Peceptron Simples

Prof. Dr. Ajalmar Rocha

Disciplina: Inteligência Computacional Aplicada (ICA)
Programa de Pós-Graduação em Eng. de Telecomunicações (PPGET)
Instituto Federal do Ceará (IFCE)

Agosto/2014

Resumo

- Introdução
- Neurônio de MucCulloch and Pitts
- Neurônio Biológico
- Perceptron Simples;

Introdução

- Tudo começou em 1943 quando foi proposto o modelo matemático para um neurônio biológico no trabalho de

McCulloch and Pitts (1943)

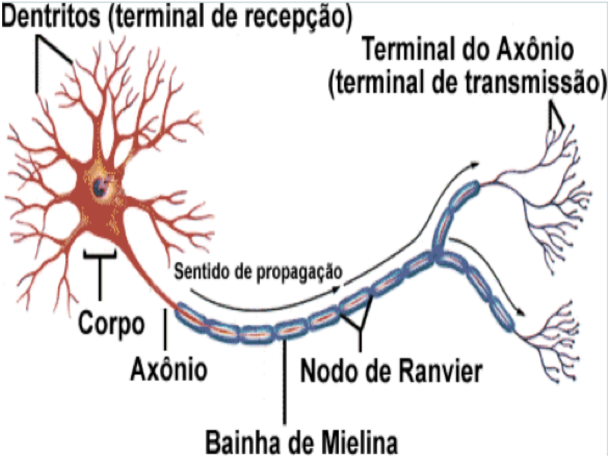
McCulloch, W. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, 7:115 - 133.

- O neurônio de McCulloch and Pitts (M-P) é uma aproximação util do neurônio biológico e serve como bloco construtivo das redes neurais artificiais.

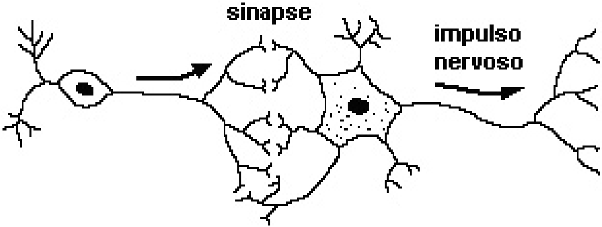
Neurônio Biológico

- Grosso modo, um neurônio biológico é composto de:
    - Dendritos;
    - Corpo Celular; e
    - Axônio.
- Não menos importante são as conexões sinápticas. As sinapses ocorrem no contato das terminações nervosas entre neurônios.
- O neurônio de M-P objetiva descrever os aspectos relacionados ao pontecial de ação que trafega:
    - de receptores (sensoriais) para um neurônio;
    - entre neurônios; e
    - de um neurônio para a um atuador (i.e., músculo).

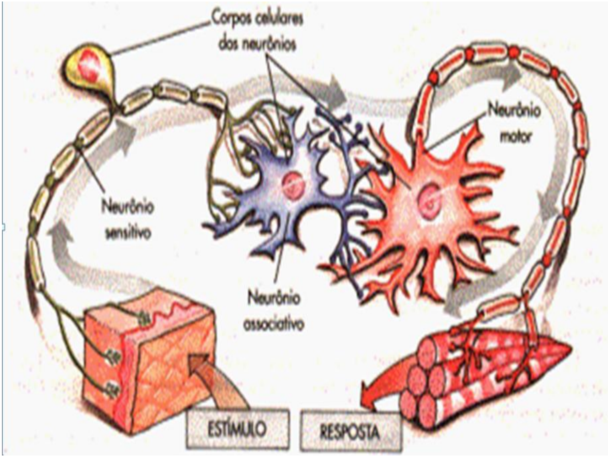
Neurônio Biológico



Interação entre Neurônios Biológicos



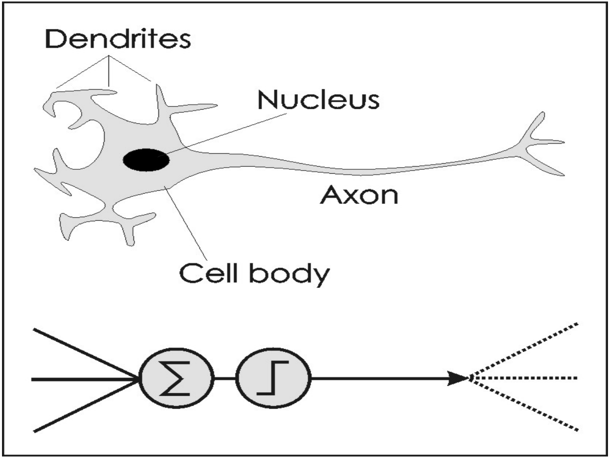
Interação entre Neurônios Biológicos



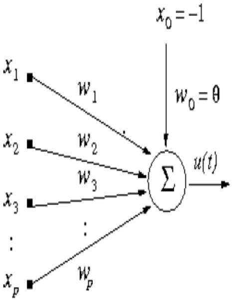
Neurônio de M-P

- Os ramos da árvore dentrítica são modelados por canais de transmissão, através dos quais flui a informação de entrada. Cada componente da informação de entrada é um escalar x_j, tal que $j = 1, \ldots, p$.
- As conexões sinápticas são excitatórias ou inibitórias. A contribuição de todos os neurônios pré-sinápticos determina se o neurônios que recebe os sinais gera ou não um impulso nervoso para o próximo neurônio.
- O acumulo energético realizado pelo corpo celular é modelado por uma operação de somatório sobre as entradas ponderadas pelos pesos sinápticos.
- Em resumo, um neurônio é modelado da seguinte forma: as suas múltiplas entradas recebem ativações excitatórias ou inibitórias dos neurônios anteriores, e caso a soma de excitações e inibições ultrapasse um determinado limite (limiar de ativação), o neurônio emite um impulso nervoso. Nesse contexto, o mesmo é modelado como uma chave *On/Off*.

Neurônio Biológico e Artificial (M-P)



Variável de Ativação do Neurônio de M-P



$$u = w_1x_1 + w_2x_2 + ... + w_px_p - \theta$$

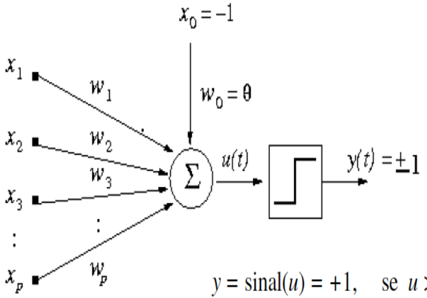
x_1, x_2: entradas

w_1, w_2: pesos sinápticos

θ: limiar (*bias*)

u : ativação

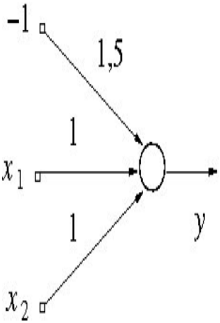
Variável de Saída do Neurônio de M-P



$y = \text{sinal}(u) = +1, \quad \text{se } u > 0$

$y = \text{sinal}(u) = -1, \quad \text{caso contrário.}$

Pesos do Neurônio de M-P para porta AND

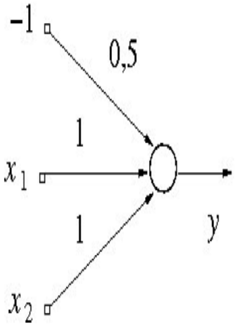


$w_1 = w_2 = 1$ e $\theta = 1{,}5$

$y = 1$, se $u \geqslant 0$.

$y = 0$, se $u < 0$.

Pesos do Neurônio de M-P para porta OR

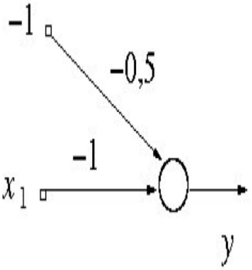


$w_1 = w_2 = 1$ e $\theta = 0{,}5$

$y = 1$, se $u \geqslant 0$.

$y = 0$, se $u < 0$.

Pesos do Neurônio de M-P para porta NOT



$w_1 = -1$ e $\theta = -0,5$

$y = 1$, se $u \geqslant 0$.

$y = 0$, se $u < 0$.

Perguntas

Pergunta 1

Como poderia se determinar cada um dos valores para os pesos sinápticos?

Resposta: na tentativa e erro, porém não seria eficiente pois o espaço de busca pode ser muito grande.

Pergunta 2

Como determinar os valores para os pesos sinápticos de forma automática?

Resposta: através de um mecanismo ou regra de aprendizagem. Esse mecanismo pode ser alcançado pela minimização de uma função custo ou por uma solução analítica em termos geométricos.

Não esqueça!

- O neurônio artificial possui p entradas $\{x_i\}_{i=1}^{p}$ e possui p pesos sinápticos $\{w_i\}_{i=1}^{p}$.
- O neurônio artificial possui ainda um limiar (thresold) de ativação θ.
- O neurônio possui uma variável de ativação u, tal que

$$u = x_1 w_1 + x_2 w_2 + \cdots - \theta = \sum_{i=1}^{p} x_i w_i - \theta. \tag{1}$$

Não esqueça!

- O neurônio possui também uma variável de saída y, tal que

$$y = sinal(u) = \begin{cases} 0, & \text{se} \quad u \leq 0, \\ +1, & \text{se} \quad u > 0. \end{cases} \tag{2}$$

- Pode-se também modelar y com os valores $+1$ ou -1.
- Dado suas características, pode-se aplicar o neurônio de M-P para categorizar problemas com duas classes.

Peceptron Simples

- Em 1958, surgiu o primeiro algoritmo de RNAs proposto por Frank Rosenblatt, chamado Perceptron Simples (PS).

McCulloch and Pitts (1943)

Rosenblatt, Frank (1958), The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Cornell Aeronautical Laboratory, Psychological Review, v65, No. 6, pp. 386-408.

- De uma forma geral, consiste do neurônio de M-P combinado com uma regra de aprendizagem.
- A inteligência surge justamente da capacidade de aprender adicionada em virtude desta regra.

Peceptron Simples

- A ativação do Perceptron Simples é calculada similarmente à do neurônio de M-P, ou seja:

$$
\begin{aligned}
u &= x_1 w_1 + x_2 w_2 + \cdots + x_p w_p - \theta \\
u &= (-1)\theta + x_1 w_1 + x_2 w_2 + \cdots + x_p w_p \\
u &= x_0 w_0 + x_1 w_1 + x_2 w_2 + \cdots + x_p w_p
\end{aligned} \tag{3}
$$

e pode ser apresentada na notação matricial, como:

$$u = \mathbf{w}^T \mathbf{x}, \tag{4}$$

Perceptron Simples

- Na equação para o cálculo da variável de ativação

$$u = \mathbf{w}^T \mathbf{x} = \sum_{i=0}^{p} w_i x_i, \tag{5}$$

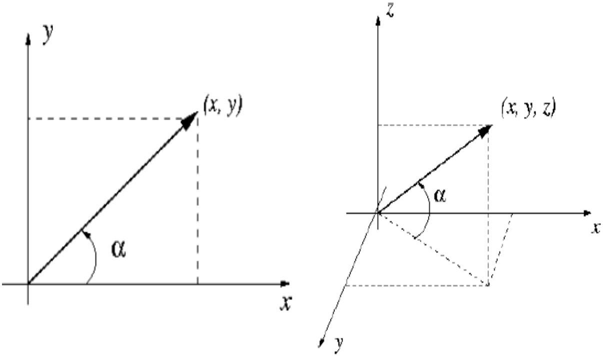
o vetor de pesos transposto $\mathbf{w}^T$ e de entrada $\mathbf{x}$ pode ser representado por

$$\mathbf{w}^{\mathbf{T}} = \begin{bmatrix} \theta & w_1 & w_2 & \dots & w_p \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \dots \\ x_p \end{bmatrix}$$

Perceptron Simples

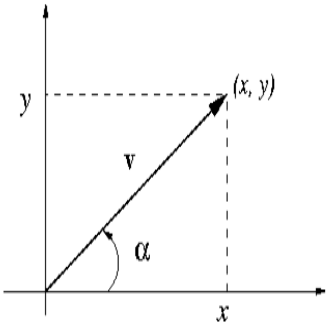
Um pouco sobre vetores

- Um vetor é uma coordenada (ponto) em um espaço de dimensão *p*.



Perceptron Simples

Comprimento de um vetor no $\mathbf{R}^2$ é apresentado abaixo.



$$\|\mathbf{v}\| = \sqrt{x^2 + y^2}$$

$$x = \|\mathbf{v}\| . \cos(\alpha)$$

$$y = \|\mathbf{v}\| . \sin(\alpha)$$

Perceptron Simples

Produto interno

- **Definição 1:**

$$u = \mathbf{w}^T \mathbf{x} = \mathbf{x}^T \mathbf{w} \tag{6}$$

O produto escalar é definido como o produto de um vetor linha por um vetor coluna, o que equivale a multiplicar cada componente de um vetor pelo seu correspondente no outro vetor e depois somar cada produto.

Perceptron Simples

Produto interno

- **Definição 2:**

$$u = \|\mathbf{x}\| \|\mathbf{y}\| \cos(\alpha) \tag{7}$$

Alternativamente, o produto escalar pode ser definido como o produto dos comprimentos dos vetores com o cosseno do menor ângulo entre eles.

Perceptron Simples

Produto interno (Definição 2)

- Lei dos Cossenos:

$$\|\mathbf{u}-\mathbf{v}\|^2 = \|\mathbf{u}\|^2 - 2\underbrace{\|\mathbf{u}\|\|\mathbf{v}\|\cos\theta}_{(I)} + \|\mathbf{v}\|^2 \tag{8}$$

- Além disto, podemos desenvolver a expressão como a seguir.

$$\|\mathbf{u}-\mathbf{v}\|^2 = \sum_{i=1}^{p}(u_i - v_i)^2 \tag{9}$$

$$\|\mathbf{u}-\mathbf{v}\|^2 = \sum_{i=1}^{p} u_i^2 - 2\sum_{i=1}^{p} u_i v_i + \sum_{i=1}^{p} v_i^2 \tag{10}$$

$$\|\mathbf{u}-\mathbf{v}\|^2 = \|\mathbf{u}\|^2 - 2\underbrace{\sum_{i=1}^{p} u_i v_i}_{(II)} + \|\mathbf{v}\|^2 \tag{11}$$

Perceptron Simples

Produto interno (Definição 2)

- Uma vez que as expressões

$$\|\mathbf{u}-\mathbf{v}\|^2 = \|\mathbf{u}\|^2 - 2\underbrace{\|\mathbf{u}\|\|\mathbf{v}\|\cos\theta}_{(I)} + \|\mathbf{v}\|^2 \qquad (12)$$

- e

$$\|\mathbf{u}-\mathbf{v}\|^2 \quad = \quad \|\mathbf{u}\|^2 - 2\underbrace{\sum_{i=1}^{p} u_i v_i}_{(II)} + \|\mathbf{v}\|^2 \qquad (13)$$

- são similares, temos que:

$$\cos\theta = \frac{\sum_{i=1}^{p} u_i v_i}{\|\mathbf{u}\|\|\mathbf{v}\|} \qquad (14)$$

Perceptron Simples

Produto interno

- O produto escalar é uma medida de similaridade entre vetores.

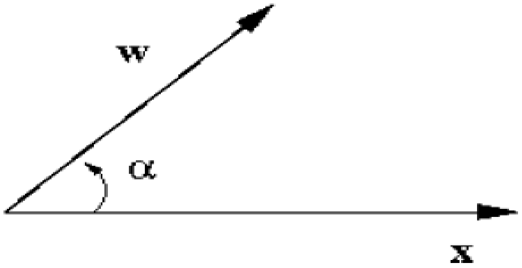
Para vetores de comprimento fixo, quanto menor o ângulo entre eles, maior é o valor resultante do produto escalar.

- O sinal do produto escalar também é um item importante na análise da orinetação entre os dois vetores.
- O sinal depende basicamente do ângulo entre os vetores.

Perceptron Simples

Produto interno

Caso 1: $0 \leq \alpha < 90^o$ e considerando que $u = \|\mathbf{x}\|\|\mathbf{y}\|\cos(\alpha)$

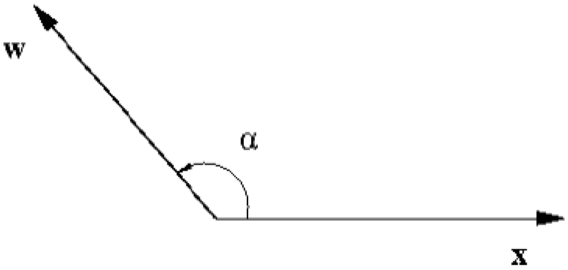


$$\cos(\alpha) > 0 \quad \Rightarrow \quad u > 0 \text{ (positivo)}$$

Perceptron Simples

Produto interno

Caso 1: $90 < \alpha \leq 180^o$ e considerando que $u = \|\mathbf{x}\| \|\mathbf{y}\| \cos(\alpha)$



$$\cos(\alpha) < 0 \quad \Rightarrow \quad u < 0 \text{ (negativo)}$$

Perceptron Simples

Variável de Saída do PS

- O cálculo para a variável de saída do Perceptron Simples é realizado da mesma maneira que o cálculo para o neurônio de M-P.

$$y \quad = \quad sinal(u) = \begin{cases} 0, & \text{se} \quad u \leq 0, \\ +1, & \text{se} \quad u > 0. \end{cases} \tag{15}$$

Perceptron Simples

Regra de Aprendizagem

- O processo de aprendizagem consiste basicamente na modificação dos pesos e do limiar do neurônio de M-P até que ele resolva o problema de interesse ou que o período de aprendizagem tenha finalizado.
- O mecanismo ou regra de aprendizagem depende:
  - do erro entre a saída desejada (d) e a saída da rede (y); e
  - da entrada ($\mathbf{x}$) apresentada ao Perceptron Simples. Isto ocorre porque u depende de $\mathbf{x}$ e y depende de u.

Perceptron Simples

Regra de Aprendizagem

- A idéia é pensar que deve haver uma variação dos valores contidos nos pesos durante o aprendizado, ou seja

$$\Delta \mathbf{w} = \mathbf{w}(t+1) - \mathbf{w}(t) \tag{16}$$
$$\mathbf{w}(t+1) = \mathbf{w}(t) + \Delta \mathbf{w}$$

em que

- $\Delta \mathbf{w}$ é o incremento memória necessário para o ajuste em relação a um vetor de entrada $\mathbf{x}$;
- $\mathbf{w}(t)$ representa o conhecimento no tempo t salvo na memória; e
- $\mathbf{w}(t+1)$ representa o conhecimento no tempo $t+1$ salvo na memória.

Perceptron Simples

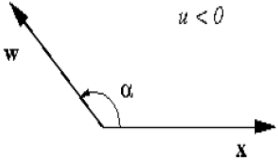
Regra de Aprendizagem

- Como definir $\Delta \mathbf{w}$?
- Vamos considerar argumentos geométricos para esta definição.
- Nesse contexto, temos 3 casos para a variável **erro**
- **Caso 1:** $e(t) = d(t) - y(t) = +1$, para $d = +1$ e $y = 0$;
- **Caso 2:** $e(t) = d(t) - y(t) = -1$, para $d = 0$ e $y = +1$
- **Caso 3:** $e(t) = d(t) - y(t) = 0$, para $(d = 0$ e $y = 0)$ ou $(d = +1$ e $y = +1)$.
- Obs: nos casos 1 e 2 houve **ERRO**, no caso 3 houve **ACERTO**.

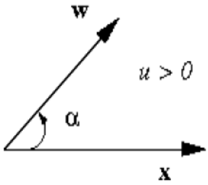
Perceptron Simples

Caso 1: $e = d - y = +1 \quad (d{=}+1 \text{ e } y{=}0)$

Situação ocorrida ($u<0$, $y=0$):



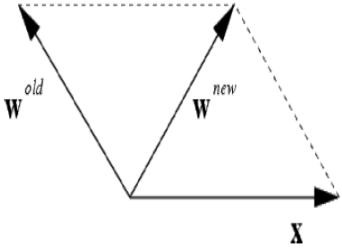
Situação desejada ($u>0$, $y=1$):



Perceptron Simples

Caso 1 $[e(t) = +1]$: O vetor $\mathbf{w}$ deve ser modificado para se aproximar de $\mathbf{x}$.

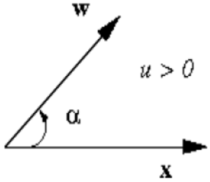
$$\mathbf{w}(t+1) = \mathbf{w}(t) + \mathbf{x}(t)$$



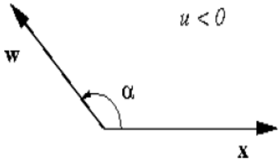
Perceptron Simples

Caso 2: $e = d - y = -1$ (d=0 e y=+1)

Situação ocorrida (u>0, y=+1):



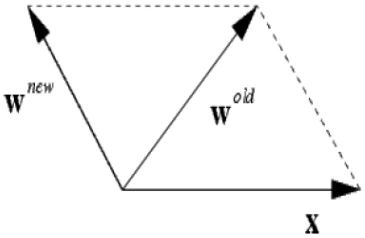
Situação desejada (u<0, y=0):



Perceptron Simples

Caso 2 [$e(t) = -1$]: O vetor **w** deve ser modificado para se afastar de **x**.

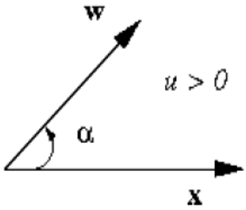
$$\mathbf{w}(t+1) = \mathbf{w}(t) - \mathbf{x}(t)$$



Perceptron Simples

Caso 3a: $e = d - y = 0 \quad (d{=}{+}1 \text{ e } y{=}{+}1)$

Situação ocorrida = Situação desejada (u>0, y=+1)

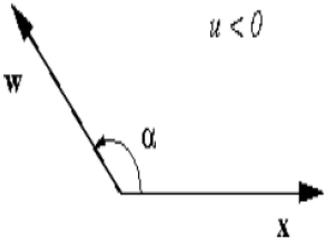


Como houve um acerto, não é preciso modificar o vetor **w**.

Perceptron Simples

Caso 3b: $e = d - y = 0$ (d=0 e y=0)

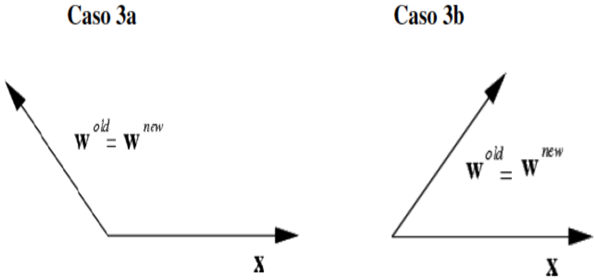
Situação ocorrida = Situação desejada (u<0, y=0)



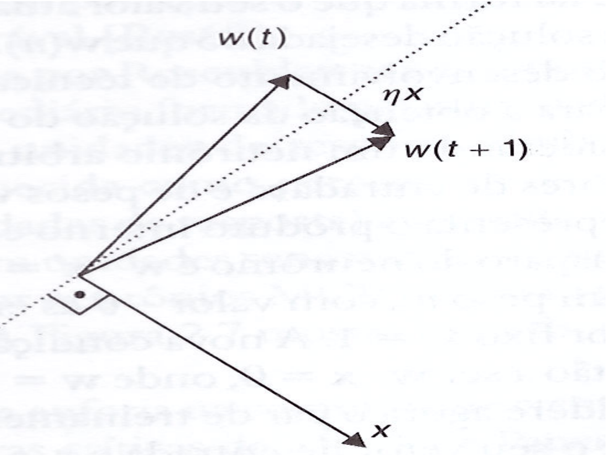
Perceptron Simples

Caso 3 $[e(t) = 0]$: O vetor $\mathbf{w}$ não deve ser modificado.

$$\mathbf{w}(t+1) = \mathbf{w}(t)$$



Perceptron Simples



Perceptron Simples

Regra de Aprendizagem

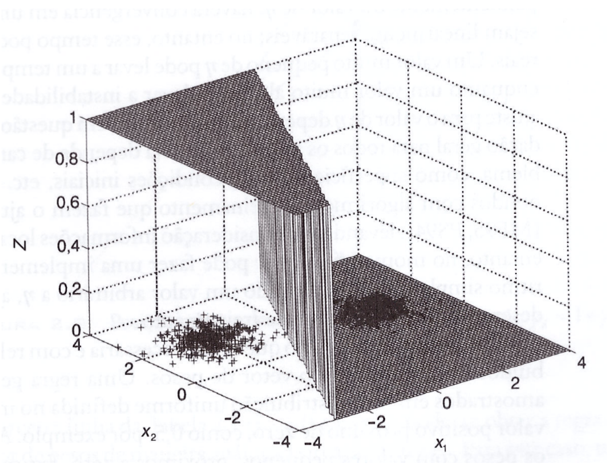
- Do exposto anteriomtente, pode-se inferir que:

$$\begin{aligned} \mathbf{w}(t+1) &= \mathbf{w}(t) + \Delta\mathbf{w}(t) \\ \mathbf{w}(t+1) &= \mathbf{w}(t) + (\eta\mathbf{x}(t))e(t) \\ \mathbf{w}(t+1) &= \mathbf{w}(t) + \eta e(t)\mathbf{x}(t) \end{aligned} \tag{17}$$

  em que $\eta\mathbf{x}(t)$ representa o vetor a ser somado para aproximação ou afastamento de **w** em relação a **x**.

- Em geral utiliza-se um valor para η, denominado fator ou taxa de aprendizagem, pequeno $(0 < \eta << 1)$.

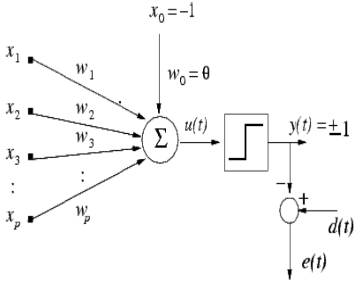
Perceptron Simples



Perceptron Simples

Resumo do Algoritmo do Percepron Simples

Perceptron Simples = Neurônio de M-P + Regra de Aprendizagem



$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta e(t)\mathbf{x}(t)$$

Perceptron Simples

Resumo do Algoritmo do Percepron Simples

1. Início (t=0)
   - 1.1 – Definir valor de η entre 0 e 1.
   - 1.2 – Iniciar $\mathbf{w}(0)$ com valores nulos ou aleatórios.
2. Funcionamento
   - 2.1 – Selecionar vetor de entrada $\mathbf{x}(t)$.
   - 2.2 – Calcular ativação $u(t)$.
   - 2.3 – Calcular saída $y(t)$.
3. Treinamento
   - 3.1 – Calcular erro: $e(t) = d(t) - y(t)$
   - 3.2 – Ajustar pesos via regra de aprendizagem.
   - 3.3 – Verificar critério de parada.
     - 3.3.1 – Se atendido, finalizar treinamento.
     - 3.3.2 – Caso contrário, fazer t=t+1 e ir para Passo 2.

Obrigado