

# Perceptrons

## Introdução

- No final da década de 1950, Rosenblatt na Universidade de Cornell, criou uma genuína rede de múltiplos neurônios do tipo *discriminadores lineares* e chamou esta rede de *perceptron*. Um perceptron é uma rede com os neurônios dispostos em camadas.
- Estes podem ser considerados o primeiro modelo de redes neurais.

# Introdução

- Perceptron aprende conceitos, ele pode aprender a responder com verdadeiro (1) ou falso (0) pelas entradas que nós apresentamos a ele, “estudando” repetidamente os exemplos que lhe são apresentados.
- O perceptron é uma rede neural cujos os pesos e inclinações podem ser treinados para produzir um vetor alvo que quando apresentamos tem que corresponder ao vetor de entrada.

# Treinamento

- Vetores de um grupo de treinamento são apresentados para a rede um após o outro. Se a saída da rede está correta, nenhuma mudança é feita. Por outro lado, os pesos e as inclinações são atualizados usando as regras de aprendizado do perceptron. Uma passagem inteira de treinamento de entrada de um vetor é chamado época.

## Limitações

- As redes perceptron tem duas limitações. Primeiro, os valores de saída do perceptron podem assumir somente dois valores (Verdadeiro ou Falso). Segundo, perceptrons somente podem classificar grupos de vetores linearmente separados.

## Perceptron de Camadas Simples

- O perceptron de camada simples é um exemplo de redes que podem ser usadas com entradas binárias e bipolares.
- Uma técnica usual para analisar o comportamento de redes como perceptron é plotar um mapa com as regiões de decisão criadas num espaço multidimensional abrangido pela variáveis de entrada.

# Perceptron de Camadas Simples

- 1º Passo: inicializar pesos e threshold com zero.
- 2º Passo: apresentar nova entrada que vai se somar a saída desejada .
- 3º Passo: calcular saída atual.
- 4º Passo: atualizar o peso. Esta atualização é feita através da fórmula:

# Perceptron de Camadas Simples

$$W_i(t+1) = W_i(t) + n[(d(t) - Y(t)) \times x_i(t)],$$

$$0 \leq i \leq N-1$$

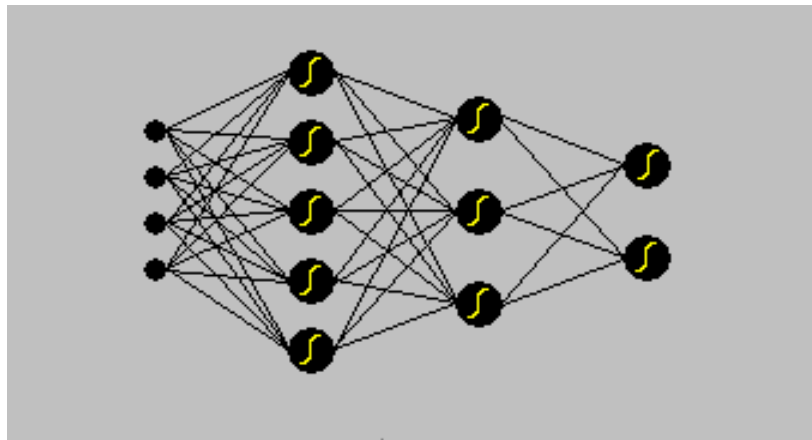
$d(t) = +1$  se entrada for da classe A e  $-1$  se entrada for da classe B.

- Nesta equação  $n$  é uma fração menor que 1 (taxa de aprendizado) e  $d(t)$  é a saída desejada para a correta entrada. Os pesos são inalterados se a rede tomar a decisão correta.

# Perceptrons de Várias Camadas

- Perceptrons de várias camadas são redes feed-forward com uma ou mais camadas entre os nós de entrada e saída . Essas camadas adicionais contêm unidades escondidas ou nós estão diretamente conectados aos nós de entrada e saída.

## Exemplo de uma rede neural perceptron de várias camadas



# Algoritmo Perceptron

- Inicializar pesos( $w$ ), bias( $b$ ) e taxa de aprendizagem ( $\alpha$ )

$$w = 0 \quad b = 0 \quad 0 < \alpha \leq 1$$

- Enquanto condição = falsa faça
  - Para cada par (S:T) faça
    - Ativar unidades de entrada

$$x_i = s_i$$

# Algoritmo Perceptron

- Calcular resposta da unidade de saída

$$y_{in} = b + \sum_i x_i w_i$$

$$y = \begin{cases} 1 & \text{se } y_{in} > \theta \\ 0 & \text{se } -\theta \leq y_{in} \leq \theta \\ -1 & \text{se } y_{in} < -\theta \end{cases}$$

# Algoritmo Perceptron

- Atualizar pesos e bias se um erro ocorreu para este padrão

Se  $y \neq t$

$$w_i(\text{nov}) = w_i(\text{velho}) + \alpha t x_i$$

$$b(\text{nov}) = b(\text{velho}) + \alpha t$$

senão

$$w_i(\text{nov}) = w_i(\text{velho})$$

$$b(\text{nov}) = b(\text{velho})$$

# Algoritmo Perceptron

- Se os pesos modificaram então

Condição = falsa

senão

Condição = verdadeira

# Conclusões quanto ao algoritmo

- Quanto mais padrões de treinamento produzirem respostas corretas menor será o treinamento;
- O threshold não sofre variações;

## Aplicações

Verificar se o ser vivo é quadrúpede ou bípede

- Suposições quanto ao resultado  
Quadrúpede = 1      Bípede = -1

- Vetor de treinamento

	Vetor	Resultado esperado
– Cão	[ 1 -1 1 1 ]	1
– Gato	[ 1 1 1 1 ]	1
– Cavalo	[ 1 1 -1 1 ]	1
– Homem	[ -1 -1 -1 1 ]	-1
– Galinha	[ -1 1 -1 1 ]	-1
– Avestruz	[ 1 -1 1 -1 ]	-1



## Aplicações

Verificar se o ser vivo é quadrúpede ou bípede

- Função de ativação

$$y = \begin{cases} 1 & \text{se } y_{in} > \theta \\ -1 & \text{se } y_{in} \leq \theta \end{cases}$$

- Taxa de aprendizagem

$$\alpha = 1$$

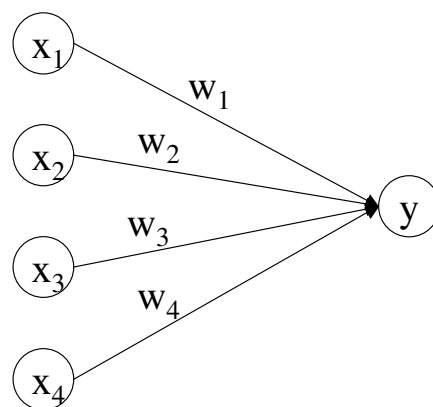
- Threshold

$$\theta = 1$$

## Aplicações

Verificar se o ser vivo é quadrúpede ou bípede

- Arquitetura



# Aplicações

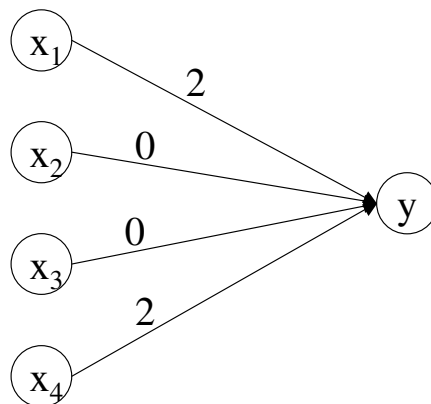
Verificar se o ser vivo é quadrúpede ou bípede

Ser-vivo	$x_1$	$x_2$	$x_3$	$x_4$	$y_{in}$	Saída Real	Saída Desejada	$w_1$	$w_2$	$w_3$	$w_4$
								0	0	0	0
Cão	1	-1	1	1	0	-1	1	1	-1	1	1
Gato	1	1	1	1	2	1	1	1	-1	1	1
Cavalo	1	1	-1	1	-1	-1	1	2	0	0	2
Homem	-1	-1	-1	1	0	-1	-1	2	0	0	2
Galinha	-1	1	-1	1	0	-1	-1	2	0	0	2
Avestruz	1	-1	1	-1	0	-1	-1	2	0	0	2
Cão	1	-1	1	1	4	1	1	2	0	0	2
Gato	1	1	1	1	4	1	1	2	0	0	2
Cavalo	1	1	-1	1	4	1	1	2	0	0	2

# Aplicações

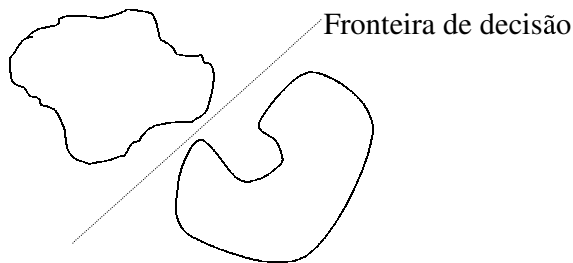
Verificar se o ser vivo é quadrúpede ou bípede

- Rede Neural Treinada

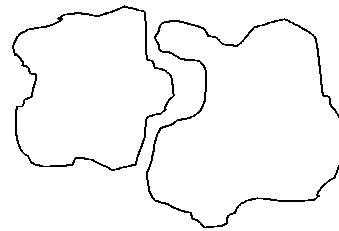


# Teorema de Convergência

- Garante que a rede será treinada
- Número finito de iterações
- O problema deve ser linearmente separável



Padrões linearmente separáveis



Padrões não linearmente separáveis

# Teorema de Convergência



# Teorema de Convergência

