

LABIC

Modelos de Neurônios

- Características Básicas
- Modelo de Neurônio
- Estrutura da Rede
- Algoritmo de Aprendizado

REDES NEURAIS
- RAFR

1

LABIC

Introdução

Neurônio:

- Cada neurônio é composto por:
 - dendritos:** conj. de terminais de entrada corpo central
 - axônio:** um longo terminal de saída
- Dentro de um neurônio as mensagens fluem dos dendritos para o axônio passando pelo corpo celular.

REDES NEURAIS
- RAFR

2

LABIC

Introdução

- O processo de informações no sistema nervoso (redes neurais naturais) é um processo de natureza eletroquímica
 - transmissão de impulsos nervosos dentro do neurônio é um processo de natureza elétrica.
 - transmissão sináptica se faz por um mecanismo de natureza química

REDES NEURAIS
- RAFR

3

LABIC

Introdução

- Sinapses: região através da qual os impulsos nervosos são transmitidos de neurônio para neurônio.
- Podem ser:
 - excitatórias - estimulam a ação do neurônio
 - inibitórias - tem efeito contrário

REDES NEURAIS
- RAFR

4

x_1, x_2, \dots, x_N ← Entradas (Dendritas)
 w_1, w_2, \dots, w_N ← Pesos Sinápticos (Efeito de inibição ou de excitação sobre a ativação dos sinais de entrada)
 $Net = \sum_{i=1}^N w_i \cdot x_i + \text{Bias}$ ← Ativação (Considera o conjunto total das entradas e dos seus pesos associados)
 $Fct(Net)$ ← Função de Ativação (Regulagem da saída da rede)
 Saída (Axônio: Conexões com as Sinapses de outras unidades)

F. OSÓRIO - UNISINOS 2002

LABIC

I - PERCEPTRON

x_1, x_2, \dots, x_n
 w_1, w_2, \dots, w_n
 Σ
 -1
 θ
 $y = f(\sum_i w_i x_i - \theta)$
 F é função sinal

REDES NEURAIS
- RAFR

6

LABIC

Estrutura Básica de um Neurônio Artificial

- **Estado de Ativação (Saída):** s_j
- **Conexões entre Processadores:** w_{ij}
 - a cada conexão existe um peso sináptico que determina o efeito da entrada sobre o processador
- **Soma:** cada processador soma os sinais de entrada ponderado pelo peso sináptico das conexões
- **Função de Ativação:** $s_j = F(net_j)$
 - determina o novo valor do Estado de Ativação do processador.

REDES NEURAIS - RAFR 7

LABIC

NEURÔNIO ARTIFICIAL

Funções de Ativação Comuns:

É a função que determina o nível de ativação do Neurônio Artificial: $s_j = F(net_j)$

$s_j = \begin{cases} 1 & net_j > 0 \\ 0 & net_j \leq 0 \end{cases}$
 $s_j = \begin{cases} 1 & net_j > 1 \\ net_j & 0 < net_j \leq 1 \\ 0 & net_j \leq 0 \end{cases}$
 $s_j = \frac{1}{1 + e^{-\alpha net_j}}$

REDES NEURAIS - RAFR 8

LABIC

Funções de Transferência

Hard Limiter- Degrau
Threshold logic
Sigmoid

REDES NEURAIS - RAFR 9

LABIC

Modelos de Neurônios

$x_0 = -1$, $w_0 = \theta$ (threshold)

$y = f(\sum_i w_i x_i - \theta)$

F é função sinal

REDES NEURAIS - RAFR 10

LABIC

Modelos de Neurônios

$x_0 = +1$, $w_0 = b$ (bias)

$y = f(\sum_i w_i x_i)$

F é função sinal

REDES NEURAIS - RAFR 11

LABIC

CARACTERÍSTICAS BASICAS

- Regra de propagação
- Função de ativação: função escada
- Topologia: uma única camada de processadores
- Algoritmo de Aprendizado: $\Delta w_{ij} = \eta x_i(t_j - y_j)$ (é do tipo supervisionado)
- Valor de Entrada/Saída: Binários $t = 0$ ou 1

REDES NEURAIS - RAFR 12

LABIC Finalidade do Termo *Bias*:

(a) without bias (b) with bias

$\sum_i x_i w_{ij} = 0$ Define um hiperplano passando pela origem

$\sum_i x_i w_{ij} + \theta_i = 0$ Desloca-se o hiperplano da origem

RAFR REDES NEURAIS - RAFR 13

LABIC **ALGORITMO DE APRENDIZADO**
Regra Delta - LMS

- 1) iniciar os pesos sinápticos com valores randomicos e pequenos ou iguais a zero;
- 2) aplicar um padrão com seu respectivo valor desejado de saída (t_j) e verificar a saída da rede (y_j);
- 3) calcula o erro na saída $E_j = t_j - y_j$;
- 4) se $E_j = 0$, volta ao passo 2; se $E_j \neq 0$, atualiza os pesos: $\Delta w_{ij} = \eta x_i E_j$;
- 5) volta ao passo 2.

RAFR REDES NEURAIS - RAFR 14

LABIC **ALGORITMO DE APRENDIZADO**

IMPORTANTE

- não ocorre variação no peso se a saída estiver correta;
- caso contrário, cada peso é incrementado de η quando a saída é menor que o valor-alvo e decrementado de η quando a saída é maior que o valor-alvo.

$$\Delta w_{ij} = \eta x_i e_j$$

RAFR REDES NEURAIS - RAFR 15

LABIC **Gradiente de uma função**

Gradiente: $\nabla f(x, y) = \left(\frac{\partial}{\partial x} f(x, y), \frac{\partial}{\partial y} f(x, y) \right)$

Derivada direcional: $D_u f(x, y) = \nabla f(x, y) \cdot \mathbf{u}$

$= \|\nabla f(x, y)\| \|\mathbf{u}\| \cos \gamma$

$= \|\nabla f(x, y)\| \cos \gamma$

$D_u f(x, y)$ é a taxa de variação de $f(x, y)$ na direção definida por \mathbf{u} .

RAFR REDES NEURAIS - RAFR 16

LABIC **Gradiente de uma função**

Teorema do gradiente: Seja f uma função de duas variáveis, diferenciáveis no ponto $P(x, y)$.

- i) O máximo de $D_u f(x, y)$ em $P(x, y)$ é $\|\nabla f(x, y)\|$.
- ii) O máximo da taxa de crescimento de $f(x, y)$ em $P(x, y)$ ocorre na direção de $\nabla f(x, y)$.

Corolário: Seja f uma função de duas variáveis, diferenciáveis no ponto $P(x, y)$.

- i) O mínimo de $D_u f(x, y)$ em $P(x, y)$ é $-\|\nabla f(x, y)\|$.
- ii) O máximo da taxa de **decréscimo** de $f(x, y)$ em $P(x, y)$ ocorre na direção de $-\nabla f(x, y)$.

RAFR REDES NEURAIS - RAFR 17

LABIC **Superfície de Erro** **Processo de Minimização**

A direção do gradiente negativo é a de descida mais íngreme ("steepest descent")

RAFR REDES NEURAIS - RAFR 18

LABIC

Método do Gradiente Descendente (GD)

$$\Delta w_{ij} = -\eta \frac{\delta E_j}{\delta w_{ij}}$$

Cada *peso sináptico* i do elemento processador j é atualizado proporcionalmente ao *negativo da derivada parcial do erro* deste processador com relação ao peso.

REDES NEURAIS - RAFR 19

LABIC

Logo:

$$\Delta w_{ij} = -\eta \frac{\delta E_j}{\delta w_{ij}} = -\eta \frac{\delta E_j}{\delta s_j} \frac{\delta s_j}{\delta w_{ij}}$$

$$E_j = \frac{1}{2} (t_j - s_j)^2$$

$$s_j = k \Sigma x_i \cdot w_{ij} + \theta$$

$$\Delta w_{ij} = -\eta \cdot [2 \cdot \frac{1}{2} (t_j - s_j) \cdot (-1)] \cdot x_i$$

$$= -\eta \cdot [-(t_j - s_j)] \cdot x_i = \eta x_i (t_j - s_j)$$

REDES NEURAIS - RAFR 20

LABIC

EXEMPLO

Simulação do Operador Lógico AND

AND	x_0	x_1	x_2	t
Entrada 1:	1	0	0	0
Entrada 2:	1	0	1	0
Entrada 3:	1	1	0	0
Entrada 4:	1	1	1	1

Peso inicial: $w_0=0, w_1=0, w_2=0$
Taxa de aprendizado: $\eta=0.5$

REDES NEURAIS - RAFR 21

LABIC

EXEMPLO

1º Ciclo

Entrada 1: $s_{out} = f(w_0 x_0 + w_1 x_1 + w_2 x_2)$
 $= f(0 \times 1 + 0 \times 0 + 0 \times 0) = f(0) = 0 \rightarrow s_{out} = t$

Entrada 2: $s_{out} = f(w_0 x_0 + w_1 x_1 + w_2 x_2)$
 $= f(0 \times 1 + 0 \times 1 + 0 \times 0) = f(0) = 0 \rightarrow s_{out} = t$

Entrada 3: $s_{out} = f(w_0 x_0 + w_1 x_1 + w_2 x_2)$
 $= f(0 \times 1 + 0 \times 0 + 0 \times 1) = f(0) = 0 \rightarrow s_{out} = t$

Entrada 4: $s_{out} = f(w_0 x_0 + w_1 x_1 + w_2 x_2)$
 $= f(0 \times 1 + 0 \times 1 + 0 \times 1) = f(0) = 0 \rightarrow s_{out} \neq t$

$w_0 = w_0 + (t - s_{out}) x_0 = 0 + 0.5 \times (1 - 0) \times 1 = 0.5$
 $w_1 = w_1 + (t - s_{out}) x_1 = 0 + 0.5 \times (1 - 0) \times 1 = 0.5$
 $w_2 = w_2 + (t - s_{out}) x_2 = 0 + 0.5 \times (1 - 0) \times 1 = 0.5$

REDES NEURAIS - RAFR 22

LABIC

EXEMPLO

2º Ciclo

Entrada 1: $s_{out} = f(w_0 x_0 + w_1 x_1 + w_2 x_2)$
 $= f(0.5 \times 1 + 0.5 \times 0 + 0.5 \times 0) = f(0.5) = 1 \rightarrow s_{out} \neq t$

$w_0 = w_0 + (t - s_{out}) x_0 = 0.5 + 0.5 \times (0 - 1) \times 1 = 0$
 $w_1 = w_1 + (t - s_{out}) x_1 = 0.5 + 0.5 \times (0 - 1) \times 0 = 0.5$
 $w_2 = w_2 + (t - s_{out}) x_2 = 0.5 + 0.5 \times (0 - 1) \times 0 = 0.5$

Entrada 2: $s_{out} = f(w_0 x_0 + w_1 x_1 + w_2 x_2)$
 $= f(0 \times 1 + 0.5 \times 0 + 0.5 \times 1) = f(0.5) = 1 \rightarrow s_{out} \neq t$

$w_0 = w_0 + (t - s_{out}) x_0 = 0 + 0.5 \times (0 - 1) \times 1 = -0.5$
 $w_1 = w_1 + (t - s_{out}) x_1 = 0.5 + 0.5 \times (0 - 1) \times 0 = 0.5$
 $w_2 = w_2 + (t - s_{out}) x_2 = 0.5 + 0.5 \times (0 - 1) \times 1 = 0$

REDES NEURAIS - RAFR 23

LABIC

EXEMPLO

2º Ciclo

Entrada 3: $s_{out} = f(w_0 x_0 + w_1 x_1 + w_2 x_2)$
 $= f(-0.5 \times 1 + 0.5 \times 1 + 0 \times 0) = f(0) = 0 \rightarrow s_{out} = t$

Entrada 4: $s_{out} = f(w_0 x_0 + w_1 x_1 + w_2 x_2)$
 $= f(-0.5 \times 1 + 0.5 \times 1 + 0 \times 1) = f(0) = 0 \rightarrow s_{out} \neq t$

$w_0 = w_0 + (t - s_{out}) x_0 = -0.5 + 0.5 \times (1 - 0) \times 1 = 0$
 $w_1 = w_1 + (t - s_{out}) x_1 = 0.5 + 0.5 \times (1 - 0) \times 1 = 1$
 $w_2 = w_2 + (t - s_{out}) x_2 = 0 + 0.5 \times (1 - 0) \times 1 = 0.5$

REDES NEURAIS - RAFR 24

EXEMPLO

LABIC

3º Ciclo

Entrada 1: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$
 $= f(0 \times 1 + 1 \times 0 + 0.5 \times 0) = f(0) = 0 \rightarrow s_{out} = t$

Entrada 2: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$
 $= f(0 \times 1 + 1 \times 0 + 0.5 \times 1) = f(0.5) = 1 \rightarrow s_{out} \neq t$

$w_0 = w_0 + (t - s_{out})x_0 = -0.5 + 0.5 \times (0 - 1) \times 1 = -1$
 $w_1 = w_1 + (t - s_{out})x_1 = 1 + 0.5 \times (0 - 1) \times 0 = 1$
 $w_2 = w_2 + (t - s_{out})x_2 = 0.5 + 0.5 \times (0 - 1) \times 1 = 0$

REDES NEURAIS - RAFR 25

EXEMPLO

LABIC

3º Ciclo

Entrada 3: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$
 $= f(-1 \times 1 + 1 \times 1 + 0 \times 0) = f(0) = 0 \rightarrow s_{out} = t$

Entrada 4: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$
 $= f(-1 \times 1 + 1 \times 1 + 0 \times 1) = f(0) = 0 \rightarrow s_{out} \neq t$

$w_0 = w_0 + (t - s_{out})x_0 = -1 + 0.5 \times (1 - 0) \times 1 = -0.5$
 $w_1 = w_1 + (t - s_{out})x_1 = 1 + 0.5 \times (1 - 0) \times 1 = 1.5$
 $w_2 = w_2 + (t - s_{out})x_2 = 0 + 0.5 \times (1 - 0) \times 1 = 0.5$

REDES NEURAIS - RAFR 26

EXEMPLO

LABIC

4º Ciclo

Entrada 1: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$
 $= f(-0.5 \times 1 + 1.5 \times 0 + 0.5 \times 0) = f(-0.5) = 0 \rightarrow s_{out} = t$

Entrada 2: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$
 $= f(-0.5 \times 1 + 1.5 \times 0 + 0.5 \times 1) = f(0) = 0 \rightarrow s_{out} = t$

Entrada 3: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$
 $= f(-0.5 \times 1 + 1.5 \times 1 + 0.5 \times 0) = f(1) = 1 \rightarrow s_{out} \neq t$

$w_0 = w_0 + (t - s_{out})x_0 = -0.5 + 0.5 \times (0 - 1) \times 1 = -1$
 $w_1 = w_1 + (t - s_{out})x_1 = 1.5 + 0.5 \times (0 - 1) \times 1 = 1$
 $w_2 = w_2 + (t - s_{out})x_2 = 0.5 + 0.5 \times (0 - 1) \times 0 = 0.5$

Entrada 4: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$
 $= f(-1 \times 1 + 1 \times 1 + 0.5 \times 1) = f(0.5) = 1 \rightarrow s_{out} = t$

REDES NEURAIS - RAFR 27

EXEMPLO

LABIC

5º Ciclo

Entrada 1: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$
 $= f(-1 \times 1 + 1 \times 0 + 0.5 \times 0) = f(-1) = 0 \rightarrow s_{out} = t$

Entrada 2: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$
 $= f(-1 \times 1 + 1 \times 0 + 0.5 \times 1) = f(-0.5) = 0 \rightarrow s_{out} = t$

Entrada 3: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$
 $= f(-1 \times 1 + 1 \times 1 + 0.5 \times 0) = f(0) = 0 \rightarrow s_{out} = t$

Entrada 4: $s_{out} = f(w_0x_0 + w_1x_1 + w_2x_2)$
 $= f(-1 \times 1 + 1 \times 1 + 0.5 \times 1) = f(0.5) = 1 \rightarrow s_{out} = t$

$w_0 = -1, w_1 = 1, w_2 = 0.5$

REDES NEURAIS - RAFR 28

INTERPRETAÇÃO GEOMÉTRICA

LABIC

Linha de Decisão:
 $x_1w_1 + x_2w_2 = -\theta$
 \downarrow
 $x_1 + 0.5x_2 = 1$

REDES NEURAIS - RAFR 29

Perceptron

LABIC

■ **OBS:** A classe de funções representadas por Perceptrons é Limitada

2) Perceptrons Não-Lineares

$Out(\underline{x}) = g(\underline{w}^t \underline{x}) \rightarrow y$

$Out(\underline{x}) = g(\underline{w}^T \underline{x}) = g(\sum_i w_i x_i)$ onde g é uma função não-linear (SIGMOID)

REDES NEURAIS - RAFR 30

O PROBLEMA DO OU-EXCLUSIVO (XOR)

LABIC

PONTO	x_1	x_2	Saída
A_0	0	0	0
A_1	0	1	1
A_2	1	0	1
A_3	1	1	0

Função Degrau

x_1 w_1 $F(\text{net})$ saída: s

x_2 w_2

De acordo com a definição do neurônio: $s = F(x_1 w_1 + x_2 w_2 + \theta)$

$\text{net} = x_1 w_1 + x_2 w_2 + \theta$

Se $\text{net} \geq 0 \rightarrow s = 1$
 Se $\text{net} < 0 \rightarrow s = 0$

A rede Perceptron divide o plano x_1, x_2 em duas regiões (através da reta net)

RAFR - RAFR 31

O PROBLEMA DO OU-EXCLUSIVO (XOR)

LABIC

Função AND

Função OR

Função OU-Exclusivo

RAFR - RAFR 32

O PROBLEMA DO OU-EXCLUSIVO (XOR)

LABIC

Conclusão

- mudando-se os valores de w_1 , w_2 e θ , muda-se a inclinação e a posição da reta;
- entretanto é impossível achar uma reta que divide o plano de forma separar os pontos A_1 e A_2 de um lado e A_0 e A_3 de outro
- redes de 1 única camada só representam **funções linearmente separáveis**

RAFR - RAFR 33

O PROBLEMA DO OU-EXCLUSIVO (XOR)

LABIC

Minsky & Papert provaram que este problema pode ser solucionado adicionando-se uma outra camada intermediária de processadores- Multi-Layer Perceptron (MLP)

REDES NEURAIS - RAFR 34

O PROBLEMA DO OU-EXCLUSIVO (XOR)

LABIC

Exemplo:

$w_{11} = w_{12} = w_{21} = w_{22} = +1$

$s_1 = 1 \rightarrow s_1 w_{11} + s_2 w_{21} + \theta_j \geq 0$
 $-2s_1 + s_2 - 0.5 \geq 0$
 $-2s_1 + s_2 \geq 0.5$

s_1 é inibitório e s_2 é excitatório

Região de $s_1 = 1$

Região de $s_2 = 0$

Região de $s_1 = 1$

RAFR - RAFR 35

MULTI-LAYER PERCEPTRON

LABIC

- Redes de apenas uma camada só representam funções linearmente separáveis
- Redes de múltiplas camadas solucionam essa restrição
- O desenvolvimento do algoritmo Back-Propagation foi um dos motivos para o ressurgimento da área de redes neurais

REDES NEURAIS - RAFR 36

LABIC

Algoritmos de Aprendizado

- **Problema**

\underline{x}_1	y_1
\underline{x}_2	y_2
\underline{x}_3	y_3
\underline{x}_N	y_N

\underline{x} denota um vetor de p componentes

RAFR REDES NEURAIS - RAFR 37

LABIC

- Perceptrons Lineares: eles são modelos multivariados lineares:
 - $\text{Out}(\underline{x}) = \underline{w}^T \underline{x} \rightarrow y$
- e o treinamento consiste de minimizar a soma dos quadrados do resíduo pelo método do “gradiente descent”.
- Regra Gradiente Descent:
 - $w \leftarrow w - \eta \partial f(w) / \partial w$

RAFR REDES NEURAIS - RAFR 38

LABIC

Adaline

- O objetivo do processo adaptativo do Adaline (perceptron) consiste em utilizar a função de ativação “hard limiter” e minimizar os pesos e θ usando o algoritmo LMS.
- Como a saída desejada pode ter apenas valores 1 ou -1, o erro pode ser +2, 0, ou -2.

RAFR REDES NEURAIS - RAFR 39

LABIC

Adaline

$$\Delta w_{jk} = -\eta \delta_j^p x_k^p$$

REGRA DELTA, REGRA LMS ou
REGRA de Widrow-Hoff (1960)

RAFR REDES NEURAIS - RAFR 40

LABIC

Exercícios

- Treinar uma rede de perceptrons constituída de uma camada intermediária constituída de um único neurônio, para implementar a função OU-EXCLUSIVO.
- Implementar um Perceptron para reconhecer a letra A e a letra A invertida.

RAFR REDES NEURAIS - RAFR 41