



Sistemas Inteligentes

UNIDADE 3 – Redes Neurais Artificiais (Perceptron)

Prof. Ivan Nunes da Silva



1. Rede Perceptron

Aspectos históricos do Perceptron

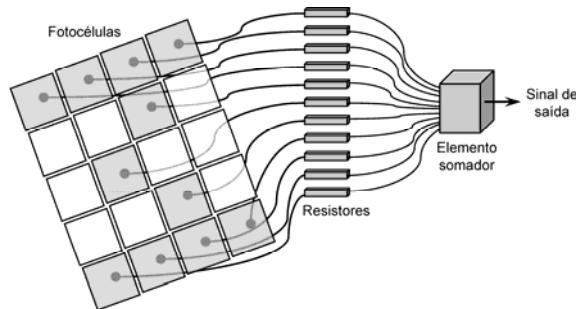
- É a forma mais simples de configuração de uma rede neural artificial (idealizada por Rosenblatt, 1958).
- Constituída de apenas uma camada, tendo-se ainda somente um neurônio nesta única camada.
- Seu propósito inicial era implementar um modelo computacional inspirado na retina, objetivando-se então um elemento de percepção eletrônica de sinais.
- Suas aplicações consistiam de identificar padrões geométricos.



1. Rede Perceptron

Concepção inicial do elemento Perceptron

- Modelo ilustrativo do *Perceptron* para reconhecimento de padrões:



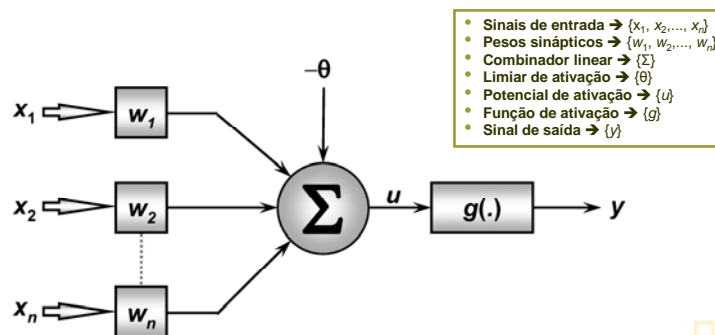
- 1) Sinais elétricos advindos de fotocélulas mapeando padrões geométricos eram ponderados por resistores sintonizáveis.
- 2) Os resistores eram ajustados durante o processo de treinamento.
- 3) Um somador efetuava a composição de todos os sinais.
- 4) Em consequência, o *Perceptron* poderia reconhecer diversos padrões geométricos, tais como letras e números.

3

2. Arquitetura do Perceptron

Aspectos topológicos

- Ilustração da rede *Perceptron*:



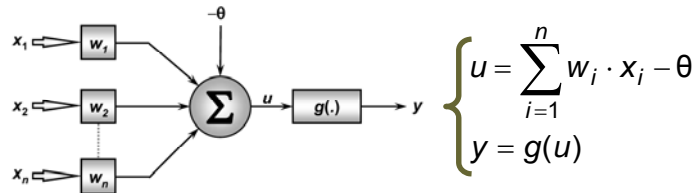
- 1) Embora seja uma rede simples, o *Perceptron* teve potencial de atrair, quando de sua proposição, diversos pesquisadores que aspiravam investigar essa promissora área de pesquisa.
- 2) Recebeu ainda especial atenção da comunidade científica que também trabalhava com inteligência artificial.
- 3) O *Perceptron* é tipicamente utilizado em problemas de “Classificação de Padrões”.

4

3. Princípio de Funcionamento

Resumo do funcionamento do Perceptron

● Passos para obtenção da resposta (saída):



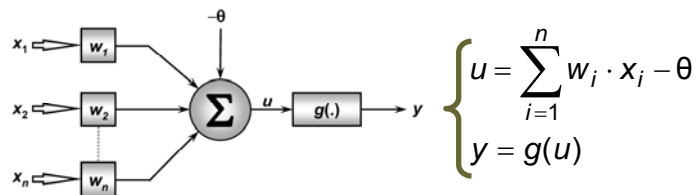
- 1) Apresentação de um conjunto de valores que representam as variáveis de entrada do neurônio.
- 2) Multiplicação de cada entrada do neurônio pelo seu respectivo peso sináptico.
- 3) Obtenção do potencial de ativação produzido pela soma ponderada dos sinais de entrada, subtraindo-se o limiar de ativação.
- 4) Aplicação de uma função de ativação apropriada, tendo-se como objetivo limitar a saída do neurônio.
- 5) Compilação da saída a partir da aplicação da função de ativação neural em relação ao seu potencial de ativação.

5

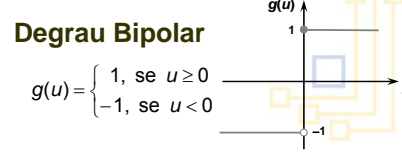
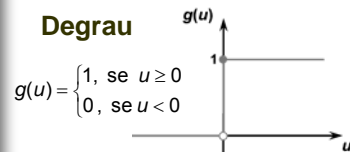
3. Princípio de Funcionamento

Aplicabilidade em classificação de padrões

● Aspectos de aplicabilidade:



- 1) Tipicamente, devido às suas características estruturais, as funções de ativação usadas no **Perceptron** são a “**degrau**” ou “**degrau bipolar**”.
- 2) Assim, tem-se apenas “**duas possibilidades**” de valores a serem produzidos pela sua saída, ou seja, valor 0 ou 1 (para a função de ativação Degrau), ou ainda, valor -1 ou 1 (para a função Degrau Bipolar).

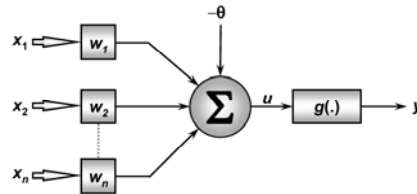


6

3. Princípio de Funcionamento

Parâmetros característicos do Perceptron

● Aspectos de aplicabilidade:



- Sinais de entrada $\rightarrow \{x_1, x_2, \dots, x_n\}$
- Pesos sinápticos $\rightarrow \{w_1, w_2, \dots, w_n\}$
- Combinador linear $\rightarrow \{\Sigma\}$
- Limiar de ativação $\rightarrow \{\theta\}$
- Potencial de ativação $\rightarrow \{u\}$
- Função de ativação $\rightarrow \{g\}$
- Sinal de saída $\rightarrow \{y\}$

Parâmetro	Variável Representativa	Tipo Característico
Entradas	x_i (i -ésima entrada)	Reais ou Binária (advindas externamente)
Pesos Sinápticos	w_i (associado a x_i)	Reais (iniciados aleatoriamente)
Limiar	θ	Real (iniciado aleatoriamente)
Saída	y	Binária
Função de Ativação	$g(\cdot)$	Degrau ou Degrau Bipolar
Processo de Treinamento	-----	Supervisionado
Regra de Aprendizado	-----	Regra de Hebb

7

3. Princípio de Funcionamento

Processo de treinamento supervisionado

● Aspectos do treinamento supervisionado:

Parâmetro	Variável Representativa	Tipo Característico
Entradas	x_i (i -ésima entrada)	Reais ou Binária (advindas externamente)
Pesos Sinápticos	w_i (associado a x_i)	Reais (iniciados aleatoriamente)
Limiar	θ	Real (iniciado aleatoriamente)
Saída	y	Binária
Função de Ativação	$g(\cdot)$	Degrau ou Degrau Bipolar
Processo de Treinamento	-----	Supervisionado
Regra de Aprendizado	-----	Regra de Hebb

- 1) Conforme tabela apresentada, o ajuste dos pesos e limiar do **Perceptron** é efetuado utilizando processo de treinamento "**Supervisionado**".
- 2) Então, para cada amostra dos sinais de entrada se tem a respectiva saída (resposta) desejada.
- 3) Como o **Perceptron** é tipicamente usado em problemas de classificação de padrões, a sua saída pode assumir somente dois valores possíveis.
- 4) Assim, cada um de tais valores será associado a uma das "**duas classes**" que o **Perceptron** estará identificando.

8

3. Princípio de Funcionamento

Mapeamento de problemas de classificação de padrões

● Aspectos de aplicabilidade:

- 1) Portanto, para problemas de classificação dos sinais de entrada, tem-se então duas classes possíveis, denominadas de **Classe A** e **Classe B**;
- 2) Paralelamente, como se tem também “duas possibilidades” de valores a serem produzidos na saída do **Perceptron**, tem-se as seguintes associações:

Degrau

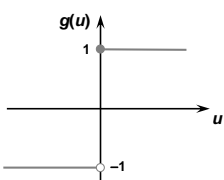
$$g(u) = \begin{cases} 1, & \text{se } u \geq 0 \\ 0, & \text{se } u < 0 \end{cases}$$



$$g(u) = \begin{cases} 1, & \text{se } u \geq 0 \Rightarrow \mathbf{x \in Classe A} \\ 0, & \text{se } u < 0 \Rightarrow \mathbf{x \in Classe B} \end{cases}$$

Degrau Bipolar

$$g(u) = \begin{cases} 1, & \text{se } u \geq 0 \\ -1, & \text{se } u < 0 \end{cases}$$



$$g(u) = \begin{cases} 1, & \text{se } u \geq 0 \Rightarrow \mathbf{x \in Classe A} \\ -1, & \text{se } u < 0 \Rightarrow \mathbf{x \in Classe B} \end{cases}$$

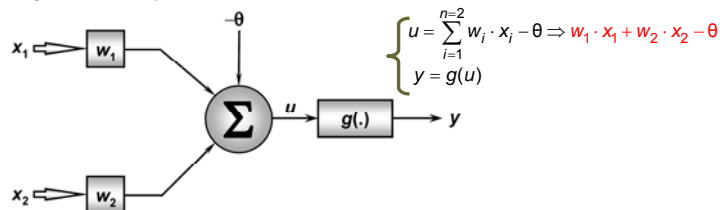
9

3. Princípio de Funcionamento

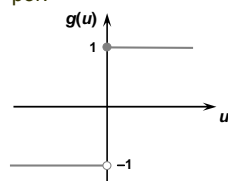
O quê acontece internamente no Perceptron?

● Análise matemática do Perceptron:

- Para mostrar o quê ocorre internamente, assume-se aqui um **Perceptron** com apenas duas entradas:



- Usando como ativação a função sinal, a saída do **Perceptron** será dada por:



$$y = g(u) = \begin{cases} 1, & \text{se } u \geq 0 \\ -1, & \text{se } u < 0 \end{cases}$$

$$y = g(u) = \begin{cases} 1, & \text{se } \sum w_i \cdot x_i - \theta \geq 0 \Leftrightarrow w_1 \cdot x_1 + w_2 \cdot x_2 - \theta \geq 0 \\ -1, & \text{se } \sum w_i \cdot x_i - \theta < 0 \Leftrightarrow w_1 \cdot x_1 + w_2 \cdot x_2 - \theta < 0 \end{cases}$$

Conclusão: O **Perceptron** é um **classificador linear** que pode ser usado para classificar duas classes **linearmente separáveis**.

10

3. Princípio de Funcionamento

O Perceptron como classificador de padrões (I)

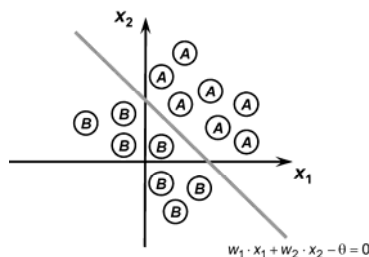
● Classificação de padrões usando o Perceptron:

- Da conclusão do slide anterior, observam-se que as desigualdades são representadas por uma expressão de primeiro grau (linear).
- A fronteira de decisão para esta instância (**Perceptron** de duas entradas) será então uma reta cuja equação é definida por:

$$w_1 \cdot x_1 + w_2 \cdot x_2 - \theta = 0$$

$$y = g(u) = \begin{cases} 1, & \text{se } \sum w_i \cdot x_i - \theta \geq 0 \Leftrightarrow w_1 \cdot x_1 + w_2 \cdot x_2 - \theta \geq 0 \\ -1, & \text{se } \sum w_i \cdot x_i - \theta < 0 \Leftrightarrow w_1 \cdot x_1 + w_2 \cdot x_2 - \theta < 0 \end{cases}$$

- Assim, tem-se a seguinte representação gráfica para a saída do **Perceptron**:



$$g(u) = \begin{cases} 1, & \text{se } u \geq 0 \Rightarrow x \in \text{Classe A} \\ -1, & \text{se } u < 0 \Rightarrow x \in \text{Classe B} \end{cases}$$

- Em suma, para a circunstância ao lado, o **Perceptron** consegue então dividir duas classes linearmente separáveis
- A saída do mesmo for 1 significa que os padrões (**Classe A**) estão localizados acima da fronteira (reta) de separação; caso contrário, quando a saída for -1 indica que os padrões (**Classe B**) estão abaixo desta fronteira.

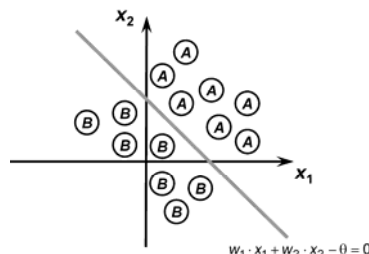
11

3. Princípio de Funcionamento

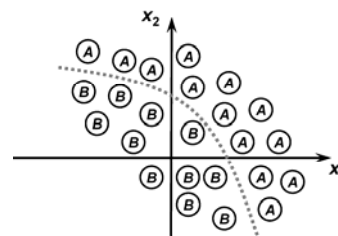
O Perceptron como classificador de padrões (II)

● Aspectos de classificação de padrões:

- Se o **Perceptron** fosse constituído de três entradas (três dimensões), a fronteira de separação seria representada por um plano.
- Se o **Perceptron** fosse constituído de quatro ou mais entradas, suas fronteiras seriam hiperplanos.
- **Conclusão:** A condição necessária para que o **Perceptron** de camada simples possa ser utilizado como um classificador de padrões é que as classes do problema a ser mapeado sejam **linearmente separáveis**.



Problema linearmente separável



Problema não-linearmente separável

12

4. Processo de Treinamento

Regra de aprendizado de Hebb

● Aspectos do processo de treinamento:

- O ajuste dos pesos $\{w_i\}$ e limiar $\{\theta\}$ do **Perceptron**, visando-se propósitos de classificação de padrões que podem pertencer a uma das duas únicas classes possíveis, é feito por meio da **regra de aprendizado de Hebb**.
- Se a saída produzida pelo **Perceptron** está coincidente com a saída desejada, os seus pesos e limiares são então **mantidos inalterados**.
- Se a saída produzida pelo **Perceptron** é diferente do valor desejado, os pesos sinápticos e limiar são então **proporcionalmente ajustados** frente aos valores de suas entradas.
- Este processo é repetido, sequencialmente, para todas as amostras de treinamento, até que a saída produzida pelo **Perceptron** seja similar à saída desejada de cada amostra.

➤ Em termos matemáticos, tem-se:

$$\begin{cases} w_i^{\text{atual}} = w_i^{\text{anterior}} + \eta \cdot (d^{(k)} - y) \cdot x_i^{(k)} \\ \theta^{\text{atual}} = \theta^{\text{anterior}} + \eta \cdot (d^{(k)} - y) \cdot (-1) \end{cases}$$

➤ A taxa de aprendizagem $\{\eta\}$ exprime o quão rápido o processo de treinamento da rede estará sendo conduzido rumo à sua convergência.

➤ w_i são os pesos sinápticos.

➤ θ é limiar do neurônio.

➤ $\mathbf{x}^{(k)}$ é o vetor contendo a k -ésima amostra de treinamento

➤ $d^{(k)}$ é saída desejada para a k -ésima amostra de treinamento.

➤ y é a saída do **Perceptron**.

➤ η é a taxa de aprendizagem da rede.

13

4. Processo de Treinamento

Aspectos de implementação computacional (I)

● Adequações algorítmicas:

➤ Em termos de implementação, torna-se mais conveniente tratar as expressões anteriores em sua forma vetorial.

➤ Como a mesma regra de ajuste é aplicada tanto para os pesos w_i como para o limiar θ , pode-se então inserir θ dentro do vetor de pesos:

$$\mathbf{w} = [\theta \ w_1 \ w_2 \dots w_n]^T$$

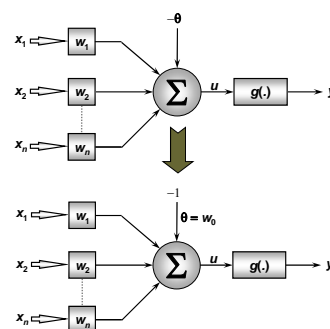
➤ De fato, o valor do limiar é também uma variável a ser ajustada a fim de se realizar o treinamento do **Perceptron**.

➤ Portanto, tem-se:

$$\begin{cases} w_i^{\text{atual}} = w_i^{\text{anterior}} + \eta \cdot (d^{(k)} - y) \cdot x_i^{(k)} \\ \theta^{\text{atual}} = \theta^{\text{anterior}} + \eta \cdot (d^{(k)} - y) \cdot (-1) \end{cases}$$

➤ Formato de cada amostra de treinamento:

$$\mathbf{x}^{(k)} = [-1 \ x_1^{(k)} \ x_2^{(k)} \dots x_n^{(k)}]^T$$



notação
vetorial

$$\mathbf{w}^{\text{atual}} = \mathbf{w}^{\text{anterior}} + \eta \cdot (d^{(k)} - y) \cdot \mathbf{x}^{(k)}$$

notação
algorítmica

$$\mathbf{w} \leftarrow \mathbf{w} + \eta \cdot (d^{(k)} - y) \cdot \mathbf{x}^{(k)}$$

14

4. Processo de Treinamento

Aspectos de implementação computacional (II)

● Montagem de conjuntos de treinamento:

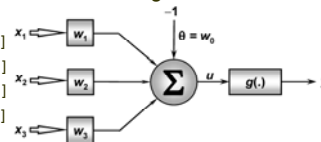
- Supõe-se que um problema a ser mapeado pelo **Perceptron** tenha três entradas $\{x_1, x_2, x_3\}$, conforme a figura ao lado (abaixo).
- Assume-se que se tem quatro amostras, constituída dos seguintes valores de entrada:

Amostra 1 → Entrada: $[0, 1, 0, 4, 0, 7]^T$ → Saída desejada: $[1]$

Amostra 2 → Entrada: $[0, 3, 0, 7, 0, 2]^T$ → Saída desejada: $[-1]$

Amostra 3 → Entrada: $[0, 6, 0, 9, 0, 8]^T$ → Saída desejada: $[-1]$

Amostra 4 → Entrada: $[0, 5, 0, 7, 0, 1]^T$ → Saída desejada: $[1]$



- Então, pode-se converter tais sinais para que os mesmos possam ser usados no treinamento do **Perceptron**:

Conjunto de treinamento

$x^{(1)} = [-1, 0, 1, 0, 4, 0, 7]^T \rightarrow \text{com } d^{(1)} = 1$
 $x^{(2)} = [-1, 0, 3, 0, 7, 0, 2]^T \rightarrow \text{com } d^{(2)} = -1$
 $x^{(3)} = [-1, 0, 6, 0, 9, 0, 8]^T \rightarrow \text{com } d^{(3)} = -1$
 $x^{(4)} = [-1, 0, 5, 0, 7, 0, 1]^T \rightarrow \text{com } d^{(4)} = 1$

forma matricial

$$\Omega(x) = \begin{bmatrix} x_0 & x^{(1)} & x^{(2)} & x^{(3)} & x^{(4)} \\ x_1 & -1 & -1 & -1 & -1 \\ x_2 & 0,1 & 0,3 & 0,6 & 0,5 \\ x_3 & 0,4 & 0,7 & 0,9 & 0,7 \\ x_4 & 0,7 & 0,2 & 0,8 & 0,1 \end{bmatrix}$$

- Geralmente, as amostras de treinamento são disponibilizadas em sua forma matricial (por meio de arquivo texto ou planilha).

15

4. Processo de Treinamento

Algoritmo de aprendizagem (fase de treinamento)

● Pseudocódigo para fase de treinamento:

Início {Algoritmo Perceptron – Fase de Treinamento}

- <1> Obter o conjunto de amostras de treinamento $\{x^{(k)}\}$;
- <2> Associar a saída desejada $\{d^{(k)}\}$ para cada amostra obtida;
- <3> Iniciar o vetor w com valores aleatórios pequenos;
- <4> Especificar a taxa de aprendizagem $\{\eta\}$;
- <5> Iniciar o contador de número de épocas $\{\text{época} \leftarrow 0\}$;
- <6> Repetir as instruções:
 - <6.1> erro \leftarrow "inexiste";
 - <6.2> Para todas as amostras de treinamento $\{x^{(k)}, d^{(k)}\}$, fazer:
 - <6.2.1> $u \leftarrow w^T \cdot x^{(k)}$;
 - <6.2.2> $y \leftarrow \text{sin}(u)$;
 - <6.2.3> Se $y \neq d^{(k)}$
 - <6.2.3.1> Então $\begin{cases} w \leftarrow w + \eta \cdot (d^{(k)} - y) \cdot x^{(k)} \\ \text{erro} \leftarrow \text{"existe"} \end{cases}$
 - <6.3> época \leftarrow época + 1;
- Até que: erro \leftarrow "inexiste"

Fim {Algoritmo Perceptron – Fase de Treinamento}

Época de treinamento → É cada apresentação completa de todas as amostras pertencentes ao subconjunto de treinamento, visando, sobretudo, o ajuste dos pesos sinápticos e limiares de seus neurônios.

16

4. Processo de Treinamento

Algoritmo de aprendizagem (fase de operação)

● Pseudocódigo para fase de operação:

Início {Algoritmo *Perceptron* – Fase de Operação}

{
 <1> Obter uma amostra a ser classificada { x };
 <2> Utilizar o vetor w ajustado durante o treinamento;
 <3> Executar as seguintes instruções:
 <3.1> $u \leftarrow w^T \cdot x$;
 <3.2> $y \leftarrow \text{sinal}(u)$;
 <3.3> Se $y = -1$
 <3.3.1> Então: amostra $x \in \{\text{Classe A}\}$
 <3.4> Se $y = 1$
 <3.4.1> Então: amostra $x \in \{\text{Classe B}\}$
}

Fim {Algoritmo *Perceptron* – Fase de Operação}

Obs. 1 → A “Fase de Operação” é usada somente após a fase de treinamento, pois aqui a rede já está apta para ser usada no processo.

Obs. 2 → A “Fase de Operação” é então utilizada para realizar a tarefa de classificação de padrões frente às novas amostras que serão apresentadas em suas entradas.

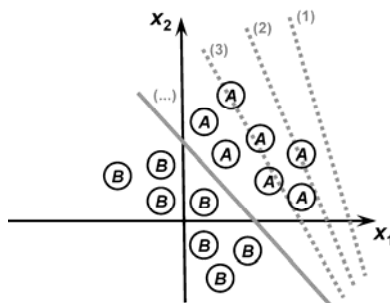
17

4. Processo de Treinamento

Aspectos de convergência (I)

● Ilustração do processo de convergência:

- Processo de treinamento tende a mover continuamente o hiperplano de classificação até que seja alcançada uma **fronteira de separação** que permite dividir as duas classes.
- Como exemplo, seja uma rede composta de apenas duas entradas $\{x_1 \text{ e } x_2\}$.



- Após a primeira época de treinamento (1), constata-se que o hiperplano está ainda bem longínquo da fronteira de separabilidade das classes.
- A distância tende a ir cada vez mais decrescendo na medida em que se transcorre as épocas de treinamento.
- Quando o *Perceptron* já estiver convergido, isto estará então significando que tal fronteira foi finalmente alcançada.

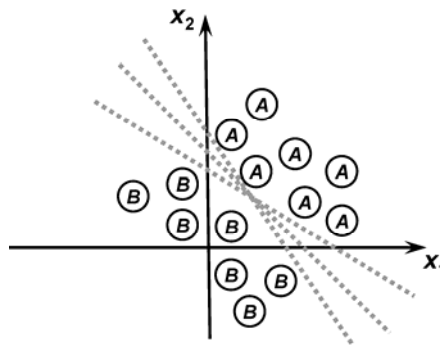
18

4. Processo de Treinamento

Aspectos de convergência (II)

● Região de separabilidade:

- A reta de separabilidade a ser produzida após o treinamento do **Perceptron** não é única.
- O número de épocas pode também variar de treinamento para treinamento.



19

5. Questões do Perceptron

Reflexões, observações e aspectos práticos

- 1) O que acontece com o algoritmo do **Perceptron** se o problema for não-linearmente separável?
- 2) Como poderia ser tratado a questão acima no algoritmo do **Perceptron**?
- 3) Qual seria os eventuais inconvenientes de se usar valores muito grandes para a taxa de aprendizagem? E para valores muito pequenos?
- 4) Dois projetistas estão implementando o mesmo problema de classificação de padrões por meio do **Perceptron**. Comente se, após o processo de treinamento, o vetor de pesos final terá que ser o mesmo em ambos os projetos.

20