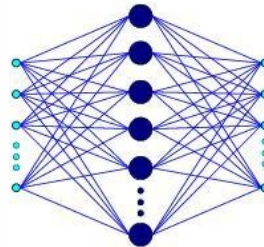


Ciência da Computação

REDE NEURAIS

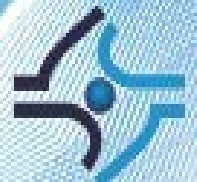
Semestre: 2010/1

AULA 05



Max Pereira

<http://paginas.unisul.br/max.pereira>



UNISUL

Aqui seu futuro acontece

Conteúdo

- Perceptron
- Exercícios

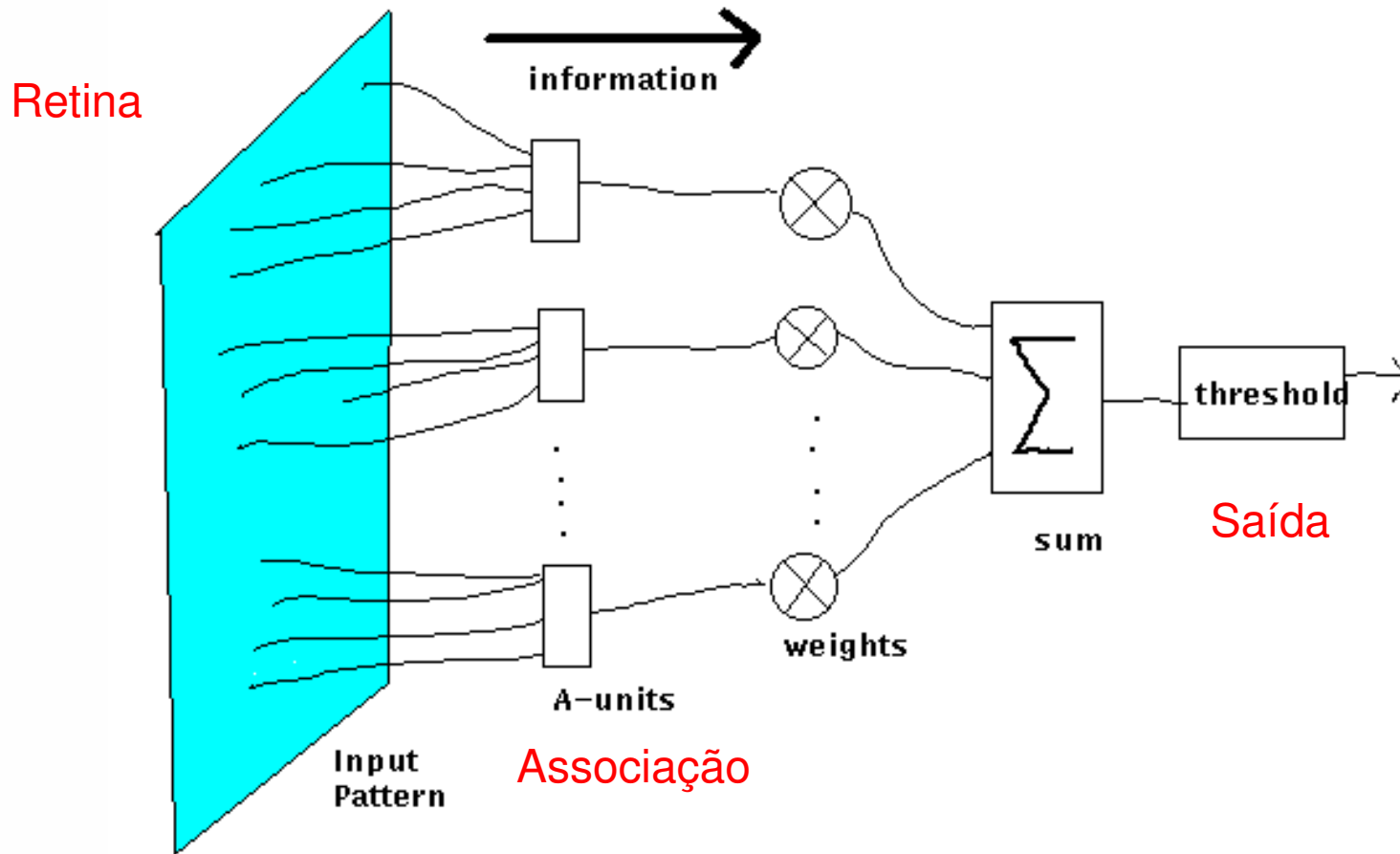
Perceptron

- Frank Rosenblatt (1958)
- Reconhecimento de letras maiúsculas do alfabeto
- A conectividade desenvolvida nas redes biológicas contém um grande número aleatório de elementos;
- Pode ser treinado;
- A topologia original era composta por unidades de entrada (retina), por um nível intermediário e por um nível de saída formado pelas unidades de resposta.

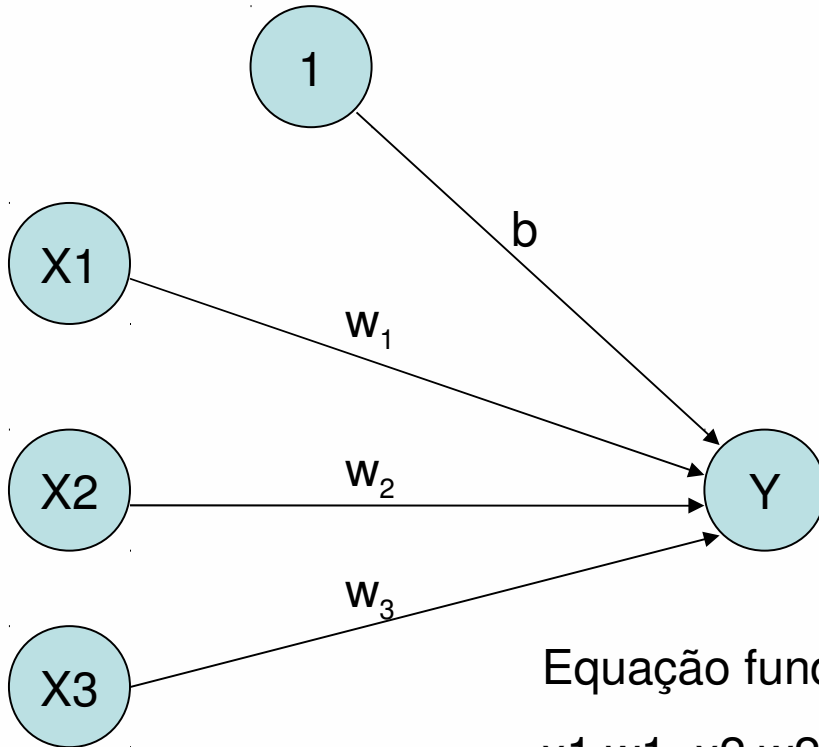
Perceptron

- Embora a topologia original possua três camadas, ela é conhecida como *perceptron* de uma única camada.
- A retina consiste basicamente em unidades sensoras, e as unidades intermediárias de associação

Perceptron



Perceptron



$$rede = b + \sum_i x_i w_i$$

$$Saída = f(rede) =$$

$$\{1 \text{ se } rede \geq \theta, 0 \text{ se } rede < \theta\}$$

Equação fundamental do Perceptron

$$x_1.w_1 + x_2.w_2 + x_3.w_3 = \theta \quad \longleftarrow \text{Equação de uma reta}$$

Perceptron (Algoritmo)

Passo 1. Inicializar pesos e *bias*

(para simplificar, setar os pesos e *bias* igual a zero e a taxa de aprendizado α ($0 < \alpha \leq 1$))

Passo 1. Enquanto a condição de parada for falsa, faça os passos 2 até 6.

Passo 2. Para cada par de treinamento $s:t$, faça os passos 3 até 5

Passo 3. Unidades de entrada (valores)

Passo 4. Calcular o valor da unidade de saída

$$rede = b + \sum_i x_i w_i$$

$$y = 1 \text{ se } rede \geq \theta, 0 \text{ se } -\theta \leq rede \leq \theta, -1 \text{ se } rede < -\theta$$

Passo 5. Ajustar pesos e *bias* se ocorrer um erro para o padrão

Se $y \neq t$

$$w_i(\text{new}) = w_i(\text{old}) + \alpha t x_i$$

$$b(\text{new}) = b(\text{old}) + \alpha t$$

senão

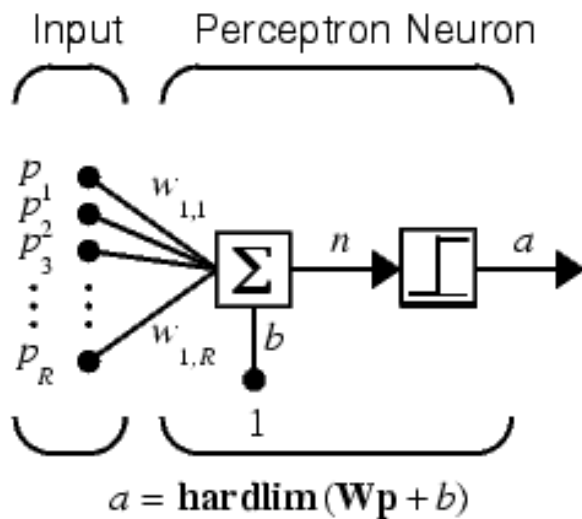
$$w_i(\text{new}) = w_i(\text{old})$$

$$b(\text{new}) = b(\text{old})$$

Passo 6. Testar condição de parada

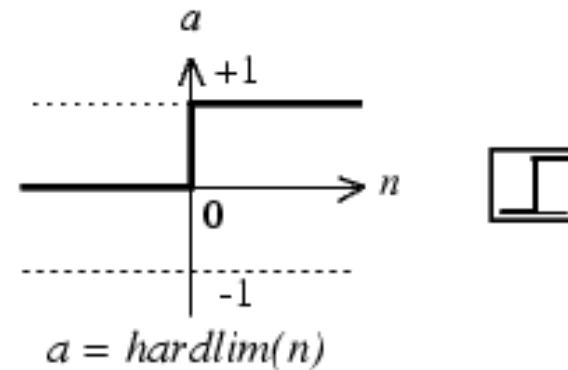
Se nenhum peso mudou no passo 2, *para*; senão *continua*

Perceptron - MatLab



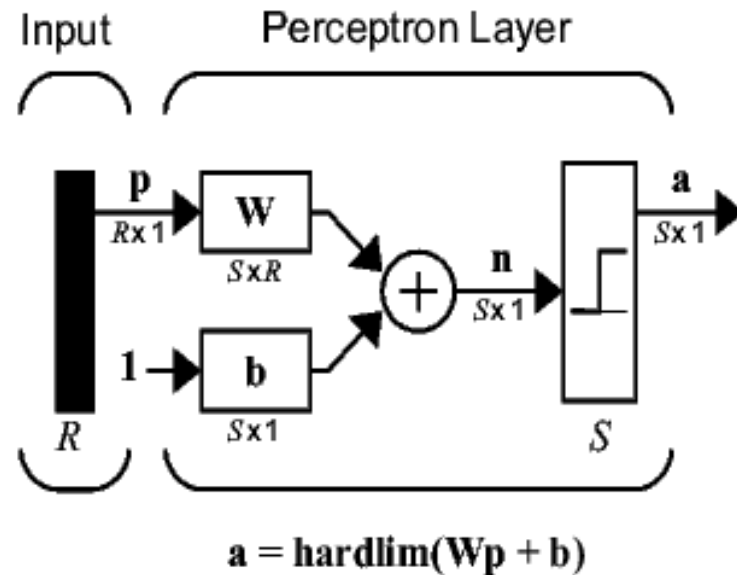
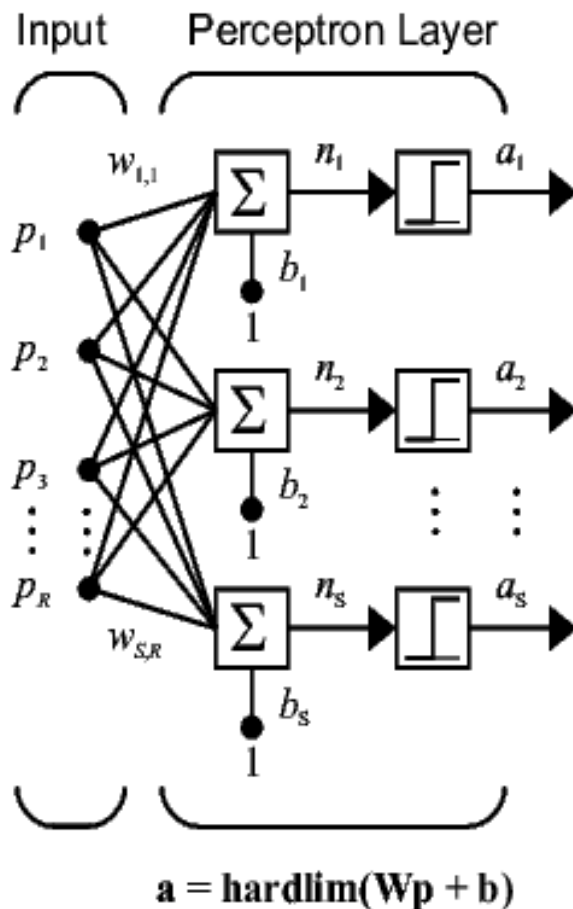
Where

R = number of elements in input vector



Hard-Limit Transfer Function

Perceptron – MatLab (Arquitetura)



Where

R = number of elements in input

S = number of neurons in layer

Perceptron - MatLab

NET = NEWP(PR,S,TF,LF),

PR - matriz Rx2 de valores máximos e mínimos para uma entrada de R elementos

S - Número de neurônios

TF – Função de transferência, default = 'hardlim'.

LF – Função de aprendizado, default = 'learnp'.

net = newp([-2 2; -2 2],1); - Cria um perceptron com 2 elementos de entrada

P2 = [0 0 1 1; 0 1 0 1];

T2 = [0 1 1 1];

net = init(net); - Inicializa os pesos e bias de forma randômica

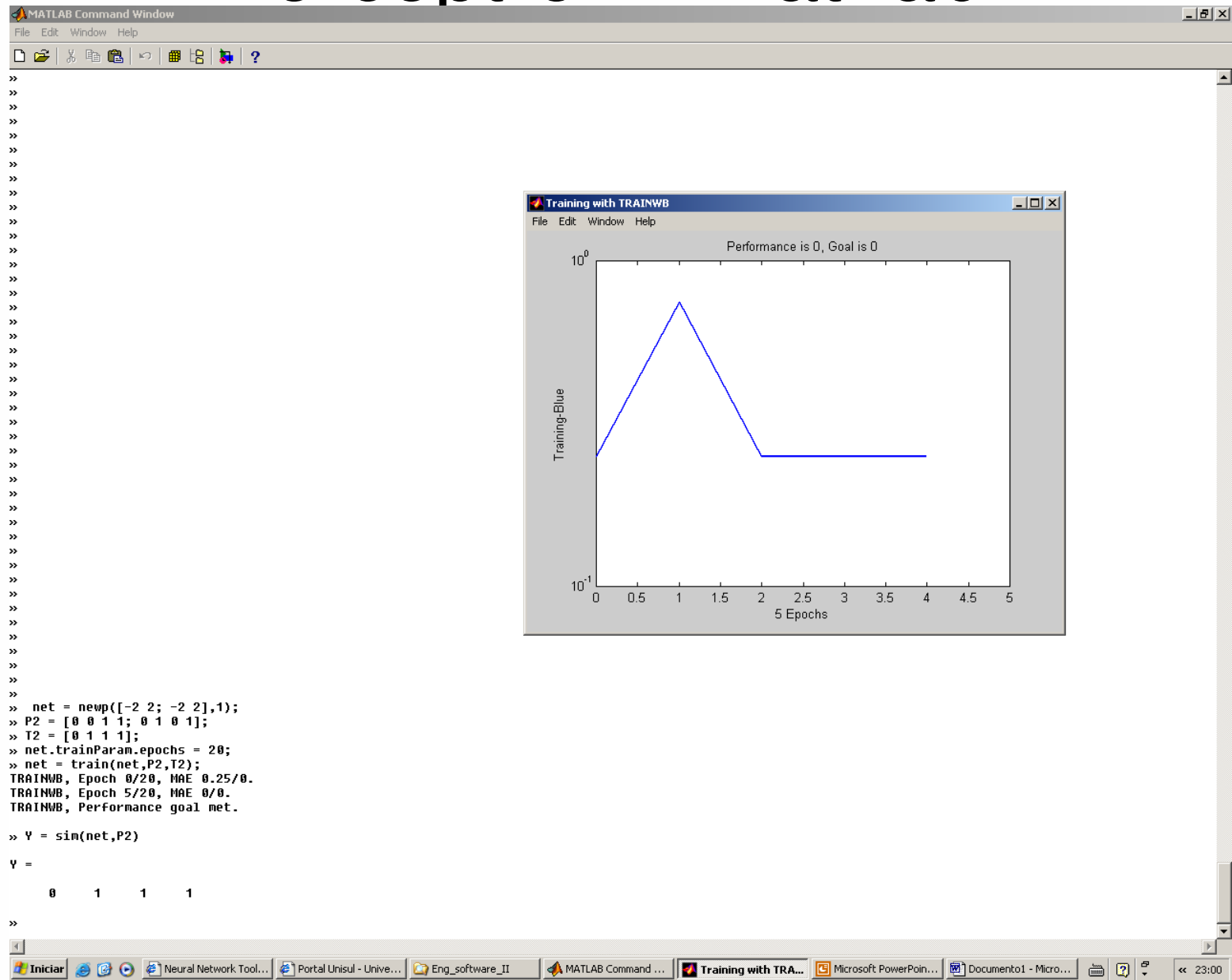
Y = sim(net,P2) – Simula a saída da rede

net.trainParam.epochs = 20; - Treina a rede com no máximo 20 épocas

net = train(net,P2,T2);

Y = sim(net,P2) – Simula a saída da rede novamente

Perceptron - MatLab



Exercício

Exercício

Reconhecimento de Caracteres

- Cada vetor de entrada é representado por um vetor de 63 posições que expressa uma matriz 7×9 .
- Os padrões de entrada estão ilustrados no material disponível no xerox (Cettal)
- Existem 7 categorias onde um padrão deve pertencer (sete padrões de saída).
- A representação deve ser bipolar (-1, 0, 1)
- Limiar = 0 e taxa de aprendizado = 1