTRON

Cristiane Neri Nobre

PERCEPTRON

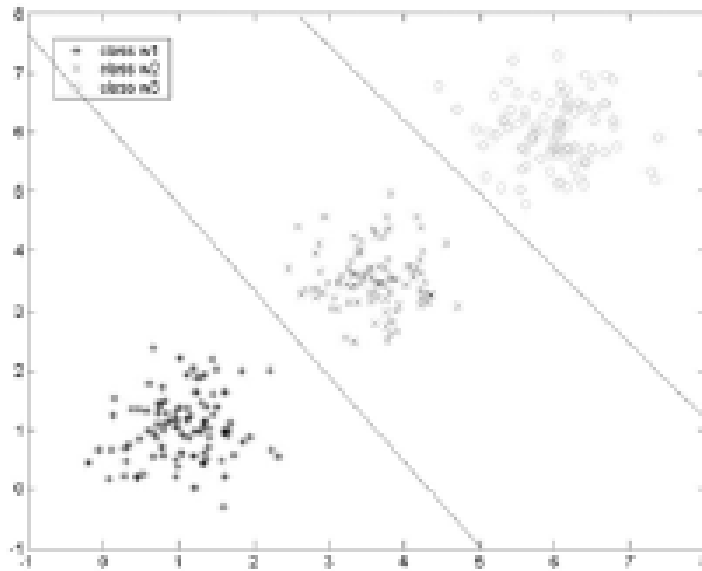
O Perceptron, proposto por Rosenblatt, é composto pelo neurônio de McCulloch-Pitts, com Função de Limiar, e Aprendizado Supervisionado. Sua arquitetura consiste na entrada e uma camada de saída.



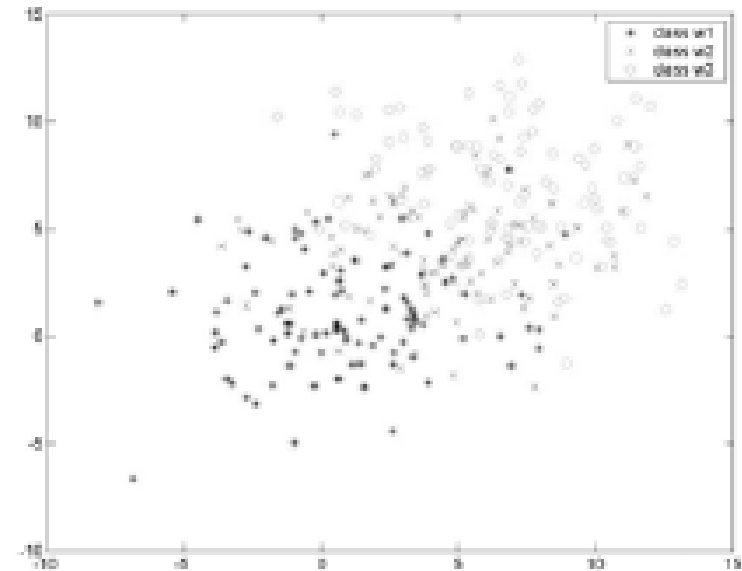
Perceptron - Características

- A superfície de decisão(curva de separação) forma um hiperplano, ou seja, para um dos lados está uma classe e para o outro lado está a outra classe.
- Podemos “ver” o perceptron como uma superfície de separação em um espaço *N-dimensional* de instâncias.
- Um único perceptron consegue separar somente *conjuntos de exemplo linearmente separáveis*.

Perceptron - Características



Linearmente Separável



Linearmente Não-Separável

Perceptron - Características

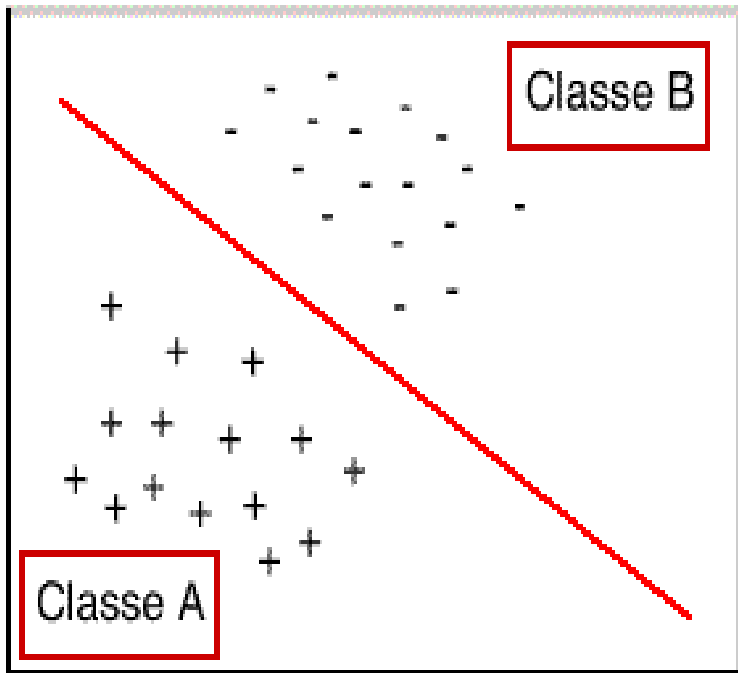


Fig. 1: Classes linearmente separáveis

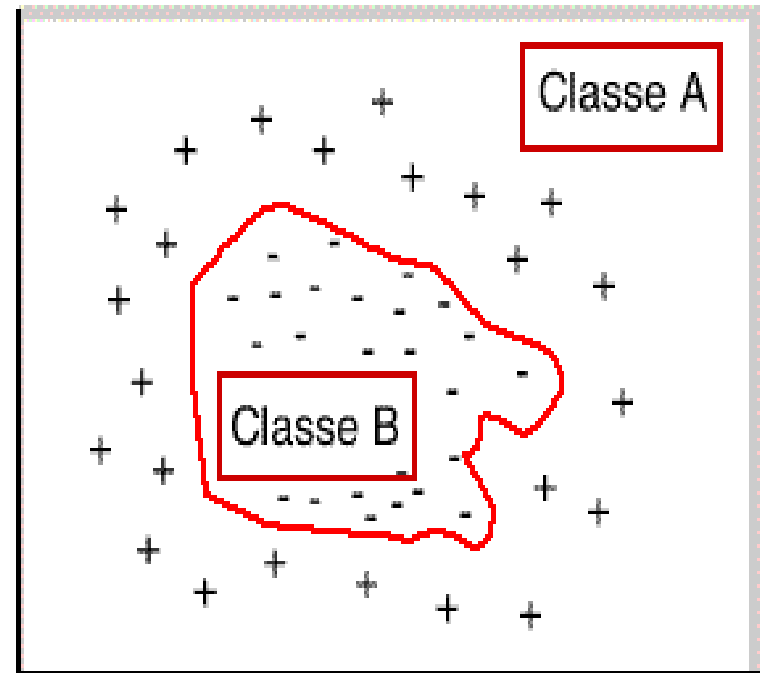


Fig. 2: Classes não linearmente separáveis

Algoritmo do Perceptron

Durante o processo de treinamento do Perceptron, busca-se encontrar um conjunto de pesos que defina uma reta que separe as diferentes classes, de forma que a Rede classifique corretamente as entradas apresentadas.

Algoritmo 7.1 Algoritmo de treinamento da rede perceptron

Entrada: Um conjunto de n objetos de treinamento

Saída: Rede perceptron com valores dos pesos ajustados

```
1 Inicializar pesos da rede com valores baixos
2 repita
3   para cada objeto  $\mathbf{x}_i$  do conjunto de treinamento faça
4     Calcular valor da saída produzida pelo neurônio,  $\hat{f}(\mathbf{x}_i)$ 
5     Calcular erro =  $y_i - \hat{f}(\mathbf{x}_i)$ 
6     se erro > 0 então
7       Ajustar pesos do neurônio utilizando Equação 7.2
8     fim
9   fim
10 até erro = 0;
```

$$w_j(t + 1) = w_j(t) + \eta x_i^j (y_i - \hat{f}(\mathbf{x}_i)) \quad (7.2)$$

Resumo do Algoritmo de Convergência do Perceptron

Variáveis e Parâmetros:

$x(n)$ = vetor de entrada $(m+1)$ -por-1
 $= [+1, x_1(n), x_2(n), \dots, x_m(n)]^T$

$w(n)$ = vetor de pesos $(m+1)$ -por-1
 $= [b(n), w_1(n), w_2(n), \dots, w_m(n)]^T$

$b(n)$ = bias;

A camada de entrada deve possuir uma unidade especial conhecida como bias, usada para aumentar os graus de liberdade, permitindo uma melhor adaptação, por parte da rede neural, ao conhecimento a ela fornecido.

$y(n)$ = resposta real (quantizada)

$d(n)$ = resposta desejada;

$e(n)$ = erro na saída da unidade;

η = parâmetro da taxa de aprendizagem, uma constante positiva entre 0 e 1.

O valor da taxa de aprendizado define a magnitude do ajuste feito no valor de cada peso. Valores altos fazem com que as variações sejam grandes, enquanto taxas pequenas implicam poucas variações nos pesos.

Esta magnitude vai definir a velocidade de convergência da rede.

Algoritmo do Perceptron

1 - *Inicialização*: Inicialize os pesos. Execute, então, os seguintes cálculos para os passos de tempo $n = 1, 2, \dots$

2. *Ativação*. No caso de tempo n , ative o perceptron aplicando o vetor de entrada de valores contínuos $\mathbf{x}(n)$ e a resposta desejada $d(n)$.

3. *Cálculo da Resposta Real*. Calcule a resposta real do perceptron:

$$y(n) = \text{função}[\mathbf{w}^T(n)\mathbf{x}(n)]$$

onde função(.) é a Função de Limiar.

4. *Adaptação do vetor de peso*. Atualize o vetor de peso do perceptron:

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

onde

$$d(n) = \begin{cases} +1 & \text{se } x(n) \text{ pertence à classe } \delta_1 \\ -1 & \text{se } x(n) \text{ pertence à classe } \delta_2 \end{cases}$$

5. *Continuação*. Incremento o passo de tempo n em um e volte para o passo 2.

Algoritmo do Perceptron

- Quando devemos parar o treinamento, ou seja, parar a atualização dos pesos?
 - Escolha óbvia: continuar o treinamento até que o erro E seja menor que o valor pré-estabelecido.
 - Porém, isto implica em sobreajuste(*overfitting*).
 - O sobreajuste diminui a generalização da rede neural.

Perceptron - Conclusões

- Se um conjunto de exemplos de treinamento E é **não-separável**, então por definição não existe um vetor de pesos W que classifique corretamente todos os exemplos de treinamento em E utilizando o algoritmo de aprendizagem do perceptron.

Referência

Veja capítulo 7 do livro texto adotado na disciplina

