

2. Redes Neurais Artificiais

Prof. Renato Tinós

Depto. de Computação e Matemática (FFCLRP/USP)

2.2. Perceptron

2.2.1. Introdução

2.2.2. Funcionamento do perceptron

2.2.3. Aprendizado

2.2.4. Teorema da convergência

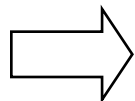
2.2.5. Aspectos do problema de reconhecimento de padrões

2.2.1. Introdução

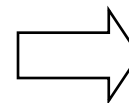
- **Classificador**

- Recebe vetores de **atributos** como entrada (exemplos)
- Atribui cada vetor a uma das **classes** $C_1, C_2, \dots, C_q$

Entrada: vetor de atributos

$$\begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_m \end{bmatrix}$$


Classificador



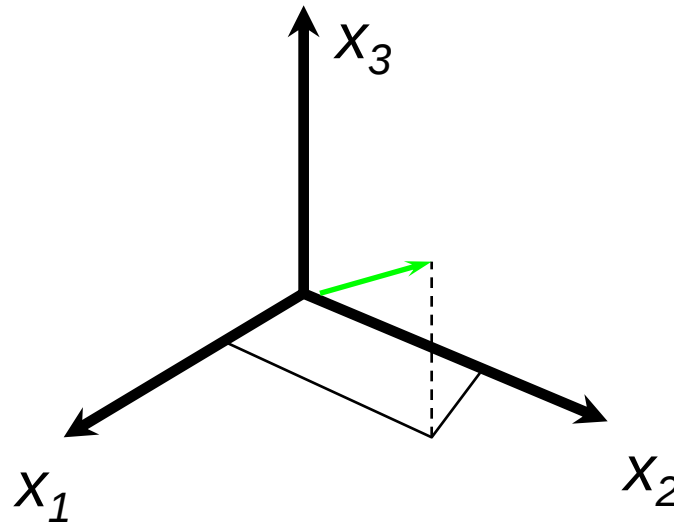
Saída: classificação

$$\begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_q \end{bmatrix}$$

2.2.1. Introdução

- **Exemplo: vetor entrada com três atributos**

Espaço de entradas ou de atributos



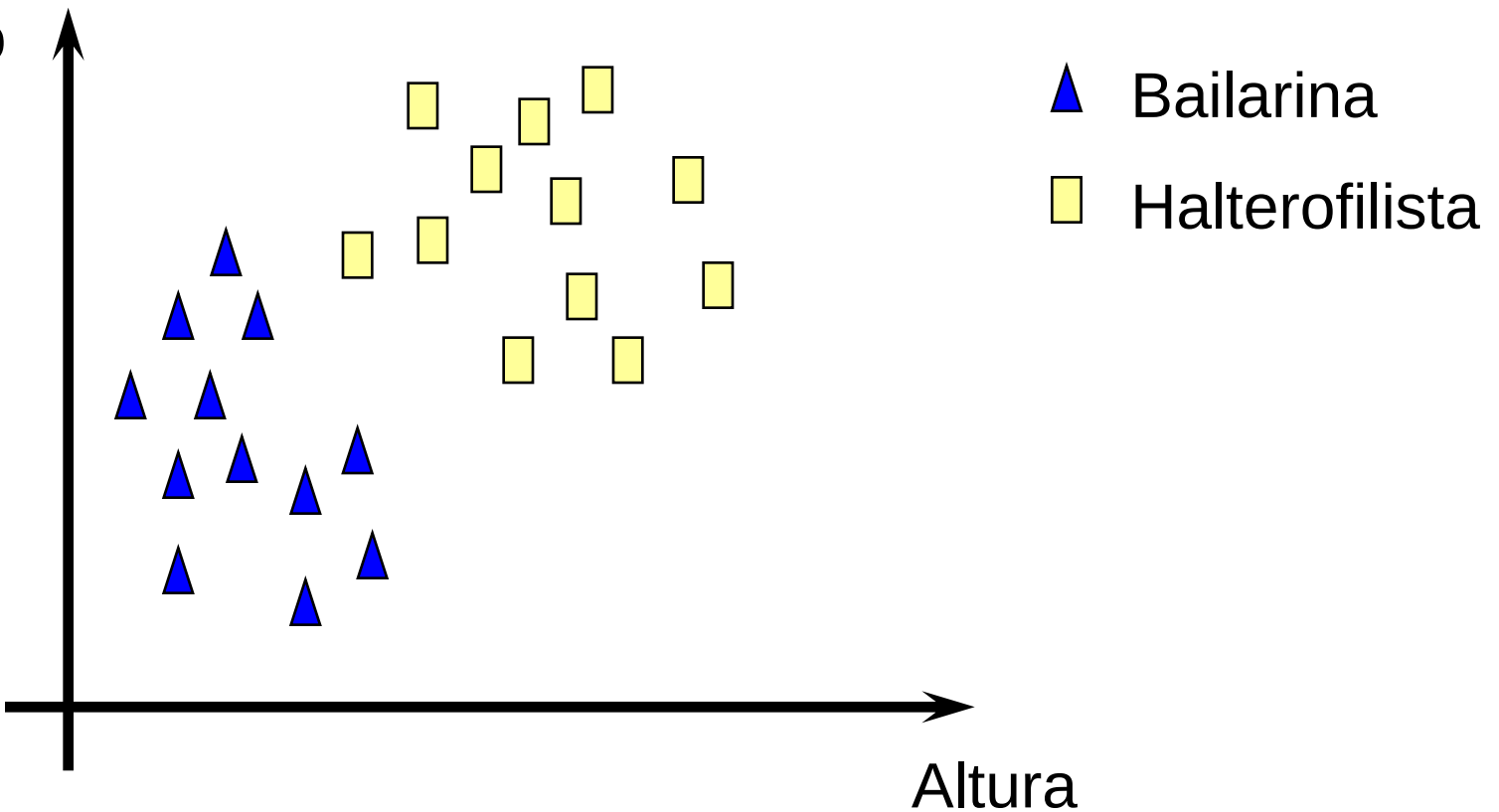
Exemplo ou padrão pode ser representado, abstratamente, como um ponto no espaço de atributos

2.2.1. Introdução

- **Vetor de atributos (exemplos)**
 - Composto por vários atributos
 - » Discretos
 - » Contínuos
 - Dimensão do vetor = número de atributos

2.2.1. Introdução

- **Exemplo: classificar entre bailarinas e halterofilistas**
 - Atributos:
 - » peso
 - » altura
 - Padrões (ou exemplos):
 - » pontos no espaço (peso, altura)

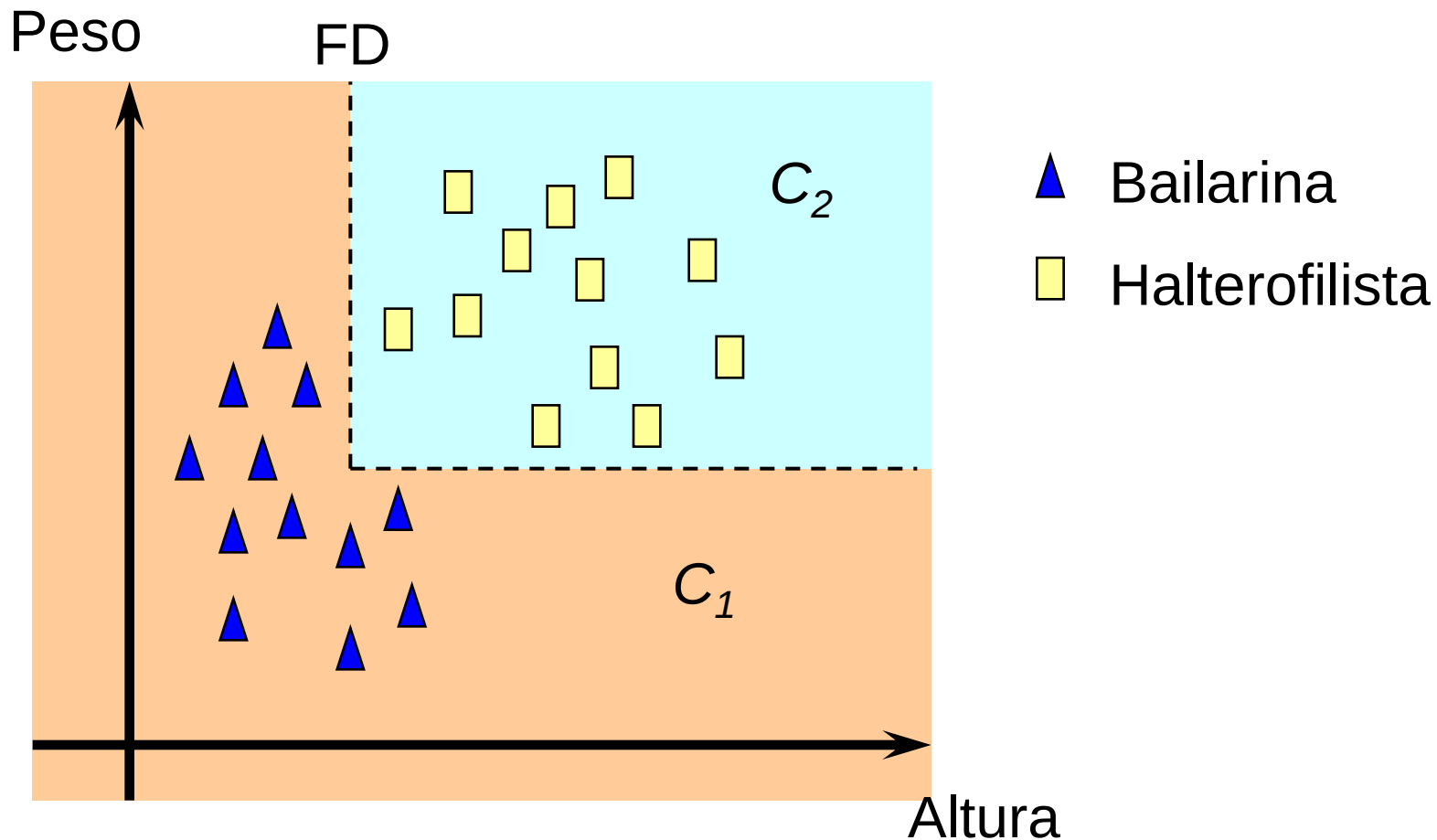


2.2.1. Introdução

- Em geral os classificadores de padrões particionam o espaço de entradas em volumes, chamados **regiões de decisão**
 - Todos os vetores de atributos dentro de uma mesma região são atribuídos a uma mesma categoria (classe)
 - Regiões são separadas por superfícies chamadas **fronteiras de decisão (FD)**
 - » Representam pontos onde não é possível definir a classe de um padrão

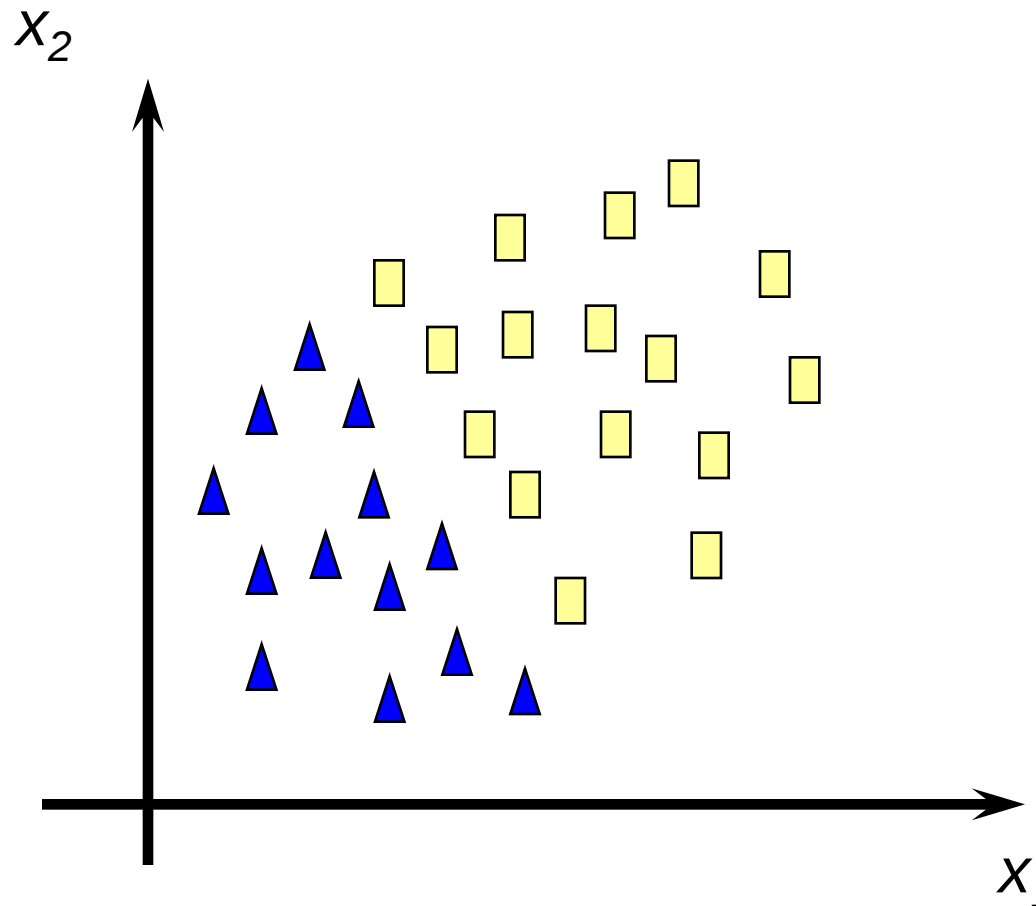
2.2.1. Introdução

- Exemplo da FD produzida por um classificador



2.2.1. Introdução

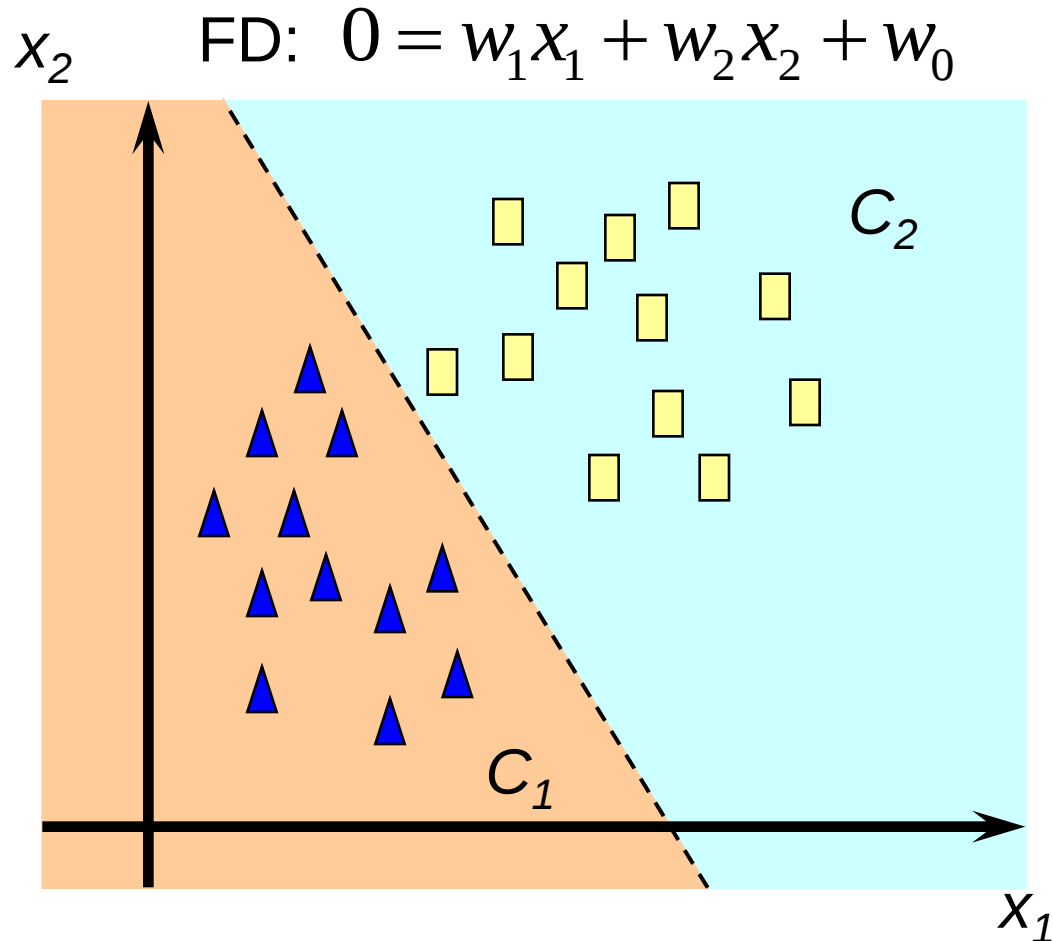
- Agora, imagine o seguinte problema de classificação



2.2.1. Introdução

- **Seria mais intuitivo e simples a utilização de um discriminante linear**
 - Utiliza uma combinação linear das entradas para produzir a FD
 - Transformação linear de um problema multidimensional em um problema unidimensional

2.2.1. Introdução



2.2.1. Introdução

- **Discriminante linear**

$$u = w_1x_1 + w_2x_2 + \dots + w_mx_m + w_0 = \sum_{i=1}^m w_ix_i + w_0$$

Na qual w_i é o peso (ponderação) da i -ésima entrada x_i
e w_0 é o bias

$$\begin{cases} u > 0 & , \text{vetor de entradas é relacionado à classe } C_1 \\ u \leq 0 & , \text{vetor de entradas é relacionado à classe } C_2 \end{cases}$$

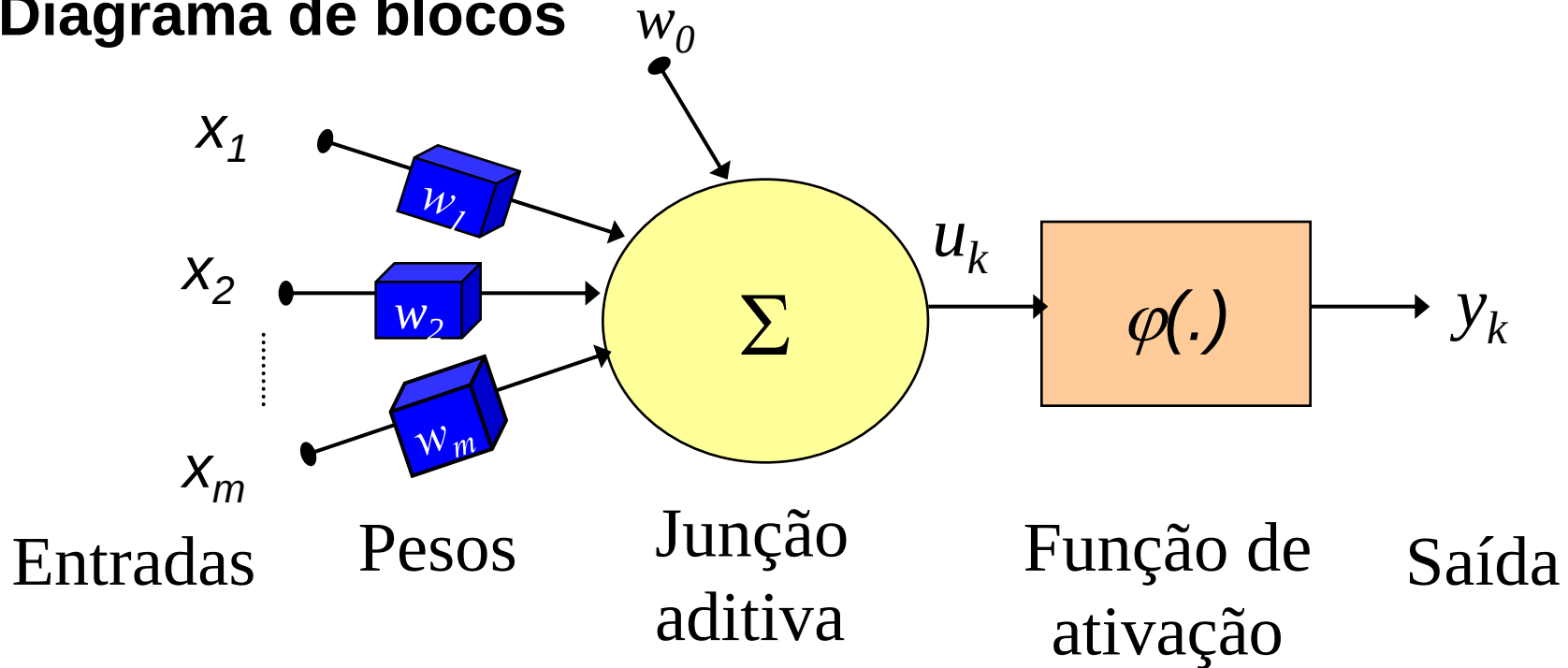
2.2.1. Introdução

- **Perceptron**

- Discriminante linear
- Mantém uma certa relação com o classificador Bayesiano
 - » Quando o ambiente é gaussiano, o classificador Bayesiano se reduz a um classificador linear
- Desenvolvido por Rosembat, 1958
 - » Problema de reconhecimento de padrões visuais
- Utiliza modelo de McCulloch-Pitts como modelo de neurônio
- Similar ao ADALINE (*ADaptive LINear Element*)

2.2.2. Funcionamento do Perceptron

- Diagrama de blocos

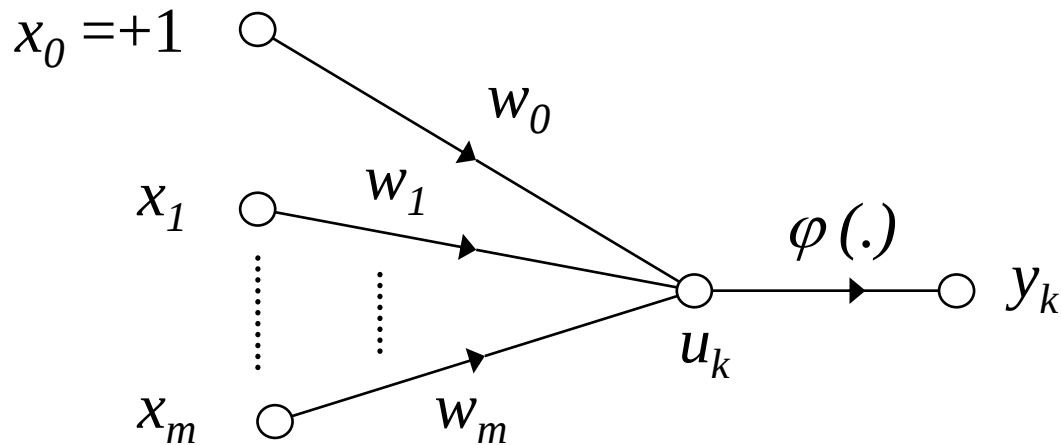


y_k : ativação (saída do neurônio k)
 u_k : ativação interna do neurônio k
 w_0 : bias

$$y_k = \varphi(u_k) = \varphi\left(\sum_{i=1}^m x_i w_i + w_0\right) \quad (2.2.1)$$

2.2.2. Funcionamento do Perceptron

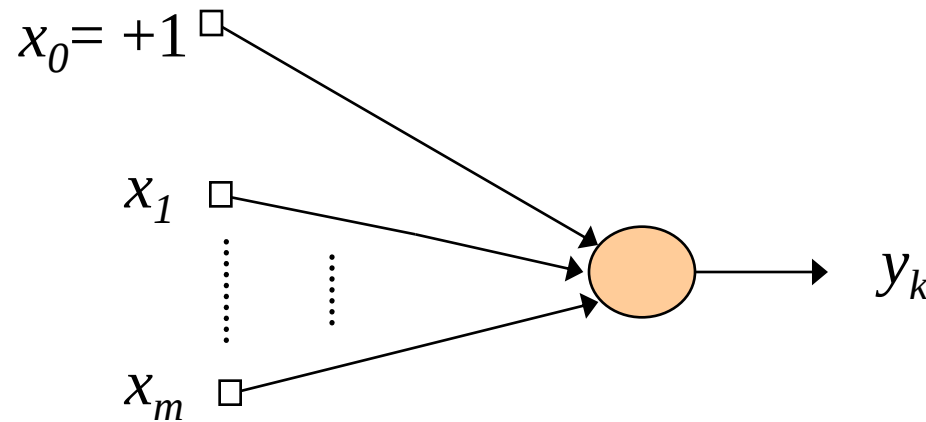
- **Perceptron visto com um grafo orientado completo**



- **Regra 1:** um sinal flui ao longo de um elo somente no sentido definido pela seta do elo
- **Regra 2:** um sinal nodal é igual à soma algébrica de todos os sinais que entram no nó via os elos incidentes
- **Regra 3:** O sinal em um nó é transmitido para cada elo de saída deste nó, sendo a transmissão inteiramente independente das funções de transferência dos elos de saída

2.2.2. Funcionamento do Perceptron

- **Perceptron visto com um grafo orientado parcialmente completo (ou arquitetural)**



- **Nós de fonte** fornecem sinais de entrada para o grafo
- Cada **neurônio** é representado por um único nó chamado de **nó computacional**
- Os elos de comunicação que conectam os nós de fonte aos nós computacionais não carregam pesos; eles meramente fornecem direções de fluxo de sinal no grafo

2.2.2. Funcionamento do Perceptron

- Como foi visto, a saída do k -ésimo *Perceptron* é dada por

$$y_k = \varphi(u_k) = \varphi\left(\sum_{i=1}^m x_i w_i + w_0\right) \quad (2.2.1)$$

Se fizermos

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_m \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_m \end{bmatrix}$$

2.2.2. Funcionamento do Perceptron

Podemos escrever a ativação interna do neurônio k como

$$u_k = \mathbf{w}^T \mathbf{x} + w_0 \quad (2.2.2)$$

e a sua saída como

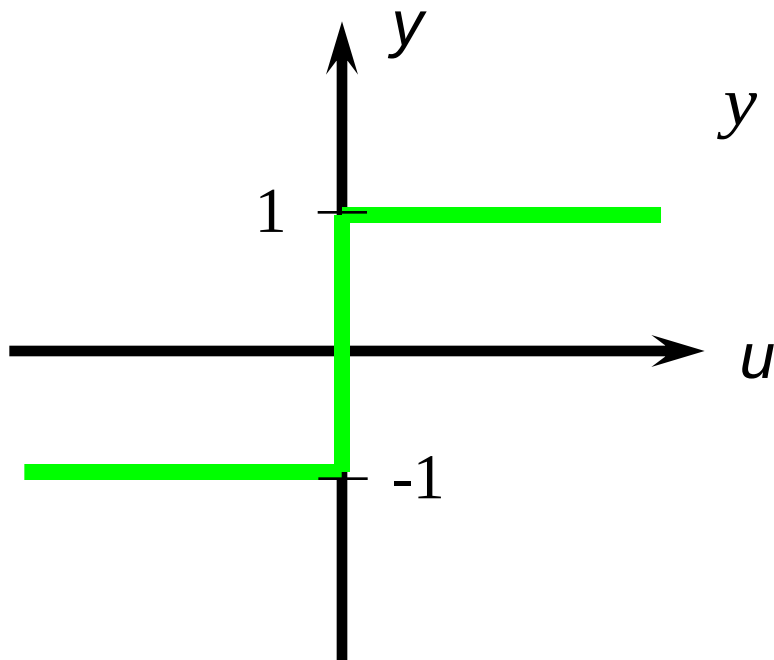
$$y_k = \varphi(u_k) = \varphi(\mathbf{w}^T \mathbf{x} + w_0) \quad (2.2.3)$$

na qual $\mathbf{x}$ é o vetor de entradas, $\mathbf{w}$ é o vetor de pesos (indica as forças da conexão das sinapses), w_0 é o bias e $\varphi(.)$ é a função de ativação do neurônio.

2.2.2. Funcionamento do Perceptron

- Costuma-se usar a função sinal como função de ativação no *perceptron*

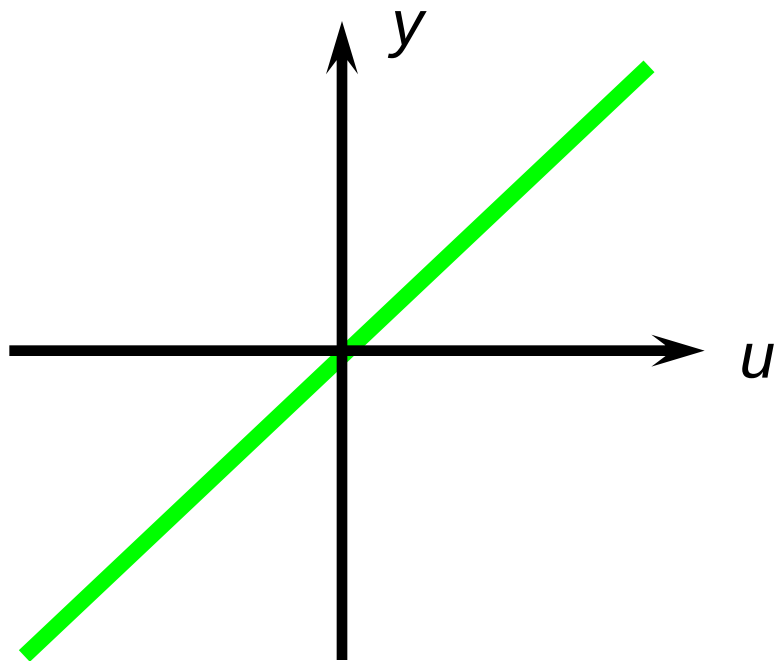
- Função sinal:



$$y = \varphi(u) = \begin{cases} +1 & \text{se } u > 0 \\ -1 & \text{se } u \leq 0 \end{cases} \quad (2.2.4)$$

2.2.2. Funcionamento do Perceptron

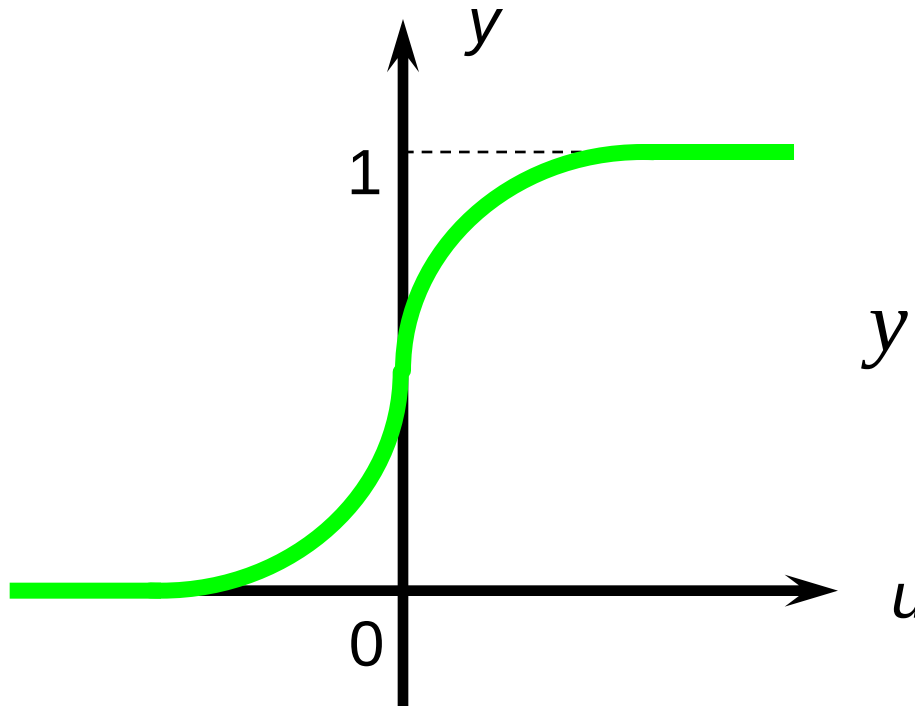
- No entanto, existem outras funções de ativação em RNAs
 - Função linear



$$y = \varphi(u) = a u \quad (2.2.5)$$

2.2.2. Funcionamento do Perceptron

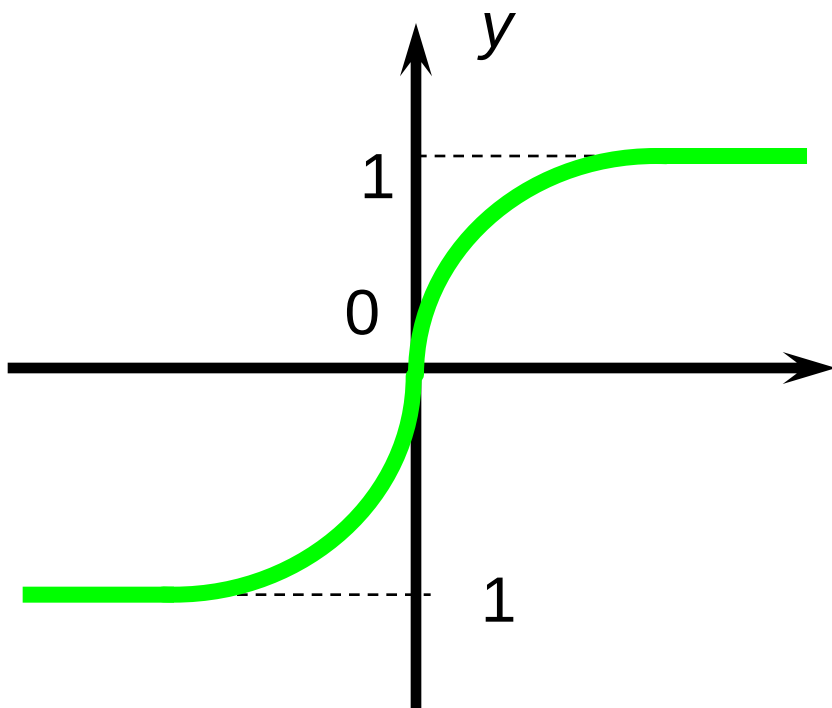
- Função logística (sigmoidal)



$$y = \varphi(u) = \frac{1}{1 + e^{-au}} \quad (2.2.6)$$

2.2.2. Funcionamento do Perceptron

- Função tangente hiperbólica



$$y = \varphi(u) = \frac{1 - e^{-au}}{1 + e^{-au}}$$

(2.2.7)

2.2.2. Funcionamento do Perceptron

- Utilizando-se a função sinal como função de ativação, a regra de decisão para a classificação de um ponto $\mathbf{x}$ é dada por

$$y = \varphi(u) = \begin{cases} +1 & \text{se } u = \mathbf{w}^T \mathbf{x} + w_0 > 0 \\ -1 & \text{se } u = \mathbf{w}^T \mathbf{x} + w_0 \leq 0 \end{cases} \quad (2.2.8)$$

$$\begin{cases} y = 1 & , \mathbf{x} \text{ é relacionado à classe } C_1 \\ y = -1 & , \mathbf{x} \text{ é relacionado à classe } C_2 \end{cases} \quad (2.2.9)$$

2.2.2. Funcionamento do Perceptron

- **Exemplo 2.2.1. função AND**

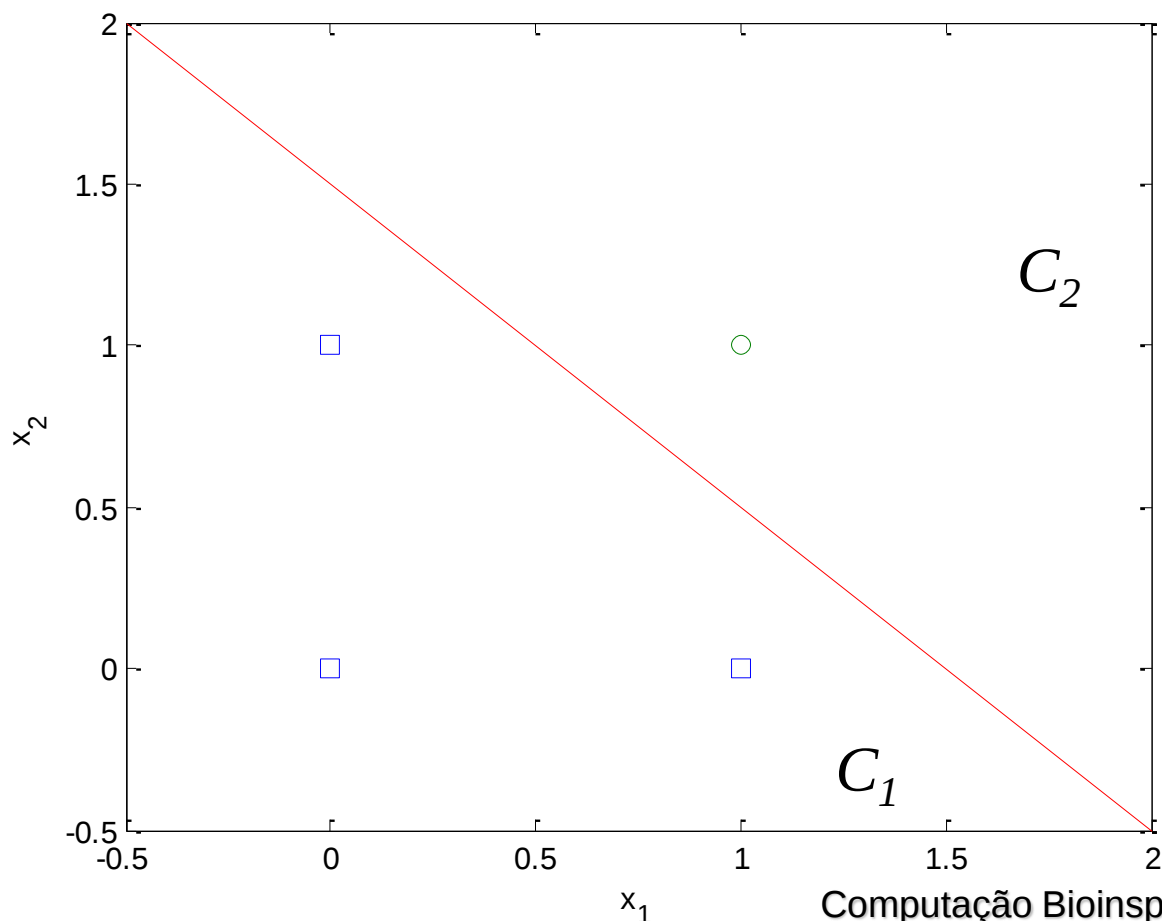
x_1	x_2	Classe
0	0	C_1
0	1	C_1
1	0	C_1
1	1	C_2

- Deve-se classificar os padrões de entrada em duas classes (C_1 e C_2)
- Cada padrão de entrada tem dois componentes (x_1 e x_2)
- Considere os pesos
 - » $w_1=1$, $w_2=1$, $w_0 = -1,5$

2.2.2. Funcionamento do Perceptron

- **Exemplo 2.2.1. função AND**

$$\text{FD: } u = 0 = w_1x_1 + w_2x_2 + w_0 = 1x_1 + 1x_2 - 1,5$$



2.2.2. Funcionamento do Perceptron

- Em um perceptron, a FD é dada por um **hiperplano** definido por

$$\mathbf{w}^T \mathbf{x} + w_0 = 0 \quad (2.2.10)$$

- O hiperplano é
 - » uma reta no caso bidimensional (duas variáveis de entrada)
 - » um plano no caso tridimensional
- Os pesos e o bias do *perceptron* definem a localização do hiperplano no espaço das entradas

2.2.2. Funcionamento do Perceptron

- **Localização do hiperplano**

- Se $\mathbf{x}(1)$ e $\mathbf{x}(2)$ são dois pontos no hiperplano, então

$$u = 0 = \mathbf{w}^T \mathbf{x}(1) + w_0 = \mathbf{w}^T \mathbf{x}(2) + w_0$$

$$\mathbf{w}^T (\mathbf{x}(1) - \mathbf{x}(2)) = 0$$

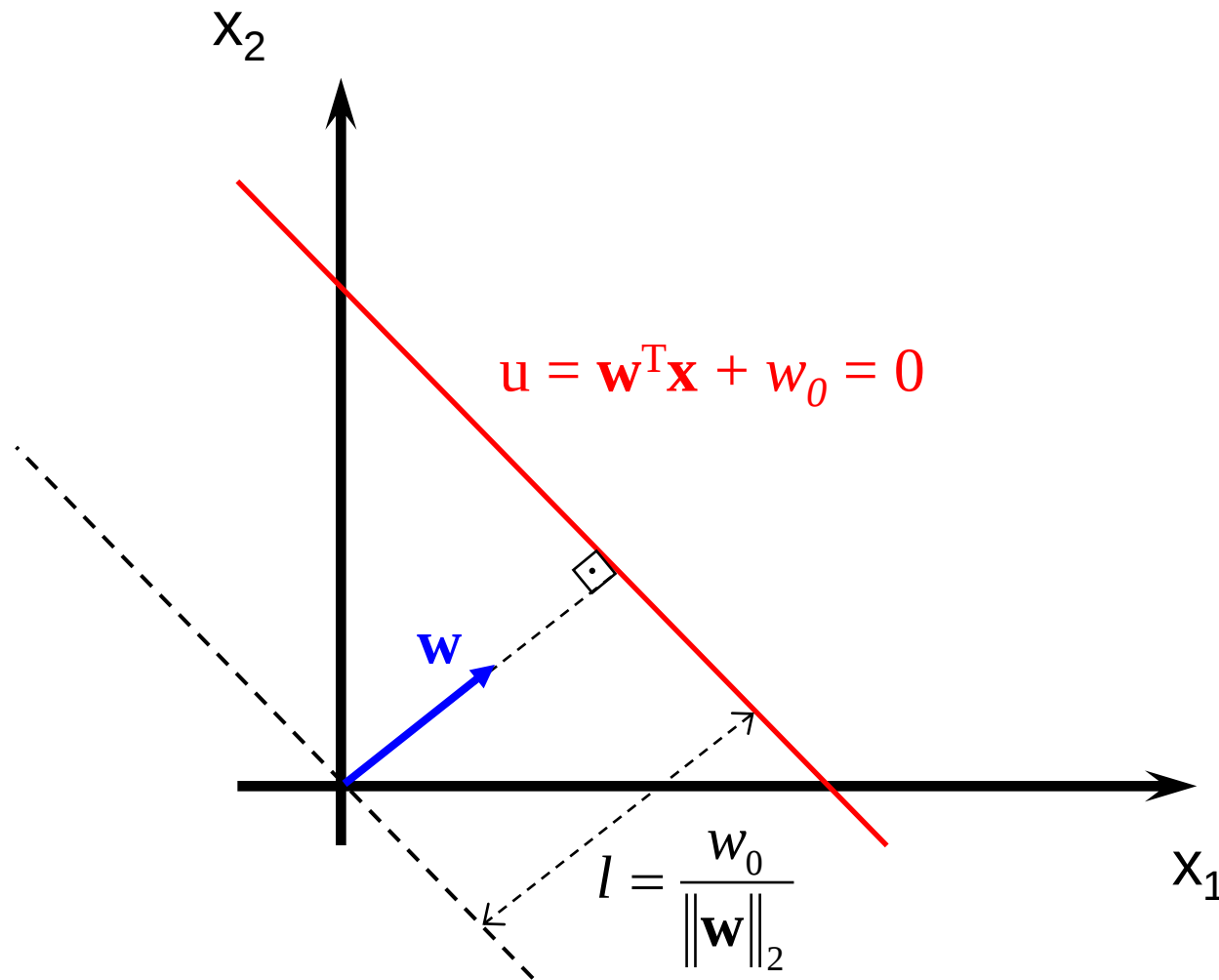
Assim, podemos concluir que $\mathbf{w}$ é normal a qualquer vetor contido no hiperplano

- Além disso, a distância normal da origem até o hiperplano é dada por

$$l = \frac{\mathbf{w}^T \mathbf{x}}{\|\mathbf{w}\|_2} = \frac{w_0}{\|\mathbf{w}\|_2} \quad (2.2.11)$$

2.2.2. Funcionamento do Perceptron

- Exemplo:



- **Referências**

- Haykin, S. S.. *Redes neurais: princípios e prática*. 2ª ed., Bookman, 2001.
 - » Capítulo 3
- Braga, A.P.; Carvalho, C.P.L.F. & Ludermir, T.B.. *Redes neurais artificiais: Teoria e Aplicações*. LTC, 2000.
 - Capítulo 3