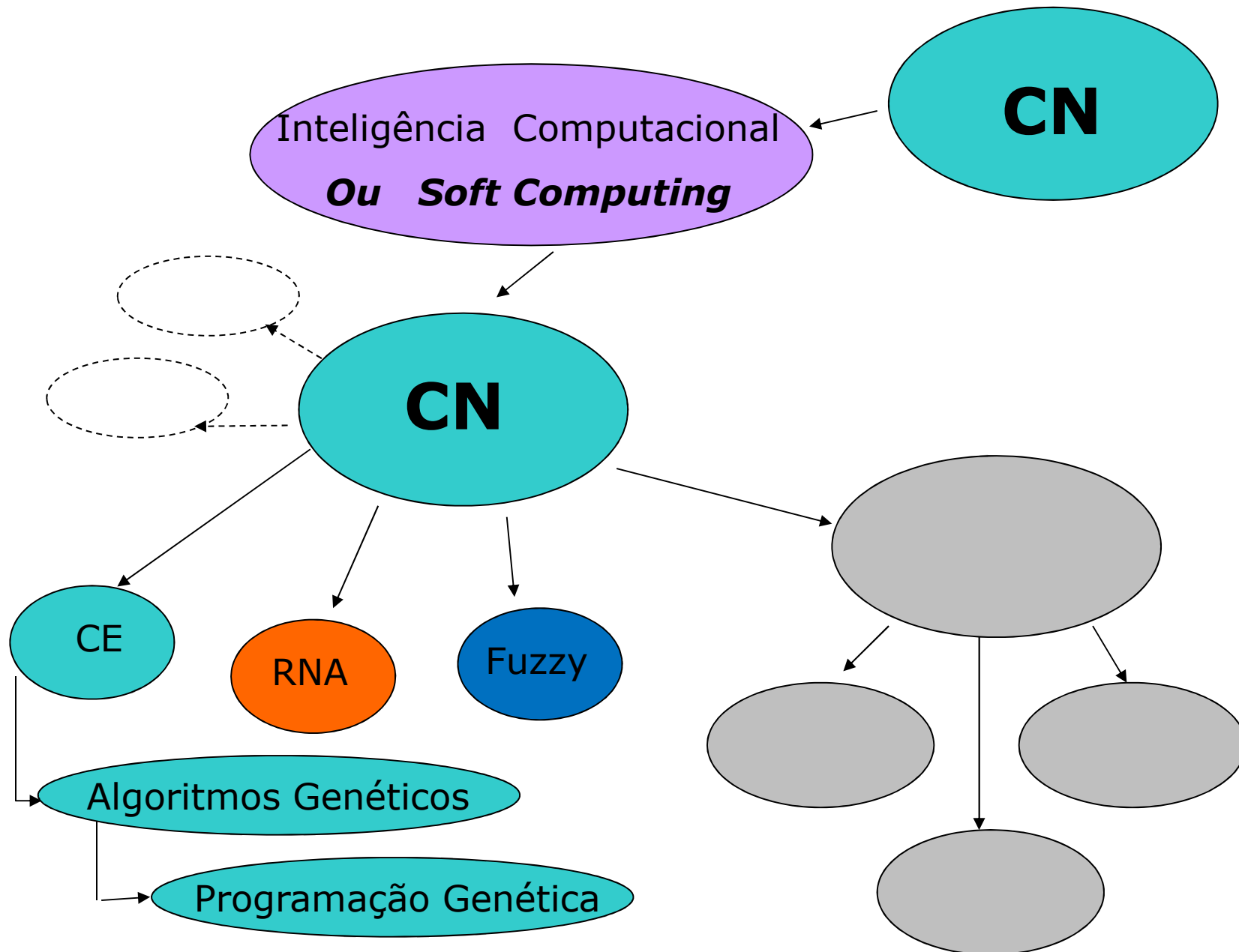


Inteligência Artificial

Redes Neurais

(Computação Natural - Soft Computing)

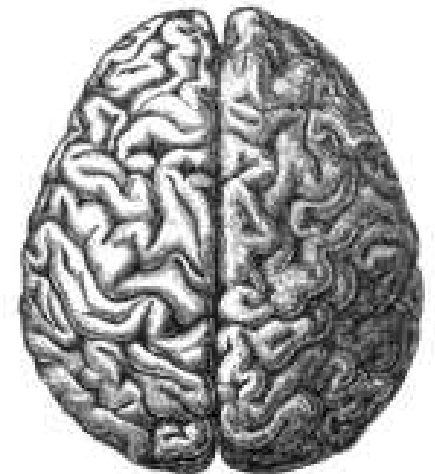
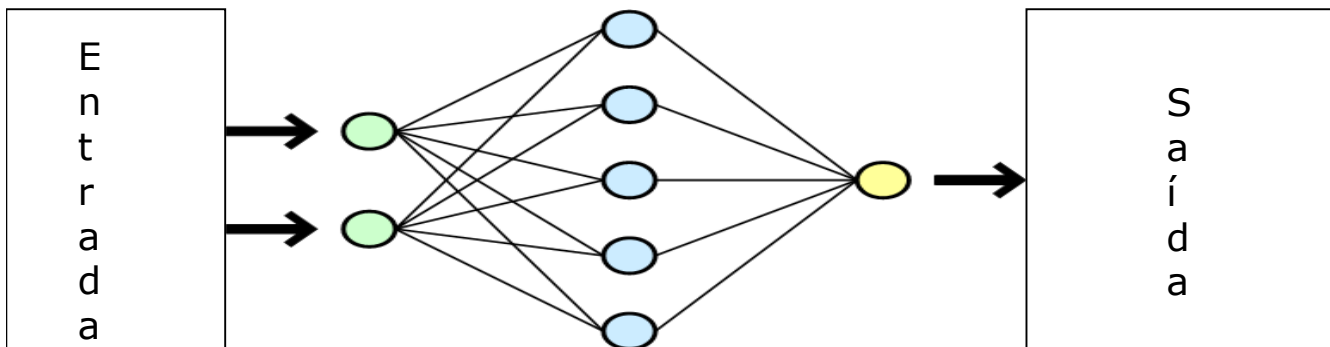
Contexto de RNA



Redes Neurais Artificiais

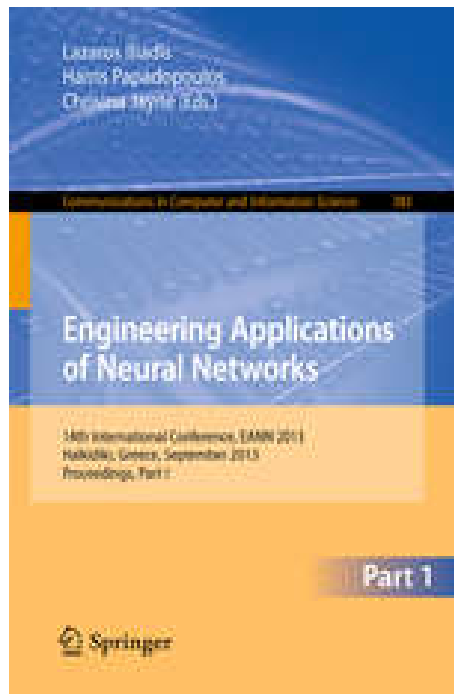
Redes Neurais (RN): Inspiradas no Funcionamento do Cérebro Humano

Uma coleção **massivamente paralela** de **unidades** de processamento pequenas e **simples**, onde as **interligações** formam a maior parte da “**inteligência**” da rede



Redes Neurais: Aplicações

Myriam Regattieri Degado



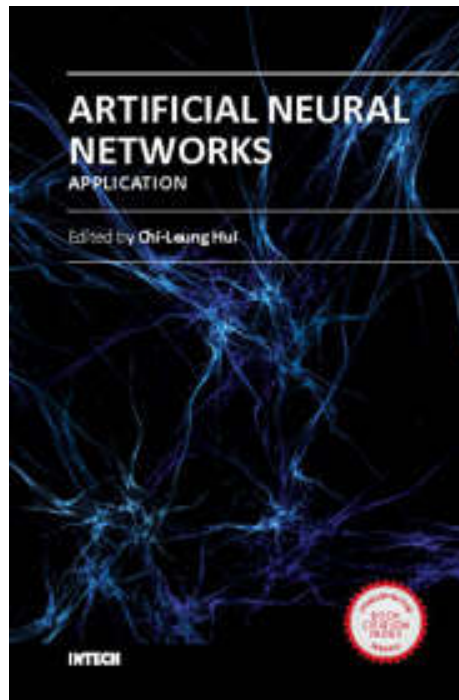
Livro RN em Aplicações de Eng:

Lazaros Iliadis, Harris Papadopoulos, Chrisina Jayne. *Engineering Applications of Neural Networks*. Springer 2013

Myriam Delgado
Dainf/CPGEI/UTFPR

Redes Neurais: Aplicações

Myriam Regattieri Degado



Livro RN em Aplicações Diversas:

Chi Leung Patrick Hui.

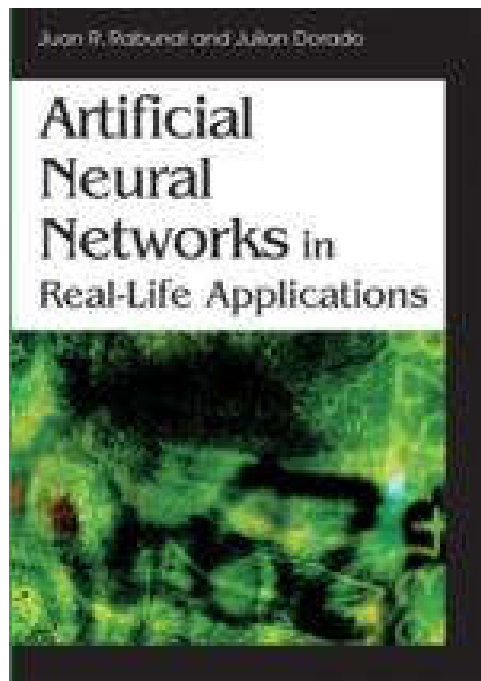
Artificial Neural Networks - Application.

InTech 2011

Myriam Delgado
Dainf/CPGEI/UTFPR

Redes Neurais: Aplicações

Myriam Regattieri Degado



Livro RN + Computação Evolucionária:

Juan R. Rabuñal; Julian Dorado.
Artificial Neural Networks in Real-Life Applications. IGI Global 2006.

Myriam Delgado
Dainf/CPGEI/UTFPR

Redes Neurais: Aplicações

Myriam Regattieri Degado



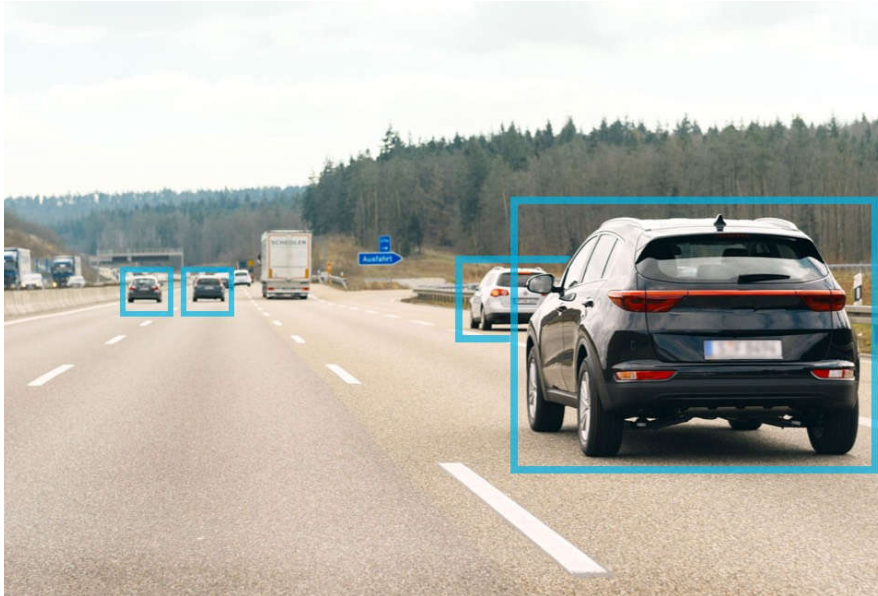
Software Prisma:

Edita fotos com o auxílio de Redes Neurais Artificiais transformando-as em arte

Myriam Delgado
Dainf/CPGEI/UTFPR

Redes Neurais: Aplicações

Myriam Regattieri Degado



Processamento de Imagens:

Reconhecimento de Objetos em
Imagens

Myriam Delgado
Dainf/CPGEI/UTFPR

Redes Neurais: Aplicações

Myriam Regattieri Degado



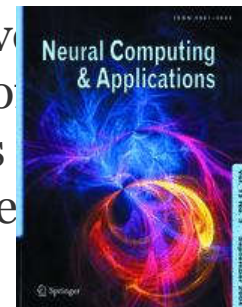
Interface Humano-computador:

Detecção de face

Reconhecimento de expressões faciais

Rastreamento do corpo humano

A Literature Survey
on Applications of
Neural Networks
Human-Computer
Interaction



Myriam Delgado
Dainf/CPGEI/UTFPR

Redes Neurais: Aplicações

Myriam Regattieri Degado



Cuidados com a saúde:
Processamento de sinais
biomédicos:

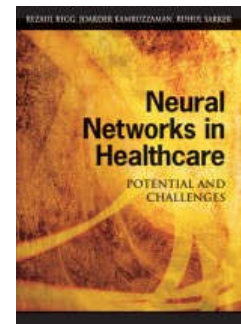
Cardiologia

Ginecologia

Controle Neuro-muscular

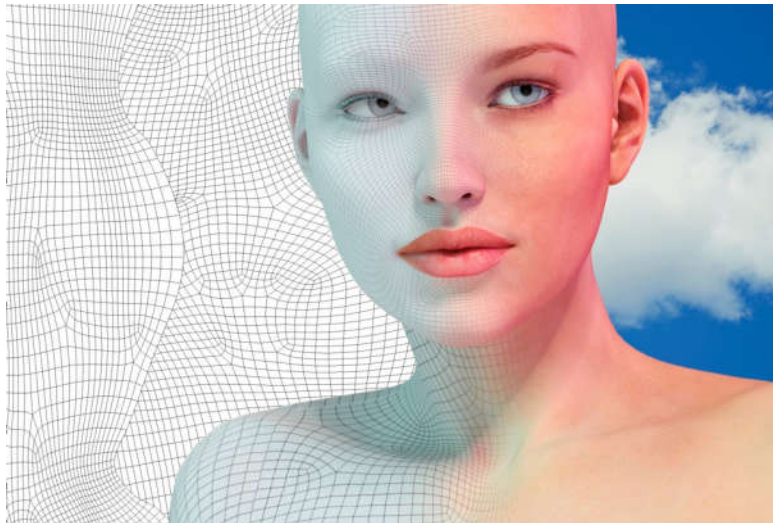
Myriam Delgado
Dainf/CPGEI/UTFPR

A Survey on
Various
Applications of
Artificial
Neural
Networks in
Selected Fields
of Healthcare



Redes Neurais: Aplicações

Myriam Regattieri Degado



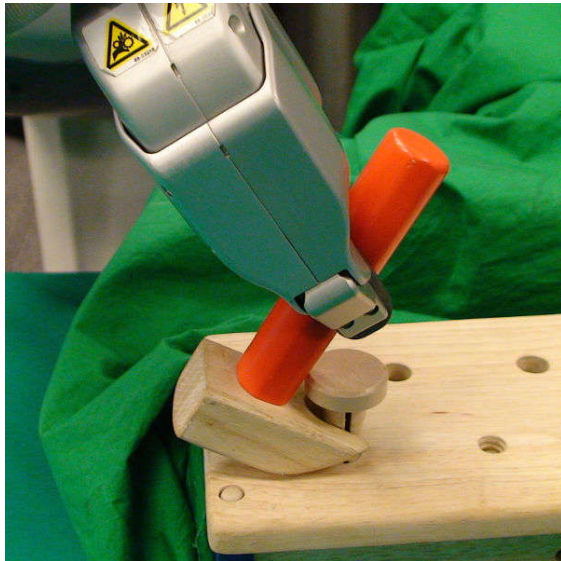
Realidade virtual:

Geração de Face

Myriam Delgado
Dainf/CPGEI/UTFPR

Redes Neurais: Aplicações

Myriam Regattieri Degado



Robótica:

Aprendizado
profundo (deep
learning)

**A Robot Learns To Do Things
Using A Deep Neural Network**

Written by Mike James

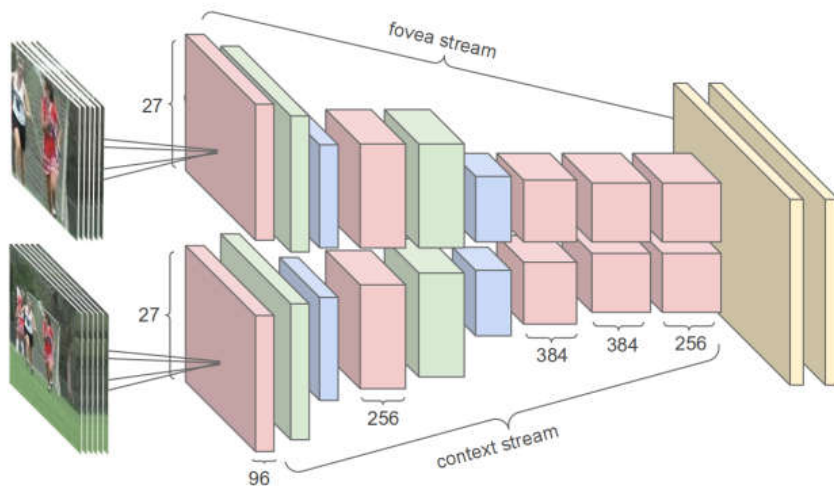
Wednesday, 27 May 2015

Myriam Delgado
Dainf/CPGEI/UTFPR

[http://www.i-
programmer.info/news/105-artificial-
intelligence/8619-a-robot-learns-to-
do-things-using-a-deep-neural-
network.html](http://www.i-programmer.info/news/105-artificial-intelligence/8619-a-robot-learns-to-do-things-using-a-deep-neural-network.html)

Redes Neurais: Aplicações

Myriam Regattieri Degado



Google search e imagens:
Redes Convolucionais: entradas
são frames de imagens

Large-scale Video Classification with Convolutional Neural Networks

Andrej Karpathy^{1,2}

karpathy@cs.stanford.edu

Thomas Leung¹

leungt@google.com

George Toderici¹

gtoderici@google.com

Rahul Sukthankar¹

sukthankar@google.com

Sanketh Shetty¹

sanketh@google.com

Li Fei-Fei²

feifeili@cs.stanford.edu

¹Google Research

²Computer Science Department, Stanford University

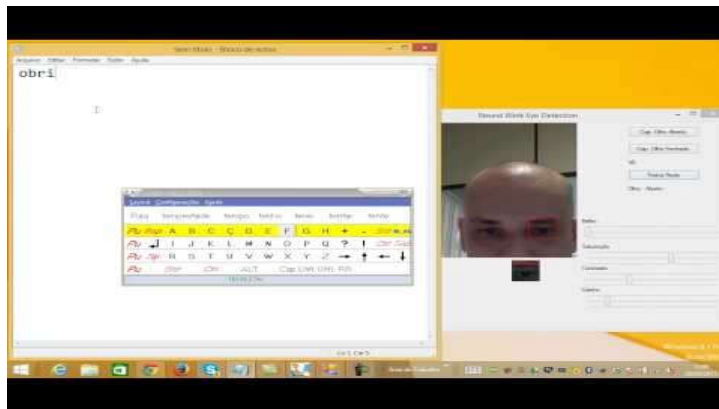
<http://cs.stanford.edu/people/karpathy/deepvideo>

Myriam Delgado
Dainf/CPGEI/UTFPR

Redes Neurais: Aplicações

Myriam Regattieri Degado

<https://www.youtube.com/watch?v=2Syw5l553GY>



Software ETM:

Para pessoas com necessidades especiais. Controla o teclado com o olho. **RN MLP** detecta o momento da **piscada** e aciona o teclado.

Myriam Delgado
Dainf/CPGEI/UTFPR

Redes Neurais são populares

Myriam Regattieri Degado

[https://www.journals.elsevier.com/
neural-networks/most-downloaded-
articles](https://www.journals.elsevier.com/neural-networks/most-downloaded-articles)

Deep learning in neural networks: An overview

January 2015

Jürgen Schmidhube

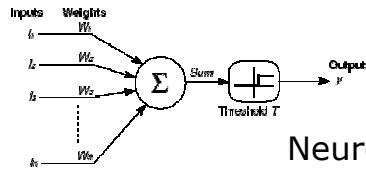
Trends in extreme learning machines: A review

January 2015

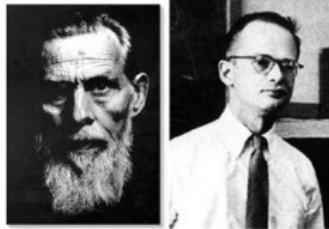
Gao Huang | Guang-Bin Huang | Shiji Song | Keyou You

Myriam Delgado
Dainf/CPGEI/UTFPR

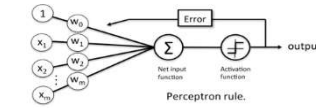
RNAs: Linha do Tempo



Neurônio MCP



1943

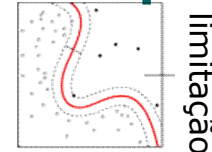


Perceptron



Rosenblatt

1958 (1957)

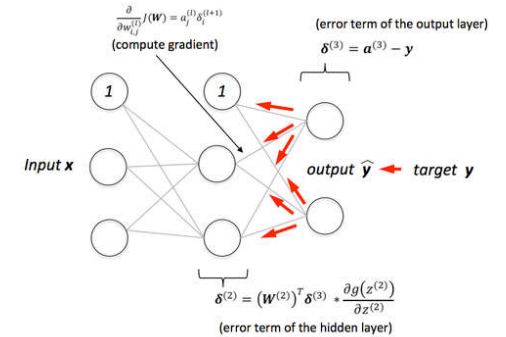


limitação



Minsky e Papert

1969

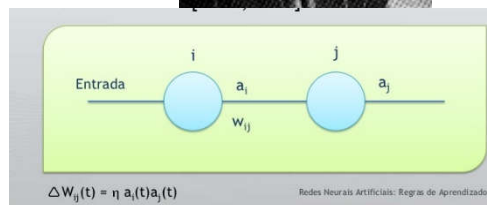


Backpropagation

1986

1949

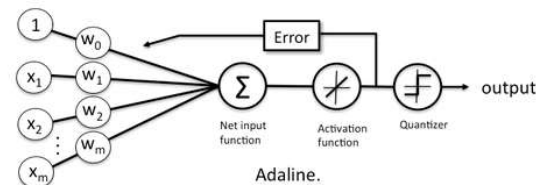
Regra de Hebb



1960

Regra Delta (Widrow-Hoff)

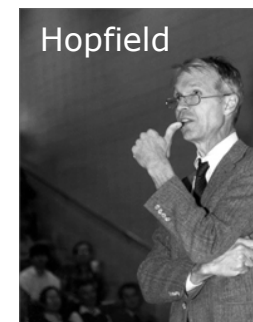
$$w_{ij}(\text{new}) = w_{ij}(\text{old}) + \alpha(t_i - y_{in_i})x_j$$



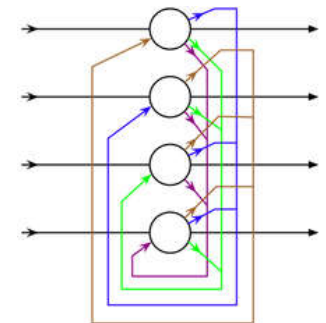
Adaline.

Prof. Myriam Delgado
Redes Neurais - UTFPR

1982

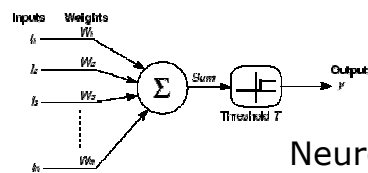


Hopfield

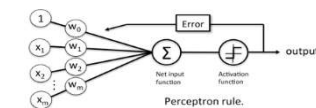


RN associativas

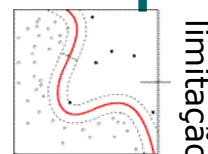
RNAs: Linha do Tempo



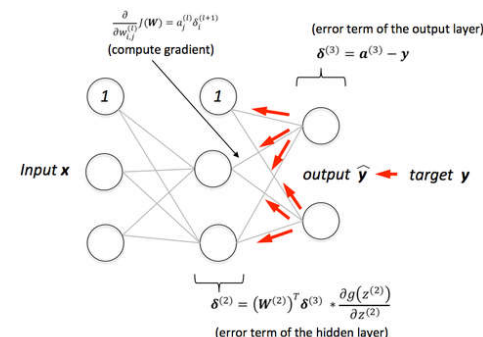
Neurônio MCP



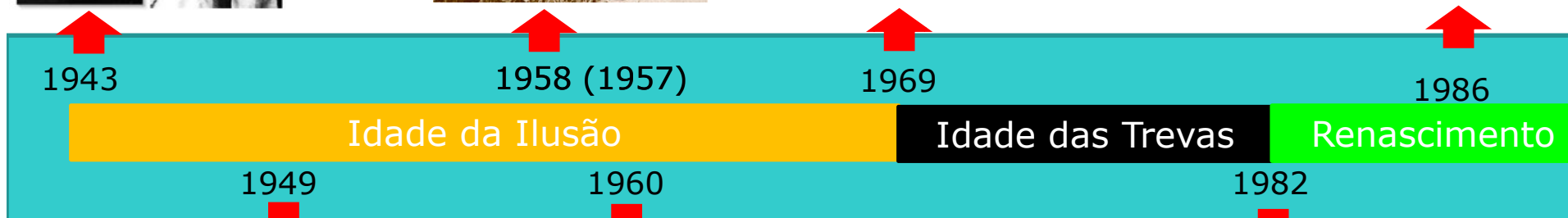
Perceptron



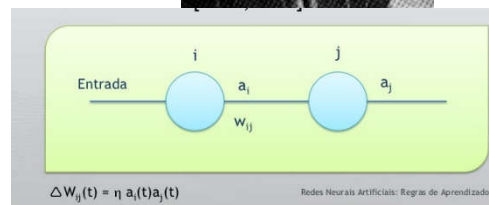
Minsky e Papert



Backpropagation

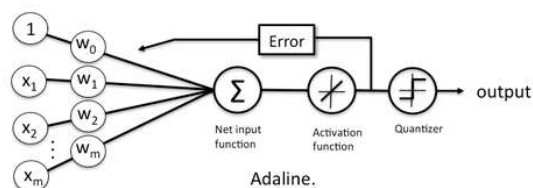


Regra de Hebb



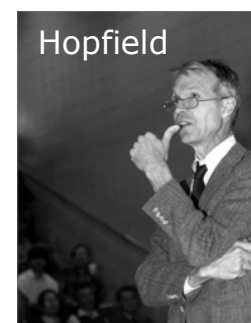
Regra Delta (Widrow-Hoff)

$$w_{ij}(\text{new}) = w_{ij}(\text{old}) + \alpha(t_i - y_{in_i})x_j$$

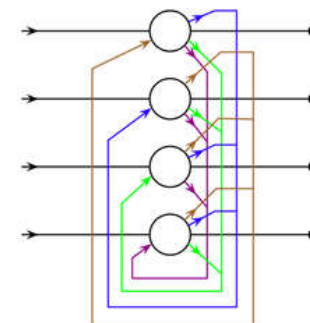


Adaline.

Prof. Myriam Delgado
Redes Neurais - UTFPR

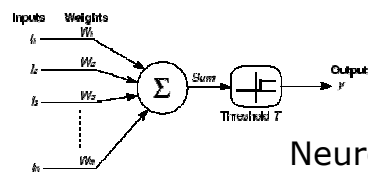


Hopfield

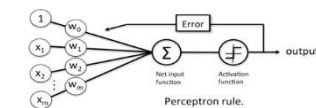


RN associativas

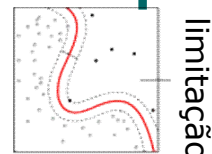
RNAs: Linha do Tempo



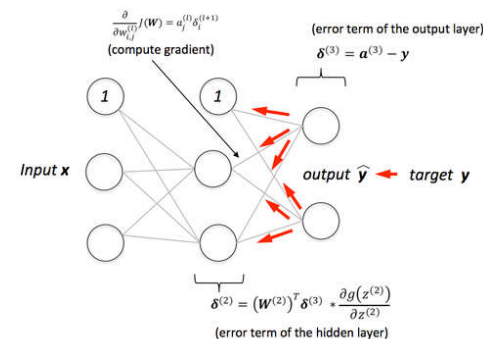
Neurônio MCP



Perceptron



Minsky e Papert



Backpropagation

1943

1958 (1957)

1969

1986

Idade da Ilusão

Idade das Trevas

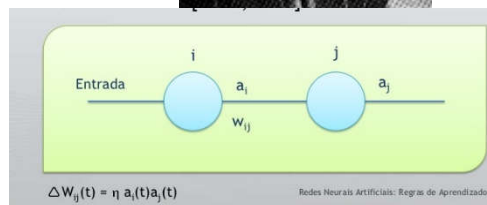
Renascimento

1949

1960

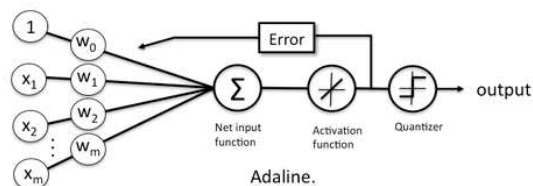
1982

Regra de Hebb



Regra Delta (Widrow-Hoff)

$$w_{ij}(\text{new}) = w_{ij}(\text{old}) + \alpha(t_i - y_{in_i})x_j$$

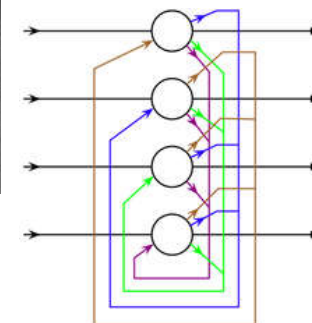


Adaline.

Prof. Myriam Delgado
Redes Neurais - UTFPR



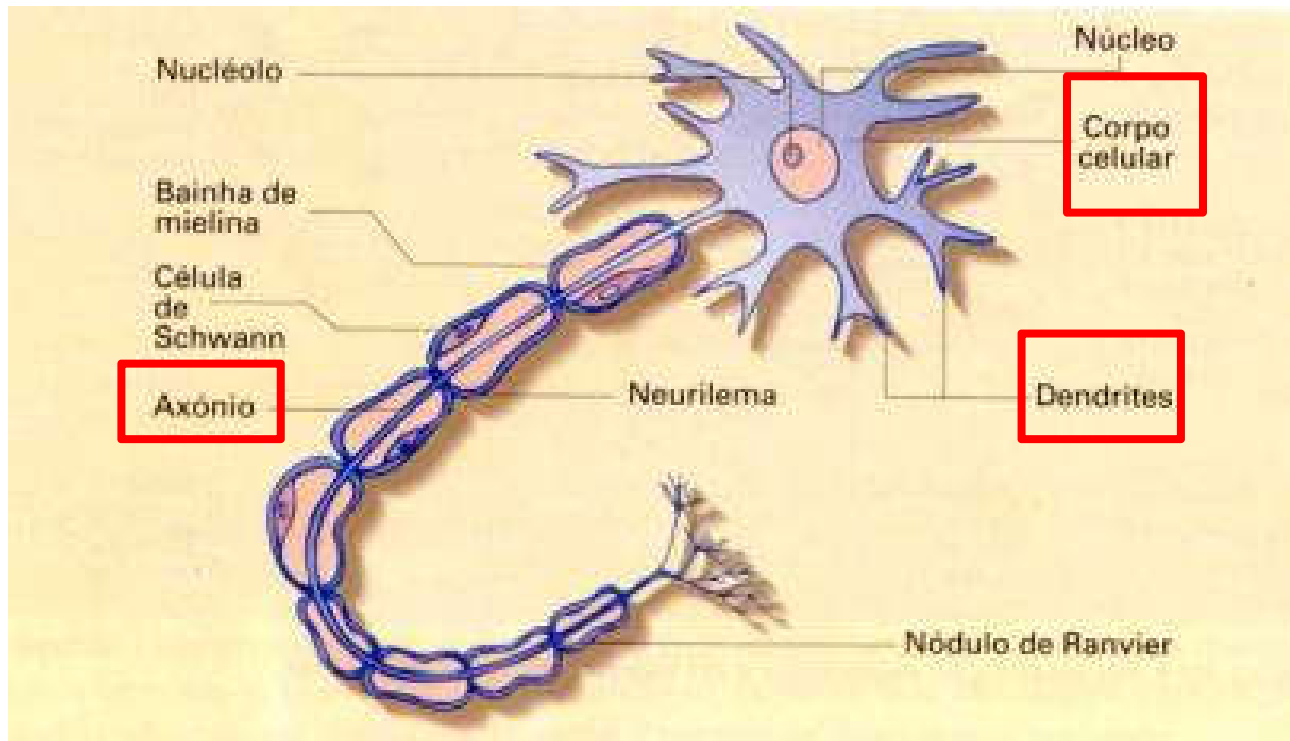
Hopfield



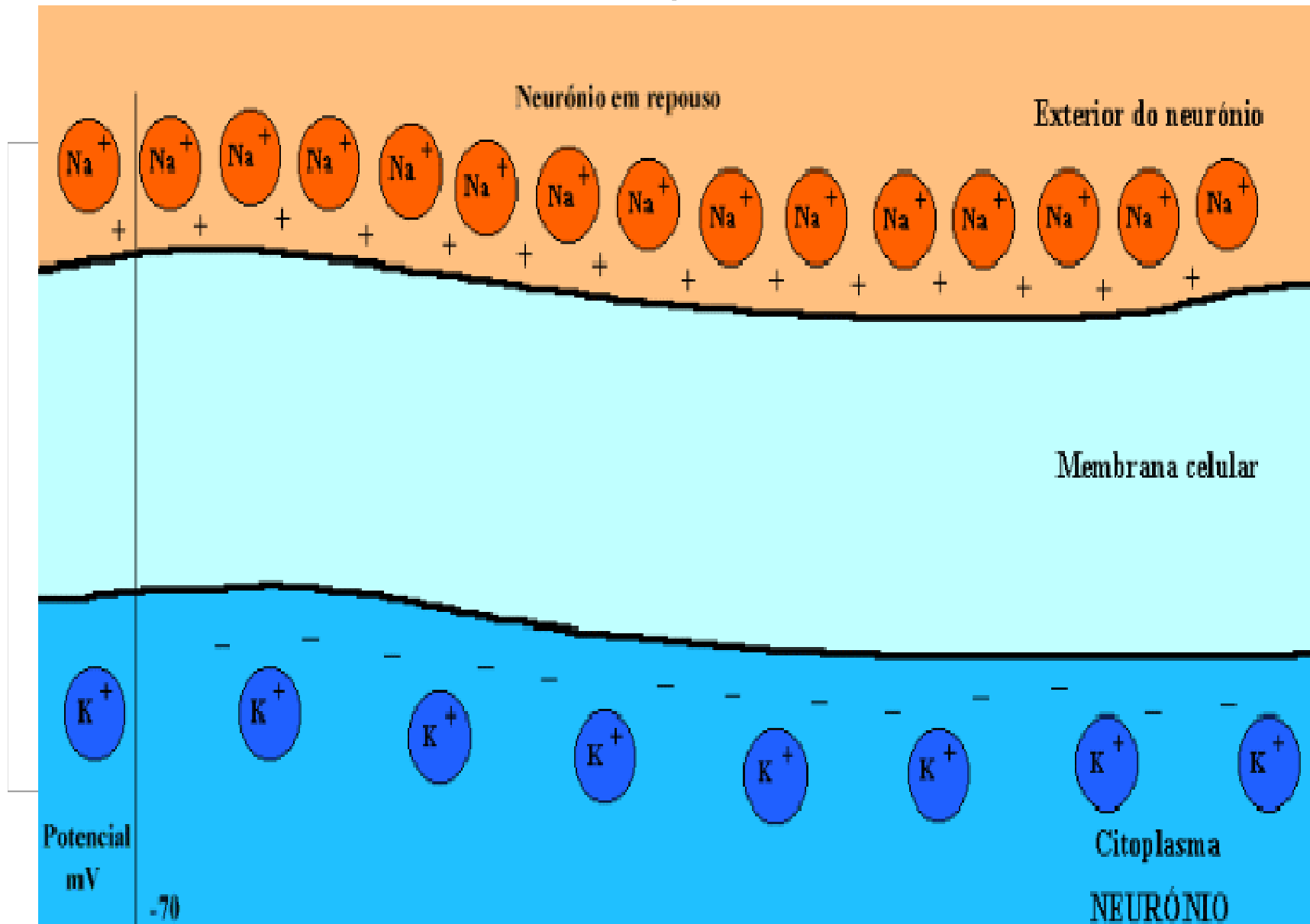
RN associativas

Neurônio MCP: Inspiração Biológica

- Neurônio Biológico



○ Neurônio e o Impulso Nervoso



Redes Neurais: Inspiração Biológica

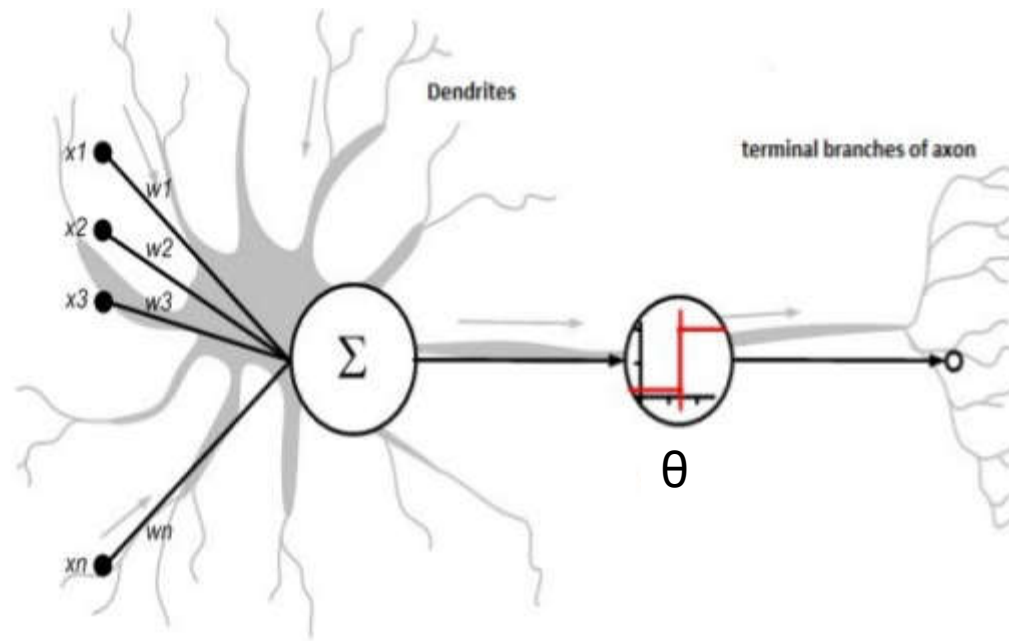
- Neurônio, Impulso Nervoso e Sinapses



<https://www.youtube.com/watch?v=2Kg0nvSpeXY>

Redes Neurais: Inspiração Biológica

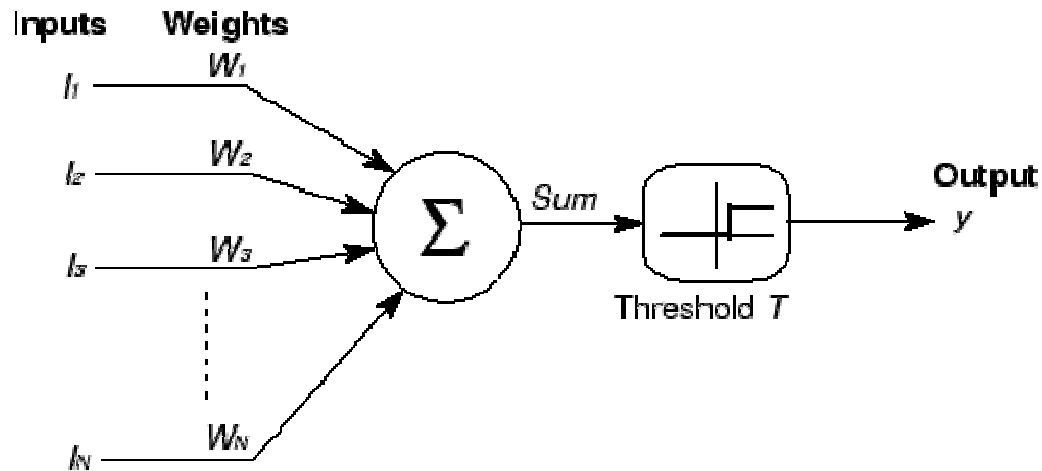
- Neurônio Biológico x Neurônio Artificial



$$y = \begin{cases} \text{dispara (1)} & \text{se } \left(\sum_{j=1}^n w_j x_j \right) \geq \theta \\ \text{não - dispara (0)} & \text{caso contrário} \end{cases}$$

Neurônio Artificial

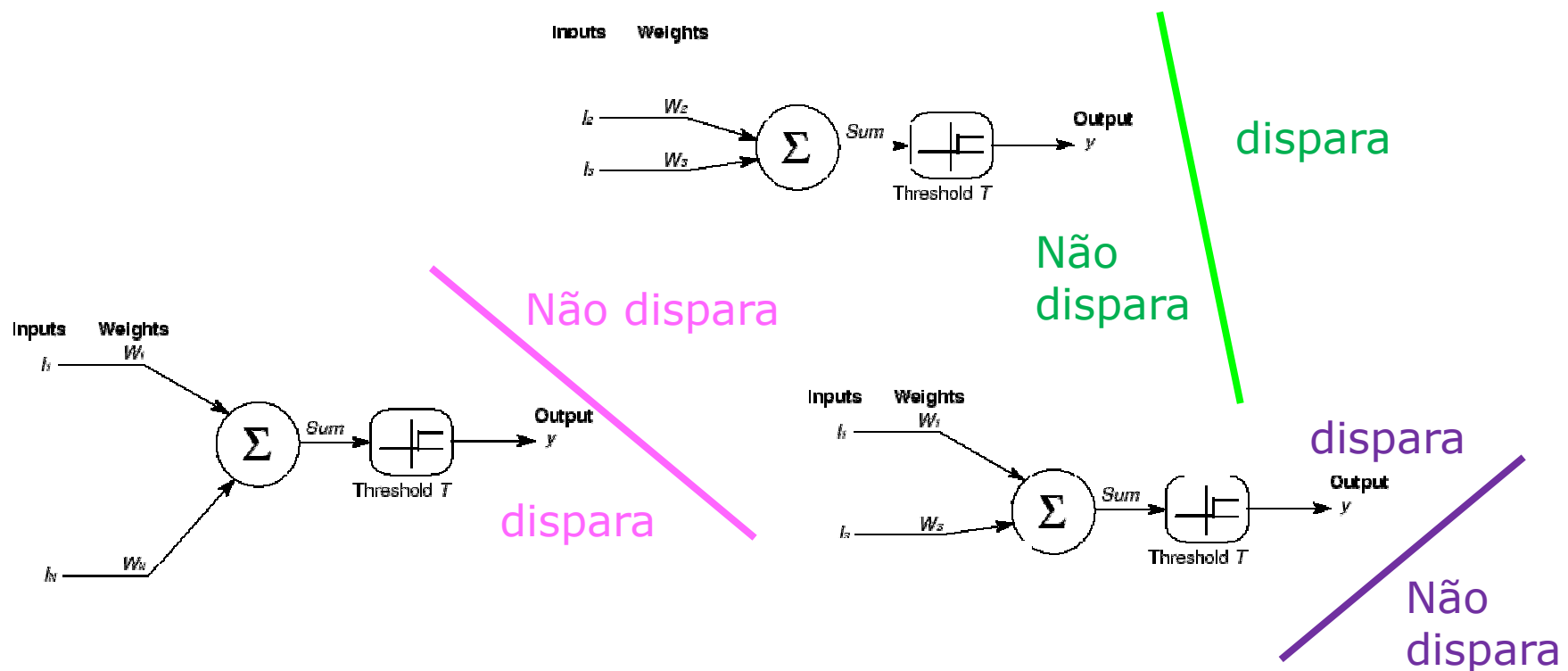
Os primeiros trabalhos na área datam de 1943:
McCulloch e Pitts desenvolveram o primeiro
modelo matemático do neurônio



$$y = \text{limiar}(\text{Soma}(I_j * W_j)), j = 1, \dots, N$$

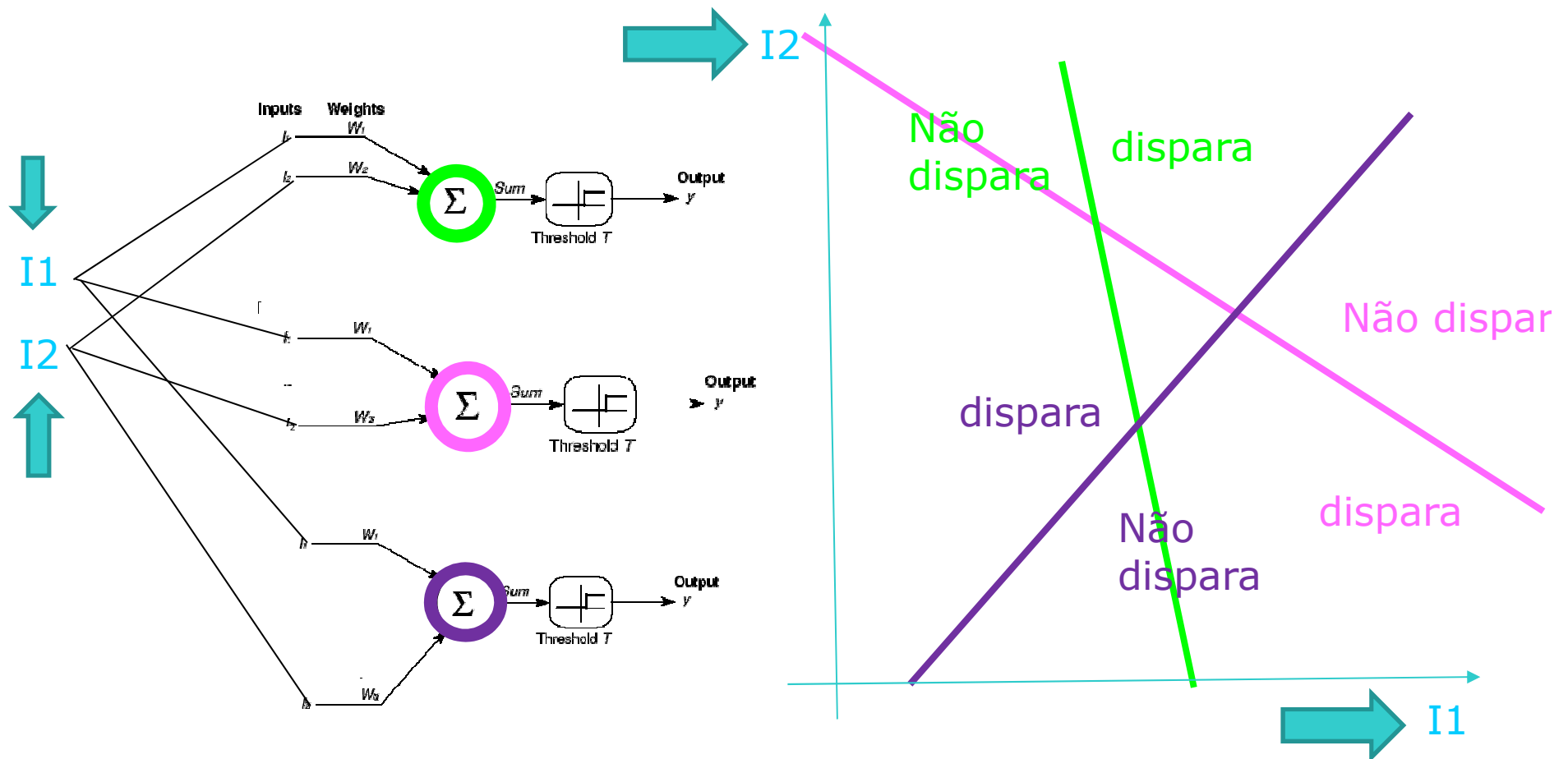
Neurônio MCP (partição do espaço)

Neurônios MCP independentes com **diferentes pesos**:
diferentes partições do espaço de entradas



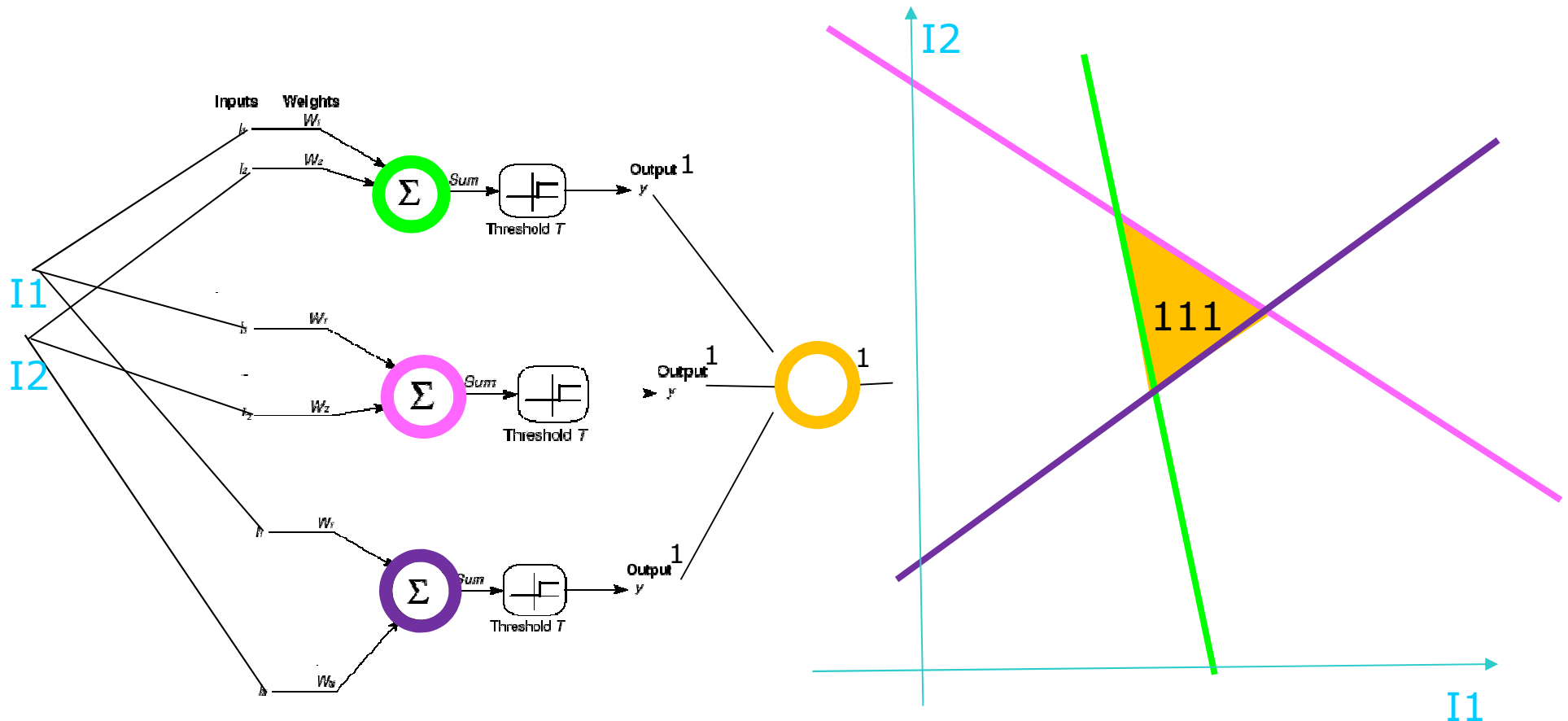
RNAs: Neurônio MCP (partição do espaço)

Rede Neural (1 camada): partições independentes no espaço $I1 \times I2$

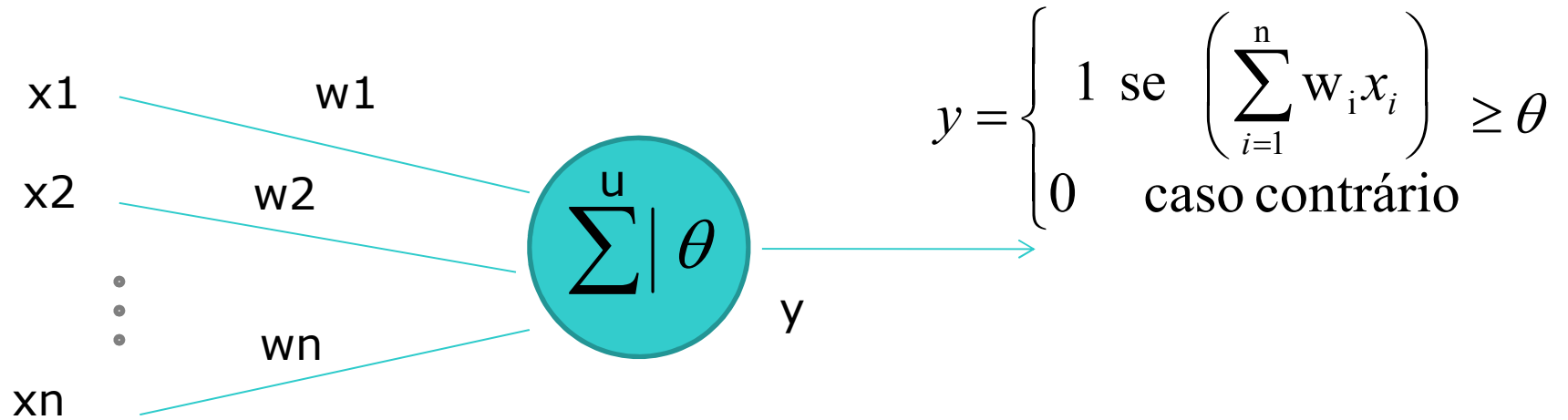


RNAs: Neurônio MCP (partição do espaço)

Rede Neural (2 camadas): combina as diferentes partições



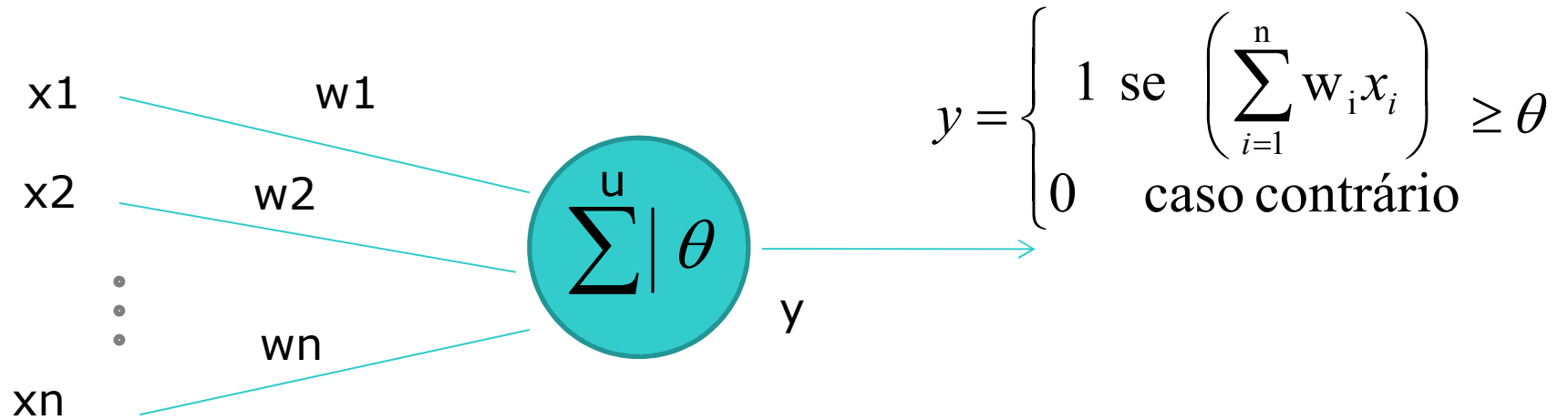
RNAs: Neurônio MCP (partição do espaço)



Há muitas formas de se entender o funcionamento do neurônio MCP

- **Mapeamento produzido por diferentes entradas**
- **Equação de Reta: Projeção da interseção de planos**
- **Projeção da função de ativação multidimensional no espaço de entrada**
- **Produto interno do vetor de pesos pelo vetor de entradas**

RNAs: Neurônio MCP (partição do espaço)

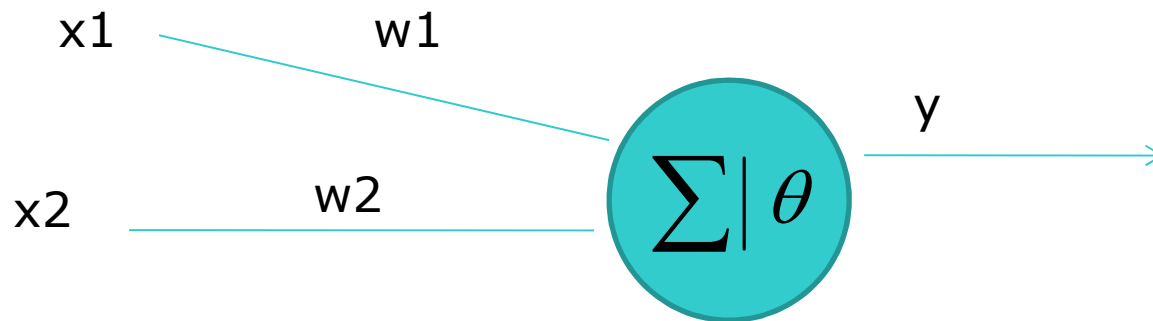


Há muitas formas de se entender o funcionamento do neurônio MCP

- **Mapeamento produzido por diferentes entradas**
- **Equação de Reta: Projeção da interseção de planos**
- **Projeção da função de ativação multidimensional no espaço de entrada**
- **Produto interno do vetor de pesos pelo vetor de entradas**

RNAs: Neurônio MCP (partição do espaço)

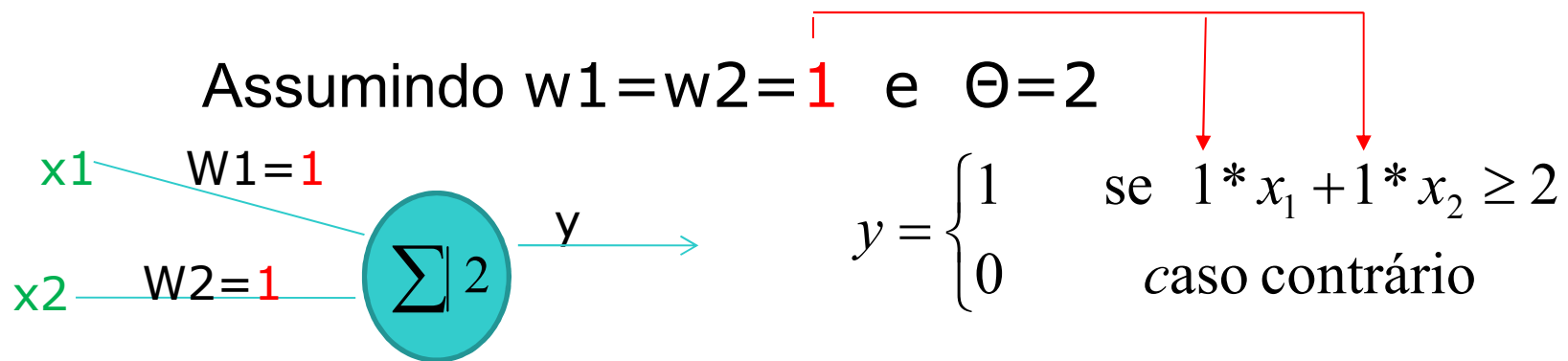
Considerando o caso particular de 1 neurônio com 2 entradas



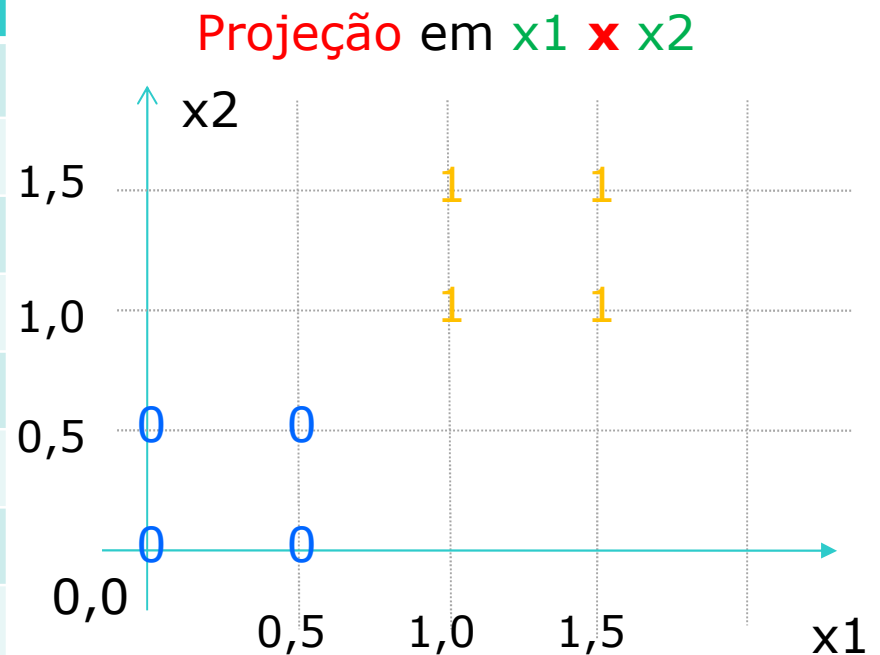
$$y = \begin{cases} 1 & \text{se } \left(\sum_{i=1}^n w_i x_i \right) \geq \theta \\ 0 & \text{caso contrário} \end{cases} \longrightarrow y = \begin{cases} 1 & \text{se } w_1 x_1 + w_2 x_2 \geq \theta \\ 0 & \text{caso contrário} \end{cases}$$

Neurônio MCP:

entendendo melhor a partição do espaço 2D

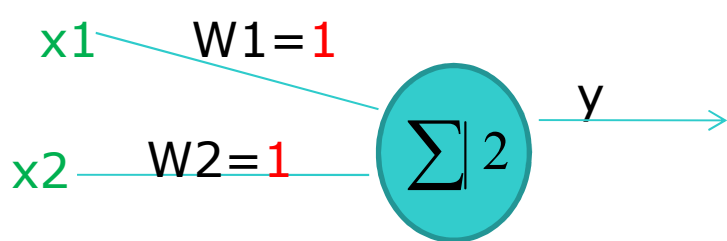


x1	x2	y
0	0	0
0	0,5	0
0,5	0	0
0,5	0,5	0
1	1	1
1	1,5	1
1,5	1	1
1,5	1,5	1



Neurônio MCP: Abstração

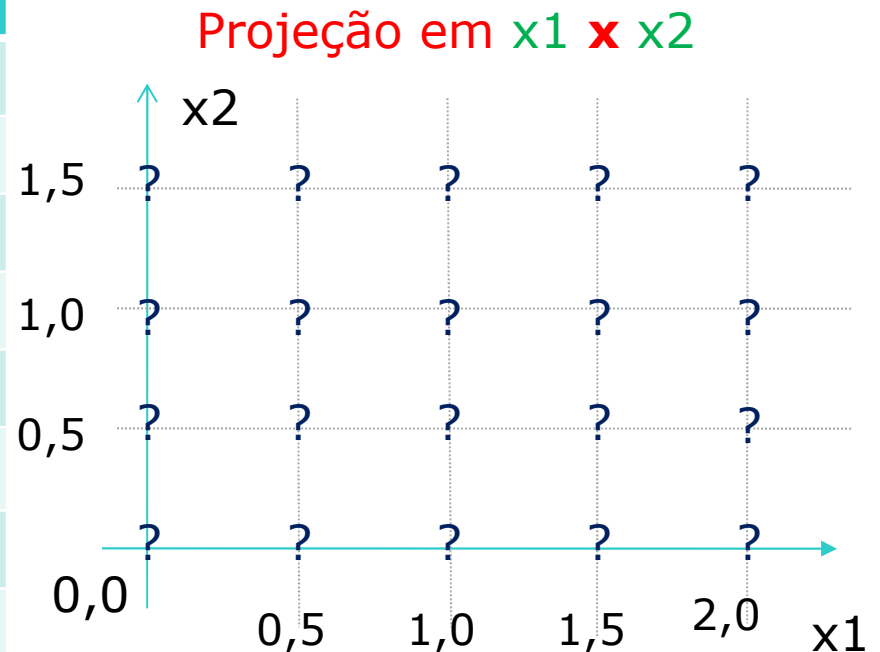
Como fica a tabela ?



$$y = \begin{cases} 1 & \text{se } 1 * x_1 + 1 * x_2 \geq 2 \\ 0 & \text{caso contrário} \end{cases}$$

Aumentando-se o número de padrões de entrada (x1,x2)

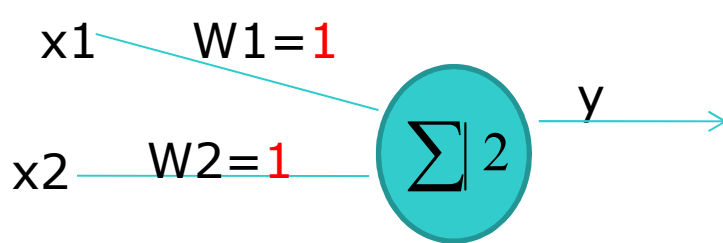
x1	x2	y
0,5	1,5	
1,0	1,0	
...	
2,0	1,5	
0	0	
...	...	
1,0	0	
1,5	0	



Neurônio MCP:

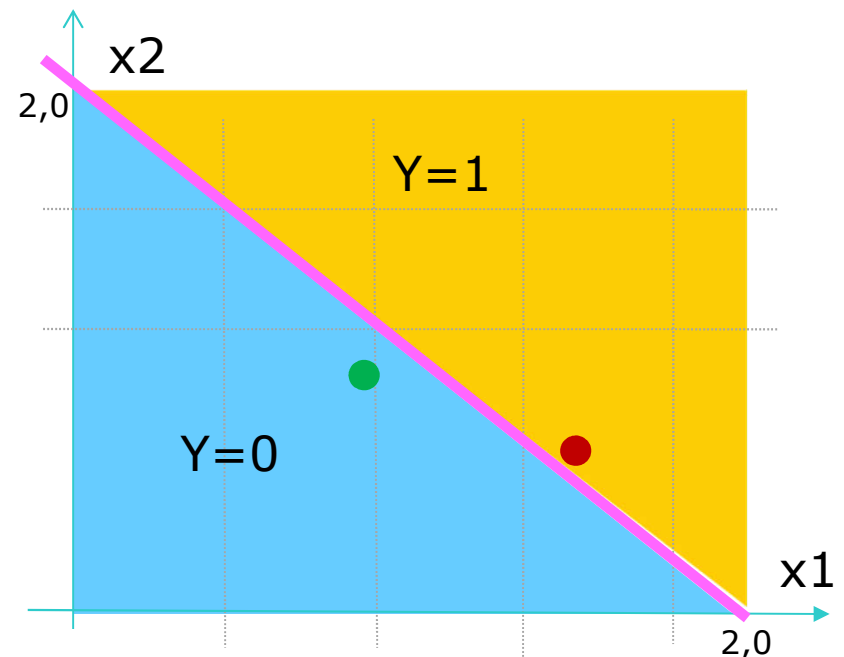
entendendo melhor a partição do espaço 2D

Para infinitos padrões de entrada (x1,x2)

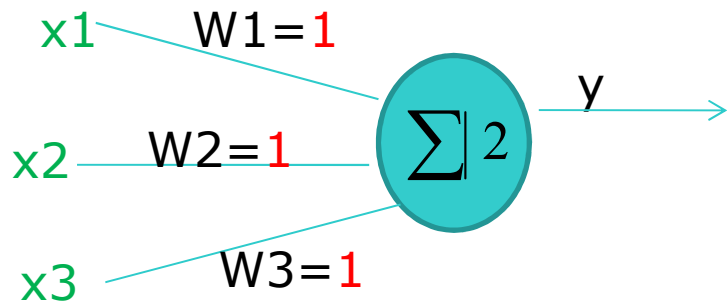


$$y = \begin{cases} 1 & \text{se } 1 * x_1 + 1 * x_2 \geq 2 \\ 0 & \text{caso contrário} \end{cases}$$

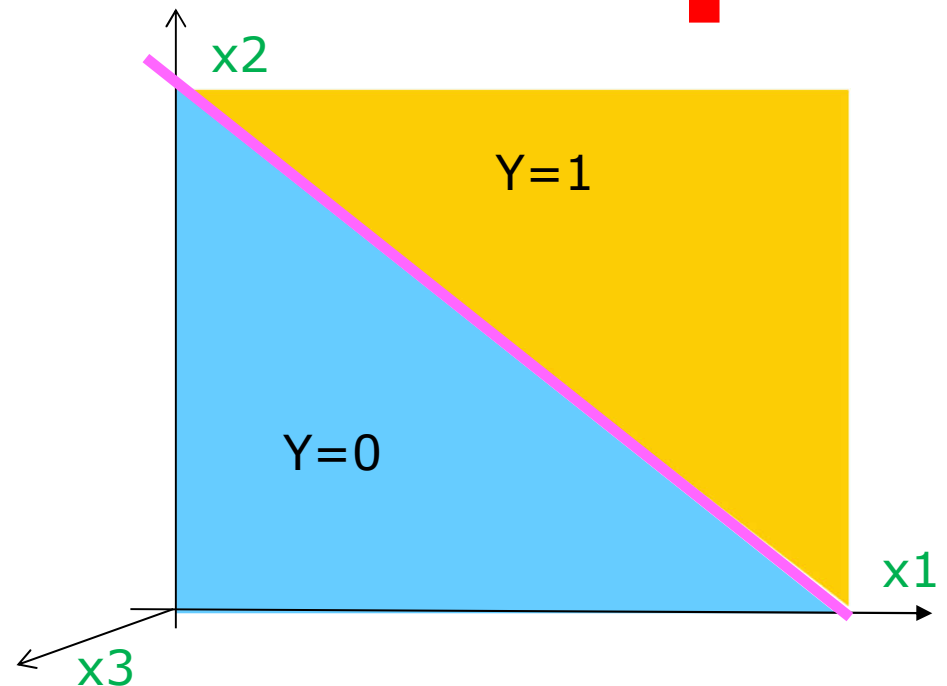
x1	x2	y
0,99	0,80	0
...
1,99	0	0
0,5	1,52	1
1,2	1,0	1
1,57	0,58	1
2,01	1,5	1
...



Neurônio MCP: **Abstração** potencial de ativação x partição do espaço **3D**

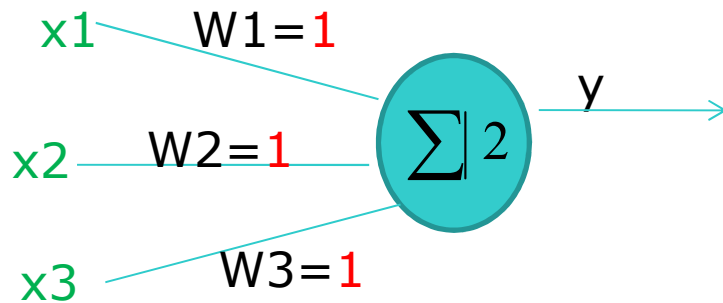


Partição em
 x_1 **x** x_2 **x** x_3

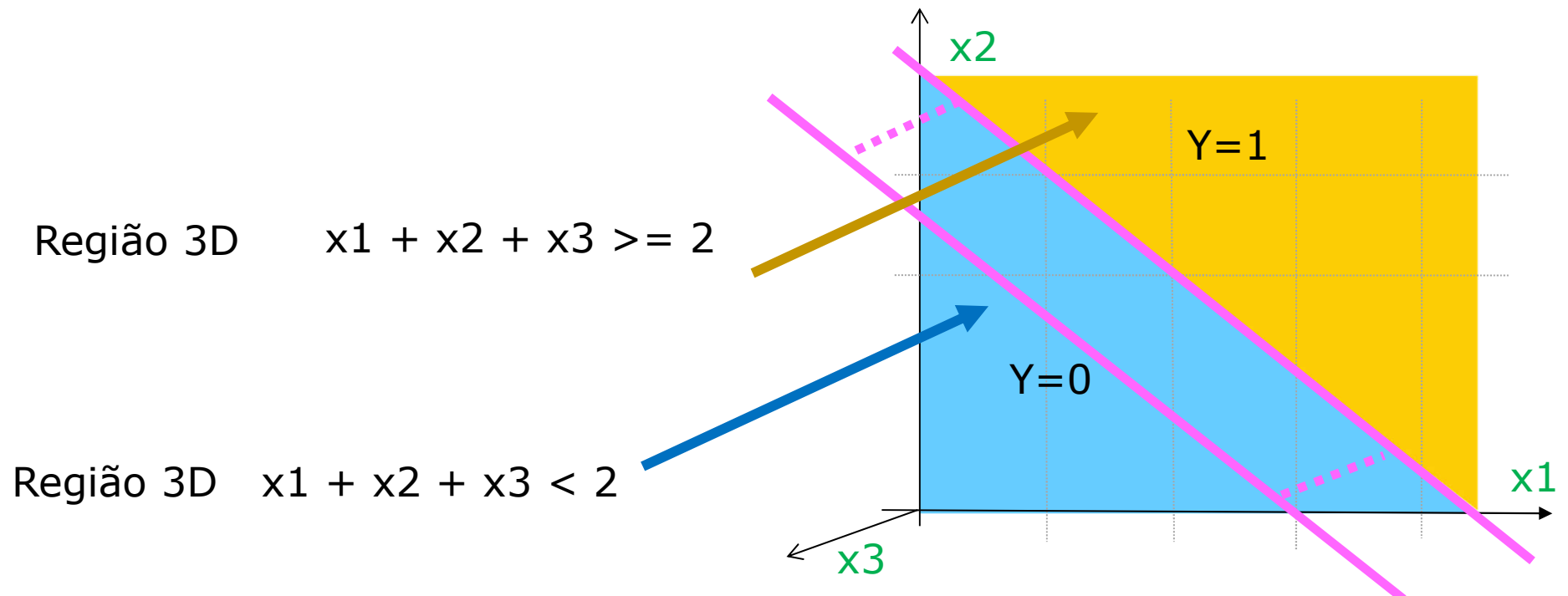


Neurônio MCP: Abstração

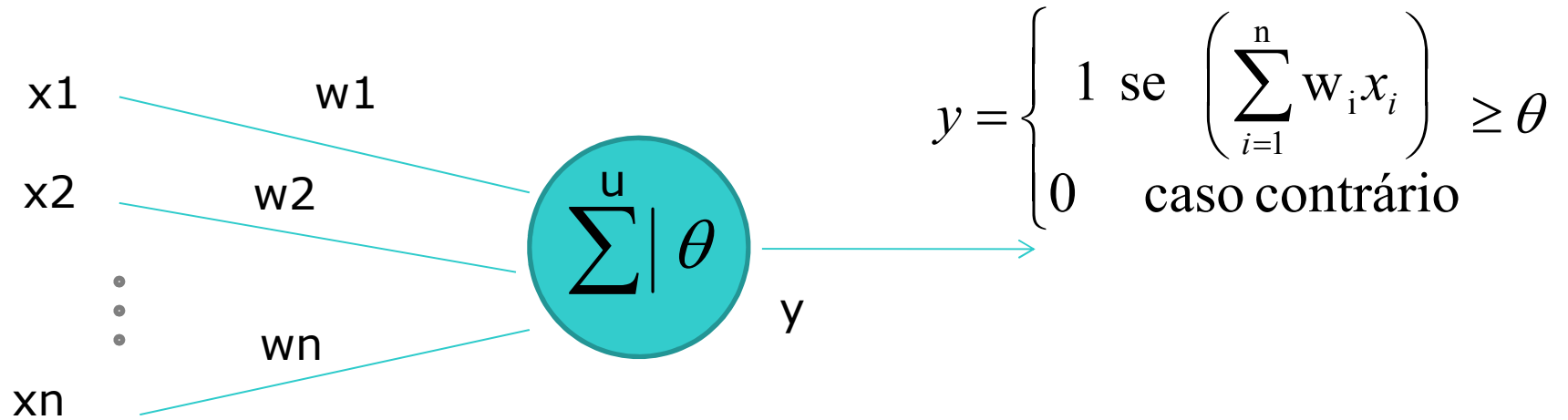
potencial de ativação x partição do espaço 3D



Partição em
 $x_1 \times x_2 \times x_3$



RNAs: Neurônio MCP (partição do espaço)



Há muitas formas de se entender o funcionamento do neurônio MCP

- Mapeamento produzido por diferentes entradas
- **Eq da Reta(hiper-plano):** **Projeção da interseção de hiper-planos**
- **$u = \sum_{i=1}^n w_i x_i$ com $u = \theta$**
- Projeção da função de ativação multidimensional no espaço de entrada
- Produto interno do vetor de pesos pelo vetor de entradas

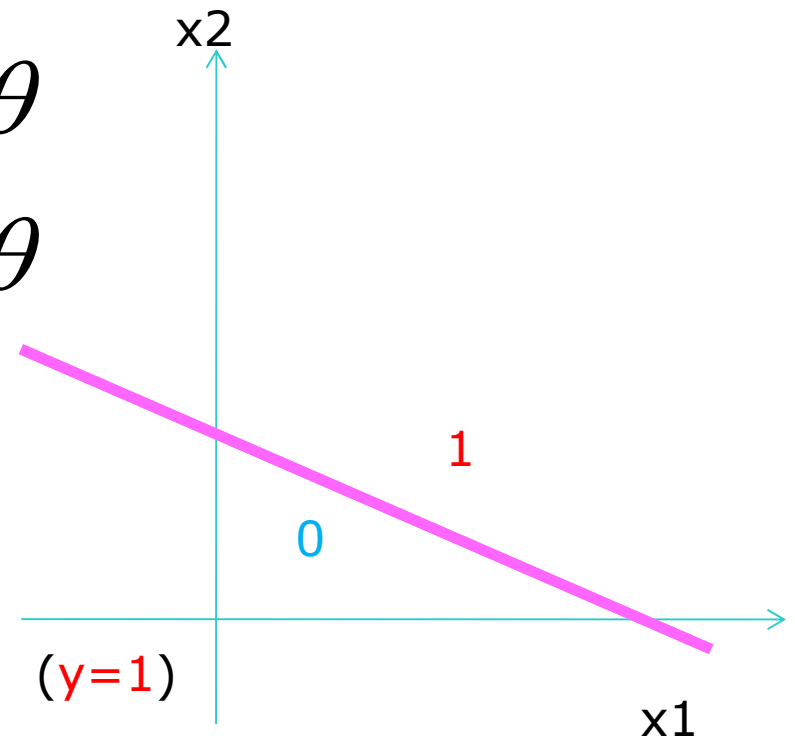
RNAs: Neurônio MCP (partição do espaço)

Saída (y) é uma função da combinação linear das entradas (\mathbf{x})

$$y = \begin{cases} 1 & \text{se } w_1x_1 + w_2x_2 \geq \theta \\ 0 & \text{se } w_1x_1 + w_2x_2 < \theta \end{cases}$$

Na condição **limite** (mudança do grau de ativação) temos:

$$w_1x_1 + w_2x_2 = \theta \begin{cases} \rightarrow \Sigma > \theta \text{ (} y=1 \text{)} \\ \rightarrow \Sigma < \theta \text{ (} y=0 \text{)} \end{cases}$$

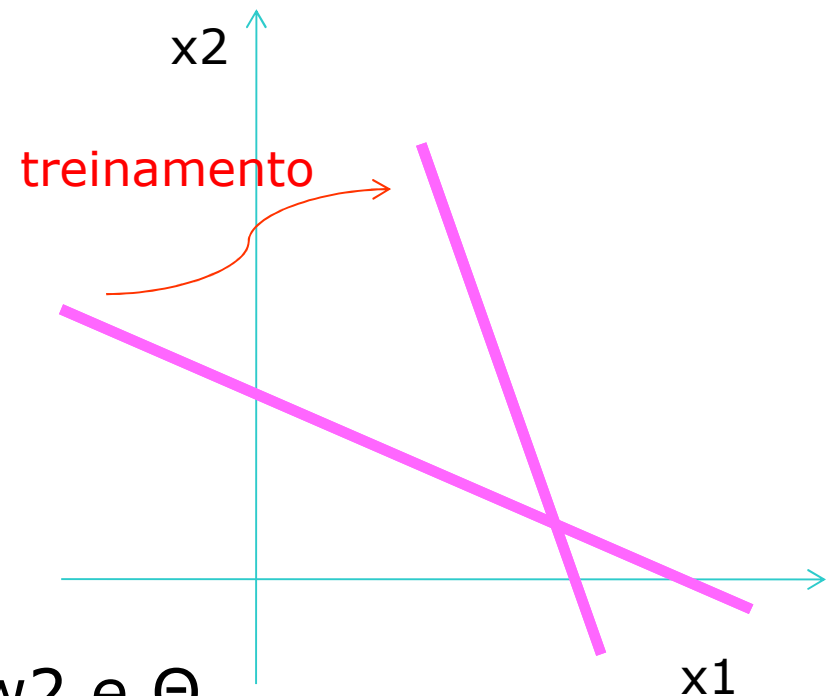


RNAs: Pesos do Neurônio MCP (treinamento)

$$w_1x_1 + w_2x_2 = \theta$$

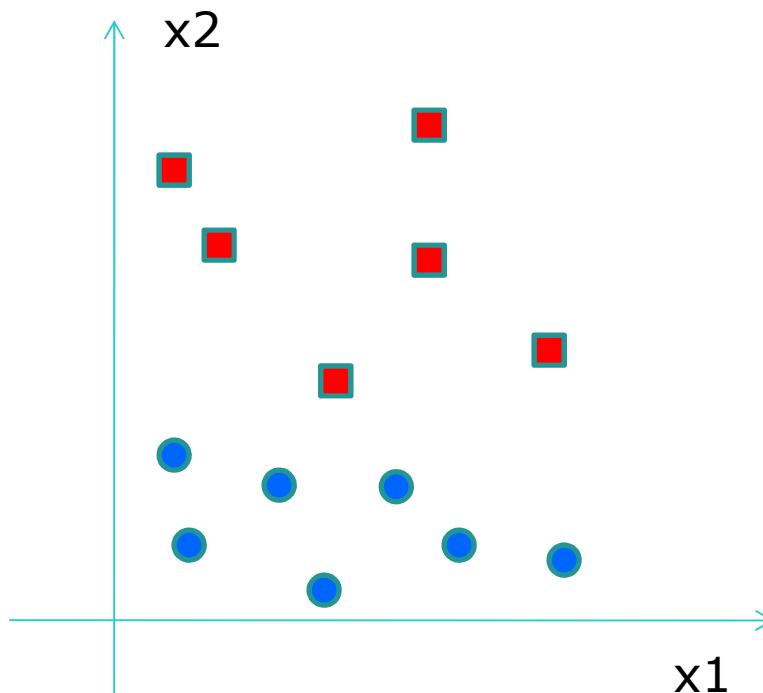
ou

$$x_2 = -\left(\frac{w_1}{w_2}\right)x_1 + \left(\frac{\theta}{w_2}\right)$$



A alteração dos parâmetros w_1 , w_2 e θ (**treinamento**) modifica a posição da reta e portanto da partição no espaço

Problema de Classificação

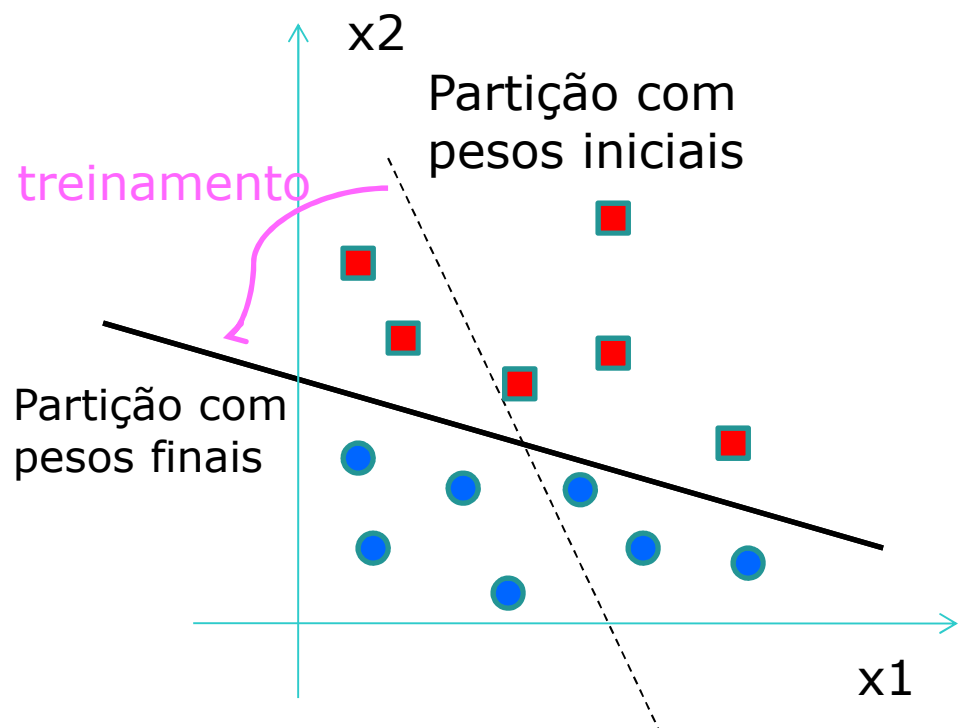


Supondo que os pontos em **vermelho** indiquem **maus pagadores** e os pontos em **azul** indiquem **bons pagadores**.

Este problema poderia ser resolvido por um único neurônio MCP?

Sim pois é linearmente separável e a alteração dos parâmetros \mathbf{w} e Θ (treinamento) modifica a posição da reta e portanto da partição linear no espaço de entrada.

RNAs: Neurônio MCP para Classificação



$$y = \begin{cases} 1 & \text{se } w_1 x_1 + w_2 x_2 \geq \theta \\ 0 & \text{se } w_1 x_1 + w_2 x_2 < \theta \end{cases}$$

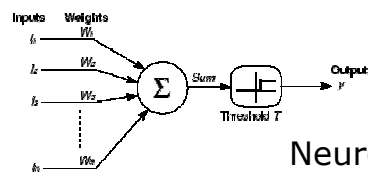
Partição com pesos iniciais

$$y = \begin{cases} 1 & \text{se } w_1^t x_1 + w_2^t x_2 \geq \theta^t \\ 0 & \text{se } w_1^t x_1 + w_2^t x_2 < \theta^t \end{cases}$$

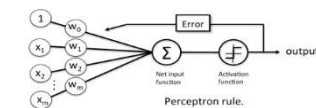
Partição com pesos finais

A alteração dos parâmetros \mathbf{w} e θ (treinamento) modifica a posição da reta e portanto da partição linear no espaço de entrada.

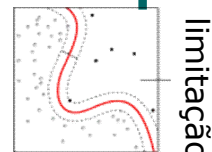
RNAs: Linha do Tempo



Neurônio MCP



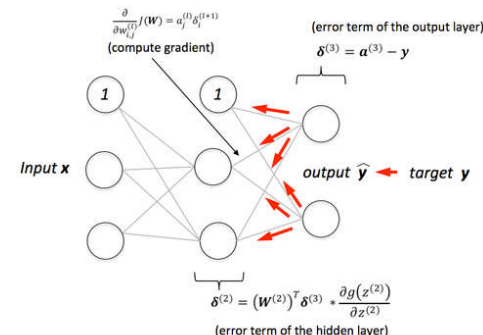
Perceptron



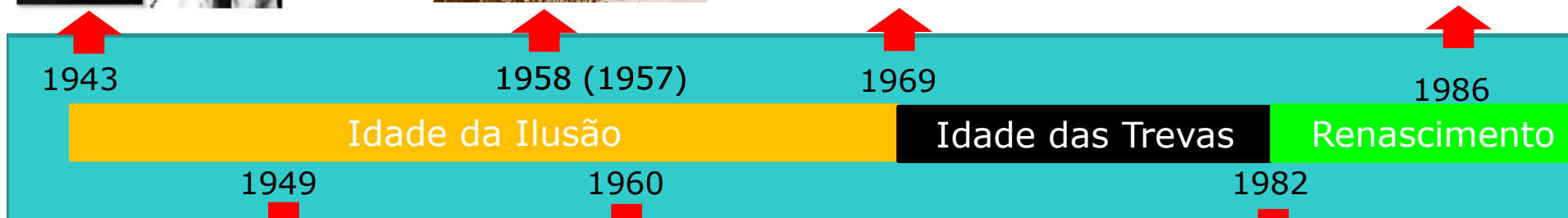
limitação



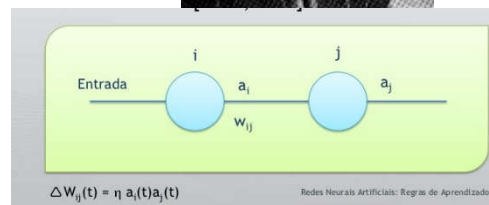
Minsky e Papert



Backpropagation

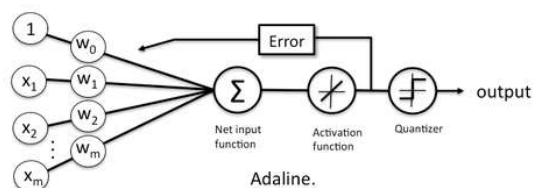


Regra de Hebb



Regra Delta (Widrow-Hoff)

$$w_{ij}(\text{new}) = w_{ij}(\text{old}) + \alpha(t_i - y_{in_i})x_j$$

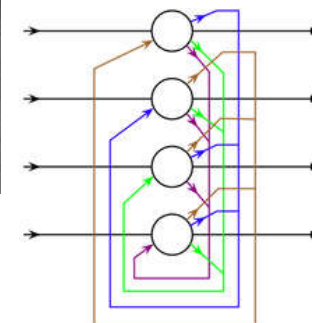


Adaline.

Prof. Myriam Delgado
Redes Neurais - UTFPR



Hopfield



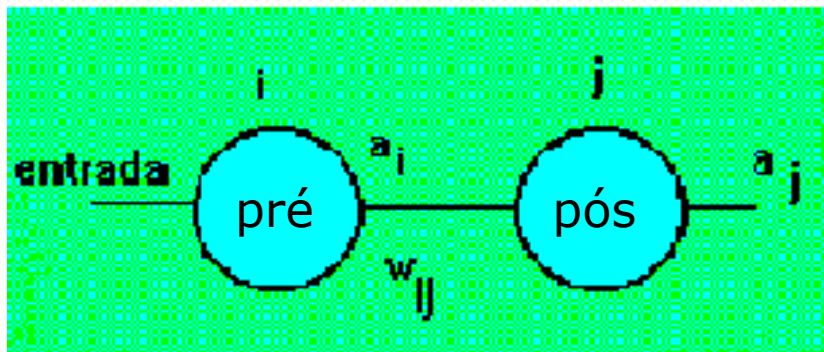
RN associativas

RNAs: Regra de Hebb

A regra de aprendizado de Hebb propõe que o peso de uma conexão sináptica deve ser

reforçada se houver **sincronismo** entre os níveis de atividade dos neurônios pré e pós-sinápticos

e **inibida** caso contrário (**assíncronos**)

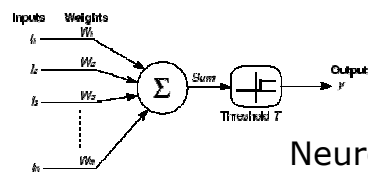


correlação + \Rightarrow o valor do peso aumenta
correlação - \Rightarrow o valor do peso diminui

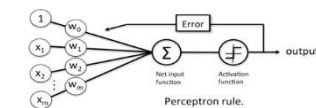
$$\Delta W_{ij}(t) = \eta a_i(t)a_j(t)$$

$a_i(t)$	$a_j(t)$	$\Delta W_{ij}(t)$
+	+	+
-	-	+
+	-	-
-	+	-

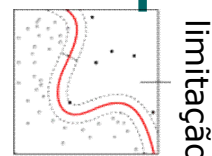
RNAs: Linha do Tempo



Neurônio MCP



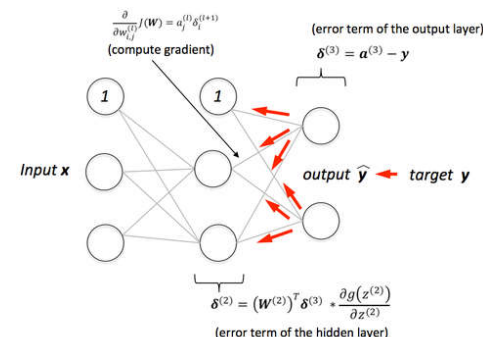
Perceptron



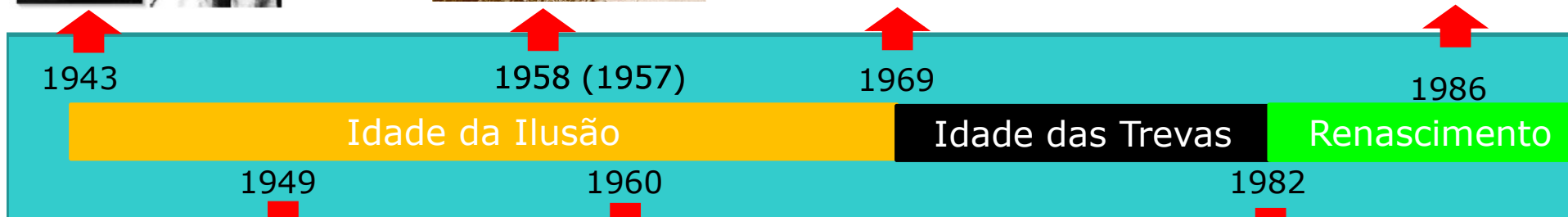
limitação



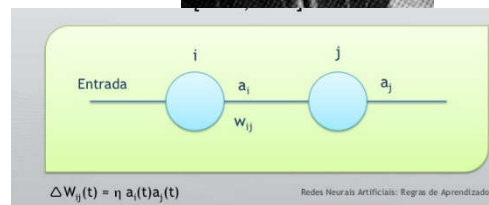
Minsky e Papert



Backpropagation

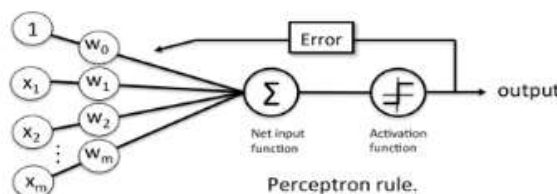


Regra de Hebb



Regra Delta (Widrow-Hoff)

$$w_{ij}(\text{new}) = w_{ij}(\text{old}) + \alpha(t_i - y_{in_i})x_j$$

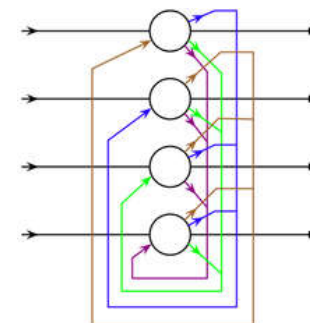


Perceptron rule.

Prof. Myriam Delgado
Redes Neurais - UTFPR

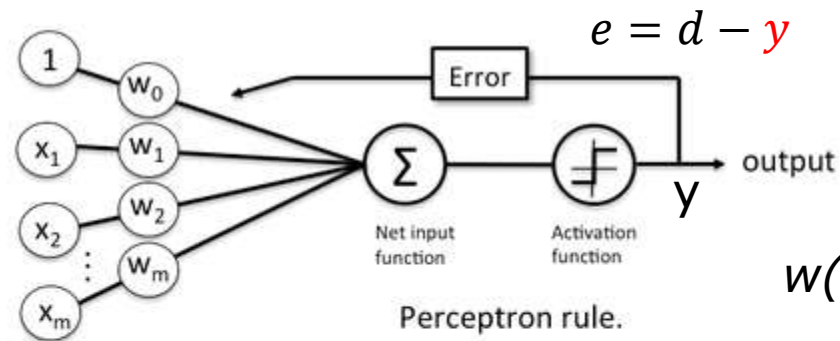


Hopfield



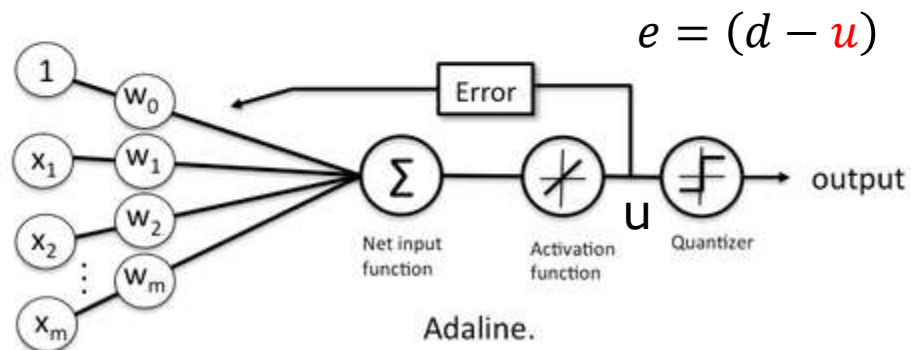
RN associativas

Perceptron e Adaline



$$E = (d - y) <\mathbf{w}, \mathbf{x}>$$

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

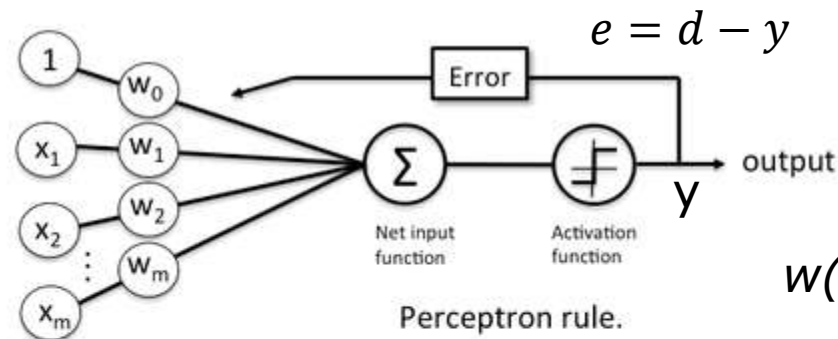


$$E = \frac{1}{2} (d - u)^2$$

$$\frac{\partial E}{\partial \mathbf{w}}$$

Perceptron e Adaline (aplicação)

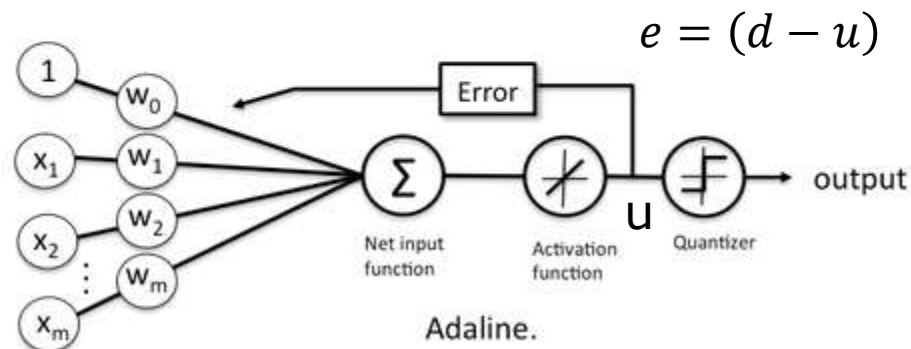
Aproximação
de
Portas lógicas
AND OR



Regra Delta

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

Combinação
Linear
de funções

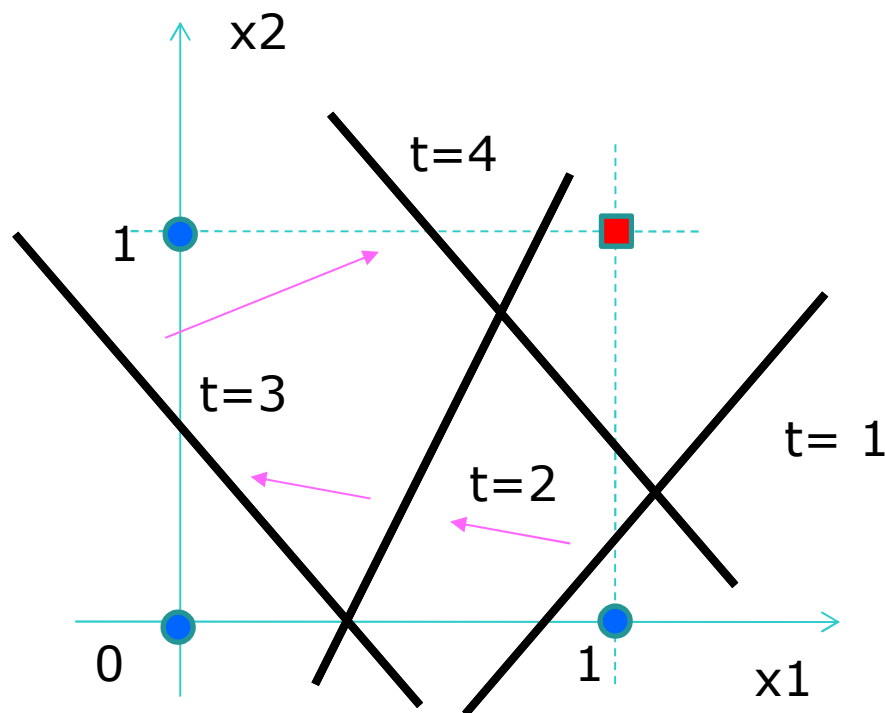


$$F(x) = a + bf_1(x) + cf_2(x) + df_3(x)$$

RNAs: Treinamento do Perceptron (2 entradas)

Treinamento: ajuste dos pesos a cada iteração muda a partição.

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \Delta\mathbf{w}(t) = \mathbf{w}(t) + \eta \text{ erro } \mathbf{x}$$



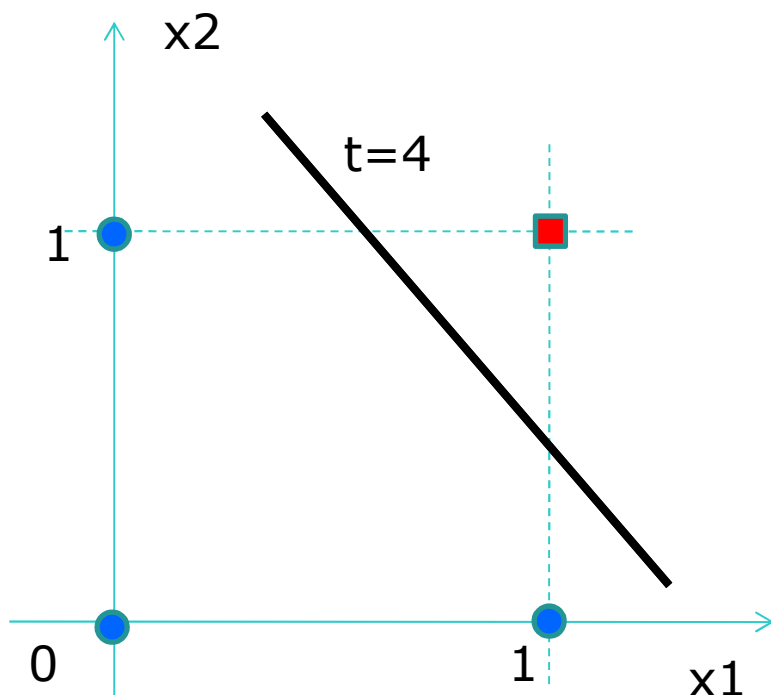
AND Lógico (Linearmente separável)

x1	x2	y
1	1	1
0	1	0
1	0	0
0	0	0

RNAs: Treinamento do Perceptron (2 entradas)

Treinamento: ajuste dos pesos a cada iteração muda a partição.

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \Delta \mathbf{w}(t) = \mathbf{w}(t) + \eta \text{ erro } \mathbf{x}$$



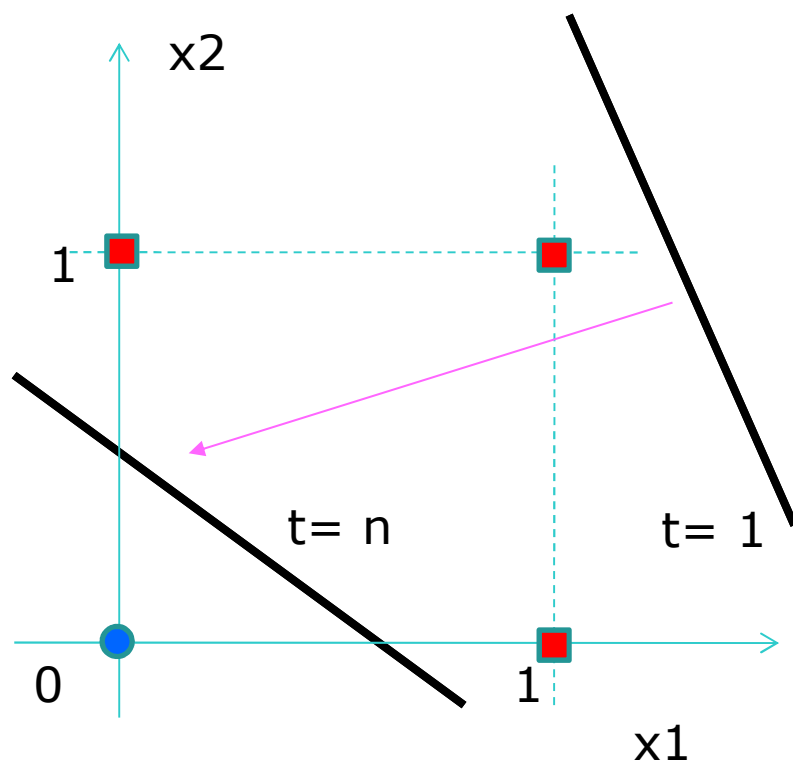
AND Lógico (Linearmente separável)

x_1	x_2	y
1	1	1
0	1	0
1	0	0
0	0	0

RNAs: Treinamento do Perceptron (2 entradas)

Treinamento: ajuste dos pesos a cada iteração muda a partição.

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \Delta \mathbf{w}(t) = \mathbf{w}(t) + \eta \text{ erro } \mathbf{x}$$

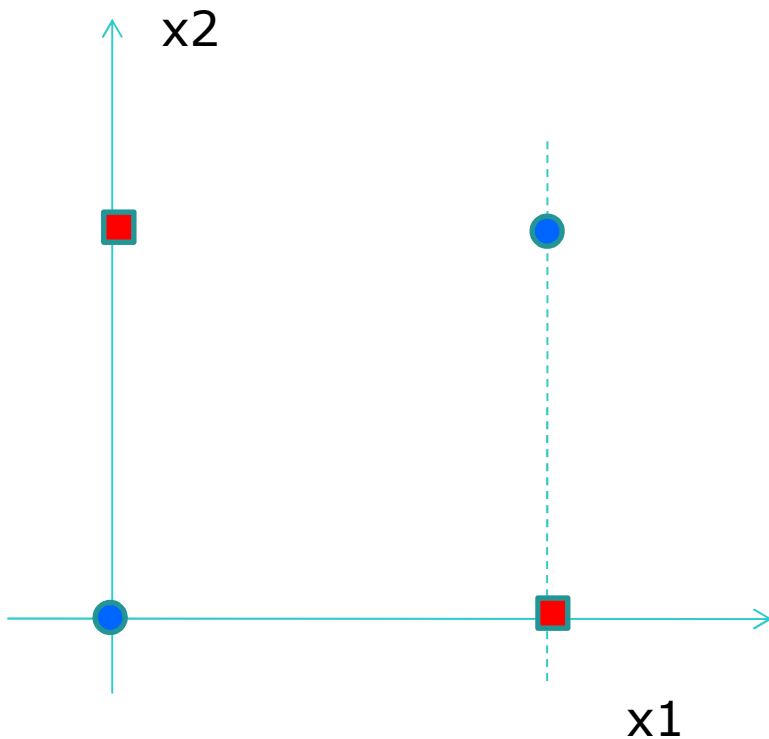


OR Lógico (Linearmente separável)

x1	x2	y
0	0	0
0	1	1
1	0	1
1	1	1

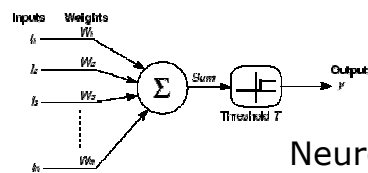
RNAs: Perceptron (resolução de problemas)

XOR lógico: ??? Qual linha separa???

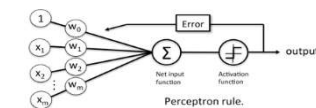


XOR	Lógico	
0	0	0
0	1	1
1	0	1
1	1	0

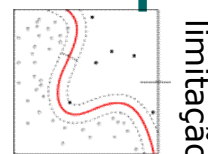
RNAs: Linha do Tempo



Neurônio MCP



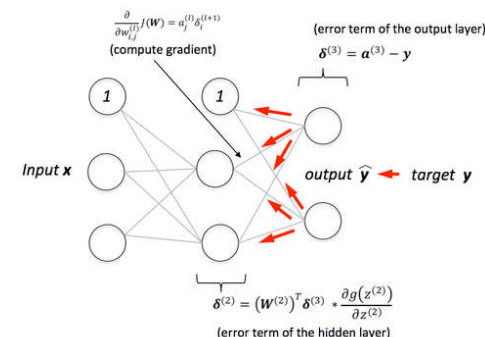
Perceptron



limitação



Minsky e Papert



Backpropagation

1943

1958 (1957)

1969

1986

Idade da Ilusão

Idade das Trevas

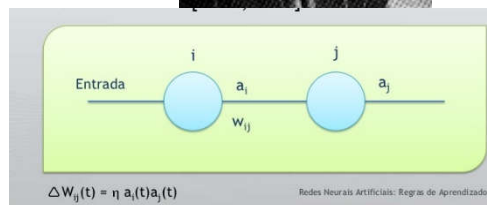
Renascimento

1949

1960

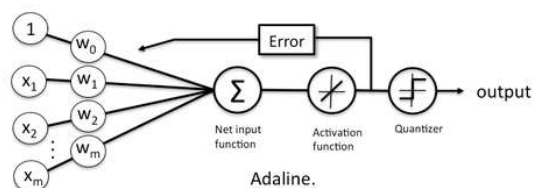
1982

Regra de Hebb



Regra Delta (Widrow-Hoff)

$$w_{ij}(\text{new}) = w_{ij}(\text{old}) + \alpha(t_i - y_{in_i})x_j$$

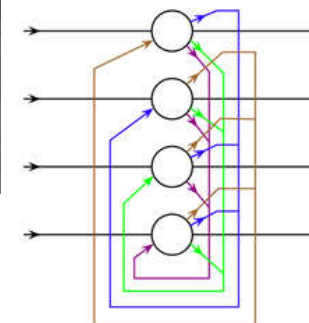


Adaline.

Prof. Myriam Delgado
Redes Neurais - UTFPR



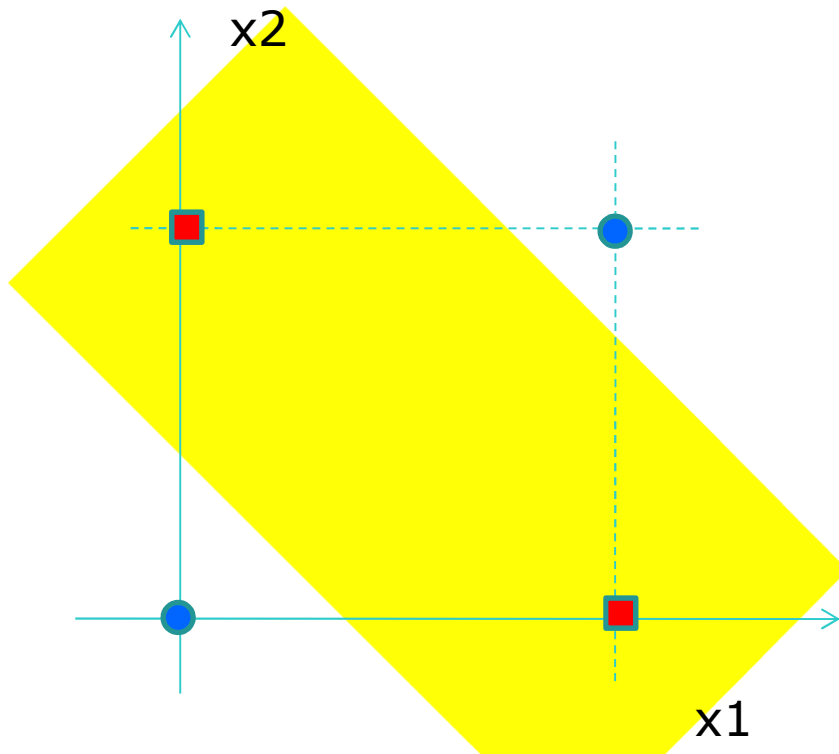
Hopfield



RN associativas

RNAs: Neurônio MCP (resolução de problemas)

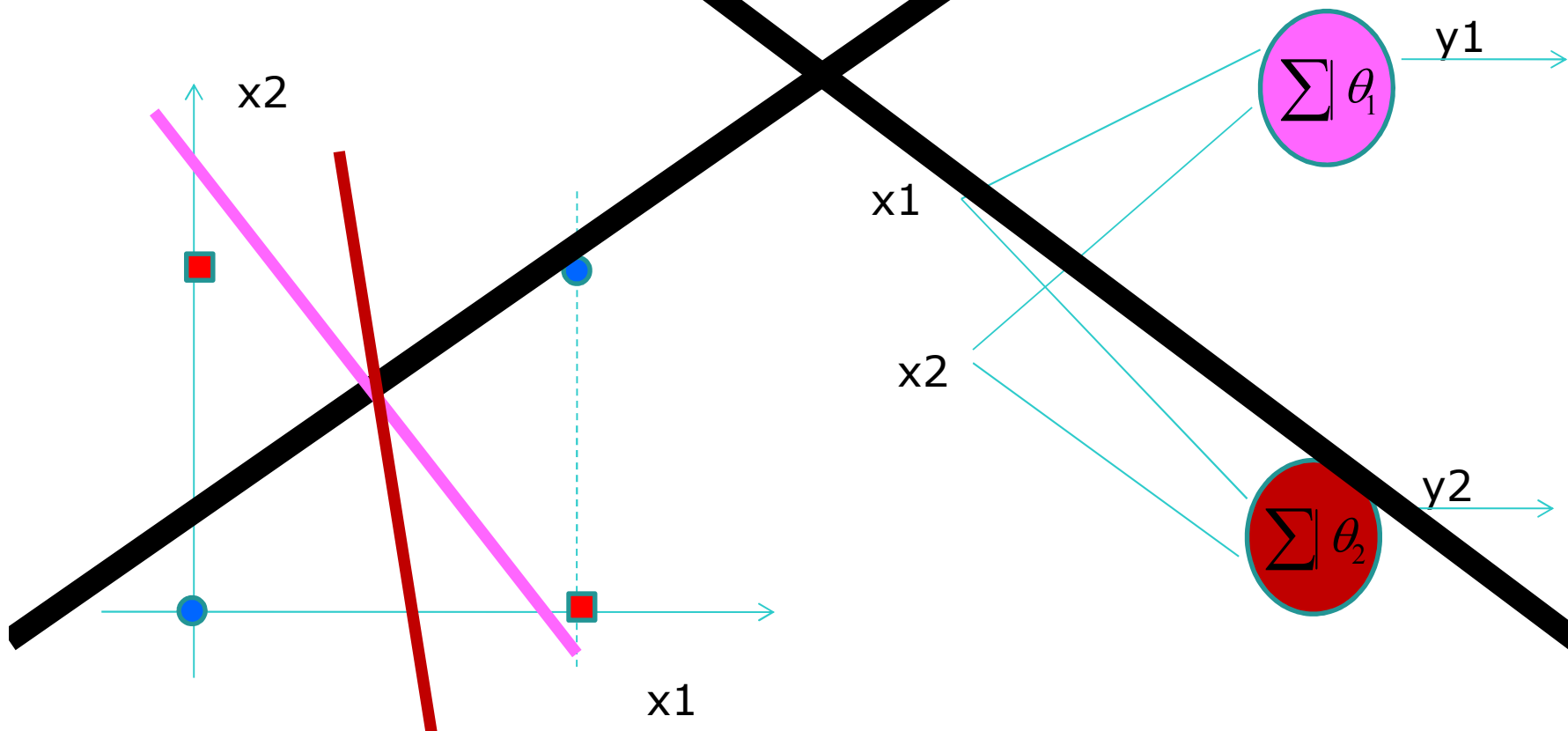
XOR lógico: não linearmente separável



XOR Lógico		
x1	x2	y
0	0	0
0	1	1
1	0	1
1	1	0

RNAs: Neurônio MCP (resolução de problemas)

XOR lógico: Mais neurônios ???

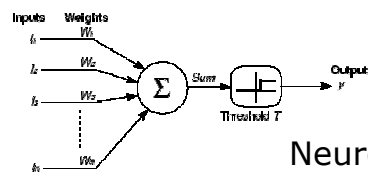


RNAs: Resolução de problemas

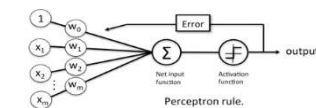
A descoberta da **limitação** dos neurônios (ou redes de uma única camada) na resolução de **problemas não-linearmente separáveis** trouxe um desânimo à comunidade.

Este período de descrença (conhecido como **idade das trevas**) teve início em 1969 e durou até o princípio dos anos 80.

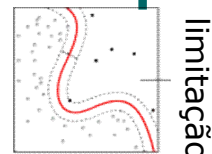
RNAs: Linha do Tempo



Neurônio MCP



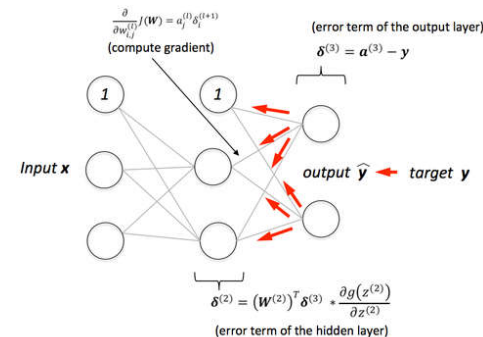
Perceptron



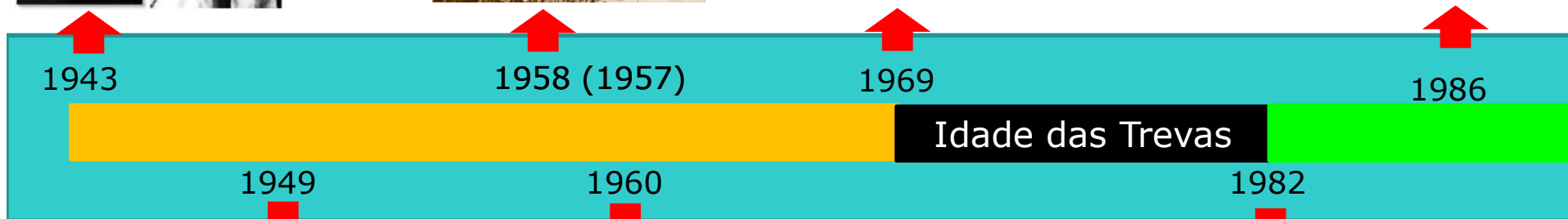
limitação



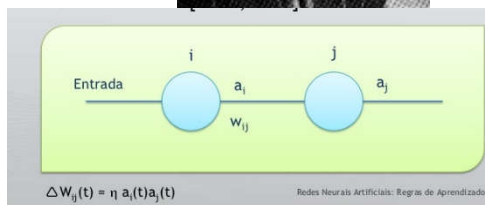
Minsky e Papert



Backpropagation

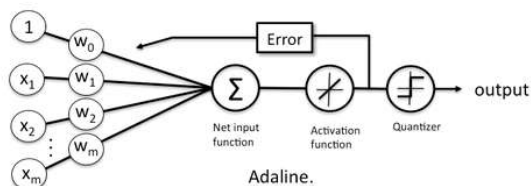


Regra de Hebb



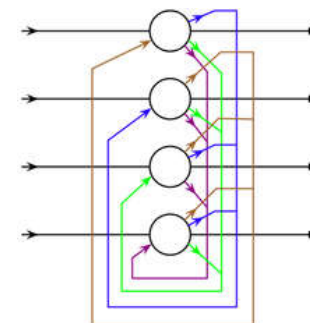
Regra Delta (Widrow-Hoff)

$$w_{ij}(\text{new}) = w_{ij}(\text{old}) + \alpha(t_i - y_{in_i})x_j$$



Adaline.

Prof. Myriam Delgado
Redes Neurais - UTFPR



RN associativas

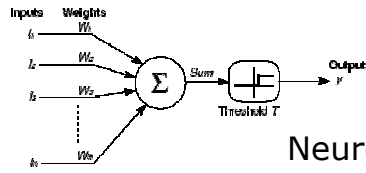
RNAs: Resolução de problemas

A descoberta da **limitação** dos neurônios (ou redes de uma única camada) na resolução de **problemas não-linearmente separáveis** trouxe um desânimo à comunidade.

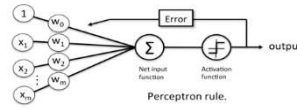
Este período de descrença (conhecido como **idade das trevas**) teve início em 1969 e durou até o princípio dos anos 80.

O **fim da idade das trevas** foi marcado por fatos importantes como a descoberta de algoritmos de **treinamento** para redes **multicamadas** – backpropagation.

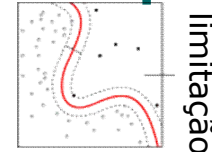
RNAs: Linha do Tempo



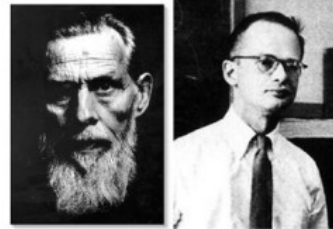
Neurônio MCP



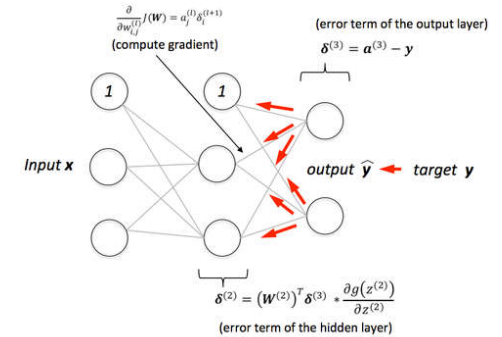
Perceptron



limitação



Minsky e Papert



Backpropagation

1943

1958 (1957)

1969

1986

Idade das Trevas

Renascimento

1949

1960

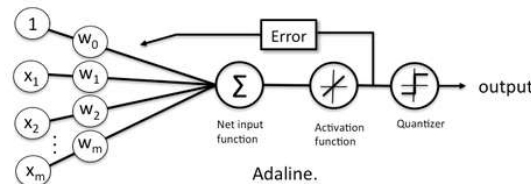
1982

Regra de Hebb

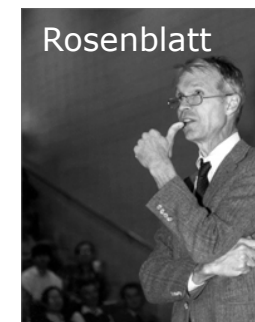


Regra Delta (Widrow-Hoff)

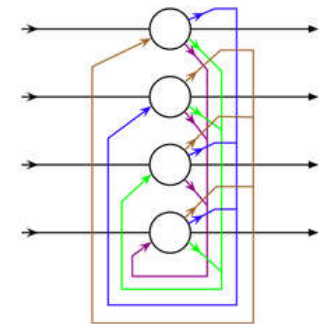
$$w_{ij}(\text{new}) = w_{ij}(\text{old}) + \alpha(t_i - y_{in_i})x_j$$



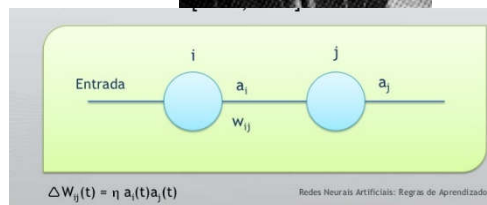
Adaline.



Rosenblatt

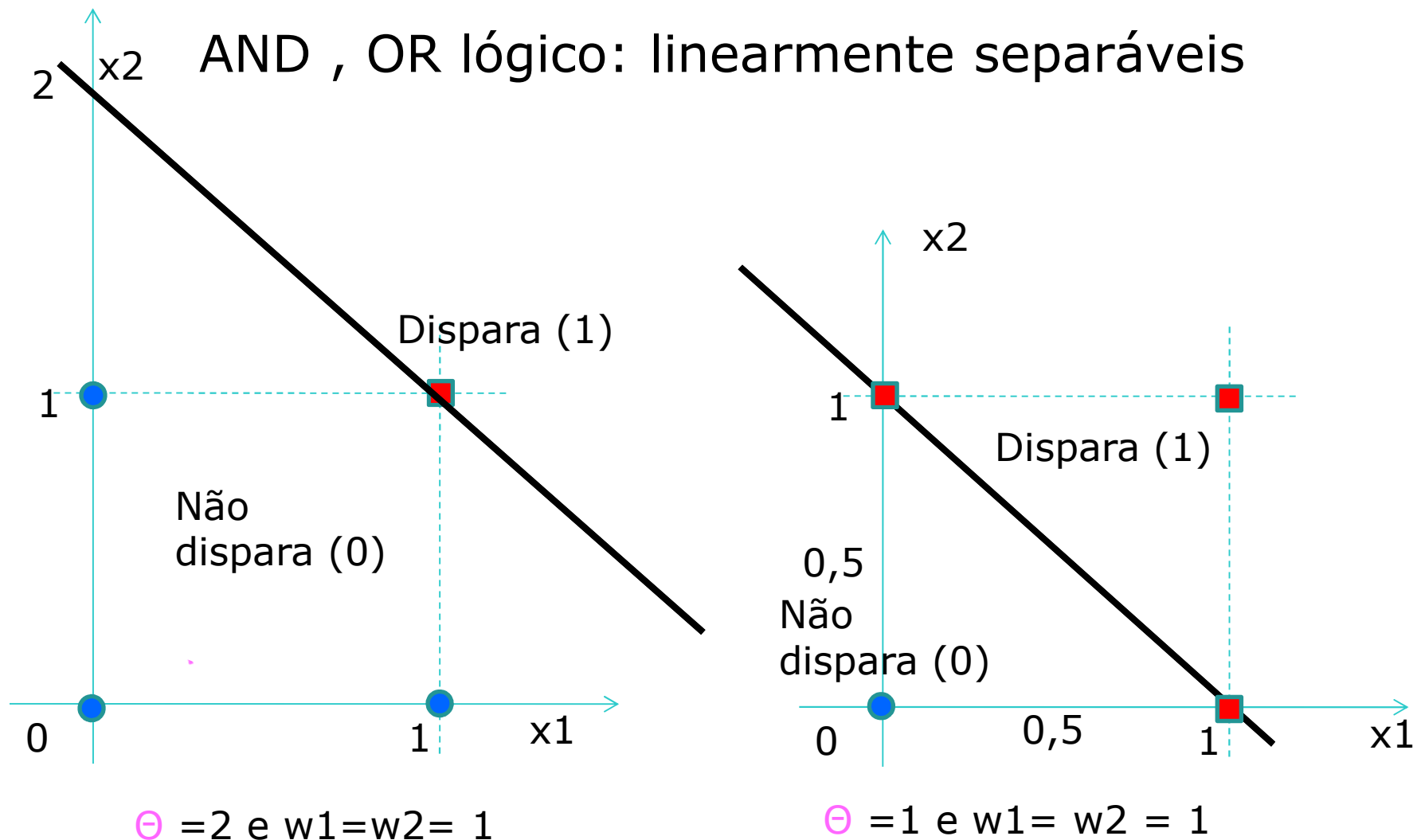


RN associativas



Prof. Myriam Delgado
Redes Neurais - UTFPR

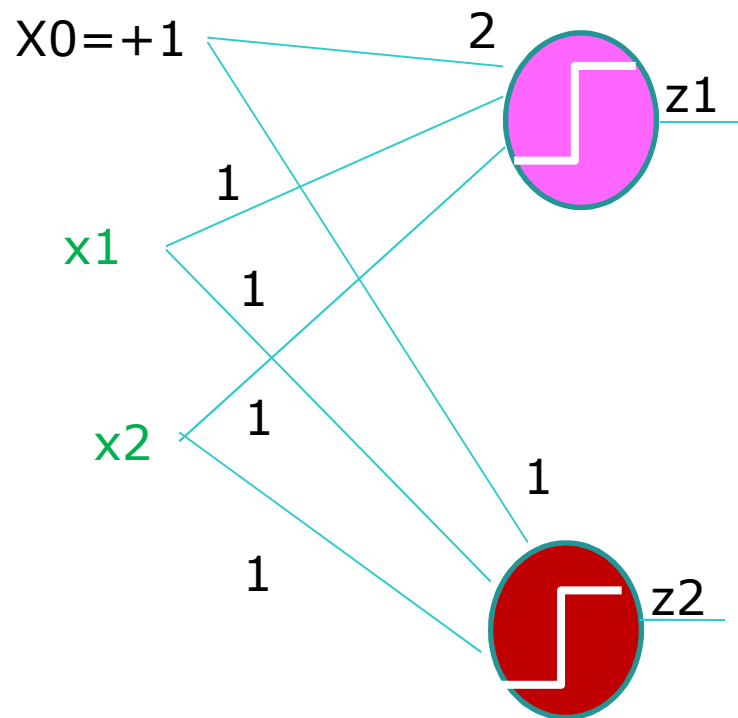
RNAs: MCP para AND e OR lógicos



RNAs: Perceptron (resolução de problemas)

AND lógico: MCP1 $Z1 = (x1 \text{ AND } x2)$

OR lógico: MCP2 $Z2 = (x1 \text{ OR } x2)$



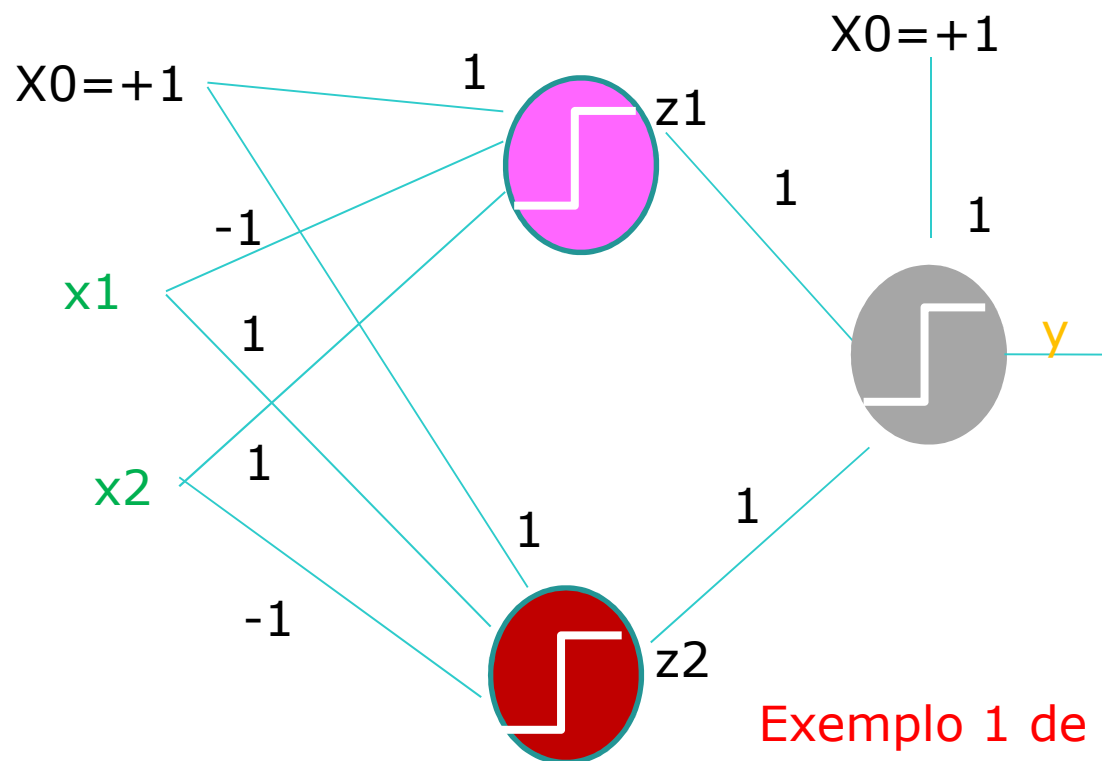
Perceptron

x1	x2	Z1 AND	Z2 OR
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

RNAs: MLP (resolução de problemas)

XOR lógico: PerceptronA + Mais uma camada

$$x1 \text{ XOR } x2 \leftrightarrow \underline{(\text{NOT } x1 \text{ AND } x2)} \text{ OR } \underline{(x1 \text{ AND NOT } x2)}$$



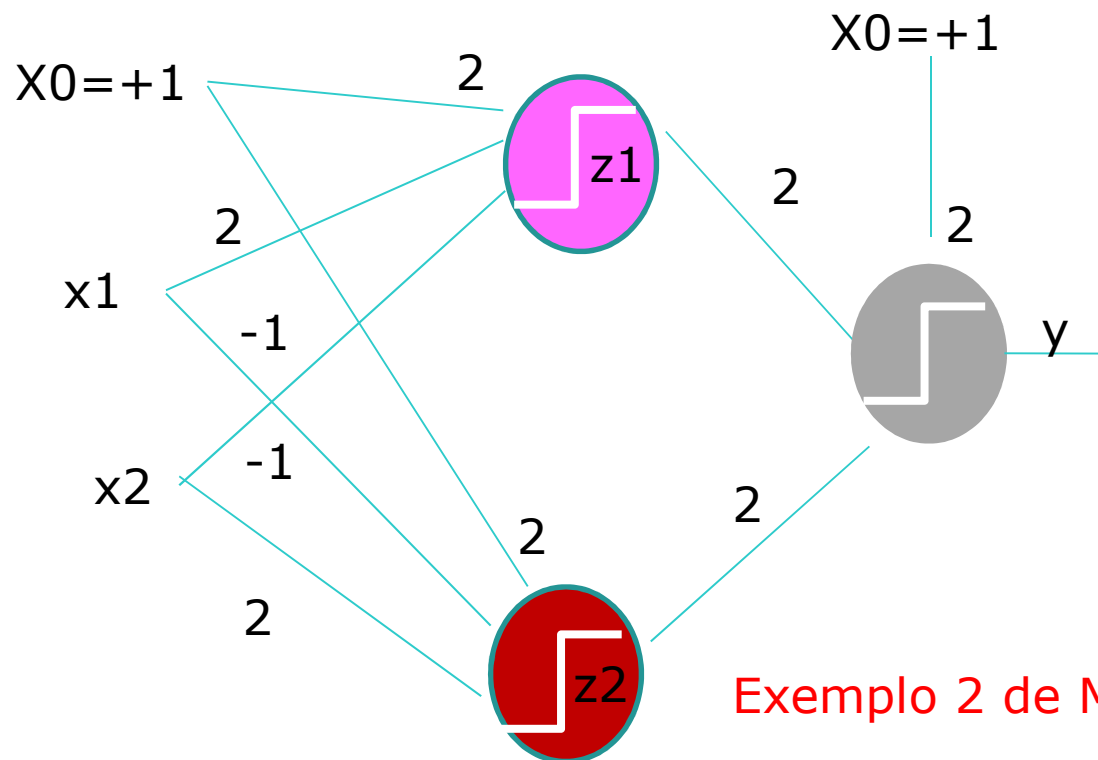
XOR		Lógico		
$x1$	$x2$	$z1$	$z2$	y
0	0	0	0	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0

Exemplo 1 de MLP

RNAs: MLP (resolução de problemas)

XOR lógico: PerceptronB + Mais uma camada

$$x1 \text{ XOR } x2 \leftrightarrow \underline{(x1 \text{ AND NOT } x2)} \text{ OR } \underline{(x2 \text{ AND NOT } x1)}$$



XOR		Lógico		
x1	x2	z1	z2	y
0	0	0	0	0
0	1	0	1	1
1	0	1	0	1
1	1	0	0	0

Exemplo 2 de MLP

RNAs: Neurônio MCP (resolução de problemas)

XOR lógico: não linearmente separável

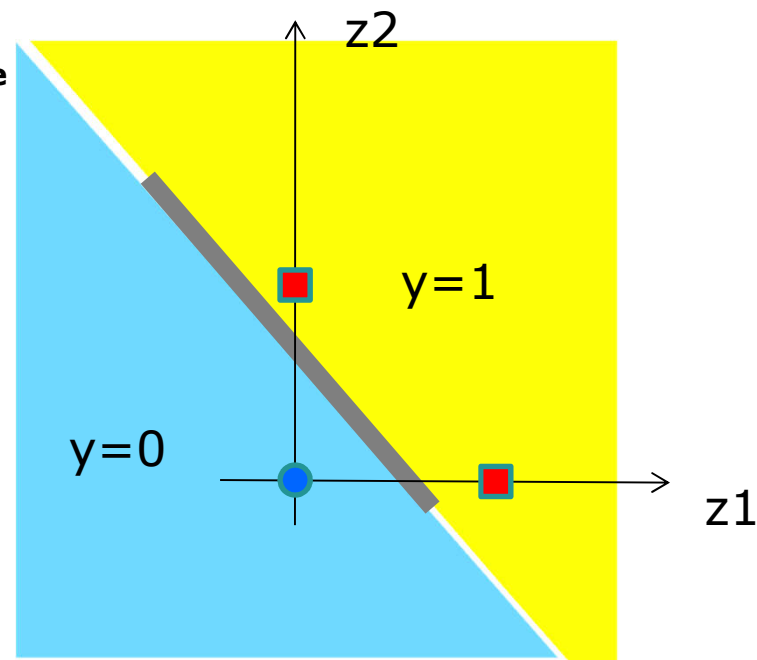
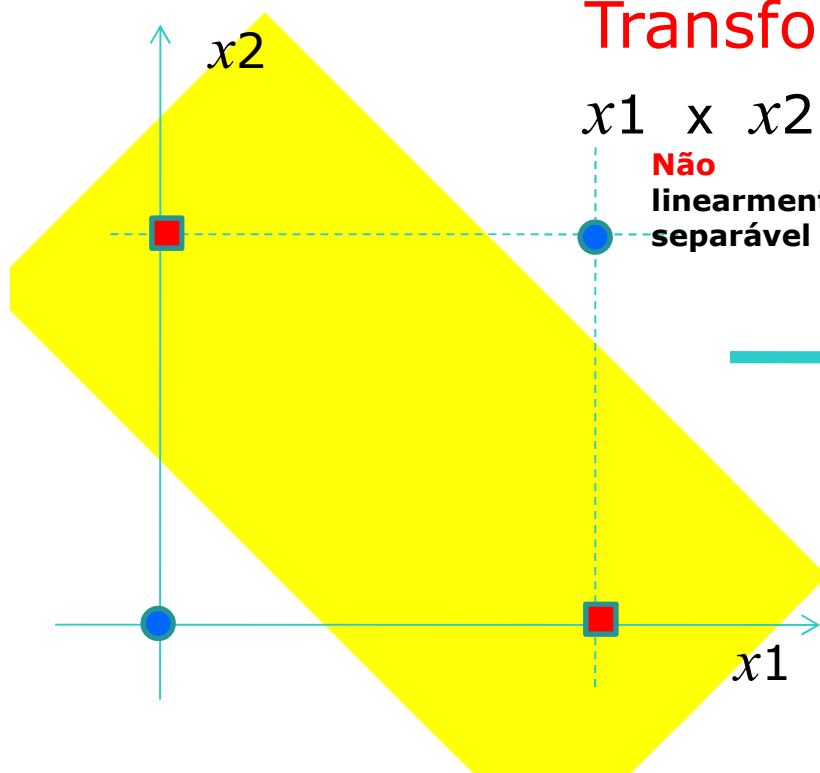
Por que as redes multicamadas são capazes de resolver XOR ?

Transformação de espaços

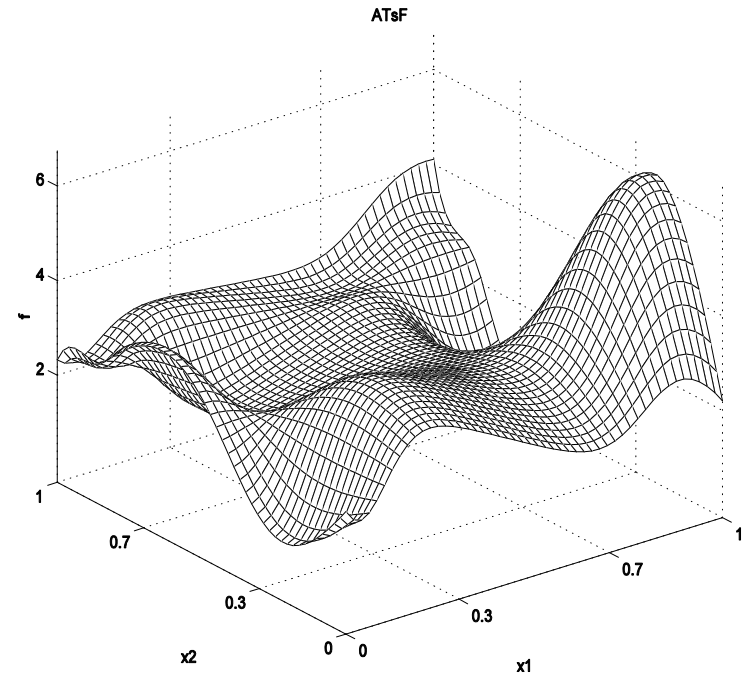
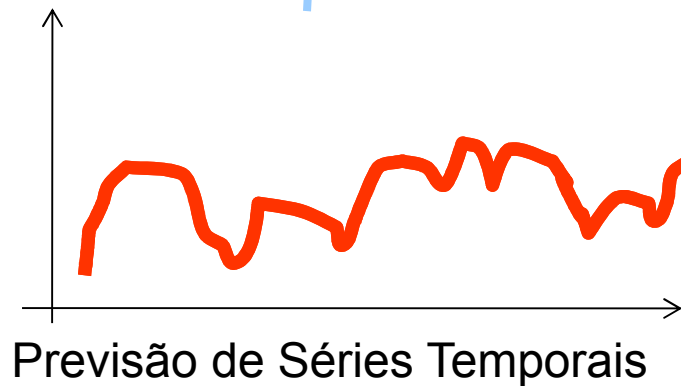
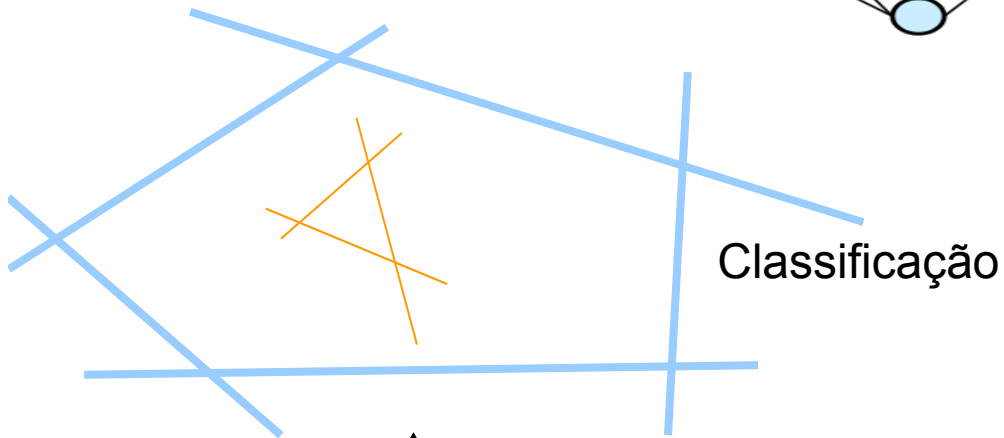
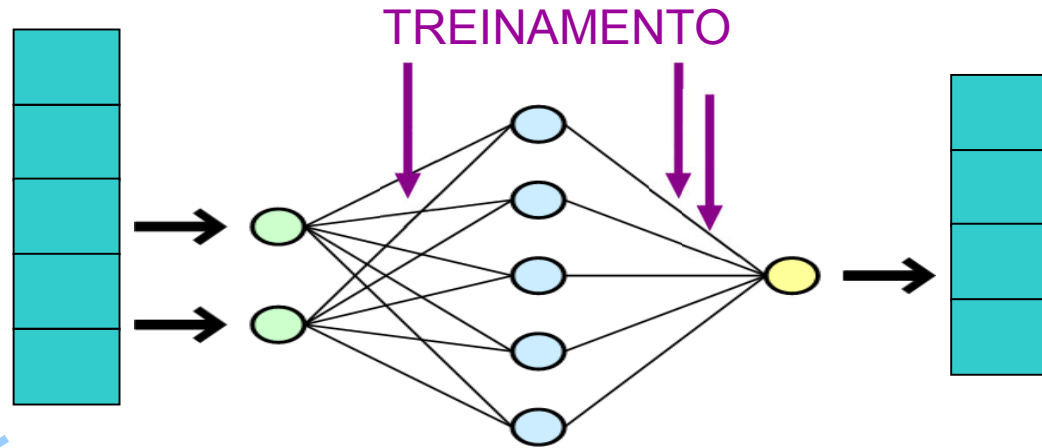
$$x1 \times x2 \rightarrow z1 \times z2$$

Não
linearmente
separável

linearmente
separável

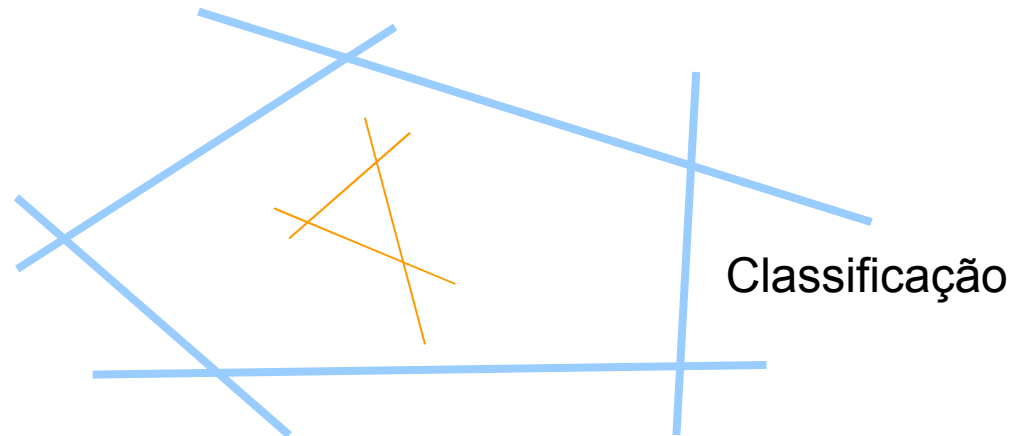
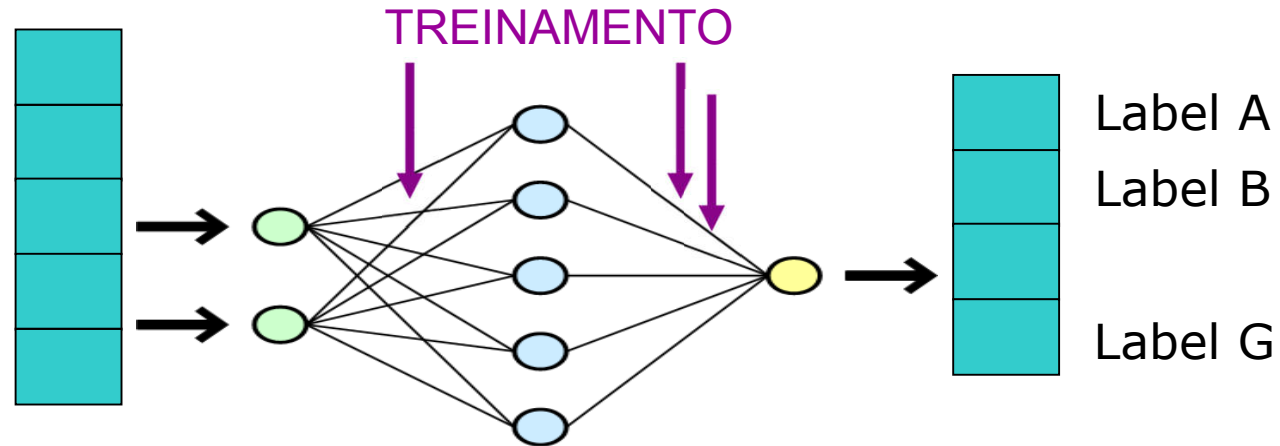


Solução por Redes Neurais (RN)

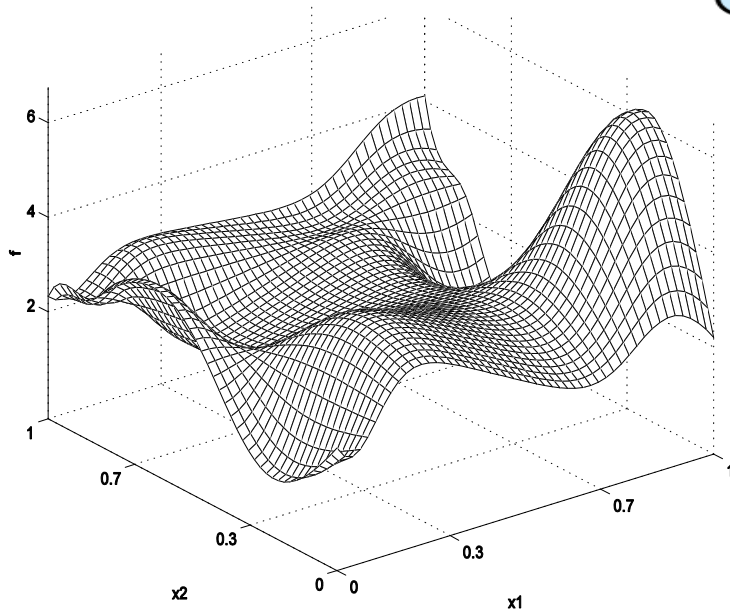
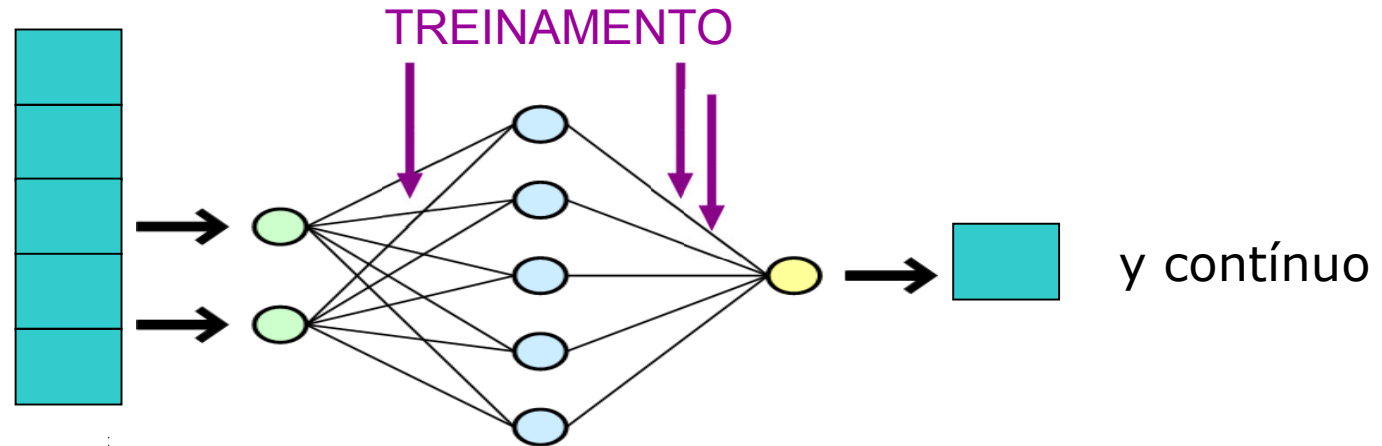


Aproximação de Funções

Solução por Redes Neurais (RN)

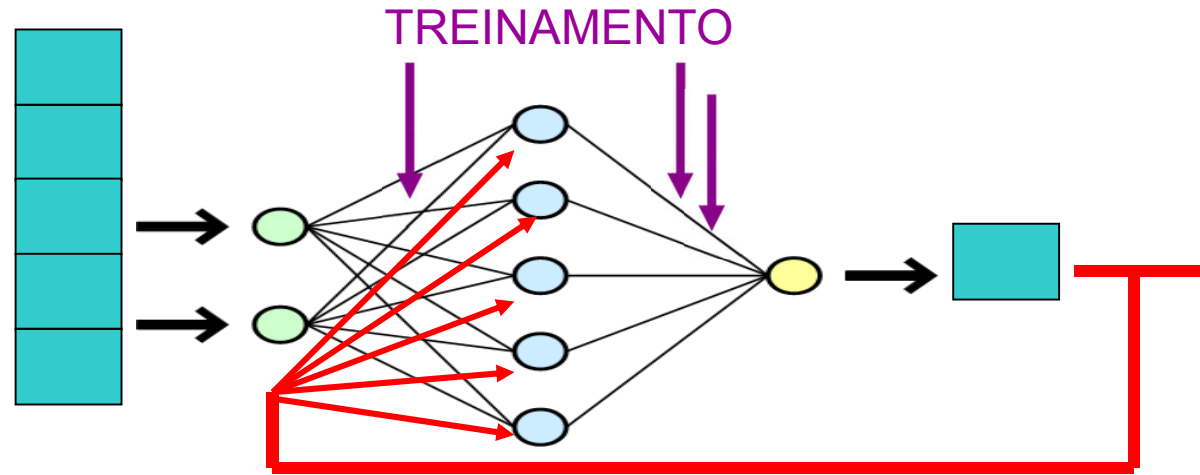


Solução por Redes Neurais (RN)

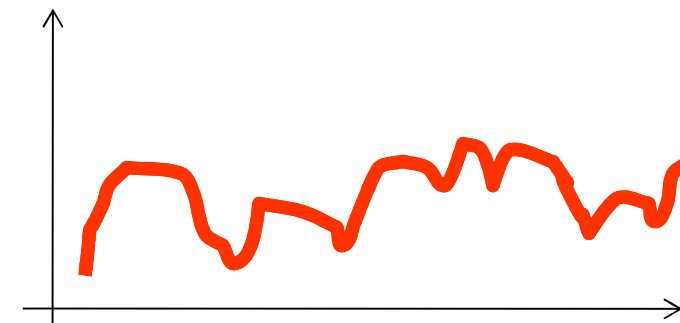


Aproximação de Funções

Solução por Redes Neurais (RN)



Redes Recorrentes



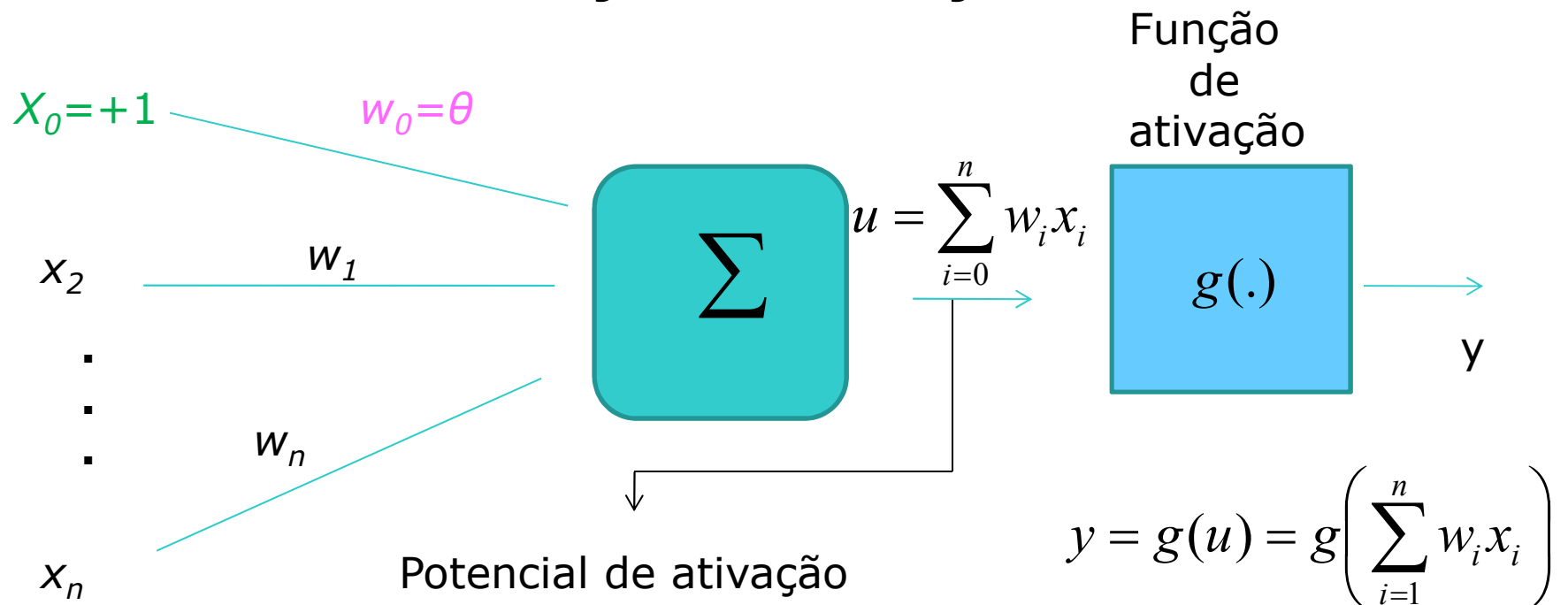
Previsão de Séries Temporais

Redes Neurais Artificiais

- Modelo do neurônio
- Topologia da rede
- Treinamento

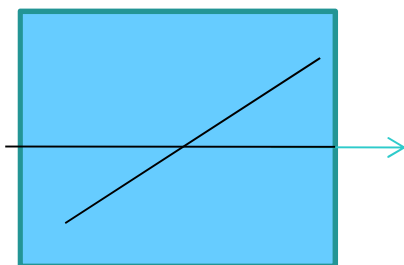
Neurônios: Funções de Ativação

A partir do Modelo MCP original, foram derivados vários outros modelos de neurônios que permitem uma saída qualquer através de **diferentes funções de ativação**



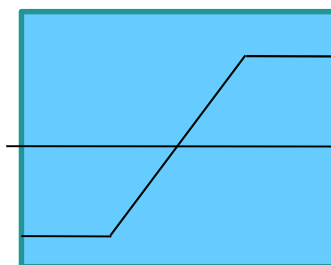
Neurônios: Funções de Ativação

Linear



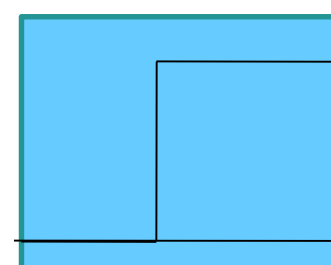
$$y \in [-\infty, +\infty]$$

Rampa



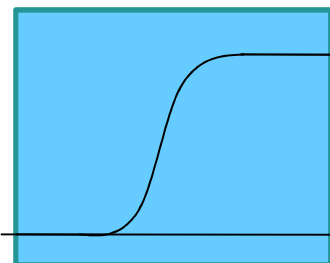
$$y \in [-1, +1]$$

Degrau



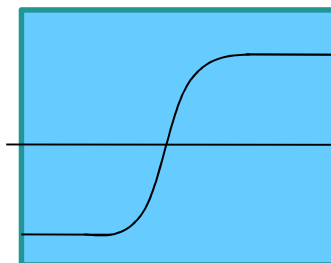
$$y \in \{0, +1\}$$

Sigmoide



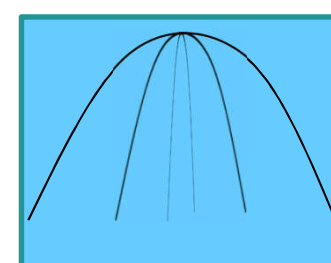
$$y \in [0, +1]$$

Tg Hiperbólica



$$y \in [-1, +1]$$

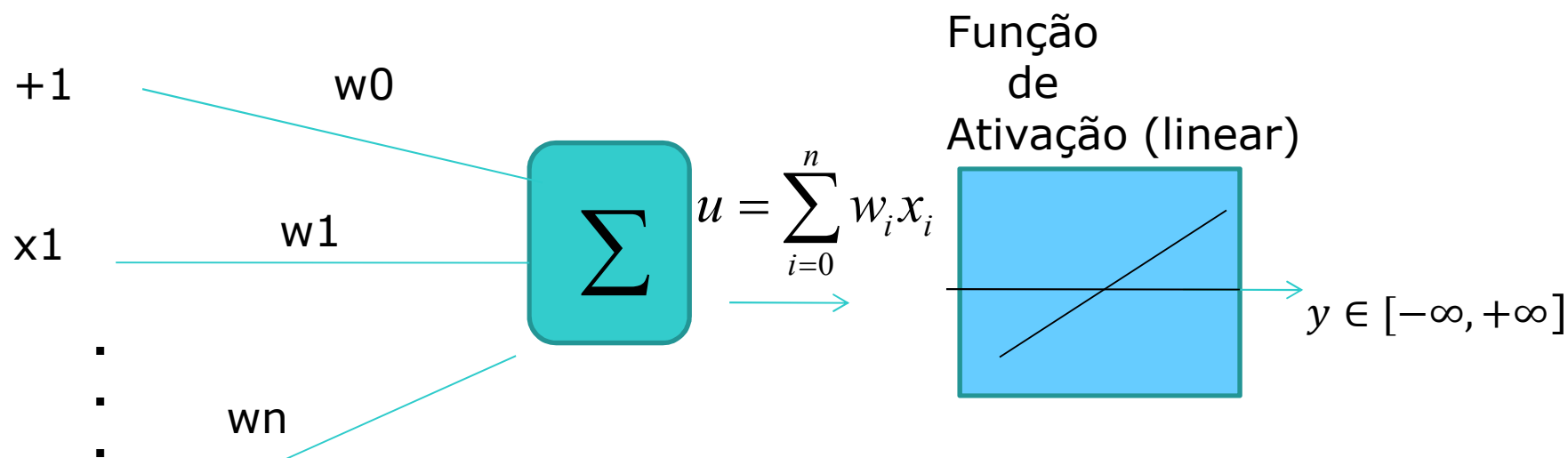
Base Radial



$$y \in [0, +1] \text{ p } N$$

Neurônio: Função de Ativação

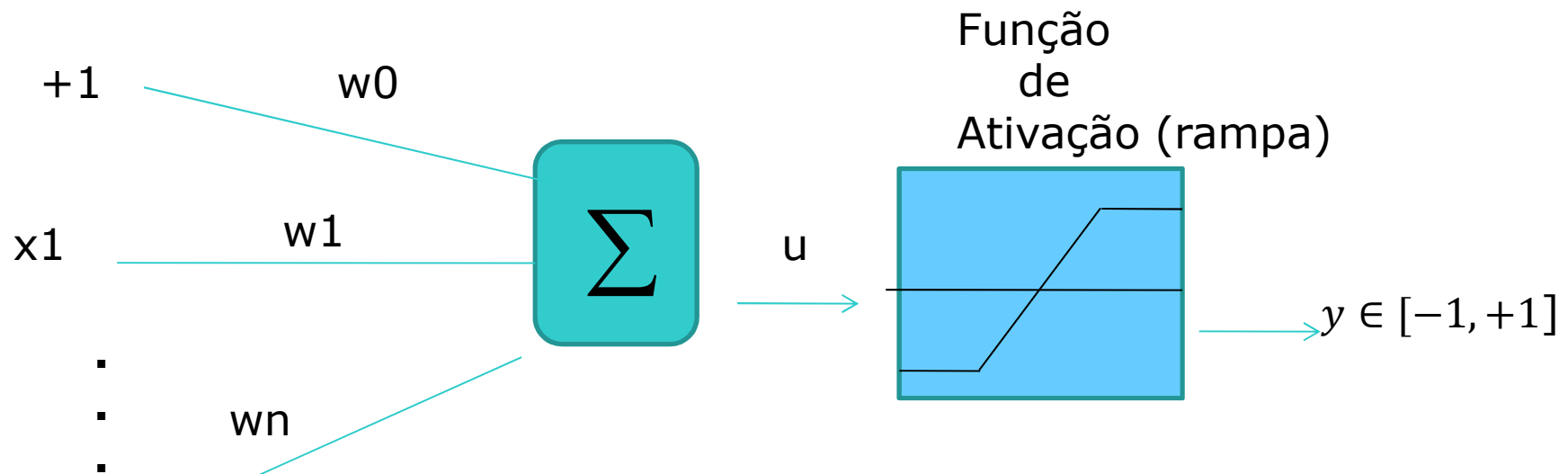
f = função linear



$$y = \left(\sum_{i=0}^n w_i x_i \right) = u$$

Neurônio: Função de Ativação

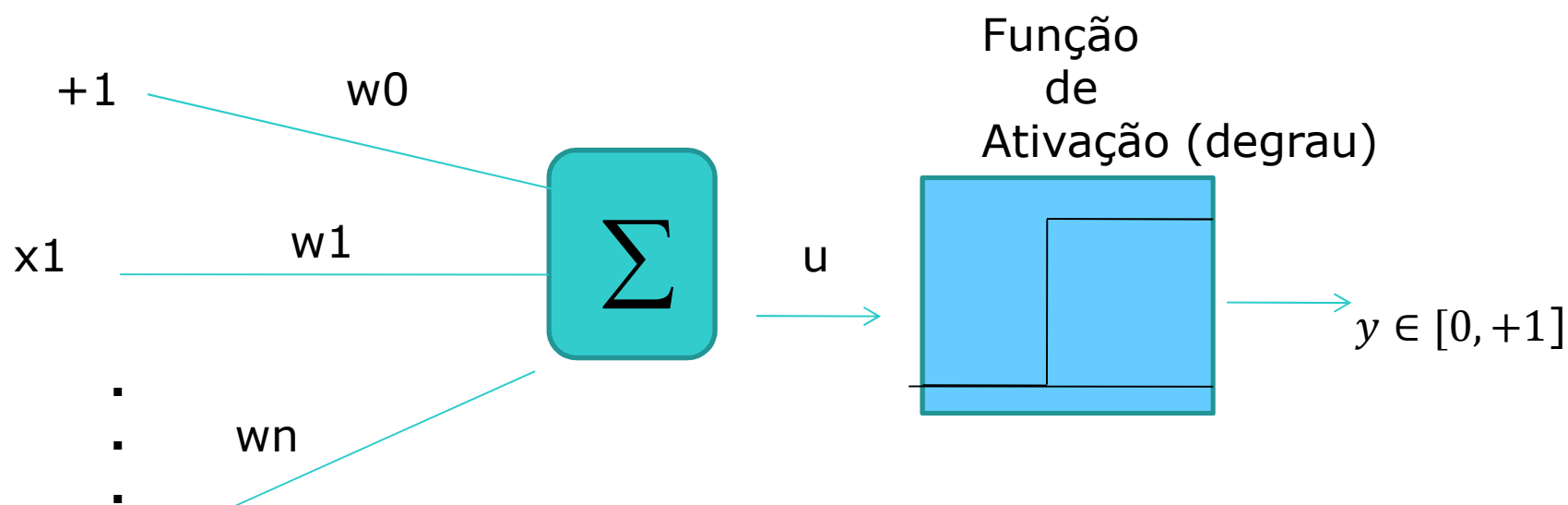
f = função rampa



$$y = \begin{cases} +1 & \text{se } u \geq 1 \\ u & \text{se } |u| < 1, \\ -1 & \text{se } u \leq -1 \end{cases} \quad \text{onde } u = \left(\sum_{i=0}^n w_i x_i \right)$$

Neurônio: Função de Ativação

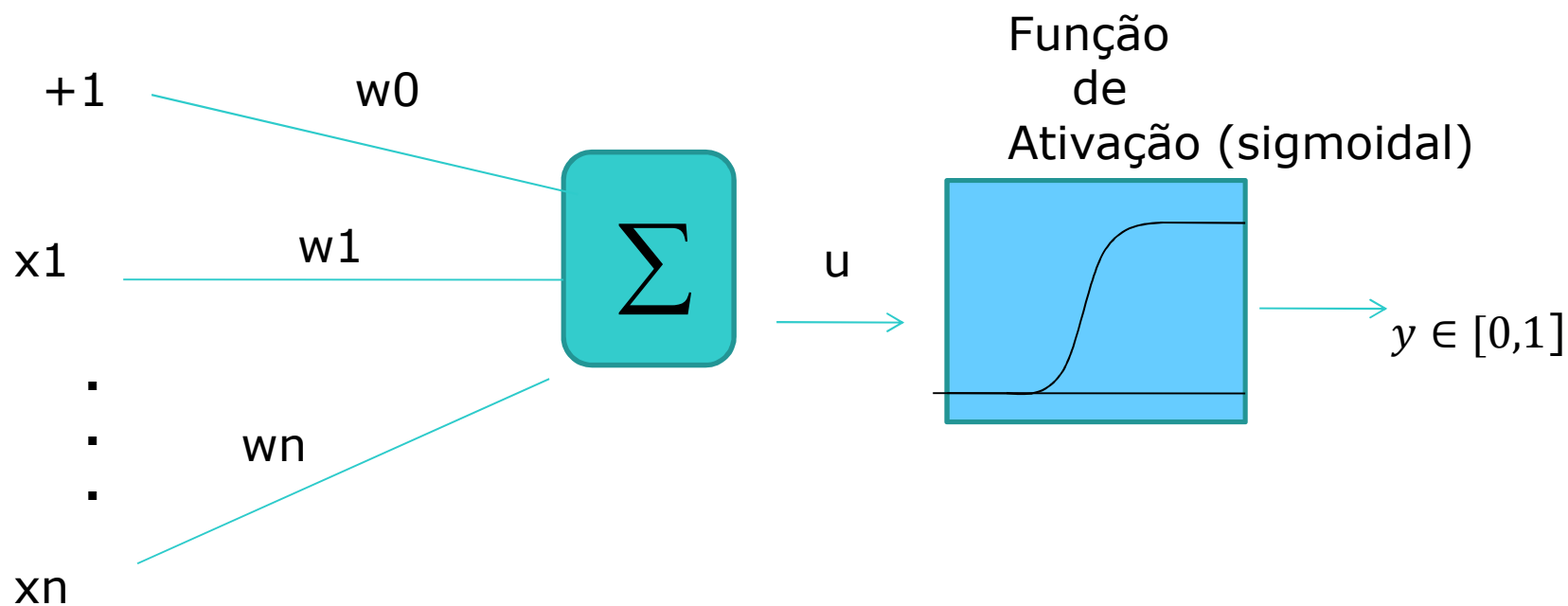
f = função degrau



$$y = \begin{cases} 1 & \text{se } u \geq 0 \\ 0 & \text{se } u < 0 \end{cases}, \quad \text{onde } u = \left(\sum_{i=0}^n w_i x_i \right)$$

Neurônio: Função de Ativação

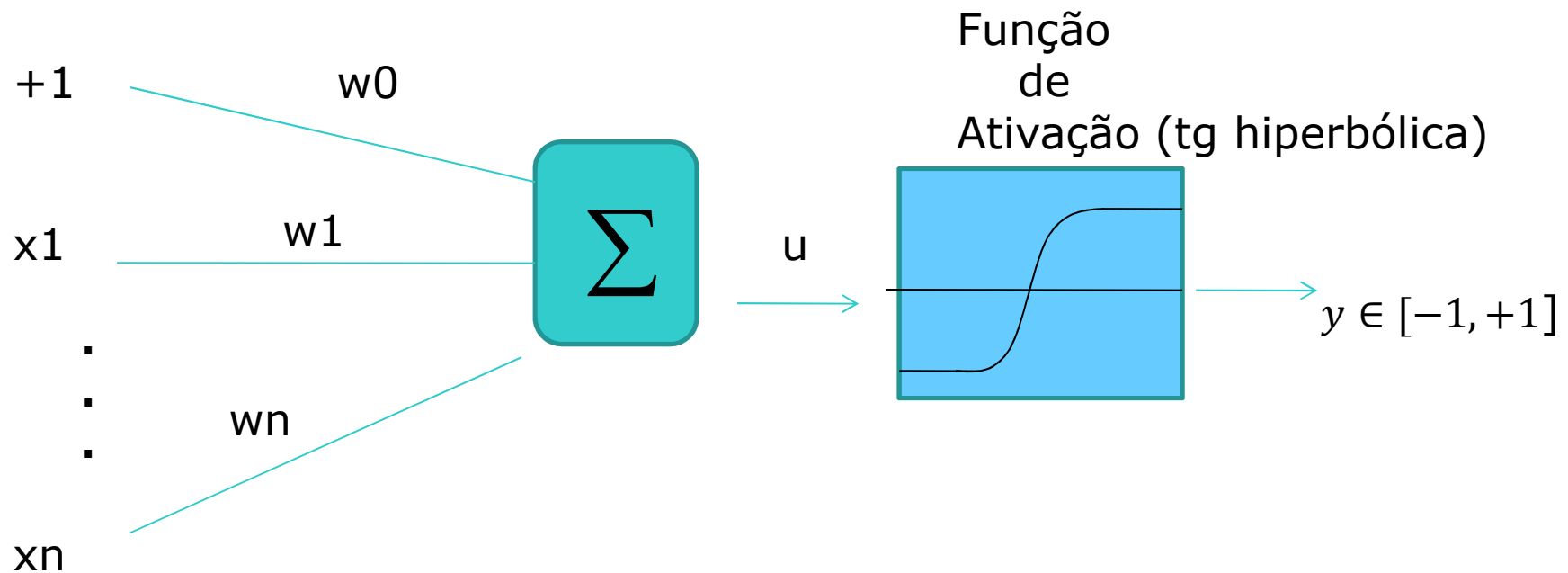
f = função sigmoidal



$$y = \frac{1}{1 + e^{(-u)}} \quad \text{onde} \quad u = \left(\sum_{i=0}^n w_i x_i \right)$$

Neurônio: Função de Ativação

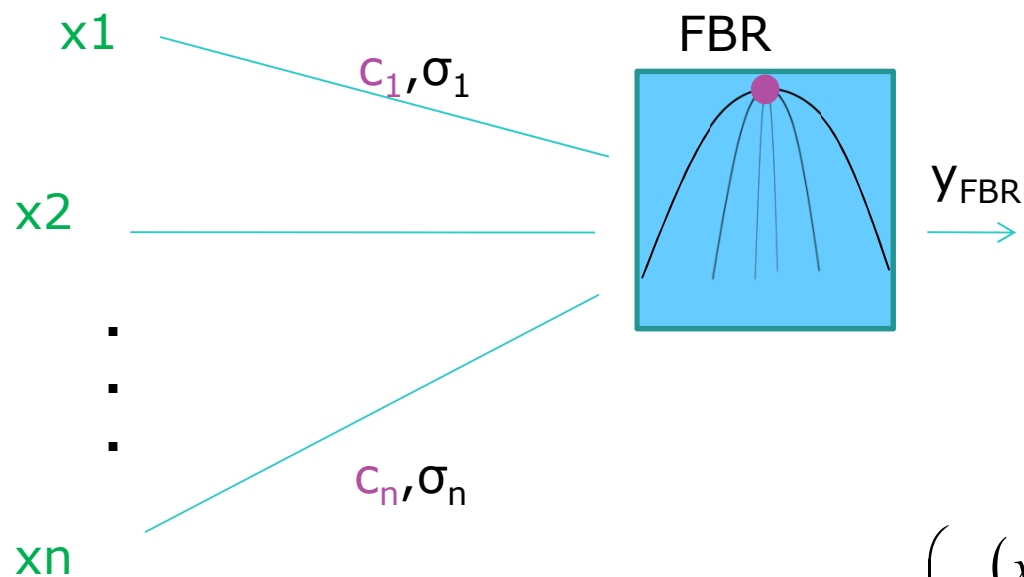
f = função tangente hiperbólica



$$y = \frac{e^{(u)} - e^{(-u)}}{e^{(u)} + e^{(-u)}}, \quad \text{onde } u = \left(\sum_{i=0}^n w_i x_i \right)$$

Neurônio: Função de Ativação

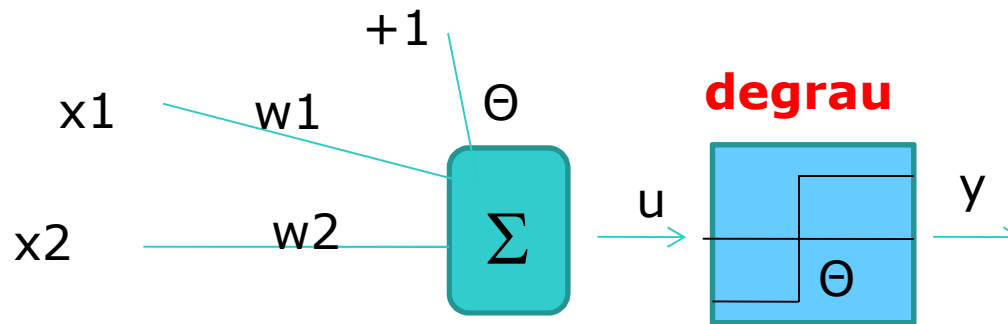
f = Função de Base Radial (FBR)



FRB avalia a distância do vetor de **entradas** ao **centro** da função:
Mais perto: $y \rightarrow 1$
Mais distante: $y \rightarrow 0$

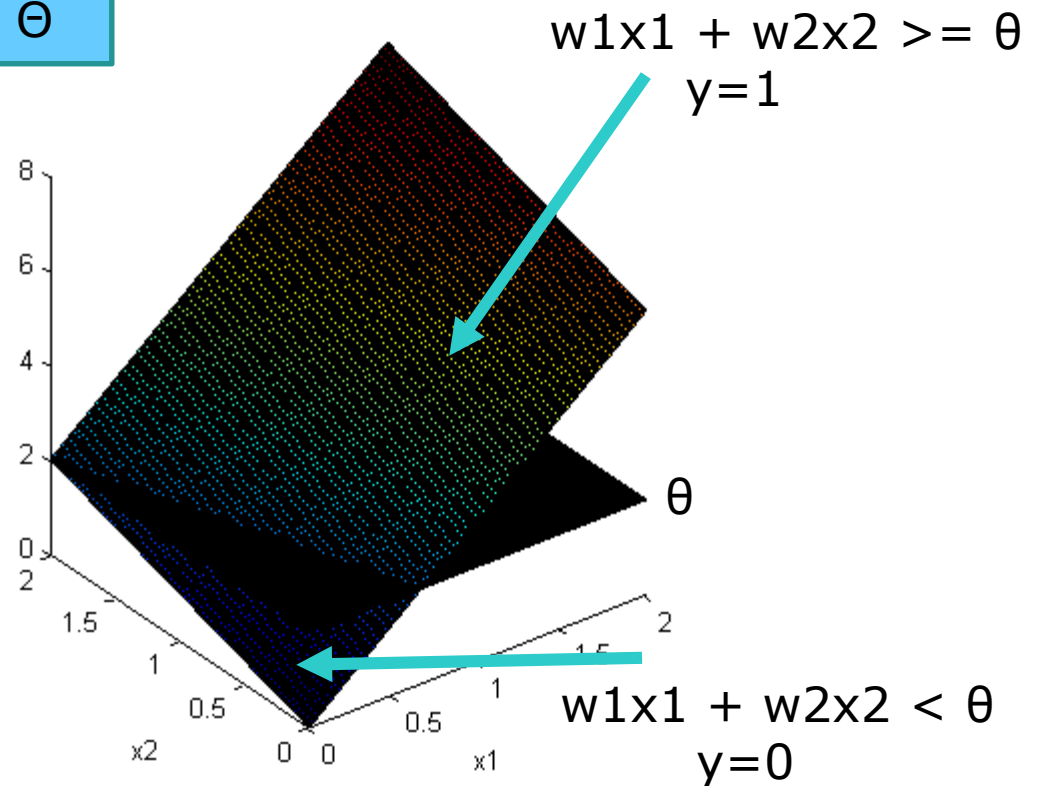
$$y_{FBR} = e^{\left(-\frac{(x_1 - c_1)^2}{\sigma_1} - \frac{(x_2 - c_2)^2}{\sigma_2} - \dots - \frac{(x_n - c_n)^2}{\sigma_n} \right)}$$

Funções de Ativação: mapeamento x_1 x x_2

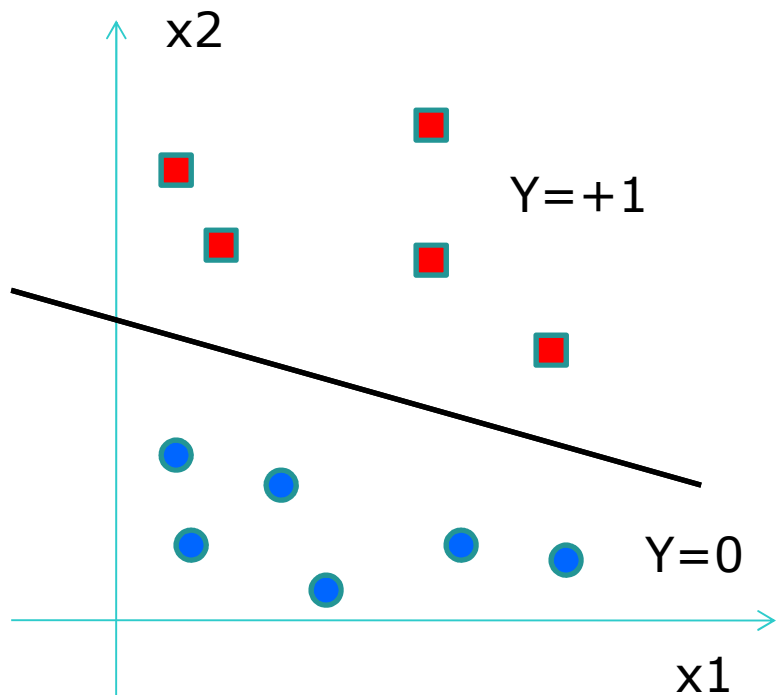


Mapeamento:
entrada – saída do neurônio

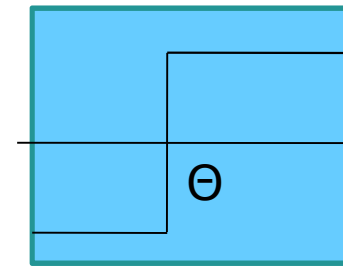
$$y = \begin{cases} 1 & \text{se } w_1x_1 + w_2x_2 \geq \theta \\ 0 & w_1x_1 + w_2x_2 < \theta \end{cases}$$



Funções de Ativação: treinamento



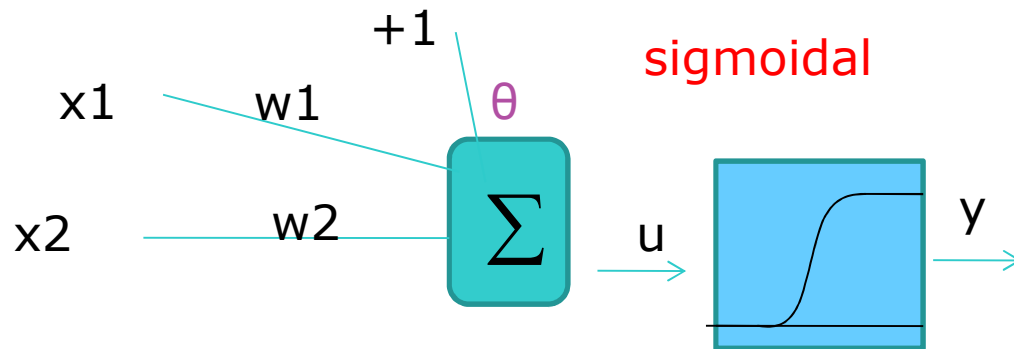
degrau



$$y = \begin{cases} +1 & \text{se } \sum_{i=1}^n w_i x_i \geq \theta \\ 0 & \text{se } \sum_{i=1}^n w_i x_i < \theta \end{cases}$$

O ajuste dos parâmetros \mathbf{w} e Θ (treinamento) altera a posição da reta (**hiperplano** para mais de duas entradas) e portanto da partição no espaço de entrada (**N-dimensional**)

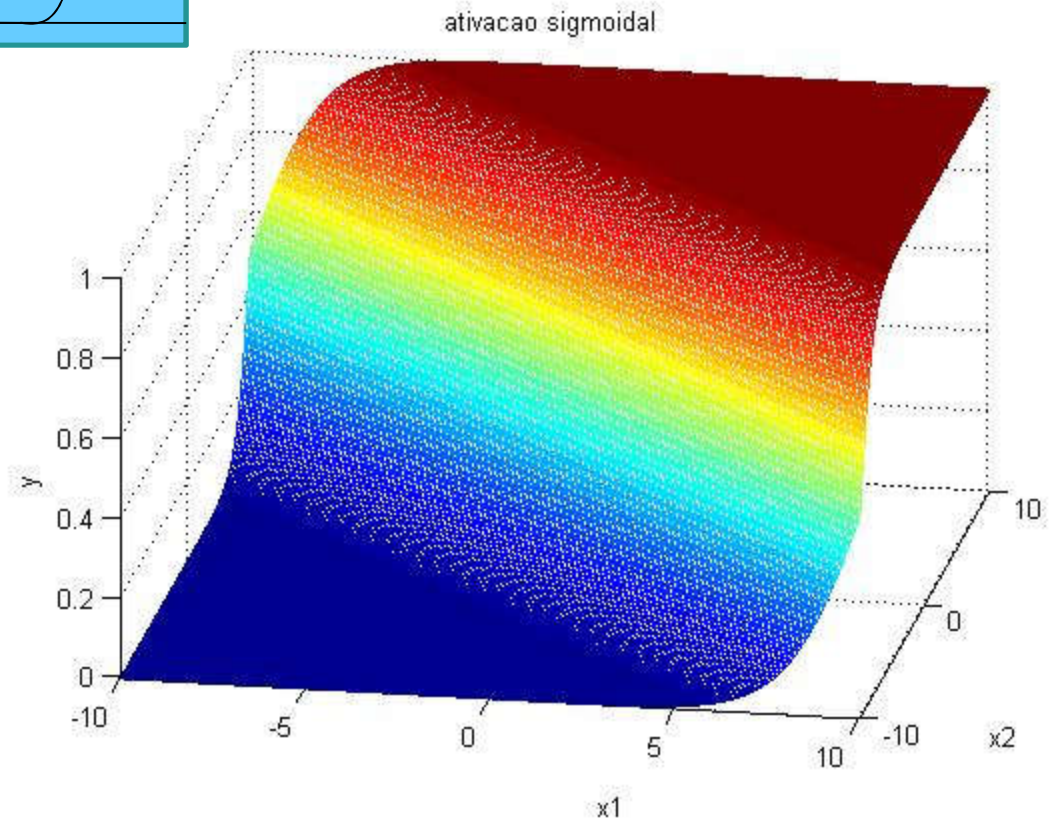
Funções de Ativação: mapeamento $x_1 \times x_2$



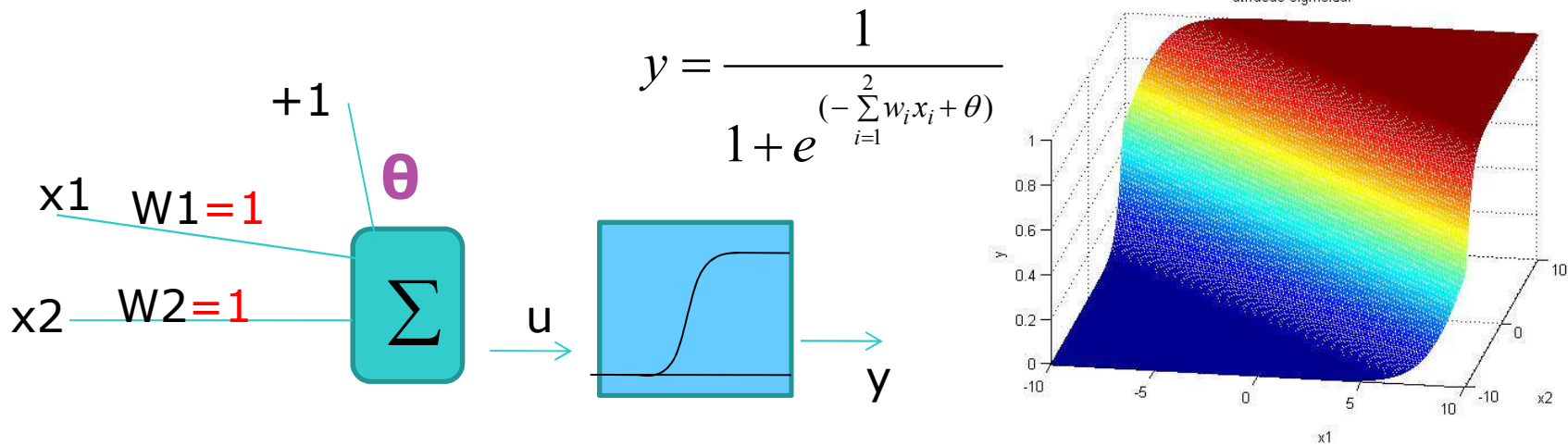
Mapeamento:
entrada – saída do neurônio

$$y = \frac{1}{1 + e^{(-u)}}$$

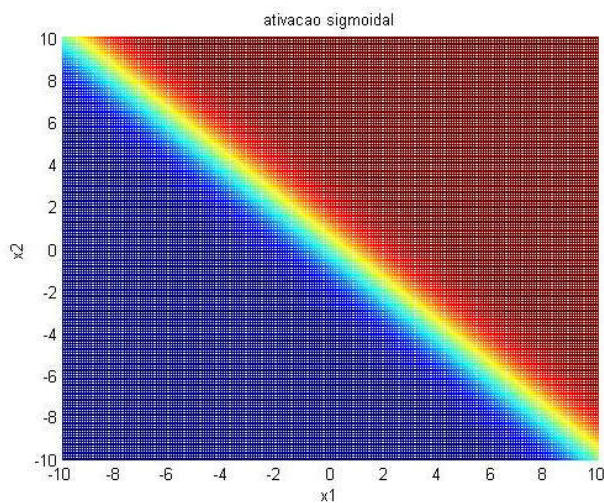
onde $u = \left(\sum_{i=1}^2 w_i x_i + \theta \right)$



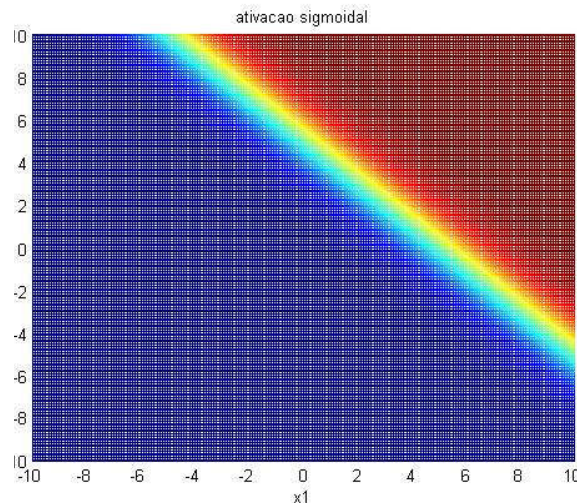
Funções de Ativação: mapeamento x_1 x x_2



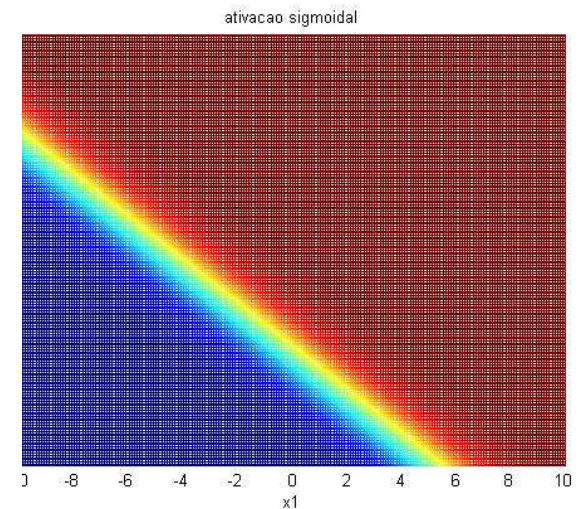
$\Theta = 0$



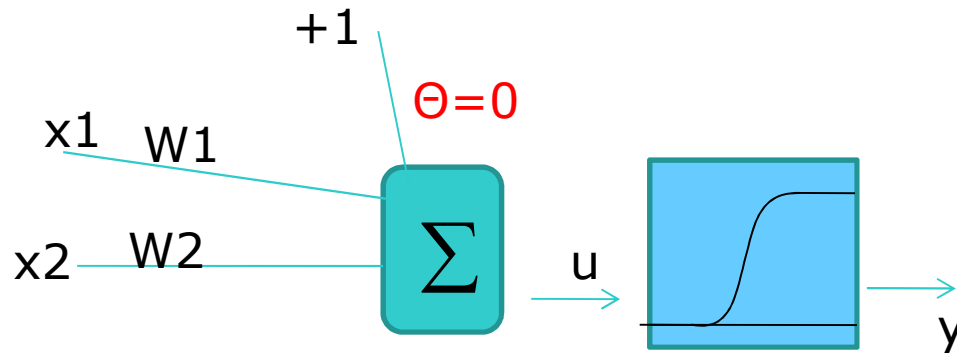
$\Theta = -5$



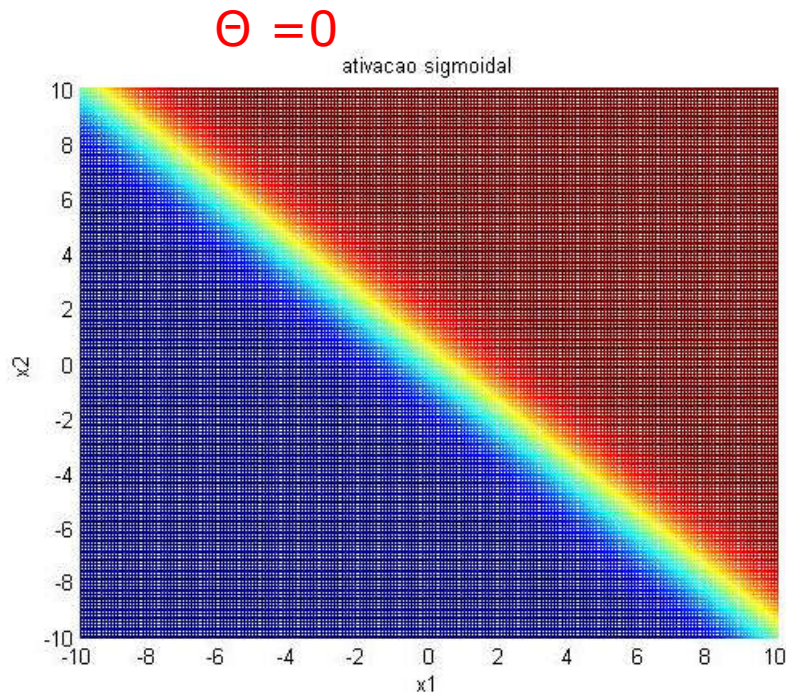
$\Theta = +5$



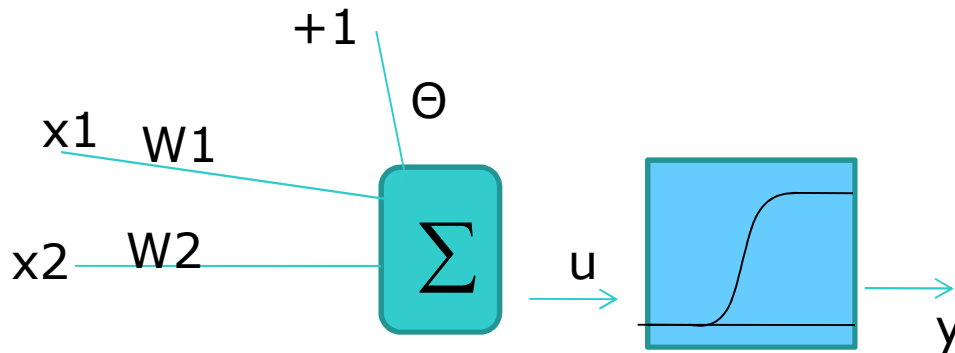
Funções de Ativação: Abstração



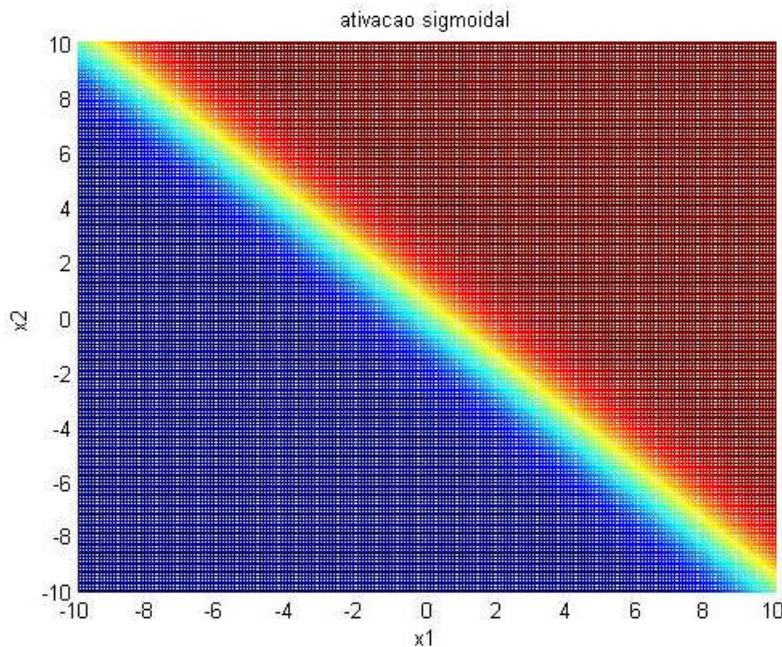
O que acontece se alterarmos os pesos w_1, w_2 ?



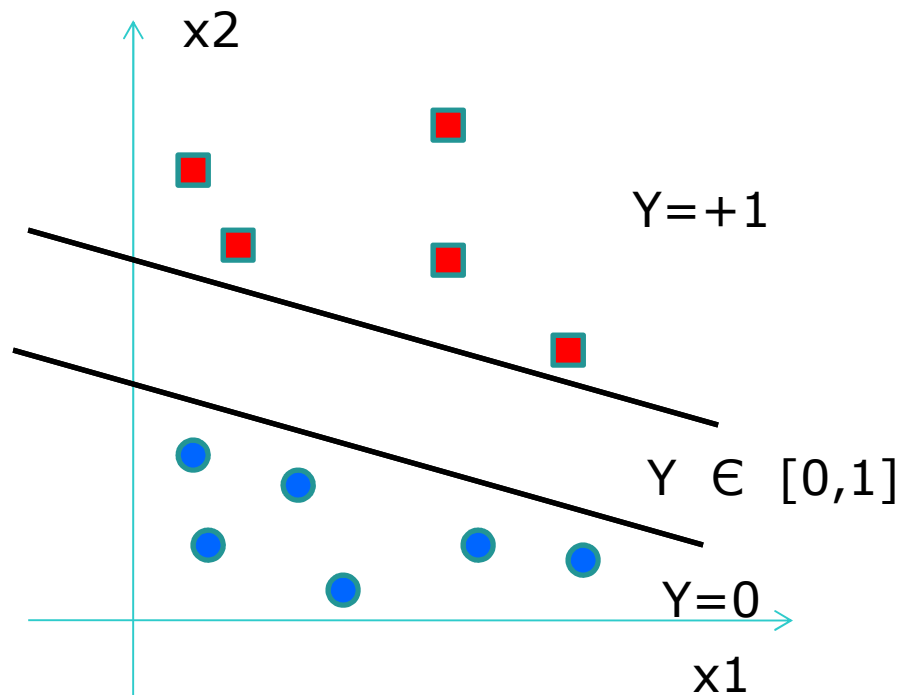
Funções de Ativação: Abstração



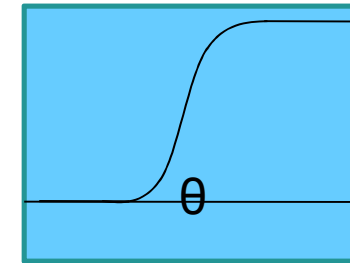
O que acontece se alterarmos o limiar Θ , e os Pesos w_1 , w_2 ?



Funções de Ativação: treinamento->map



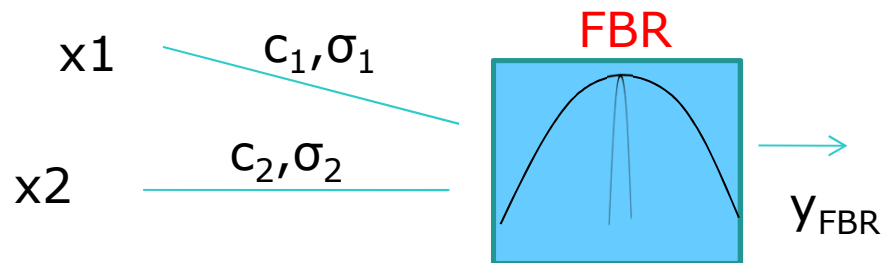
sigmoidal



$$y = \frac{1}{1 + e^{(-\gamma \sum_{i=0}^n w_i x_i)}}$$

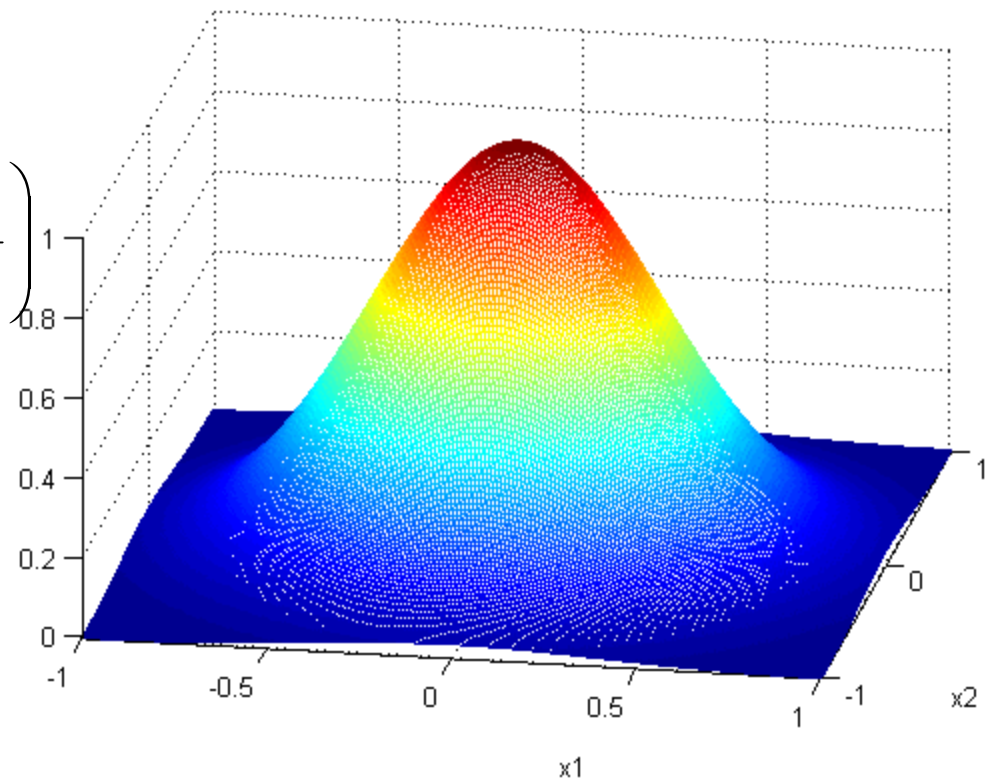
O ajuste dos parâmetros \mathbf{w} e θ (treinamento) altera o mapeamento entrada saída

Funções de Ativação: mapeamento x_1 x x_2

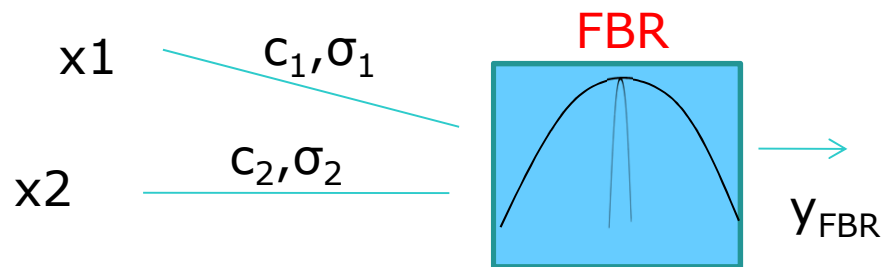


Mapeamento:
entrada – saída do neurônio

$$y_{FBR} = e^{\left(-\frac{(x_1 - c_1)^2}{\sigma_1} - \frac{(x_2 - c_2)^2}{\sigma_2} \right)}$$

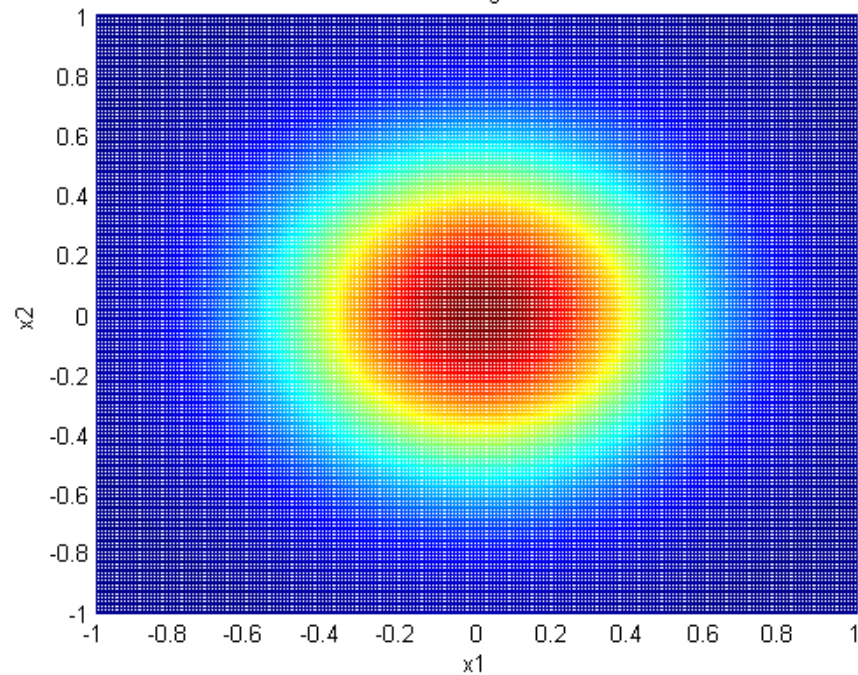


Funções de Ativação: mapeamento $x1 \times x2$



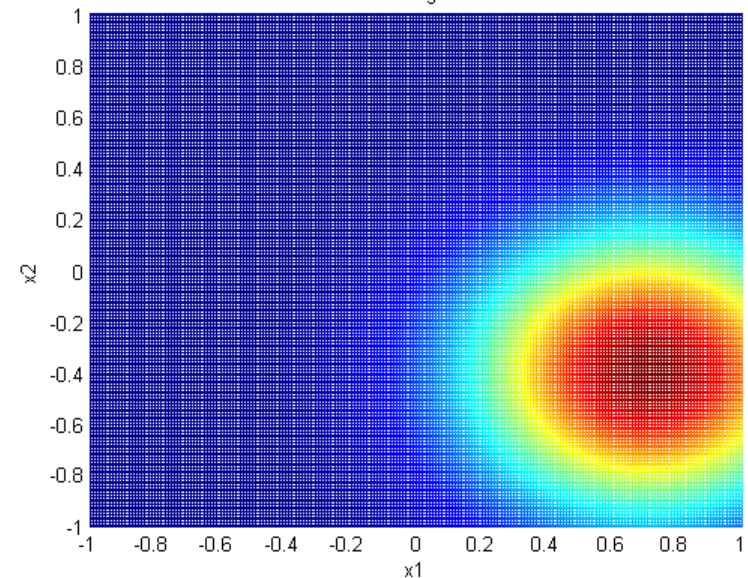
$$c_1 = c_2 = 0$$

ativacao sigmoidal

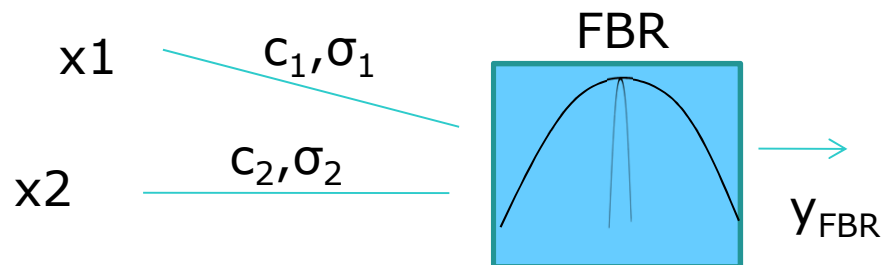


$$c_1 = 0,7$$
$$c_2 = -0,4$$

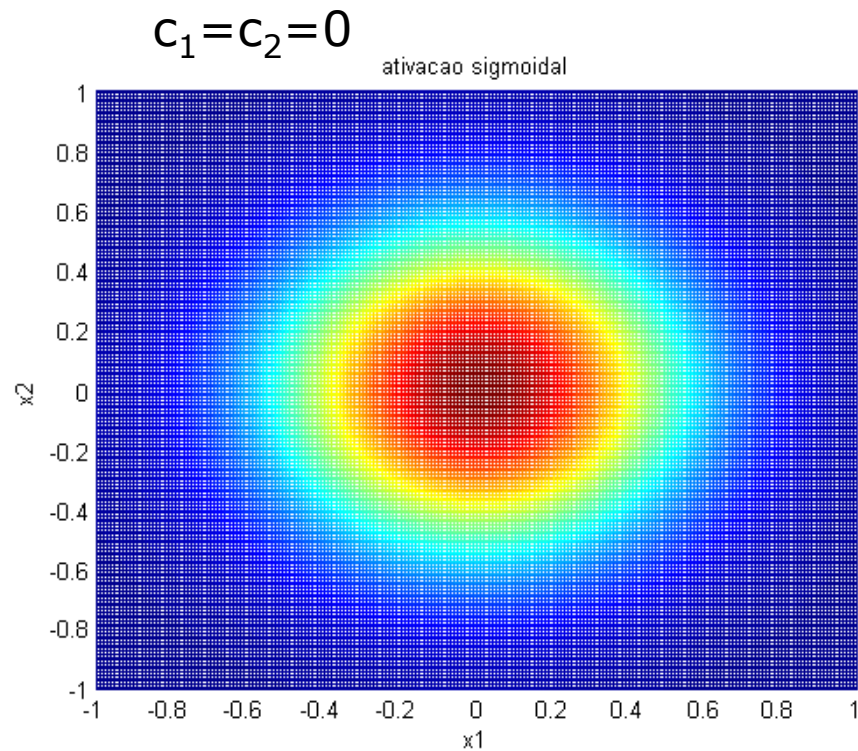
ativacao sigmoidal



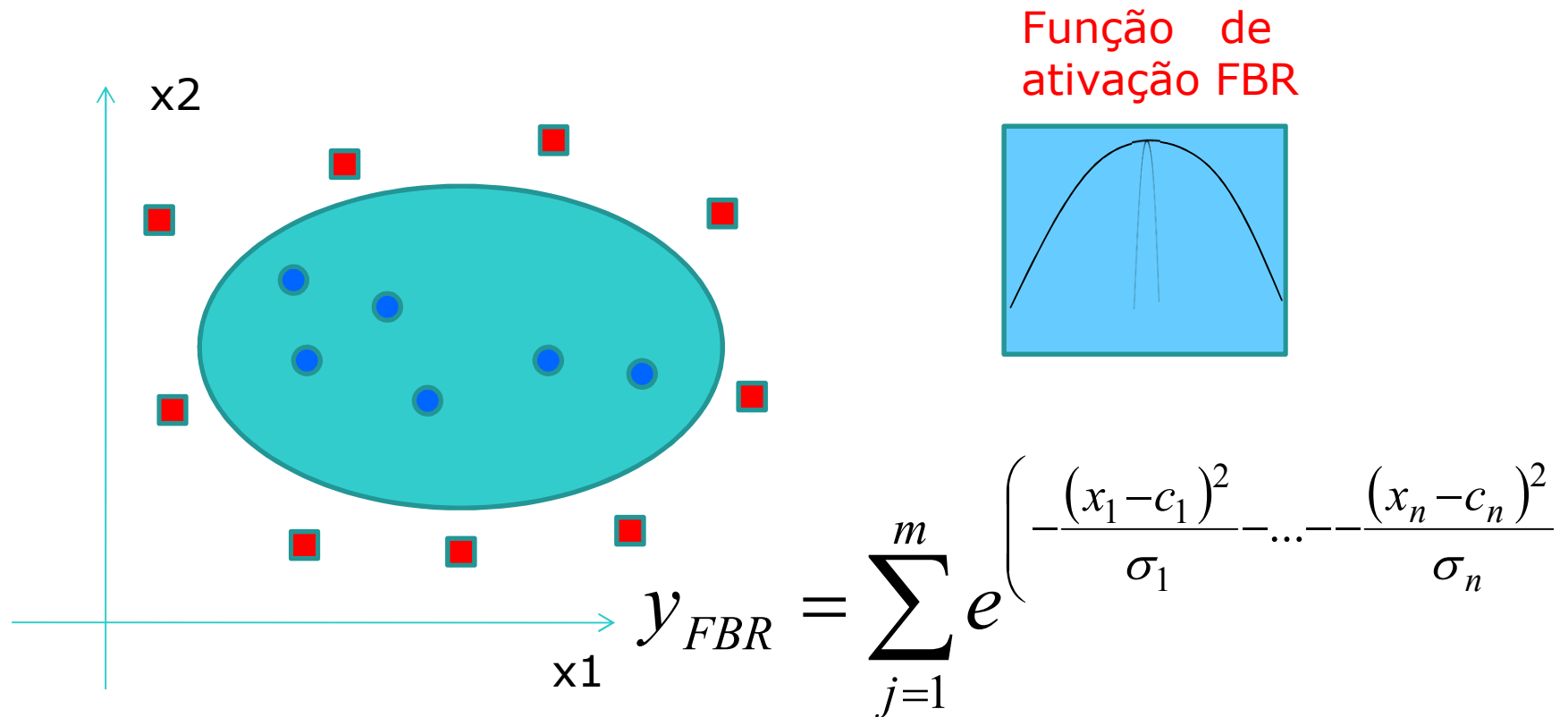
Funções de Ativação: Abstração



O que acontece se alterarmos os parâmetros σ_1 e σ_2 ?

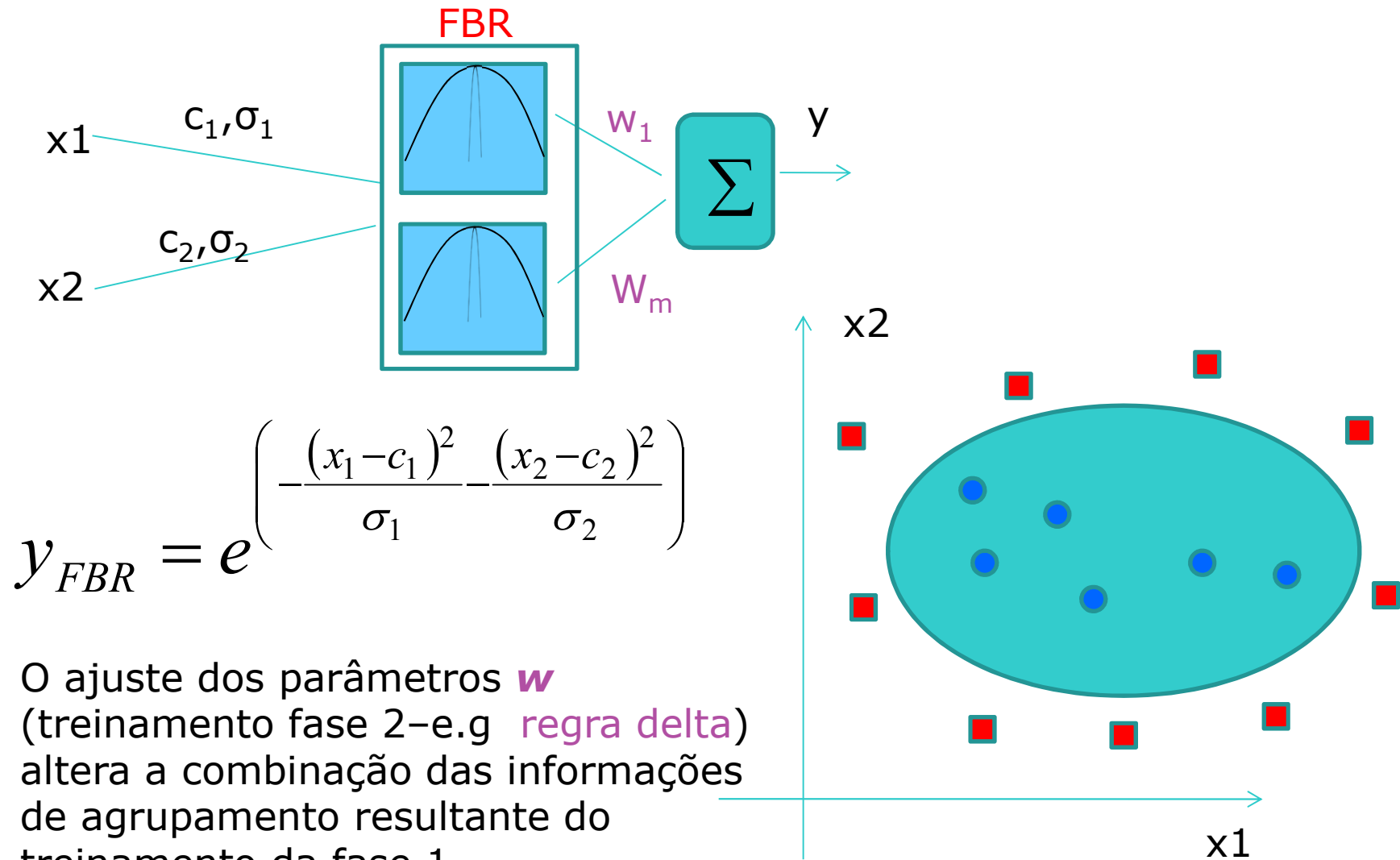


Funções de Ativação: treinamento->map



O ajuste dos parâmetros \mathbf{c} e $\boldsymbol{\sigma}$ (treinamento fase 1– e.g. clustering) altera a posição e cobertura da elipse (**hiperelipsóide**) e portanto o agrupamento no espaço de entrada (**N-dimensional**)

Funções de Ativação: treinamento->map



Redes Neurais Artificiais

- Modelo do neurônio: função de ativação
- **Topologia da rede**
- Treinamento

Redes Neurais Artificiais

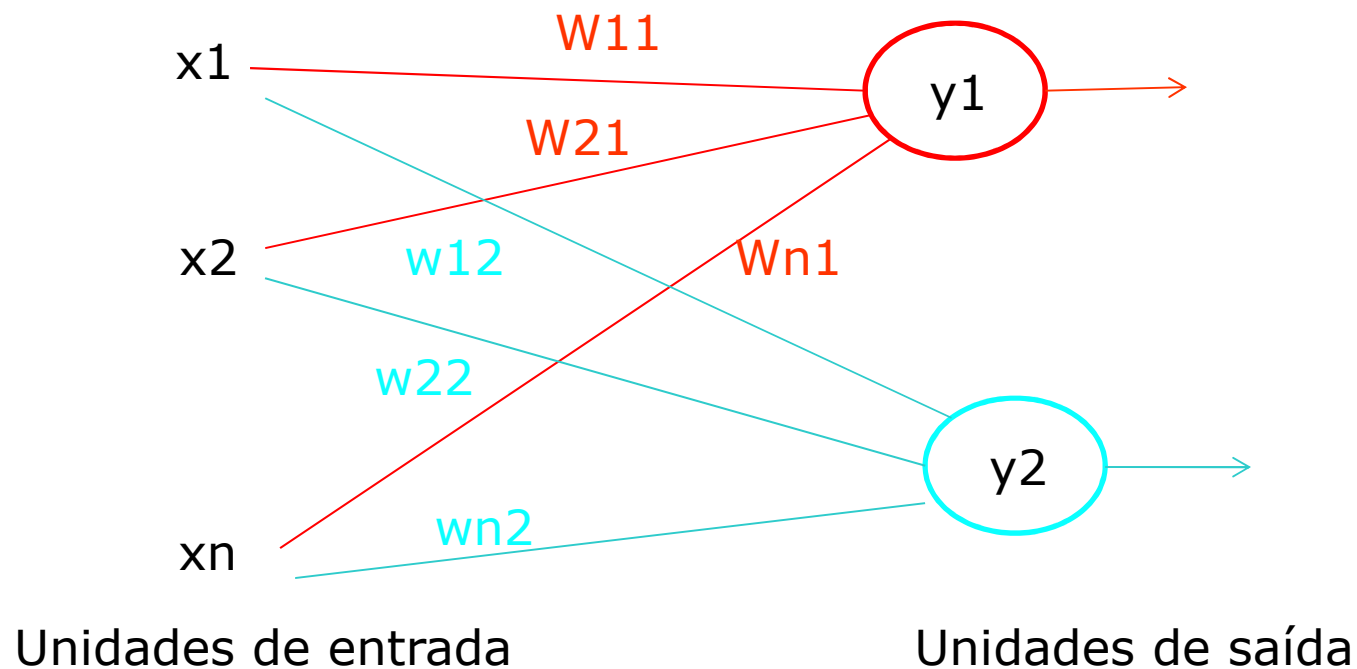
- Modelo do neurônio
- **Topologia da rede**
- **Diferentes modelos:**
 - função de ativação
 - estrutura
- Treinamento

Redes Neurais Artificiais: Arquitetura

- Parâmetros que definem a arquitetura:
 - Número de camadas
 - Número de neurônios em cada camada
 - Tipo de conexão entre os neurônios
 - Conectividade da rede

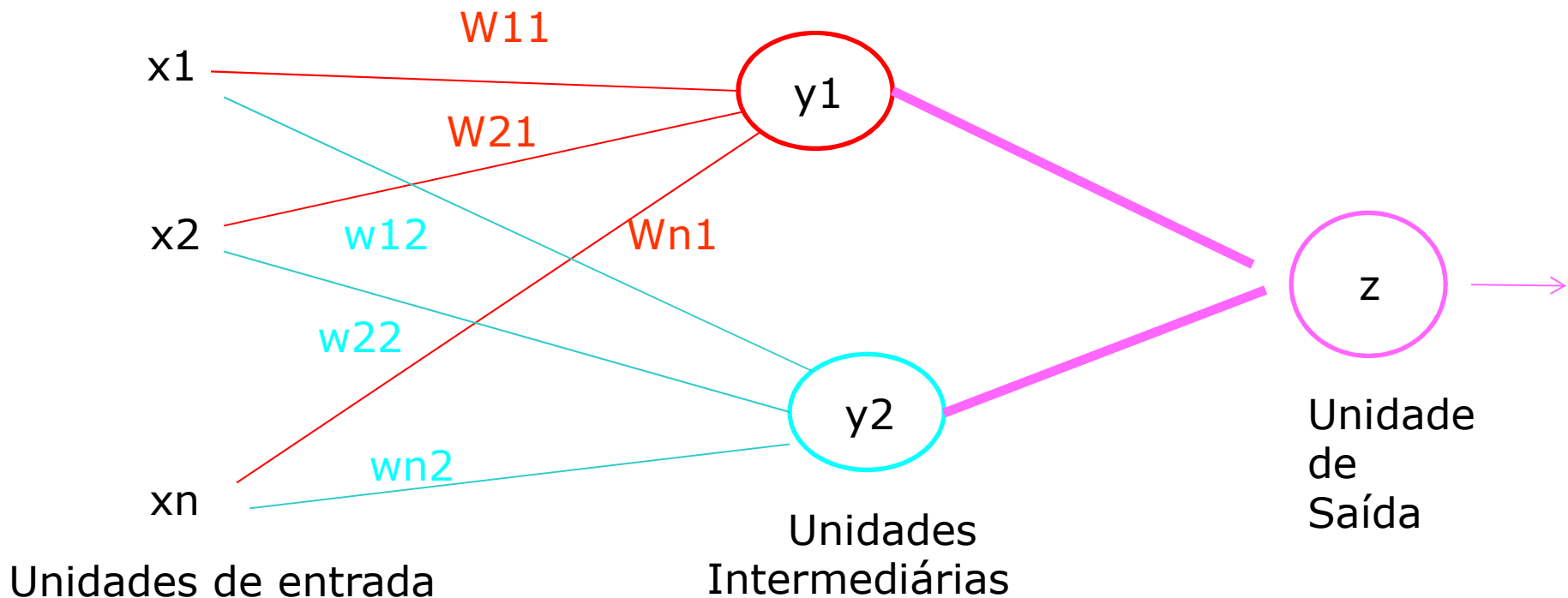
Redes Neurais Artificiais: Arquitetura

- Número de camadas: **Rede de Camada Única (Perceptron)**



Redes Neurais Artificiais: Arquitetura

- Número de camadas: **Rede de Múltiplas Camadas (Multilayer Perceptron – MLP)**



Redes Neurais Artificiais: Arquitetura

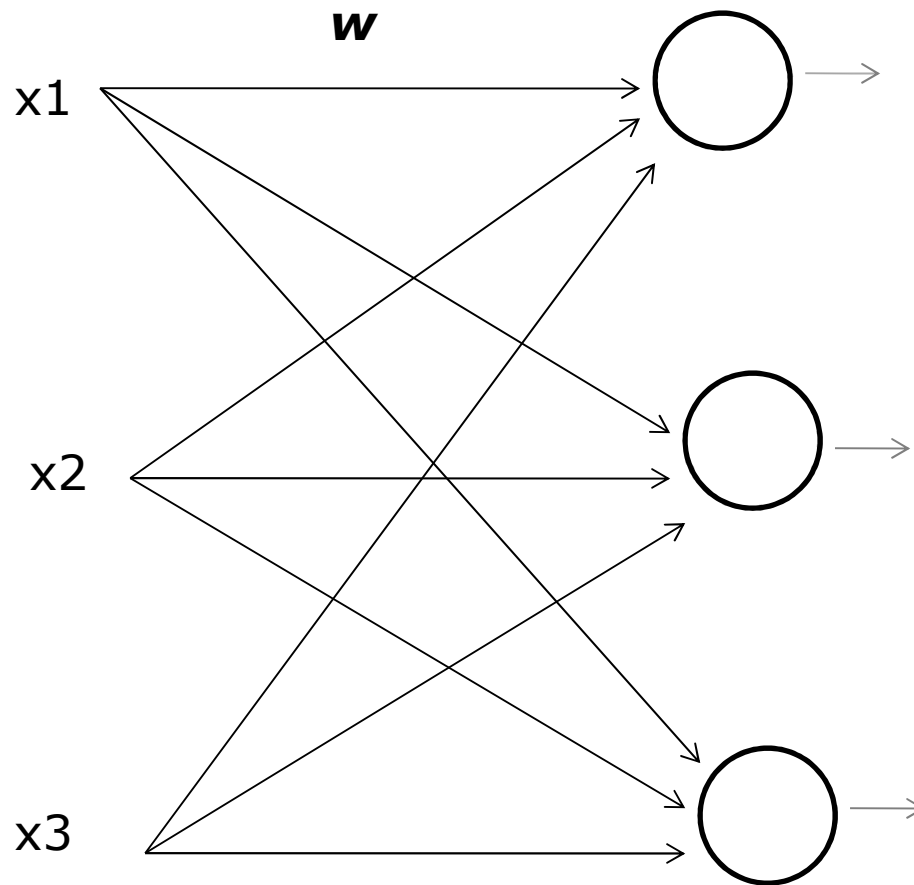
○ Tipos de Conexões:

- Redes do tipo **feedforward** (acíclicas): A saída de um neurônio na k -ésima camada **não** pode ser usada como entrada de neurônios em camadas de índices menores ou iguais a k (não há recorrência).
- Redes do tipo feedback (cíclicas ou **recorrentes**): A saída de algum neurônio na k -ésima camada **é** usada como como entrada de neurônios em camadas de índices menores ou iguais a k
 - autômatos: saída final única com recorrência
 - auto-associativas: todas as ligações são cíclicas

Redes Neurais Artificiais: Arquitetura

- **Conectividade:**
 - Parcialmente conectada
 - Completamente conectada

Redes Neurais Artificiais: Arquitetura

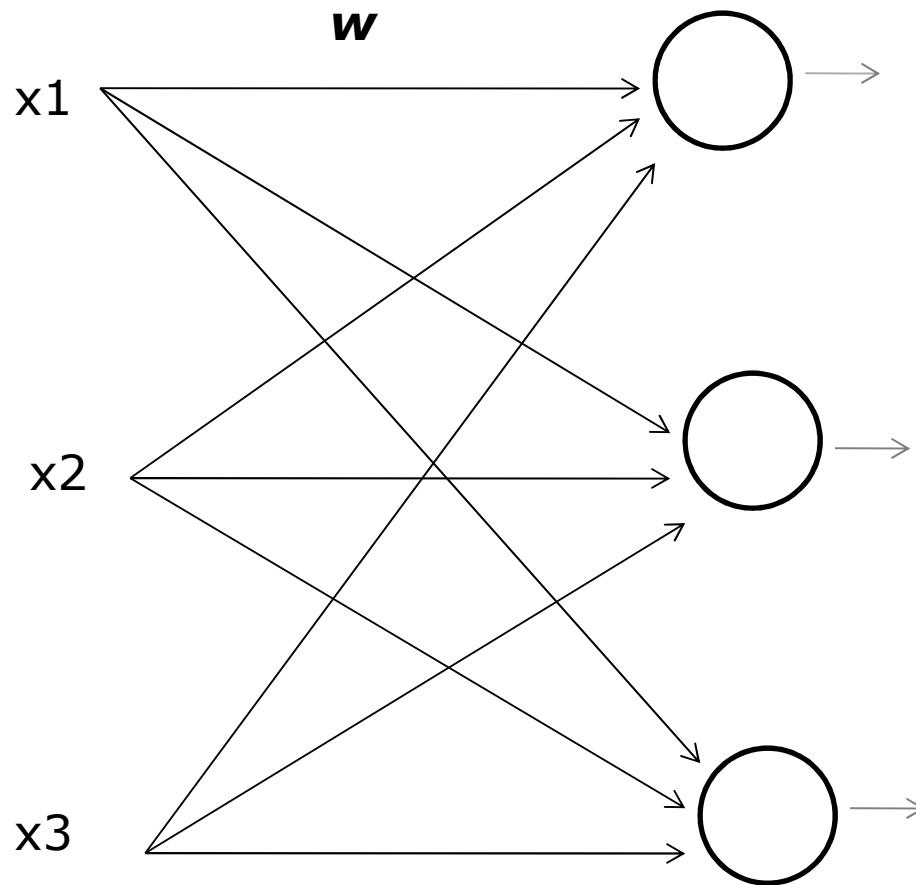


Quantas camadas?

Feedforward ou
recorrente?

Parcial ou totalmente
conectada?

Redes Neurais Artificiais: Arquitetura



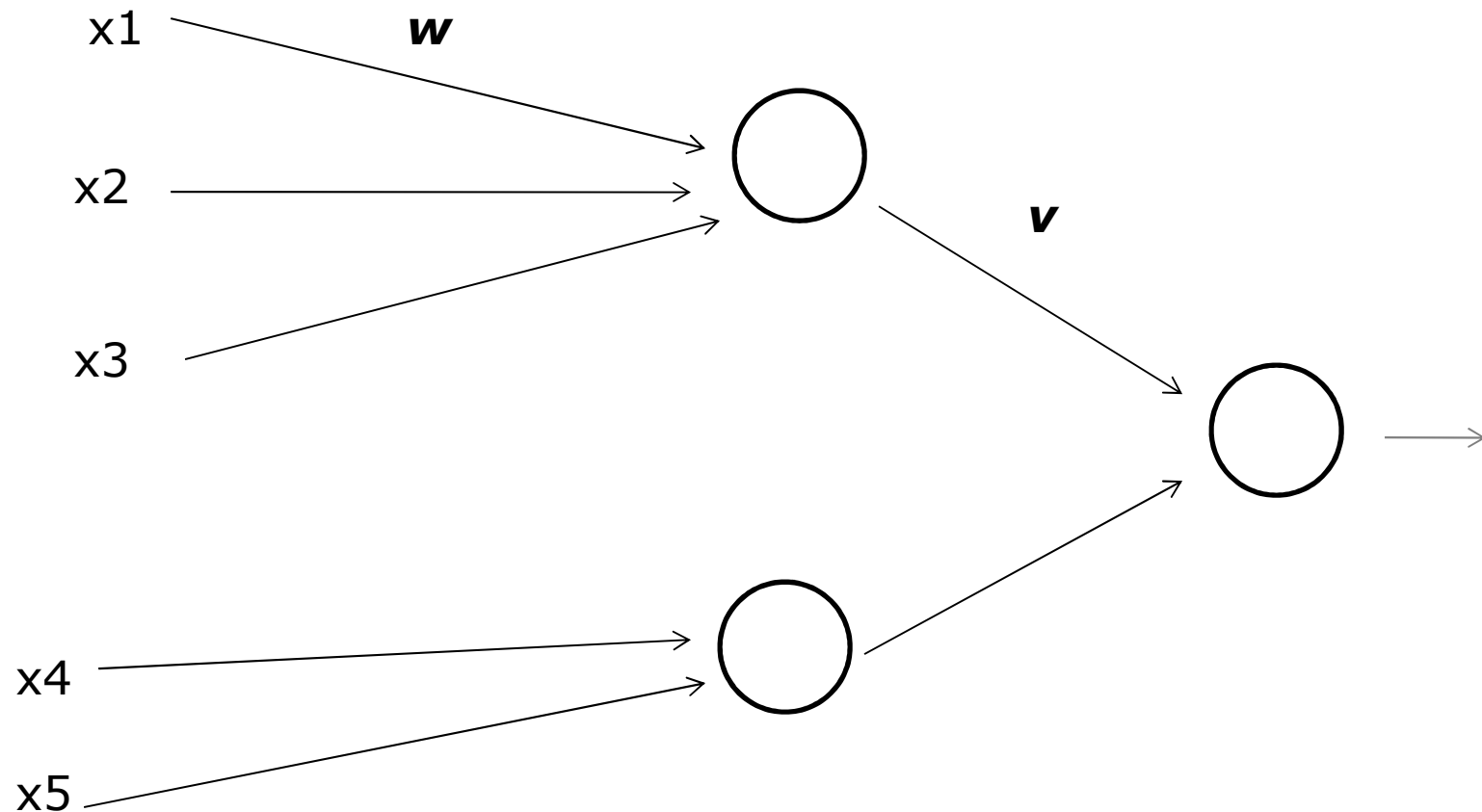
Rede de
Camada Única

Feedforward

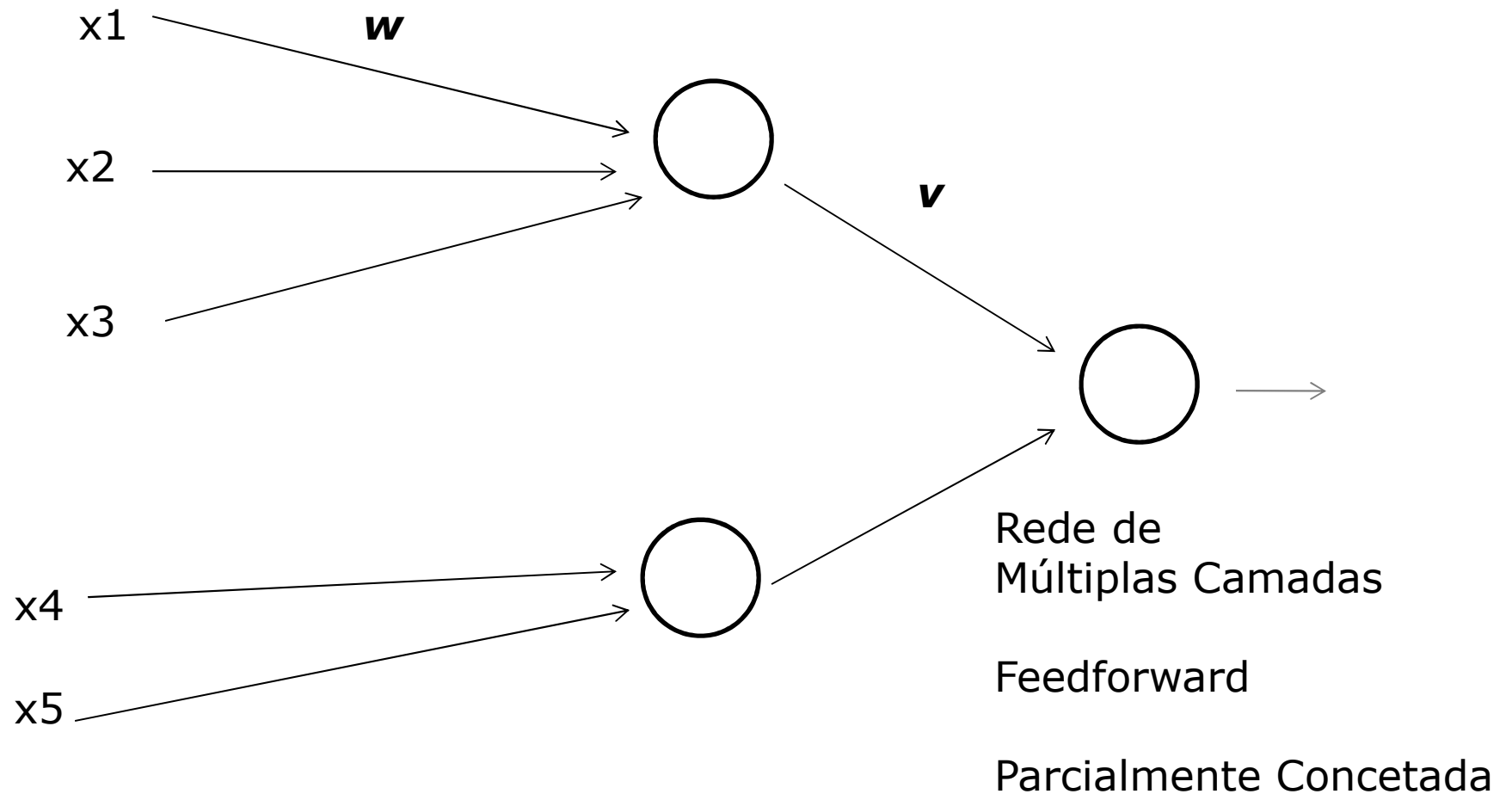
Complemente
Conectada

Exemplo:
Perceptron

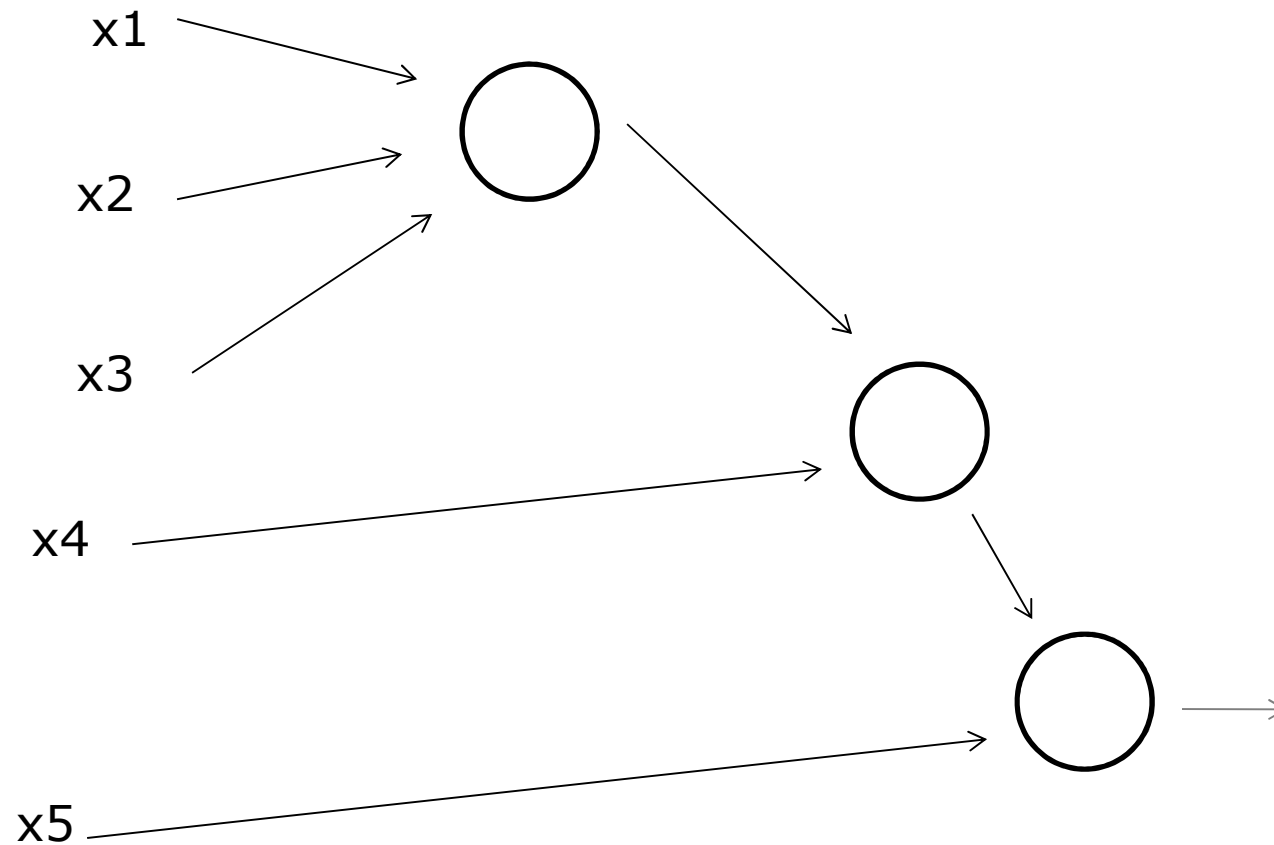
Redes Neurais Artificiais: Arquitetura



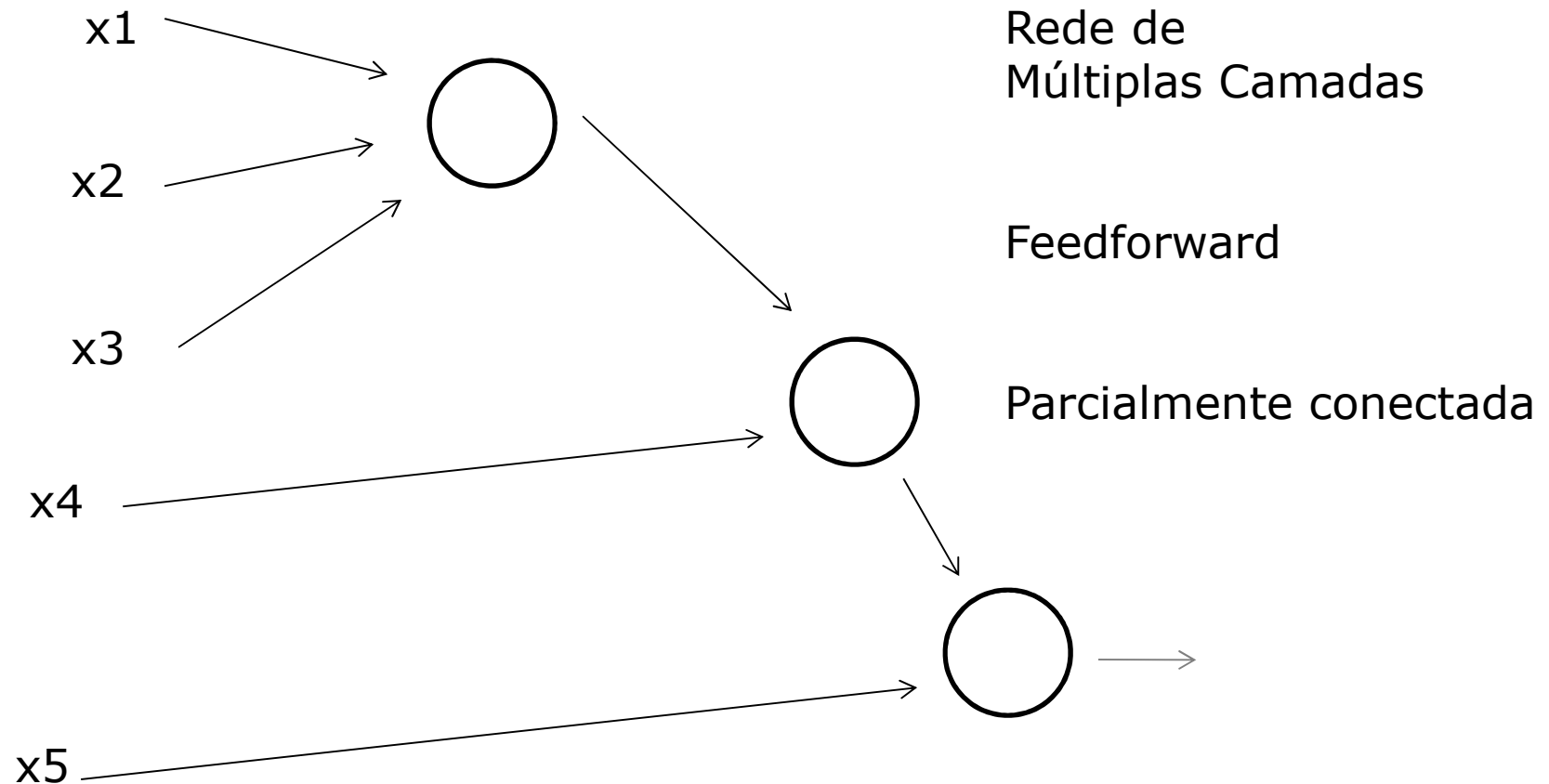
Redes Neurais Artificiais: Arquitetura



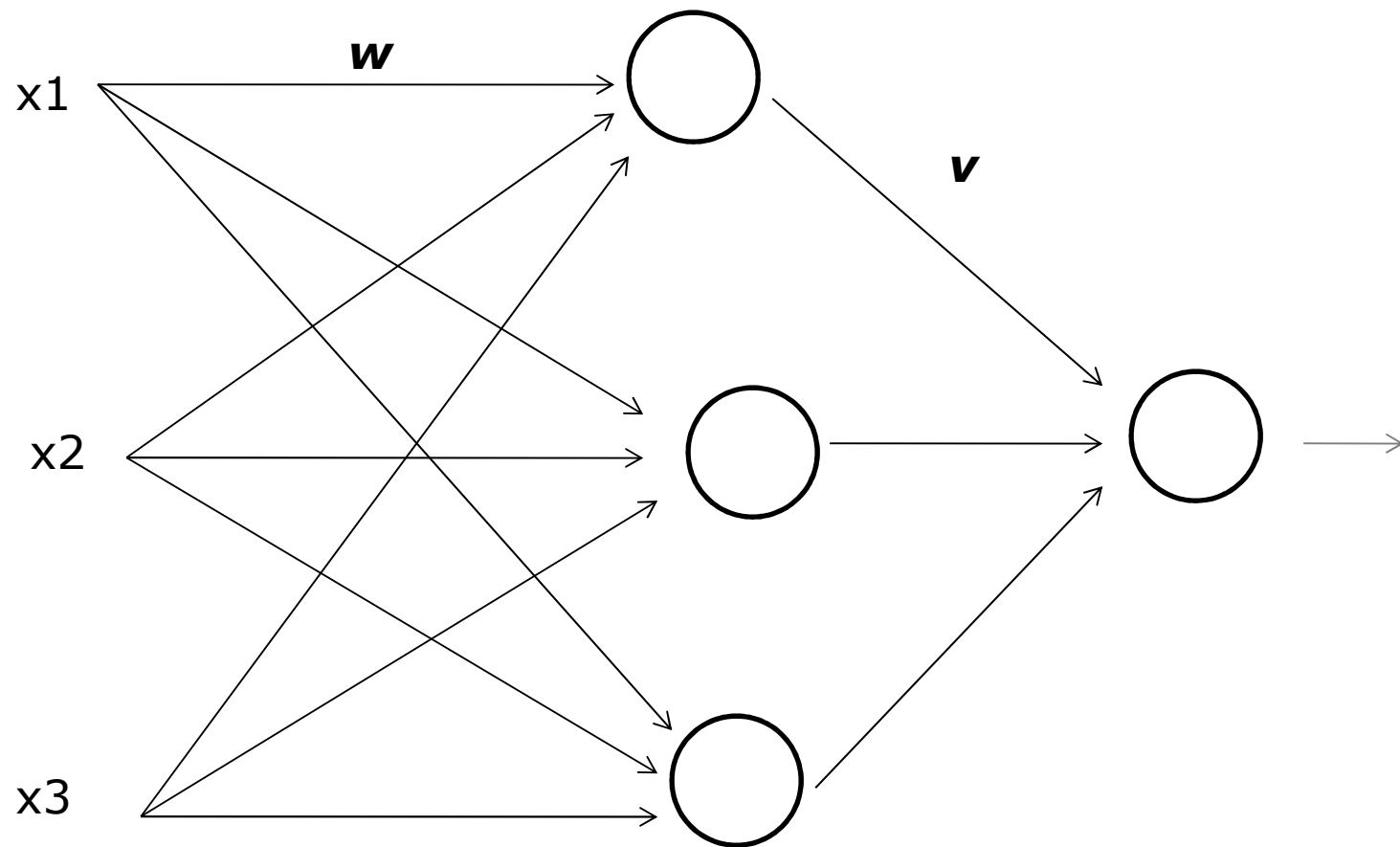
Redes Neurais Artificiais: Arquitetura



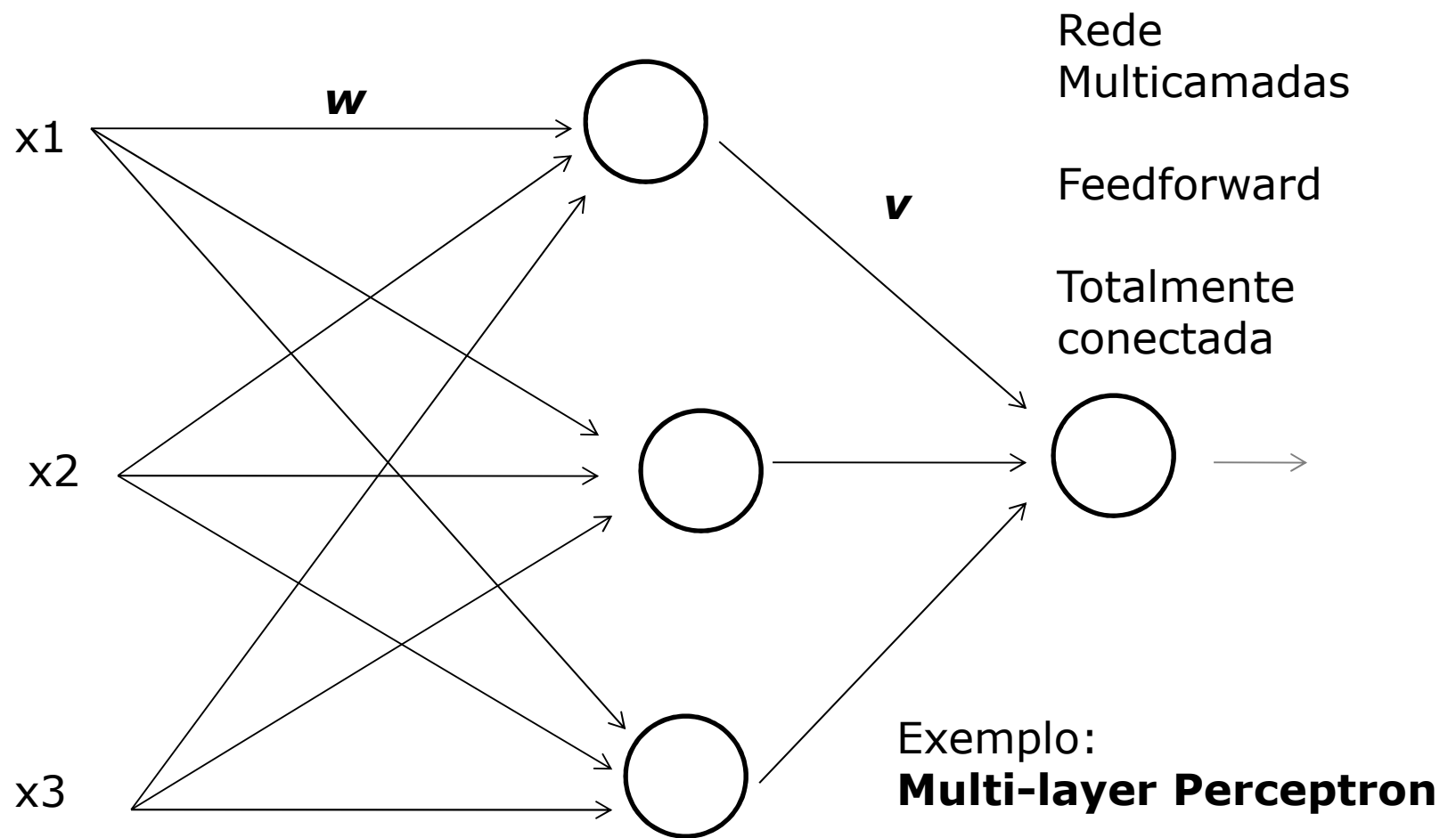
Redes Neurais Artificiais: Arquitetura



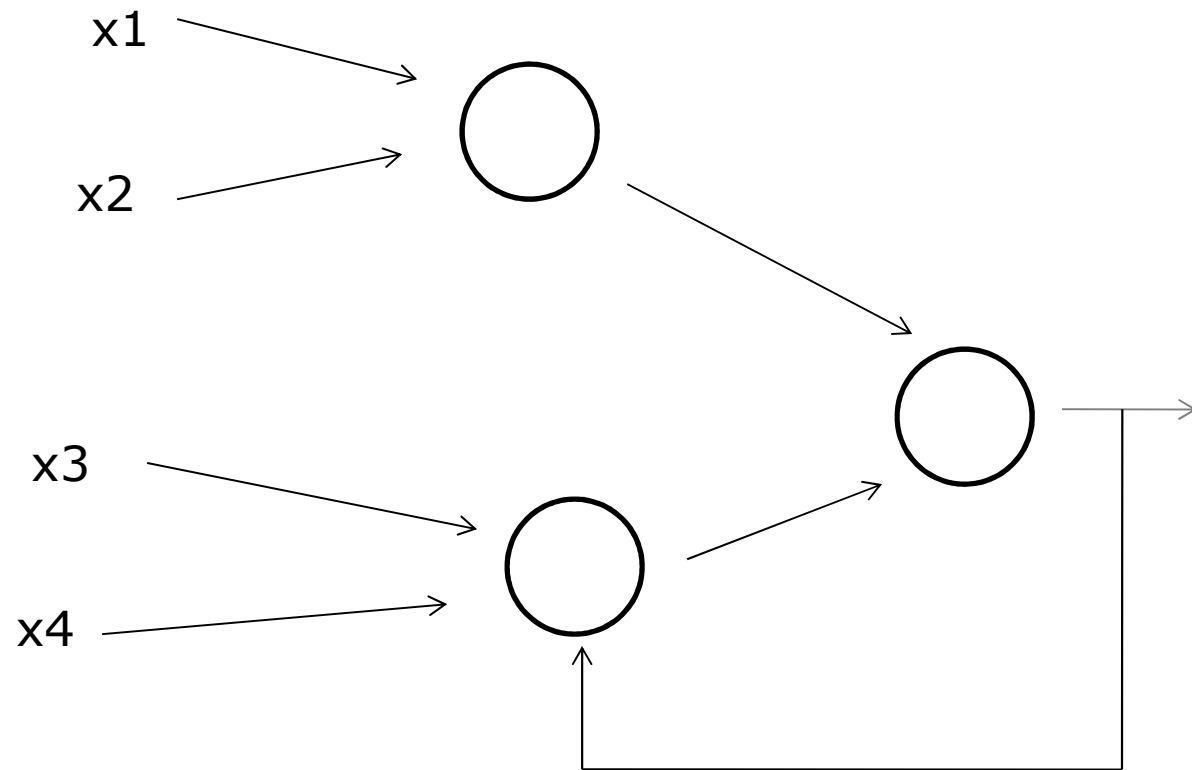
Redes Neurais Artificiais: Arquitetura



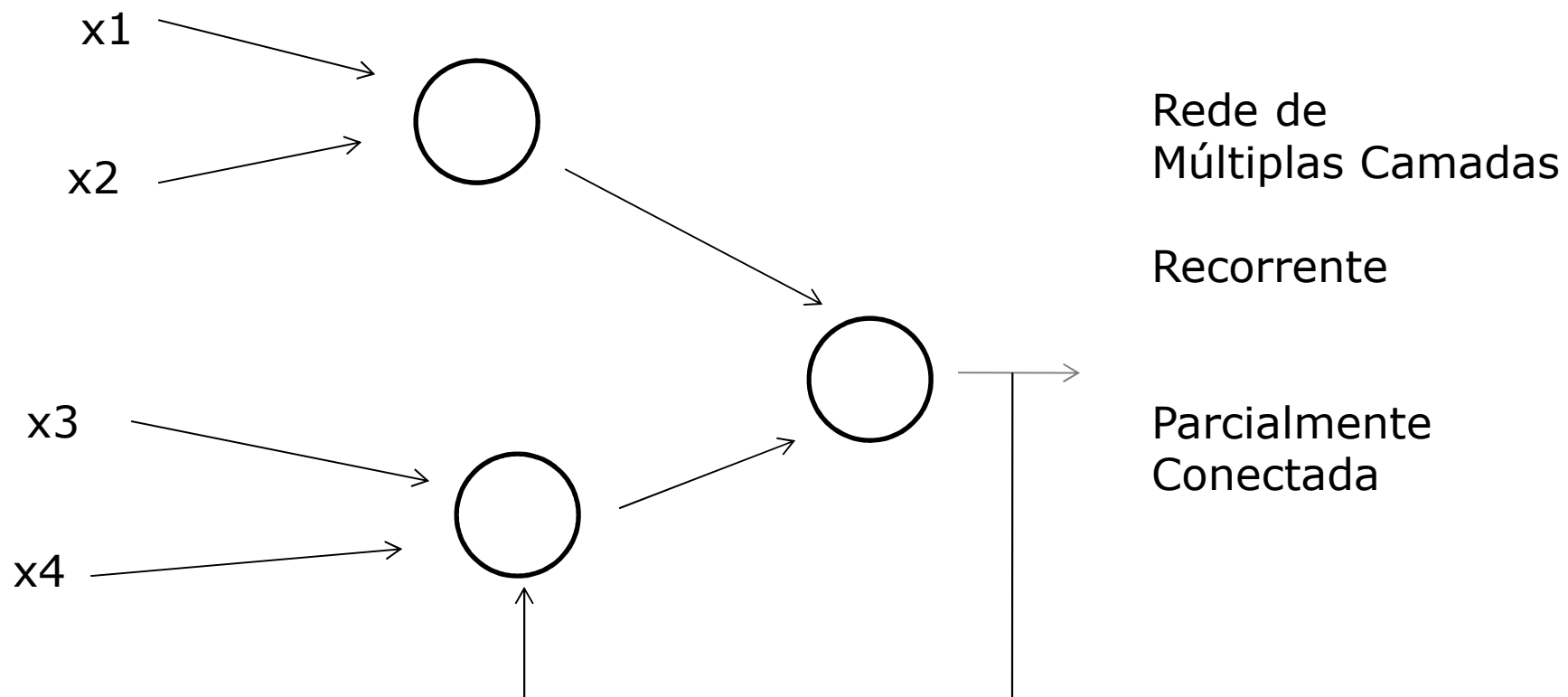
Redes Neurais Artificiais: Arquitetura



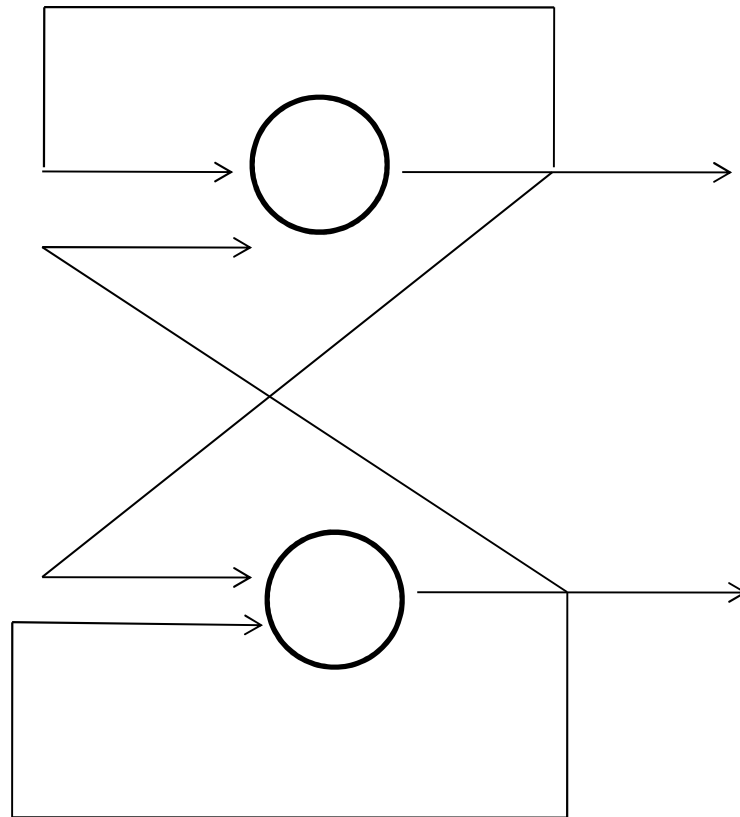
Redes Neurais Artificiais: Arquitetura



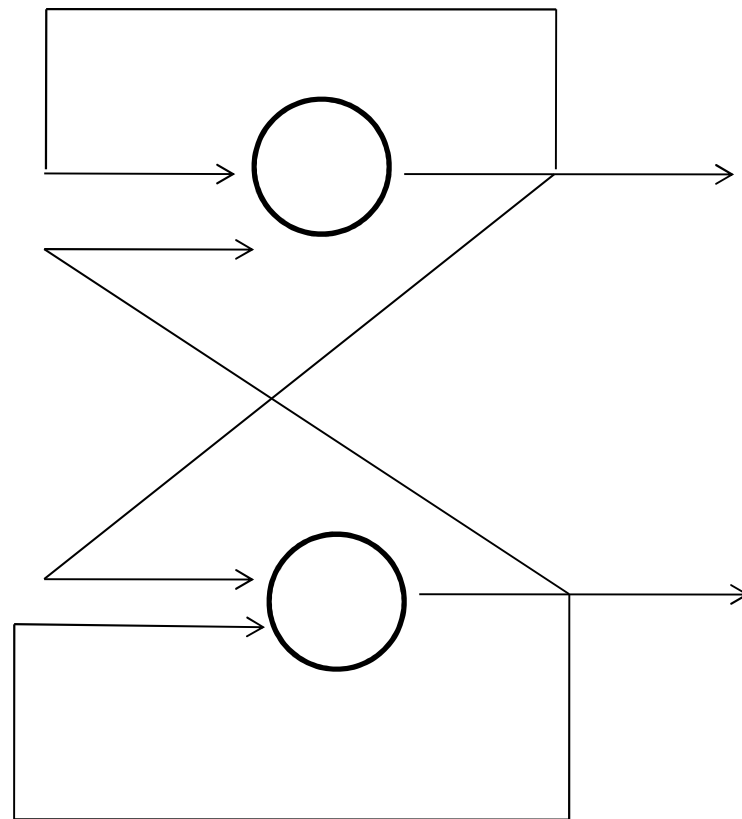
Redes Neurais Artificiais: Arquitetura



Redes Neurais Artificiais: Arquitetura



Redes Neurais Artificiais: Arquitetura

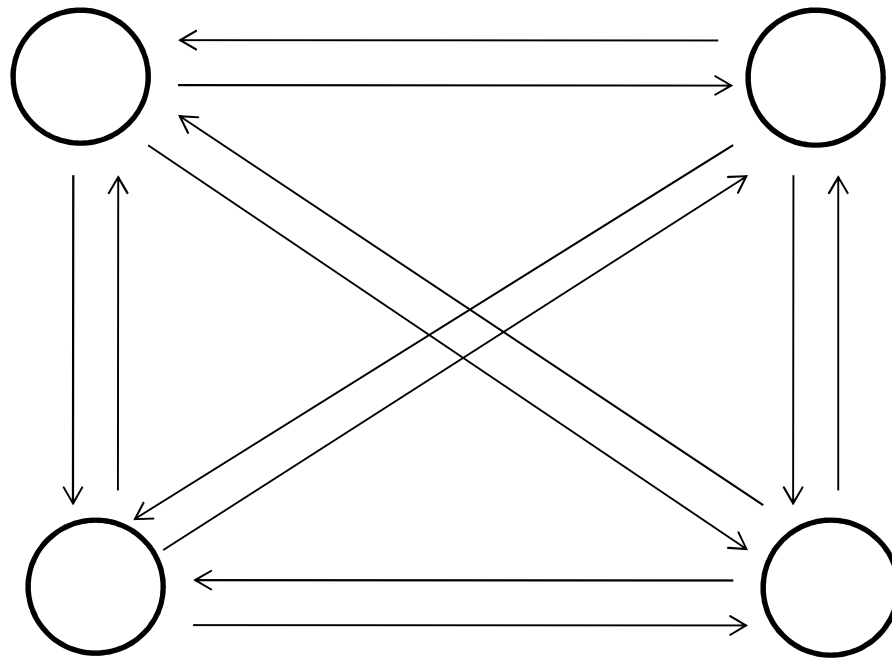


Rede de
Camada Única

Recorrente

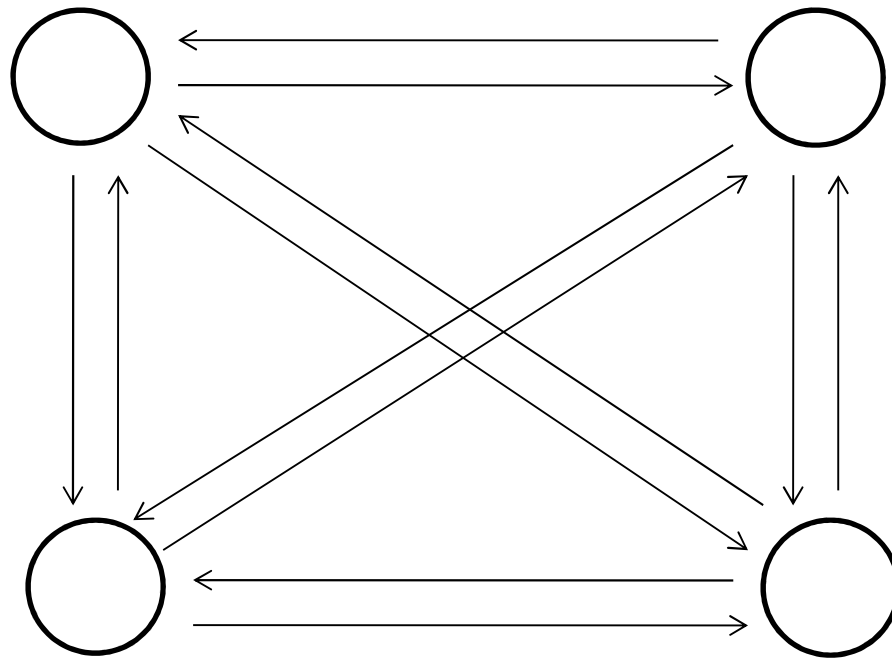
Completamente
Conectada

Redes Neurais Artificiais: Arquitetura



?????

Redes Neurais Artificiais: Arquitetura



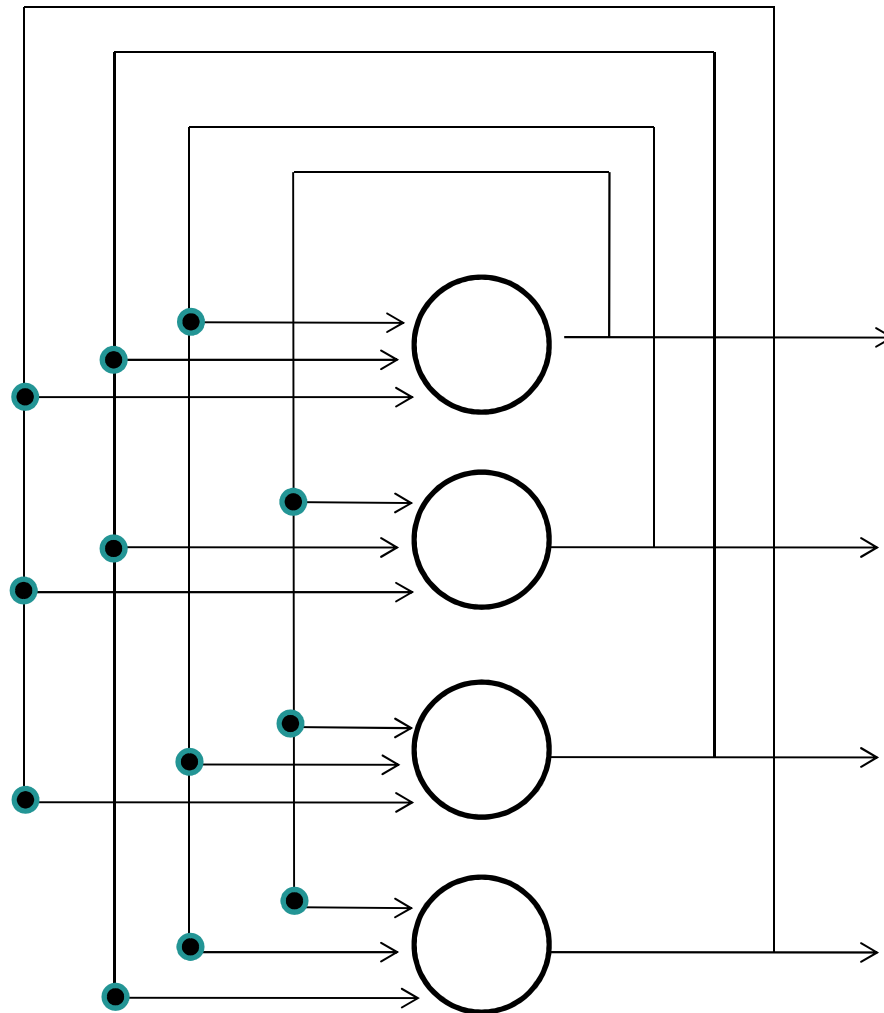
Rede de Camada Única

Recorrente

Parcialmente Conectada

Rede de Hopfield
(com 4 neurônios)

Redes Neurais Artificiais: Arquitetura

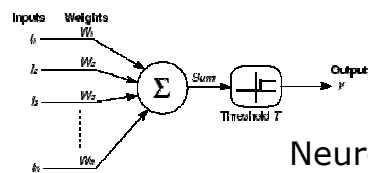


Outra forma de
visualizar a

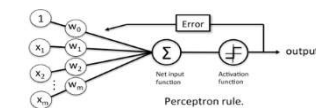
Rede de Hopfield

Mostrada
anteriormente

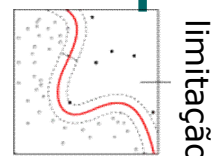
RNAs: Linha do Tempo



Neurônio MCP



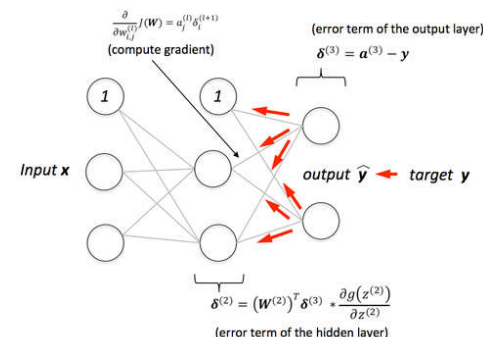
Perceptron



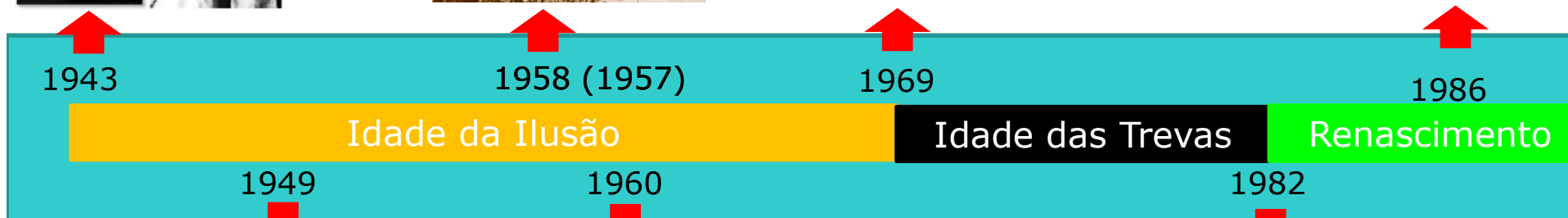
limitação



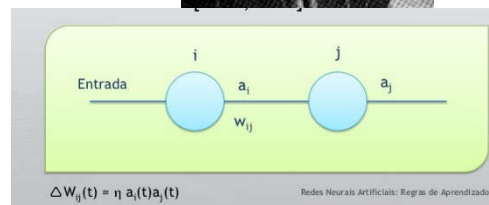
Minsky e Papert



Backpropagation

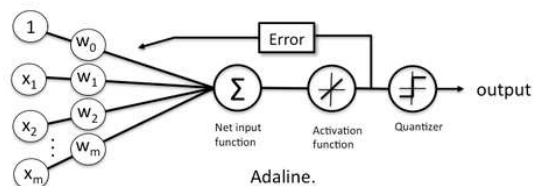


Regra de Hebb



Regra Delta (Widrow-Hoff)

$$w_{ij}(\text{new}) = w_{ij}(\text{old}) + \alpha(t_i - y_{in_i})x_j$$

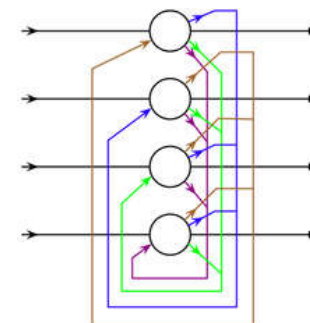


Adaline.

Prof. Myriam Delgado
Redes Neurais - UTFPR



Hopfield



RN associativas

Redes Neurais Artificiais (Modelos)

Redes Feedforward:

Perceptron

Multilayer Perceptron (MLP)

Redes de Funções de Base Radial (RBF)

Redes Associativas e Recorrentes

Hopfield

Redes Auto-organizáveis

Mapas de Kohonen