

# Aula 3

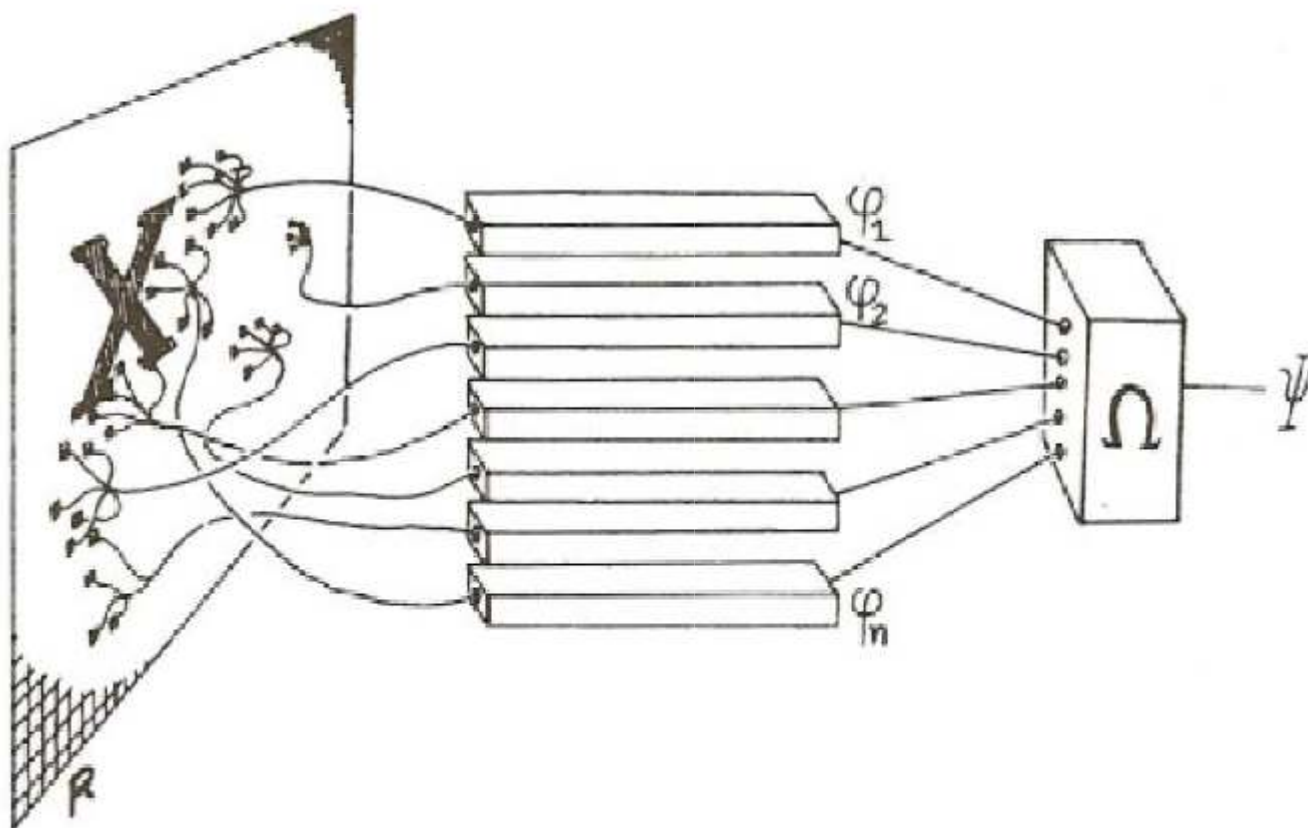
## RNA – Perceptron

## Sumário

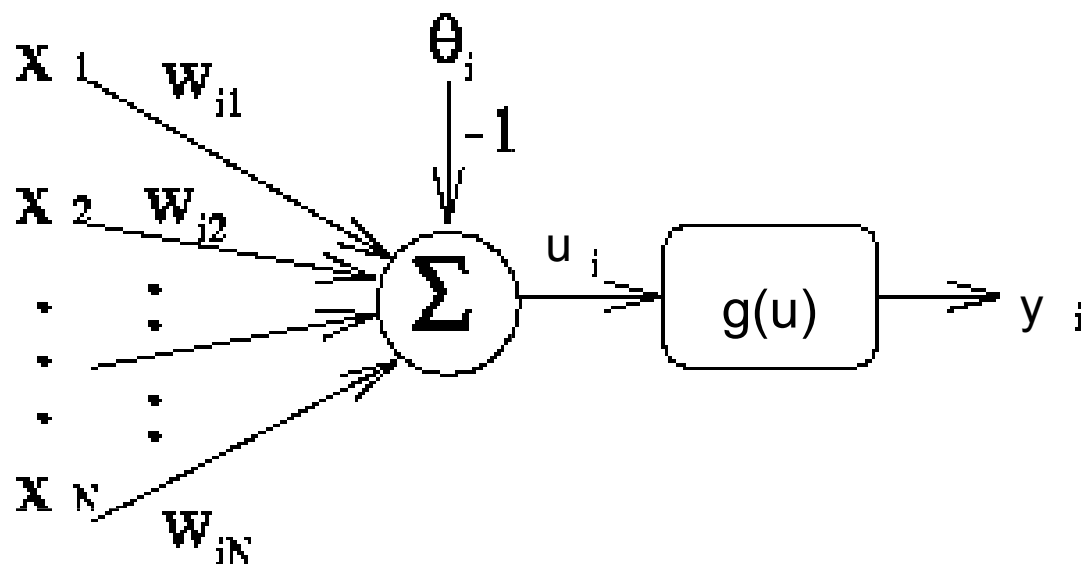
- 1- Introdução;
- 2- Algoritmo da Rede Perceptron;
- 3- Aspectos Práticos;
- 3- Projeto Prático.

## 1- Introdução

O *Perceptron* (1958), é a forma mais simples de configuração de uma rede neural artificial;



## 1- Introdução



### Características:

- $x$ : entradas
- $w$ : pesos sinápticos
- $n$ : quantidade de características
- $\theta$ : limiar de ativação
- $i$ : quantidade de neurônios
- $u$ : potencial de ativação
- $g(u)$ : função de ativação
- $y$ : saída

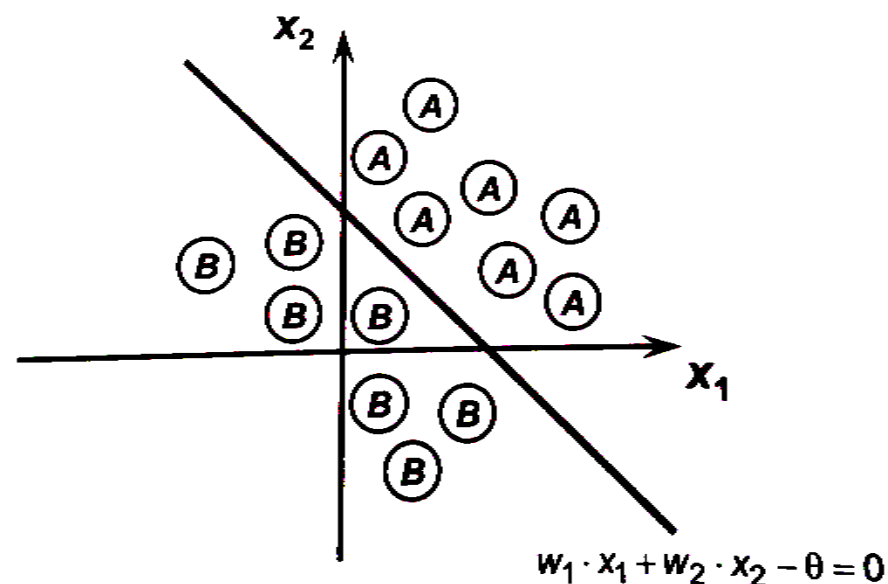
$$\begin{cases} u = \sum_{i=1}^n w_i x_i - \theta \\ y = g(u) \end{cases}$$

## 1- Introdução

$$y = \begin{cases} 1, & \text{se } \sum w_i x_i - \theta \geq 0 \\ -1, & \text{se } \sum w_i x_i - \theta < 0 \end{cases}$$

Características:

- X: entradas
- W: pesos sinápticos
- n: quantidade de características
- $\theta$ : limiar de ativação
- i: quantidade de neurônios
- u: potencial de ativação
- $g(u)$ : função de ativação
- y: saída

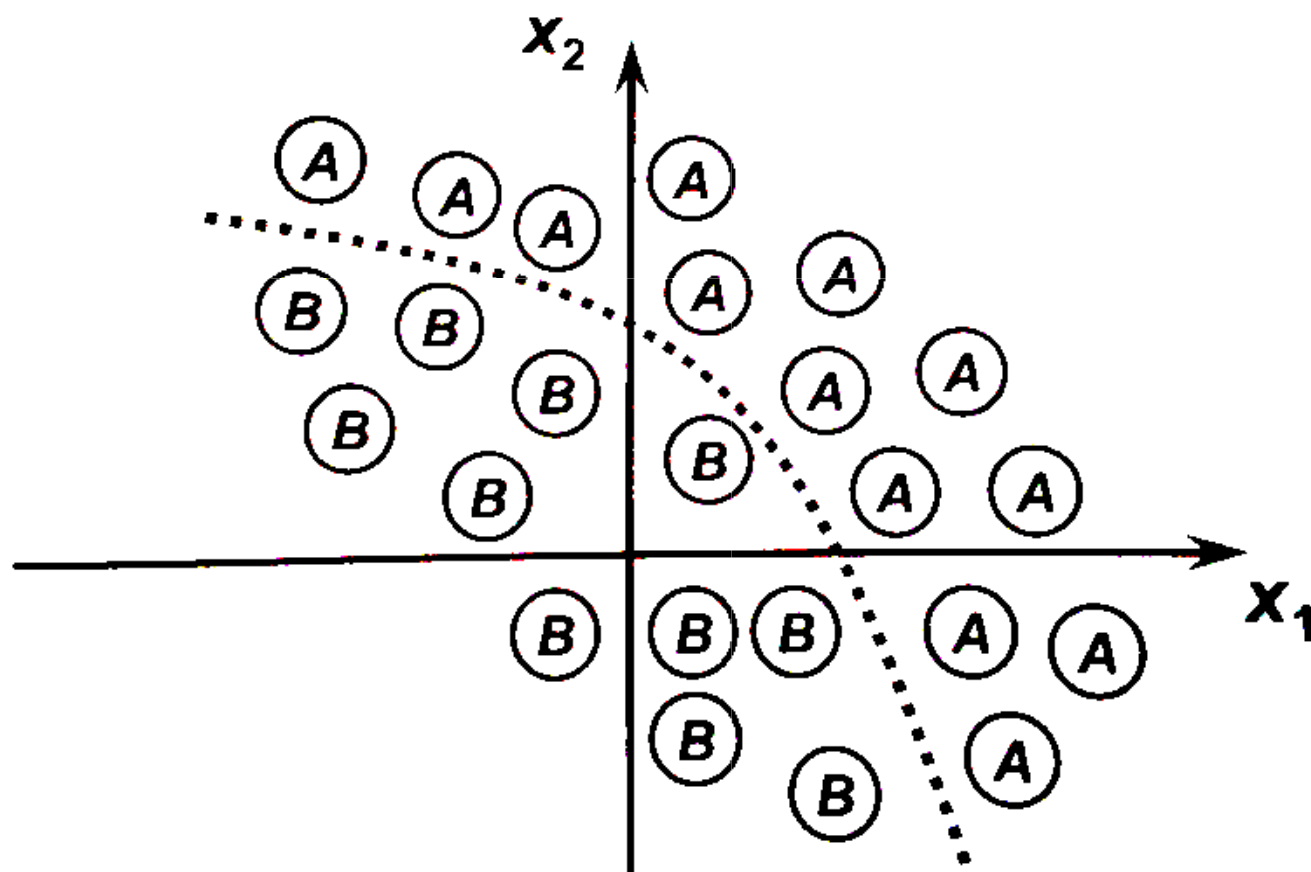


## 1- Introdução

- Um Perceptron constituído pode trêz entradas (três dimensões), a fronteira de separação seria um plano;
- Para quantidades de entradas superiores a trêz, as fronteiras seriam hiperplanos;
- Para que o Perceptron seja utilizado como classificador de padrões é necessário que a solução para o problema seja linear.
- O Perceptron não soluciona problemas que não são linearmente separáveis.

## 1- Introdução

O *Perceptron Single Layer* não soluciona este problema!



## 2- Algoritmo – Rede Perceptron

- Ajustes de pesos pela regra de aprendizado de Hebb;
- Caso a saída esteja coincidente com a saída desejada os pesos sinápticos e limiares da rede serão incrementados – ajuste excitatório;
- Caso a saída não seja coincidente, os pesos e limiares da rede serão decrementados – ajuste inibitório;
- Em termos matemáticos:

Peso sináptico

limiar

taxa de aprendizagem

valor desejado

$$\begin{cases} w_i^{atual} = w_i^{anterior} + \eta(d^{(k)} - y)x^{(k)} \\ \theta_i^{atual} = \theta_i^{anterior} + \eta(d^{(k)} - y)x^{(k)} \end{cases}$$



## 2- Algoritmo – Rede Perceptron

- A taxa de aprendizado ( $\eta$ ) define o quão rápido o processo de treinamento será conduzido;
- $0 < \eta < 1$
- Início (Algoritmos Perceptron – Treinamento)
  - (1) Obter o conjunto de treinamento;
  - (2) Associar a saída desejada  $\{d^k\}$ ;
  - (3) Iniciar  $\mathbf{w}$  com valores aleatórios pequenos;
  - (4) Especificar a taxa de aprendizado;
  - (5) Iniciar o contador de épocas com 0 e erro com 0;
  - (6) Repetir enquanto
    - (1) Erro = inexiste;
    - (2) Para todas as amostras de treinamento, fazer:
      - (1) Encontrar  $u$ ;
      - (2) Encontrar  $y$ ;
      - (3) Se  $y \neq d$ ;
        - (1) Então ajusta  $w$  e Erro = existe;
    - (3) Época = Época + 1;
    - (4) Até que erro = inexiste

## 2- Algoritmo – Rede Perceptron

- Início (Algoritmos Perceptron – Operação)
  - (1) Obter amostra a ser classificada  $\{x\}$
  - (2) Utilizar  $w$  ajustado durante fase de treinamento;
  - (3) Executar operação
    - (1) Obter  $u$ ;
    - (2) Obter  $y$ ;
    - (3) Se  $y = -1$ 
      - (1) Então amostra  $x \in \{\text{Classe A}\}$
    - (4) Se  $y = 1$ 
      - (1) Então amostra  $x \in \{\text{Classe B}\}$

FIM

### 3- Aspectos Práticos

- A rede divergir se o problema for não linearmente separável. O que se pode fazer é colocar um limite para as épocas e chegar a um resultado aproximado;
- Quando a faixa de separabilidade entre as duas classes for muito estreita, o processo de treinamento pode ser instável. Nesses casos deve-se assumir uma taxa de aprendizado pequena;
- A quantidade de épocas necessárias para a convergência varia em função dos pesos e da taxa de aprendizado;
- Quanto mais próxima a superfície de decisão estiver da fronteira de separabilidade, menos épocas são necessárias.
- A normalização das entradas acelera o processo de treinamento;

## Referências:

Silva, IN da, Danilo Hernane Spatti, and Rogério Andrade Flauzino. "Redes neurais artificiais para engenharia e ciências aplicadas." São Paulo: Artliber (2010).