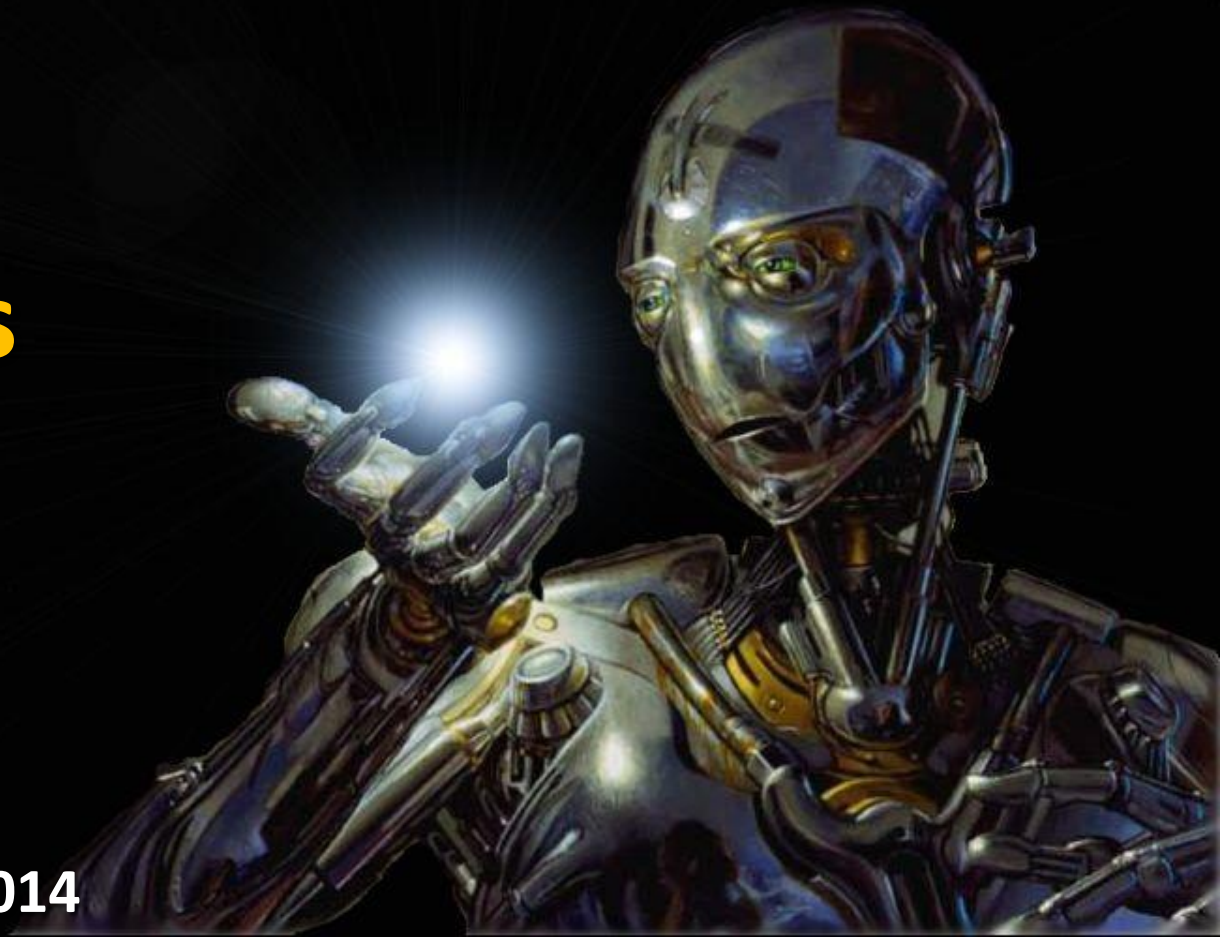


INTELIGÊNCIA ARTIFICIAL

Parte 11

**Introdução às
Redes Neurais**



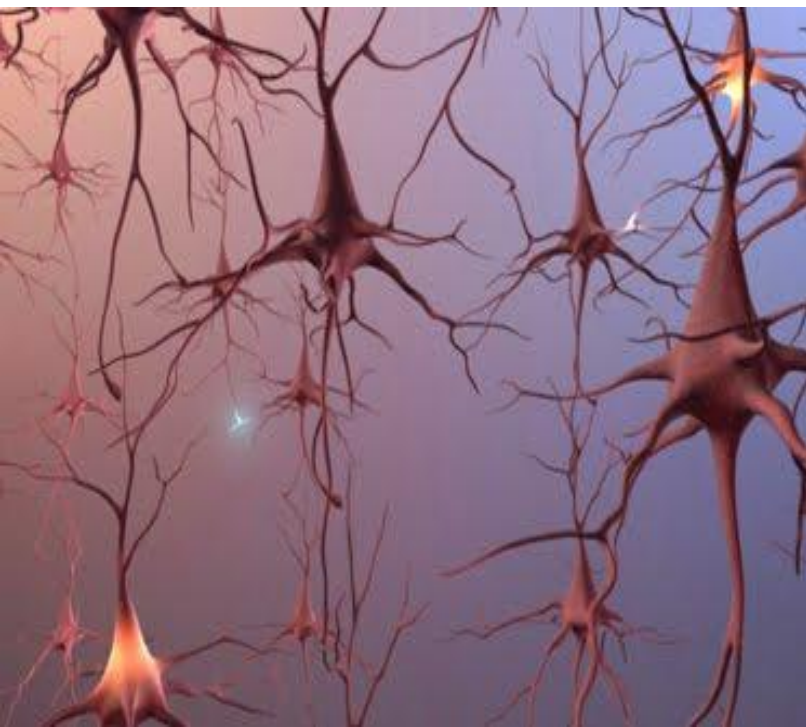
1

Redes Neurais

1 – Redes Neurais

1.1 – Introdução:

- O cérebro humano é mais fascinante processador baseado em carbono existente, sendo composto por quase 100 bilhões neurônios e possivelmente cerca de 100 trilhões de conexões.

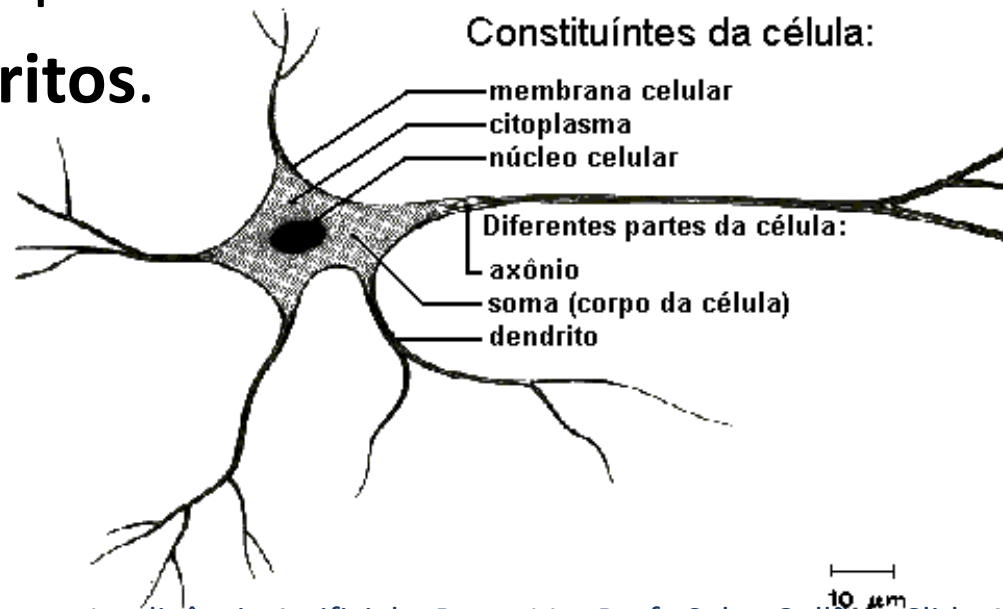


- Todas as funções do organismo estão relacionados ao funcionamento destas pequenas células.

1 – Redes Neurais

1.1 – Introdução:

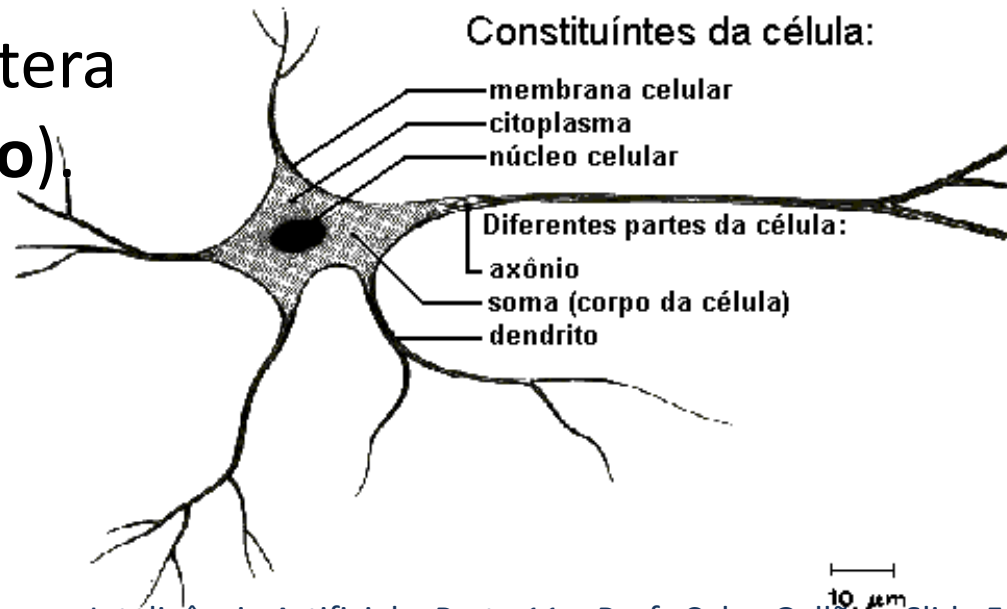
- Neurônios se conectam uns aos outros através de **sinapses**.
- As sinapses transmitem estímulos elétricos ($\cong 0,07$ volts) gerados por reações químicas através de diferentes concentrações de sódio e potássio.
- Sinais entram pelos **dentritos**.
- Após processados, sinais saem pelo **axônio**.



1 – Redes Neurais

1.1 – Introdução:

- Uma substância neurotransmissora controla os potenciais elétricos de saída, fluindo do corpo celular para o **axônio**, diminuindo ou aumentando a polaridade da **membrana pós-sináptica**, inibindo ou excitando os pulsos elétricos.
- Após vivenciar novas experiência, o **cérebro** altera as conexões (**aprendizado**).
- Experiências repetidas várias vezes reforçam as ligações.



1 – Redes Neurais

1.1 – Introdução:

- Uma rede neural é um modelo computacional que compartilha algumas das propriedades do cérebro.
- Consiste em muitas unidades simples trabalhando em paralelo sem nenhum controle central.
- As conexões entre as unidades têm pesos numéricos que podem ser modificados pelo aprendizado.

1 – Redes Neurais

1.1 – Introdução:

- Microprocessadores digitais processam a uma velocidade 1 milhão de vezes mais rápida do que o cérebro humano, no que se refere à sequência de instruções.
- No entanto, o cérebro faz processamento extremamente mais rápido no reconhecimento de padrões, por exemplo.
- Busca-se estudar a teoria e a implementação de sistemas massivamente paralelos que possam processar informações com eficiência comparável ao cérebro humano.

1 – Redes Neurais

1.1 – Introdução:

- Comparação entre o computador de *von Neumann* e o sistema neural biológico [JAIN e MAO, 1996].

	Computador de <i>von Neumann</i>	Sistema Neural Biológico
Processador	<i>Complexo</i> <i>Alta velocidade</i> <i>Um ou poucos</i>	<i>Simples</i> <i>Baixa Velocidade</i> <i>Um grande número</i>
Memória	<i>Separado do processador</i> <i>Localizado</i> <i>Não endereçável pelo conteúdo</i>	<i>Integrada ao processador</i> <i>Distribuída</i> <i>Endereçável pelo conteúdo</i>
Computação	<i>Centralizada</i> <i>Sequencial</i> <i>Programas armazenados</i>	<i>Distribuída</i> <i>Paralela</i> <i>Autoaprendizado</i>
Confiabilidade	<i>Muito vulnerável</i>	<i>Robusta</i>
Especialidade	<i>Manipulações numéricas e simbólica</i>	<i>Problemas perceptuais</i>
Ambiente Operacional	<i>Bem definido, bem restrito</i>	<i>Pobremamente definido, irrestrito</i>

1 – Redes Neurais

1.1 – Introdução:

Em quais problemas deve-se usar Redes Neurais?

- Geralmente em problemas de reconhecimento de padrões onde os dados são ruidosos (ou incompletos) e/ou quando regras claras não podem ser facilmente formuladas.

1 – Redes Neurais

1.1 – Introdução:

Vantagens

- Permite modelagem de sistemas não lineares;
- Aprendizado automático;
- Tolerante a dados ruidosos ou incompletos;
- Resposta rápida e precisa;
- Modelos compactos.

1 – Redes Neurais

1.1 – Introdução:

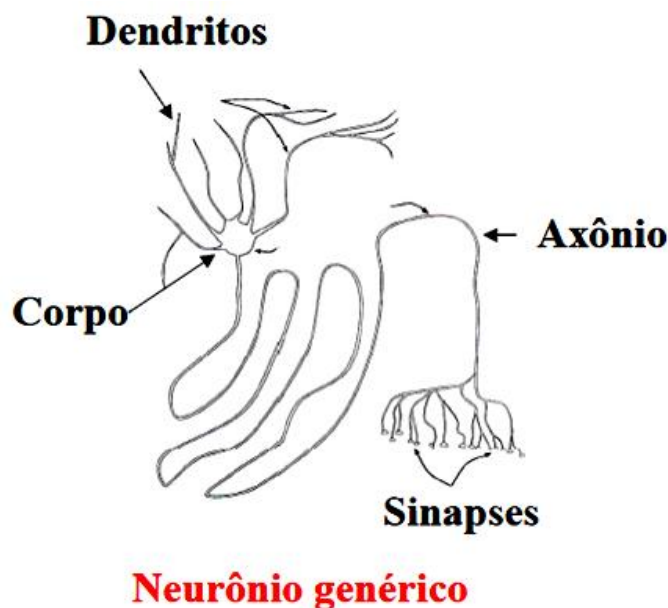
Desvantagens

- Ausência de explicações explícitas (subsistema de explanação);
- Sensível à quantidade disponível de dados;
- Processo de treinamento (aprendizado) pode ser lento.

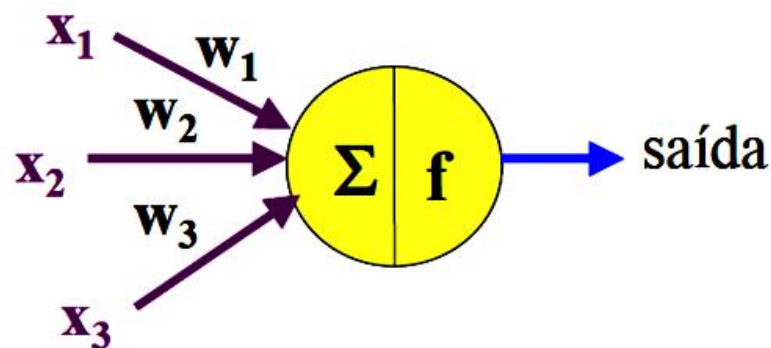
1 – Redes Neurais

1.2 – Histórico:

- **1943**: Trabalho pioneiro de **McCulloch** (psiquiatra e neuroanatomista) e **Pitts** (matemático) - Descrição do modelo formal de um neurônio.



≡



Elemento processador

1 – Redes Neurais

1.2 – Histórico:

- **1949:** *Hebb* publica o livro “*The Organization of Behavior*” - formulação da primeira regra de aprendizado (*Hebbian Learning Rule*).
- **1957:** *Rosenblatt* propõe o modelo *Perceptron* como um método inovador de aprendizado supervisionado.

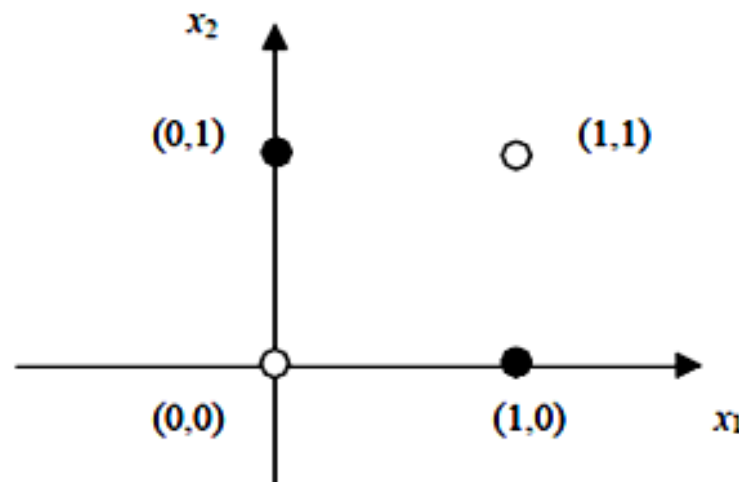
1 – Redes Neurais

1.2 – Histórico:

- **1969**, o ano fatídico: **Minsky** e **Papert** (**MIT**) demonstram as limitações do **Perceptron** provando matematicamente que são incapazes de solucionar problemas simples tipo o **Xor** (aplicável apenas a problemas linearmente separáveis).

XOR

x_1	x_2	$x_1 \otimes x_2$
0	0	0
0	1	1
1	0	1
1	1	0



1 – Redes Neurais

1.2 – Histórico:

- Anos **1970**, a **década da hibernação**: adormecimento generalizado das pesquisas em Redes Neurais, com a publicação do livro de ***Minsky*** e ***Papert***.

1 – Redes Neurais

1.2 – Histórico:

Anos **1980**, entusiasmo renovado:

- **1982:** *Hopfield* publica estudo sobre propriedades associativas das Redes Neurais.
- **1986:** *Rumelhart, Hinton* e *Williams* desenvolvem o algoritmo *Backpropagation* que supera os problemas dos *Perceptrons*.

1 – Redes Neurais

1.3 – Características Similares ao Cérebro Humano:

- **Busca paralela e endereçamento pelo conteúdo:**
O cérebro não possui endereço de memória e não procura a informação sequencialmente.
- **Aprendizado por experiências:**
Analisando seus “erros”, não necessitando explicitar os algoritmos (ou funções analíticas) para executar uma determinada tarefa.

1 – Redes Neurais

1.3 – Características Similares ao Cérebro Humano:

- **Capacidade de generalizar o seu conhecimento:**
Habilidade de lidar com ruídos e distorções, a partir de exemplos anteriores, e respondendo corretamente aos novos padrões.
- **Abstração:**
Capacidade de extrair a informação de padrões sem ruído, a partir de padrões ruidosos de entrada, abstraindo a essência do conjunto de entradas.

1 – Redes Neurais

1.3 – Características Similares ao Cérebro Humano:

- **Robustez:**

A perda de alguns elementos processadores ou conexões sinápticas não causa o mal funcionamento da rede neural.

- A característica fundamental das Redes Neurais é a **habilidade de aprender a partir de seu ambiente**. O processo de aprendizado é iterativo e tende a melhorar o desempenho gradativamente, à medida que interage com o meio ambiente.

1 – Redes Neurais

1.3 – Características Similares ao Cérebro Humano:

“Aprendizagem é um processo pelo qual os parâmetros livres (ou pesos) de uma rede neural são adaptados através de estímulos do ambiente no qual a rede está inserida. O tipo de aprendizagem é determinado pela maneira como ocorre a modificação dos parâmetros.”

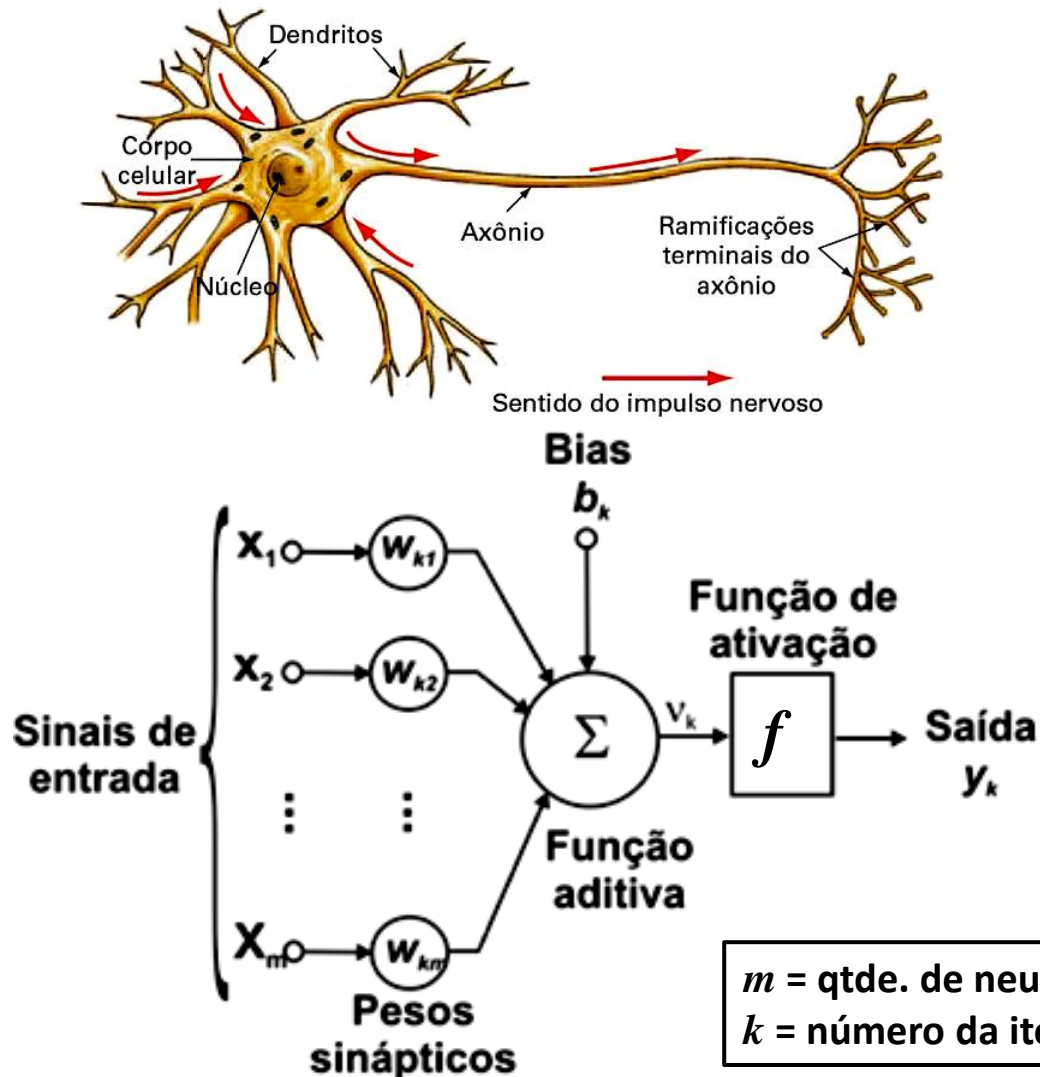
[Haykin , 2003]

2

Neurônio Artificial

2 – Neurônio Artificial

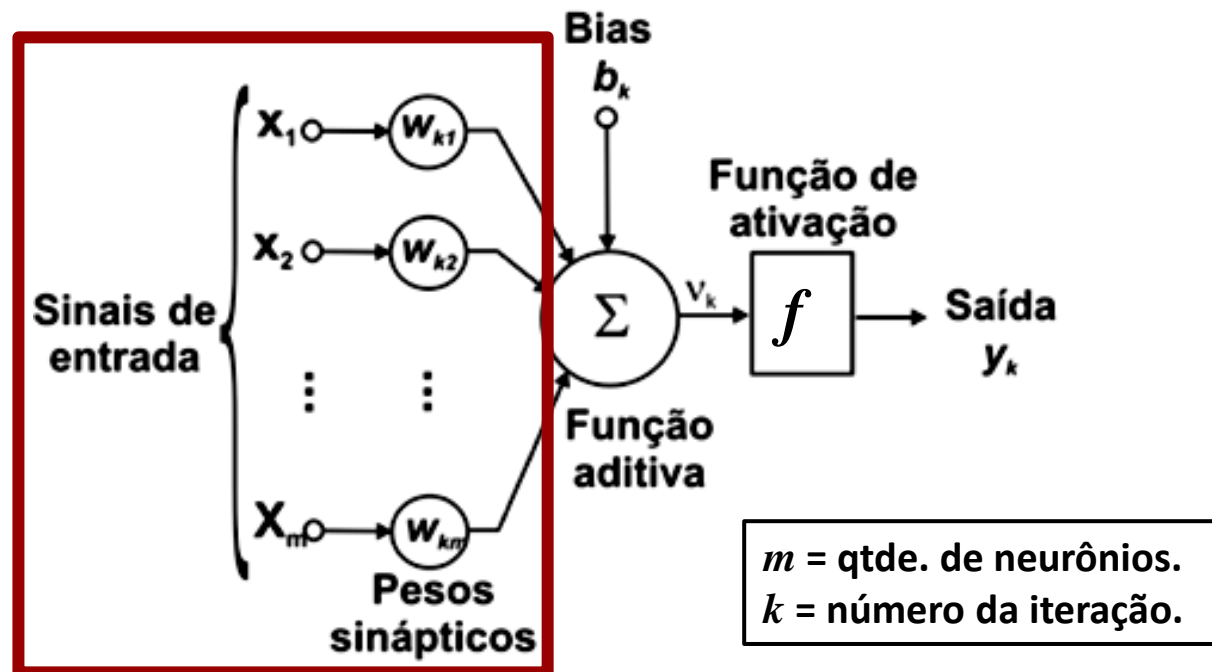
2.1 – Modelo Matemático (McCulloch & Pitts):



2 – Neurônio Artificial

2.1 – Modelo Matemático (McCulloch & Pitts):

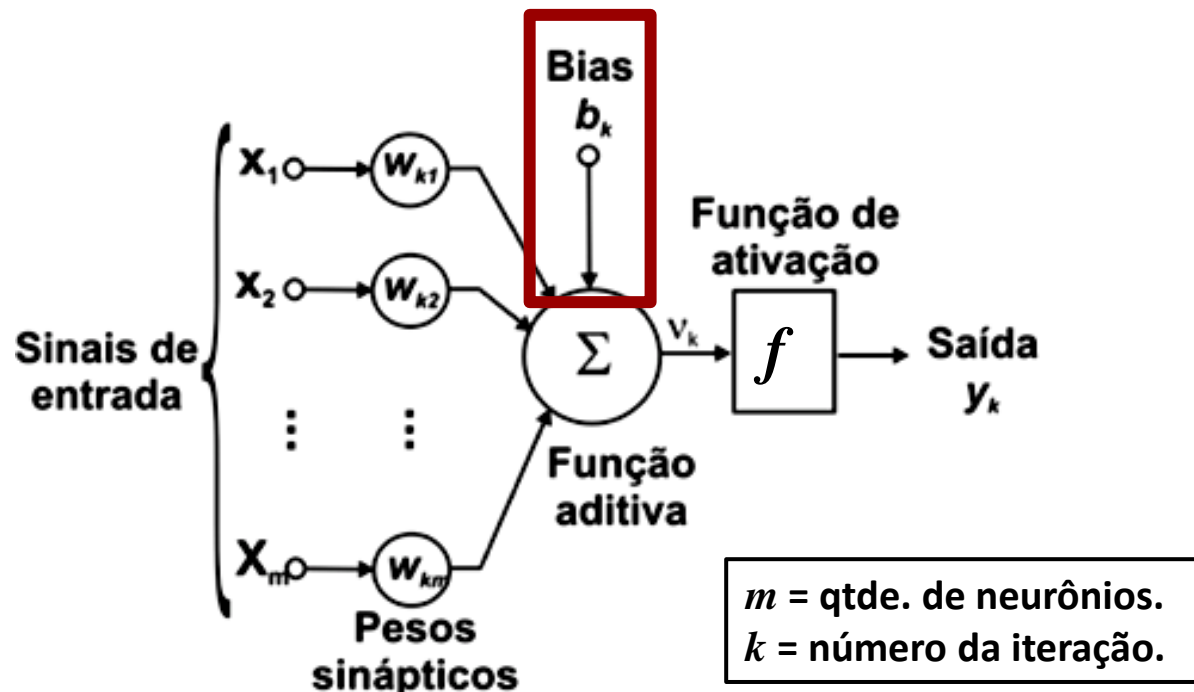
- **Conjunto de Sinapses ou Elos de Conexão**, onde cada elo de conexão é caracterizado por um estímulo de entrada (x_m) e por um peso (w_{km}).



2 – Neurônio Artificial

2.1 – Modelo Matemático (McCulloch & Pitts):

- **Bias** (viés): uma entrada extra, opcional, com valor sempre igual a **1**. Não é identificada nos neurônios biológicos, mas sua aplicação vem produzindo resultados muito úteis em Redes Neurais artificiais.

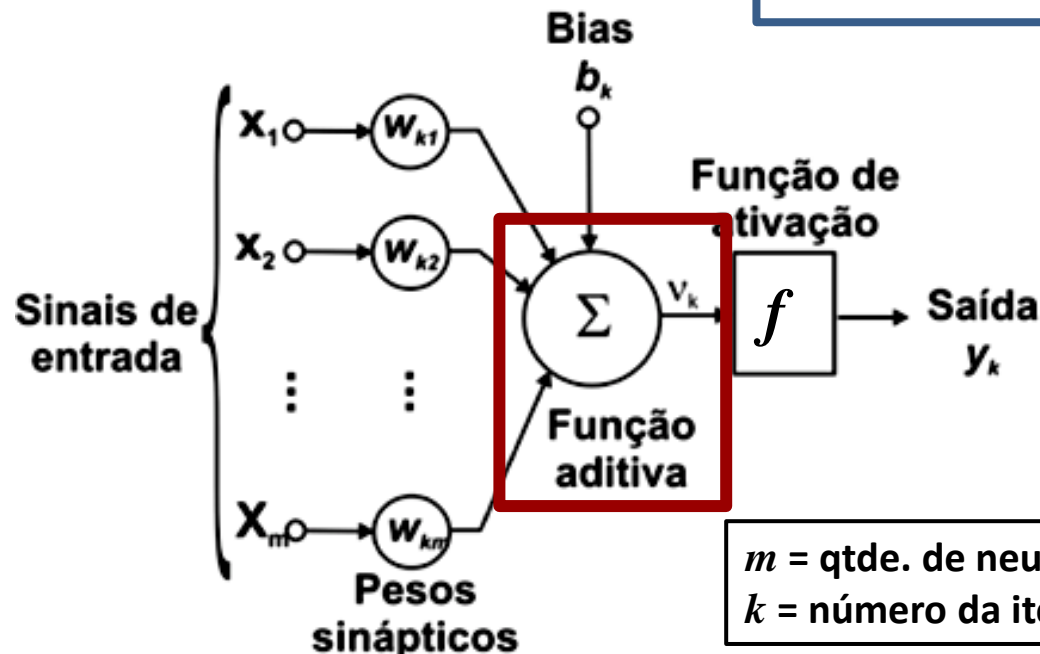


2 – Neurônio Artificial

2.1 – Modelo Matemático (McCulloch & Pitts):

- **Combinador Linear (Σ):** efetua a somatória dos sinais ponderados de entrada no neurônio. Representada pela variável *net*, dada por:

$$net = \sum w_{km} x_m + b_k$$

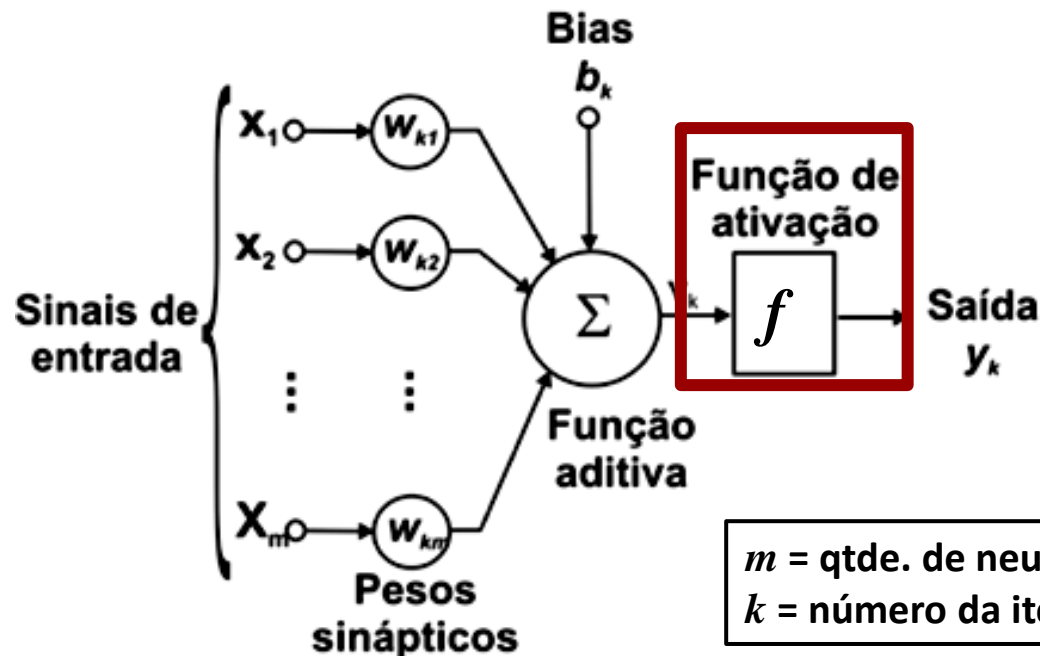


m = qtde. de neurônios.
 k = número da iteração.

2 – Neurônio Artificial

2.1 – Modelo Matemático (McCulloch & Pitts):

- **Função de Ativação:** determina o novo valor do estado de ativação do processador, utilizando a função aditiva como parâmetro de entrada para produzir a saída do neurônio. Podem ser aplicados diferentes tipos de funções.

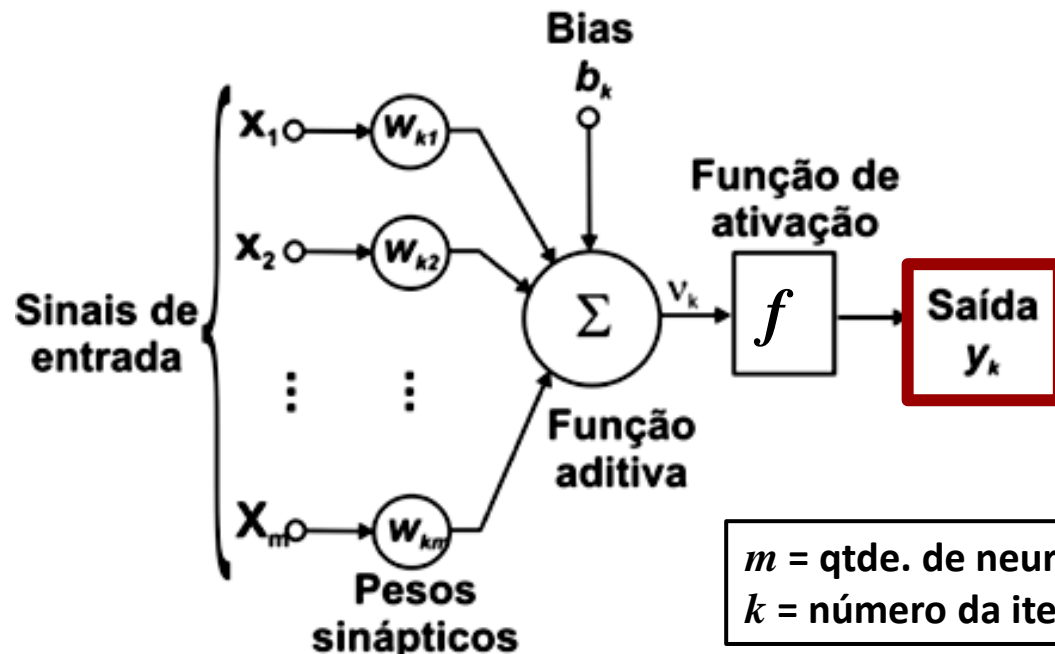


2 – Neurônio Artificial

2.1 – Modelo Matemático (McCulloch & Pitts):

- **Saída do Neurônio:** apresenta o resultado de 1 iteração no processamento da rede neural. É calculada por:

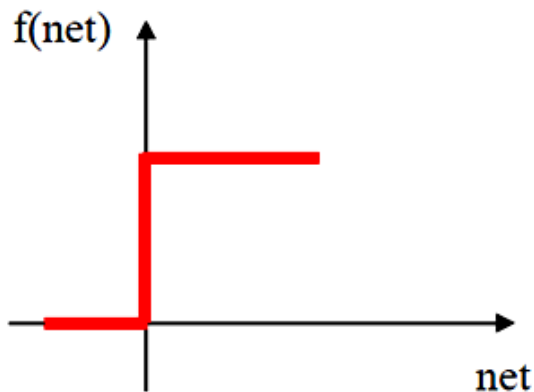
$$y_k = f(net) = f(\sum w_{km} x_m + b_k)$$



2 – Neurônio Artificial

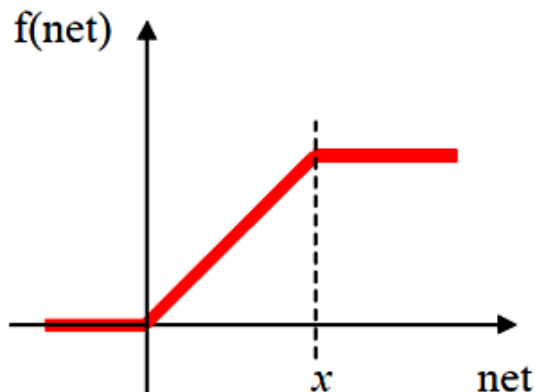
2.2 – Funções de Ativação:

Degrau



$$f(\text{net}) = \begin{cases} 1 & \text{se } \text{net} > 0 \\ 0 & \text{se } \text{net} \leq 0 \end{cases}$$

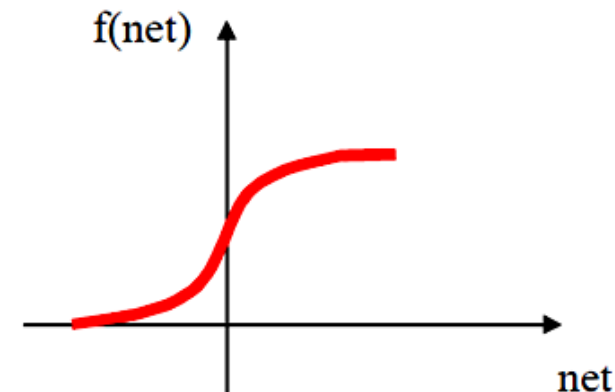
Pseudo-Linear



$$f(\text{net}) = \begin{cases} 1 & \text{se } \text{net} > x \\ \text{net} & \text{se } 0 < \text{net} \leq x \\ 0 & \text{se } \text{net} \leq 0 \end{cases}$$

$$\text{Linear: } f(\text{net}) = a.\text{net} + b$$

Sigmoid



$$f(\text{net}) = \frac{1}{1 + e^{-\text{net}}}$$

3

Projeto de Redes Neurais

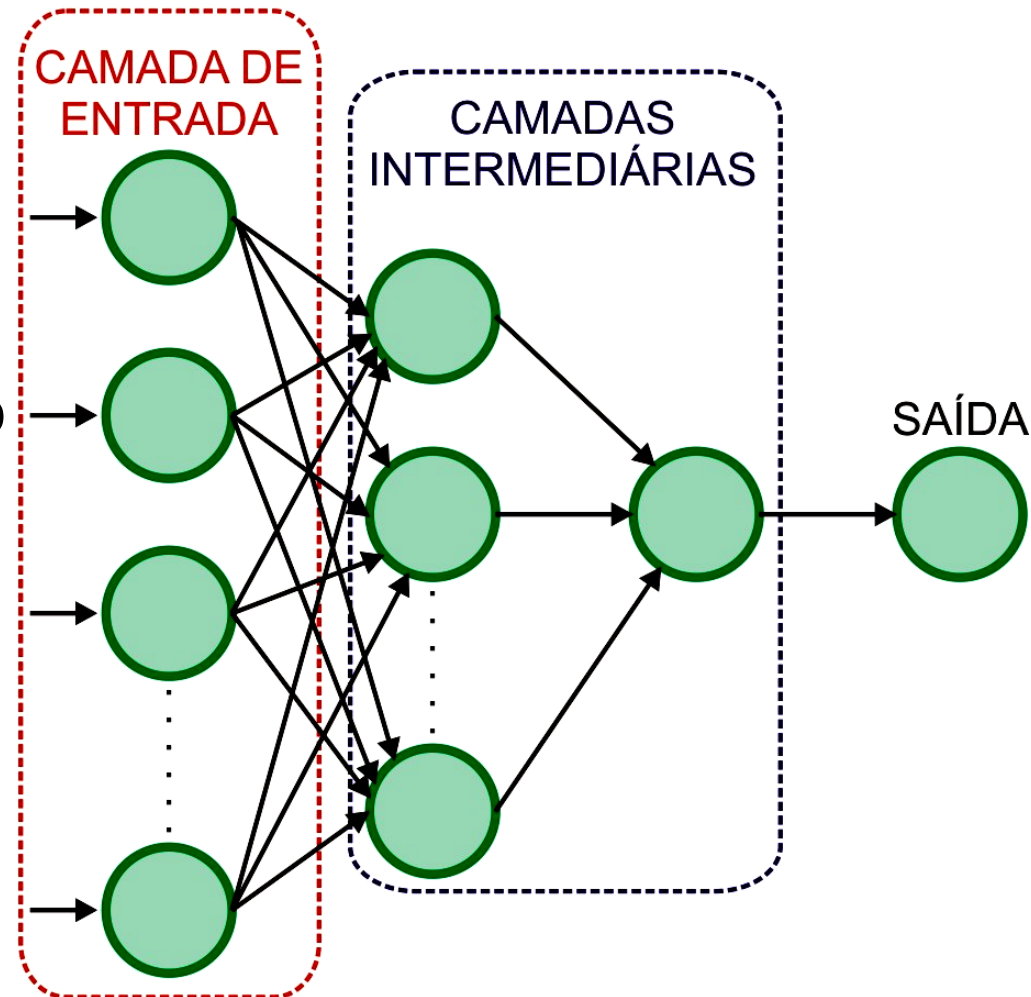
3 – Projeto de Redes Neurais

3.1 – Topologias:

Redes *Feed-Forward*: (alimentação à diante)

- São redes de uma ou mais camadas de processadores cujo fluxo de dados é sempre em uma única direção, isto é, não existe realimentação.
- Possui camadas intermediárias (ocultas).

REDE NEURAL - Topologia Feed-Forward



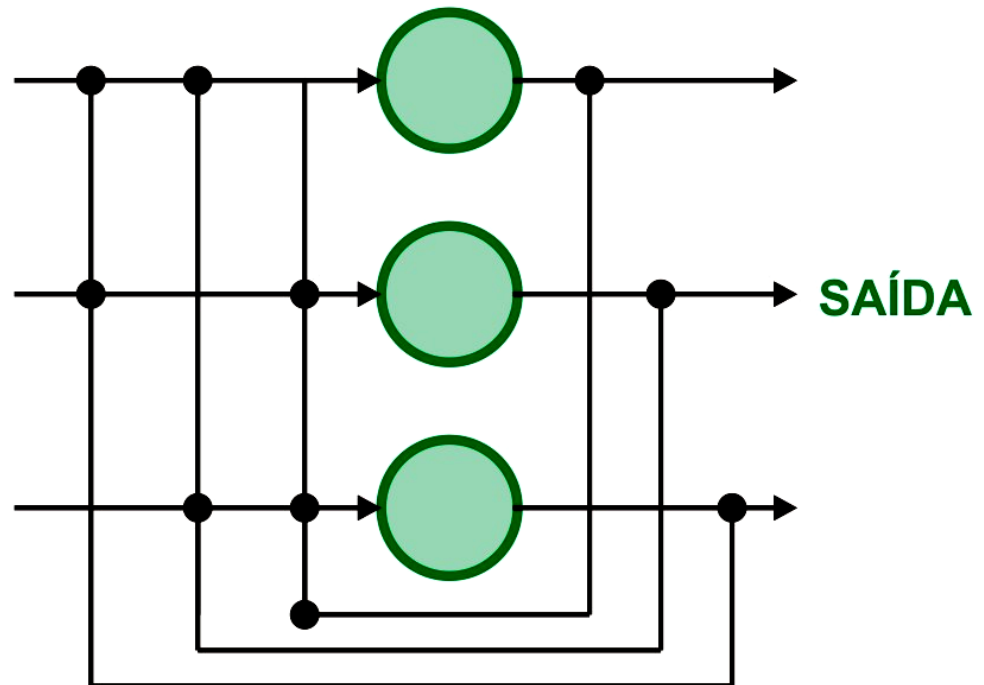
3 – Projeto de Redes Neurais

3.1 – Topologias:

Redes Recorrentes: (realimentação)

- São redes com conexões entre processadores da mesma camada e/ou processadores das camadas anteriores (*feedback* ou *realimentação*).

REDE NEURAL - Topologia Recorrente



3 – Projeto de Redes Neurais

3.1 – Topologias:

Totalmente Conectadas:

- Quando todos os neurônios de uma camada estão conectados a todos os neurônios da camada imediatamente a frente.

Parcialmente Conectadas:

- Quando não são totalmente conectadas.

3 – Projeto de Redes Neurais

3.1 – Topologias:

Multicamadas:

- Quando há mais de uma camada com neurônios processadores. A primeira camada se chama de **entrada**, a última se chama de **saída** e as camadas intermediárias são chamadas de **ocultas**.
- A quantidade de camadas, a quantidade de neurônios em cada camada e a forma de conexão entre os neurônios devem ser definidos antes do treinamento e dependem do problema a ser definido.

3 – Projeto de Redes Neurais

3.2 – Aprendizagem Supervisionada:

- Redes cujo treinamento de aprendizagem **depende da inserção** dos valores de **entrada** e de **saída desejada**.
- Durante o treinamento, a rede deve **compreender o padrão** embutido nos dados **de entrada e saída**.
- O treinamento ocorre através da **minimização do erro** encontrado na saída.
- Após o treinamento a rede deve ser capaz de **generalizar o que aprendeu**, estendendo os conceitos para os registros que não foram usados no treinamento.
- **Indicada para classificação dos dados.**

3 – Projeto de Redes Neurais

3.3 – Aprendizagem Não Supervisionada:

- Redes cujo treinamento de aprendizagem **depende da inserção apenas** dos valores de **entrada**, sendo a saída definida pela própria rede.
- O sistema **extraí as características** do conjunto de padrões, agrupando-os em classes inerentes aos dados.
- **Indicada para agrupamento dos dados (clusterização).**

3 – Projeto de Redes Neurais

3.4 – Aprendizagem por Reforço:

- Redes semelhantes ao treinamento supervisionado, no entanto, não contém um *target* para cada padrão de entrada.
- Possui um processo de realimentação com **signal de reforço (positivo ou negativo)** que avalia a resposta como boa ou ruim.
- O objetivo é **maximizar** a quantidade de **reforço positivo**.
- O sistema aprende somente com base nos resultados de sua **interação com o ambiente**.
- **Indicada para automação.**

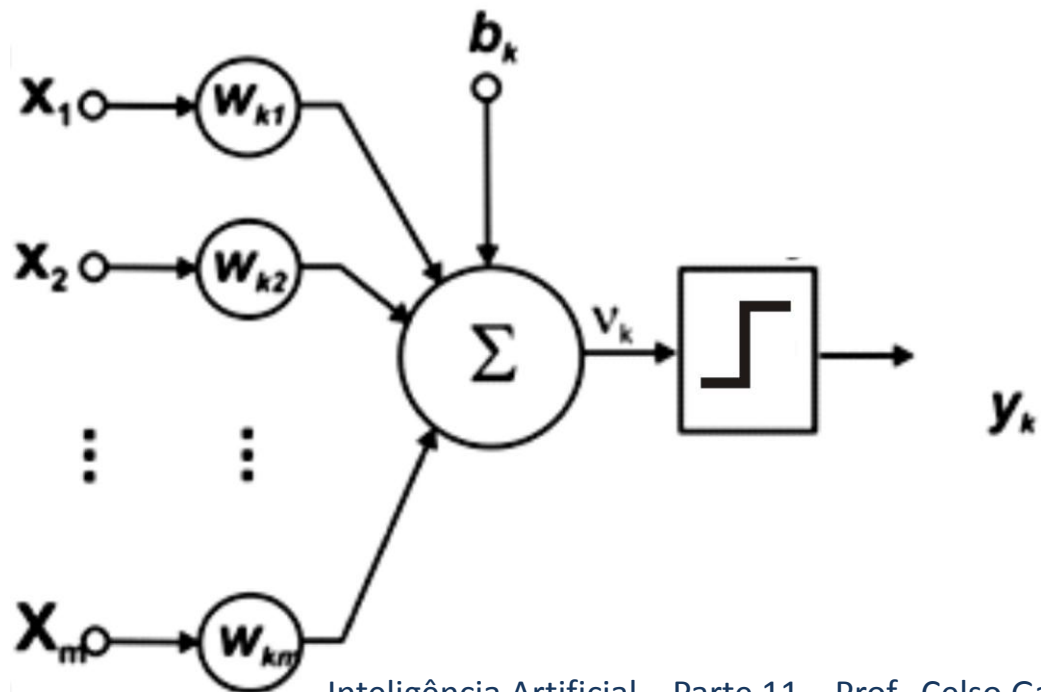
4

Rede *Perceptron*

4 – Rede *Perceptron*

4.1 – O Modelo *Perceptron* (Rosemblat, 1957):

- **Objetivo principal:** é classificar corretamente padrões de entrada em uma das duas classes possíveis.
- **Arquitetura básica de um neurônio artificial *Perceptron*:**



4 – Rede *Perceptron*

4.1 – O Modelo *Perceptron* (Rosemblat, 1957):

- Representação binária: valores de entrada/saída $\{-1, 1\}$.
- Topologia: uma única camada de pesos ajustáveis.
- Função de ativação: **degrau**.
- Apesar de não oferecer grande potencial seu estudo justifica-se por sua simplicidade e por ser historicamente importante.
- Cálculo da variável *net* é dado da forma como já vista anteriormente:

$$net = \sum w_{km} x_m + b_k$$

4 – Rede *Perceptron*

4.1 – O Modelo *Perceptron* (Rosemblat, 1957):

- O aprendizado ocorre alterando os valores dos pesos à cada iteração, a partir do erro da rede dado por:

$$\delta = d - y$$

- d é o valor de **saída desejado**.
- Y é o valor de **saída obtido** pela rede.

4 – Rede *Perceptron*

4.1 – O Modelo *Perceptron* (Rosemblat, 1957):

- Os pesos são alterados da seguinte forma:

$$\Delta x_i = \eta \cdot \delta \cdot x_i$$
$$W_{i(k+1)} = w_{i(k)} + \Delta x_i$$

- η = taxa de aprendizado:** valor próximo a zero que controla a velocidade com que os pesos são ajustados.
- δ = erro da rede** na iteração atual.
- x_i = valor da entrada.**

4 – Rede *Perceptron*

4.2 – O Algoritmo *Perceptron*:

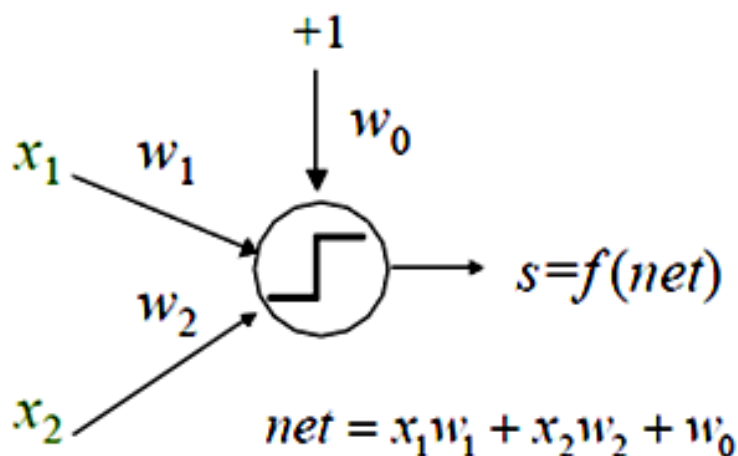
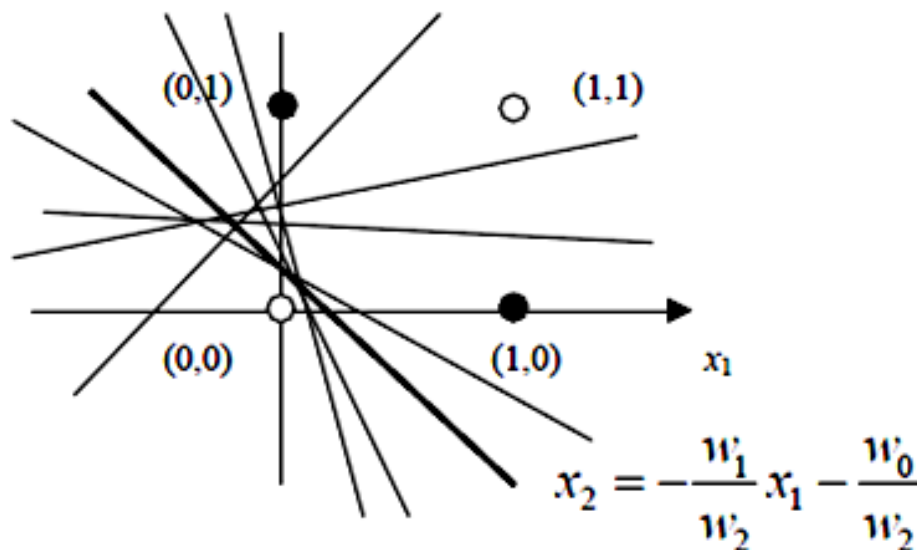
1. Iniciar os pesos sinápticos (w_i) com valores randômicos e pequenos;
2. Aplicar um padrão de entrada com seu respectivo valor desejado de saída (d)
3. Calcular a saída da rede (y) ;
4. Calcular o erro na saída ($\delta = d - y$) ;
5. Se $\delta \neq 0$ então atualizar os pesos ($w_{i(k+1)} = w_{i(k)} + \eta \cdot \delta x_i$) e voltar ao passo 3.
Se $\delta = 0$ mas treinamento não acabou voltar ao passo 2.
6. Encerrar treinamento.

4 – Rede *Perceptron*

4.3 – O problema do *Xor* no *Perceptron*:

- É impossível encontrar uma reta que separe os pontos corretamente, no caso do ou exclusivo:

x_1	x_2	$x_1 \otimes x_2$
0	0	0
0	1	1
1	0	1
1	1	0



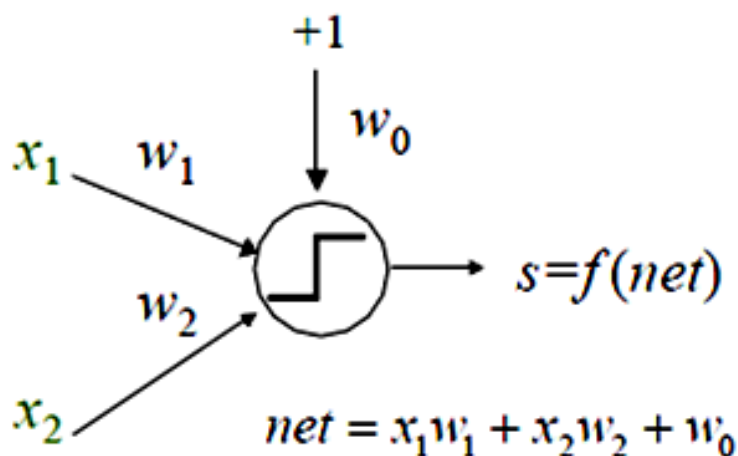
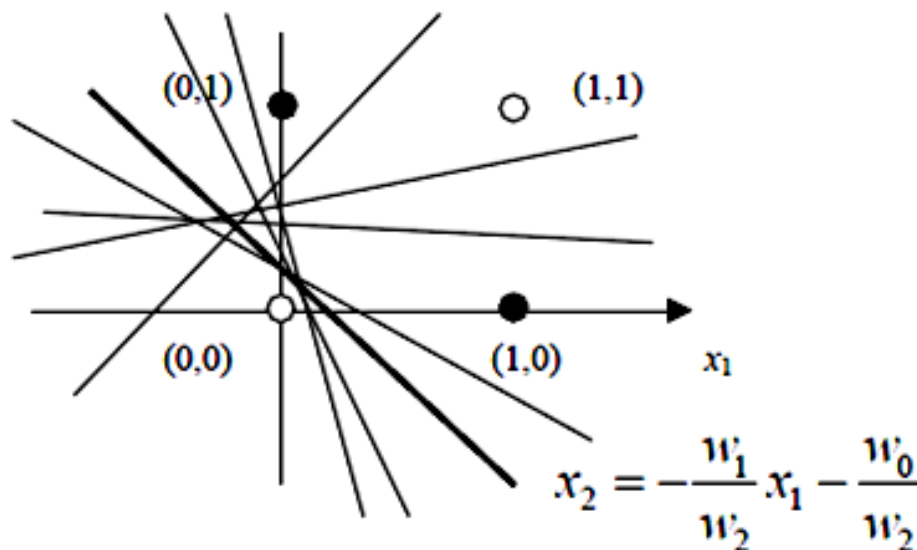
$$s = \begin{cases} 1 & \text{se } net > 0 \\ 0 & \text{se } net \leq 0 \end{cases}$$

4 – Rede *Perceptron*

4.3 – O problema do *Xor* no *Perceptron*:

- É impossível encontrar uma reta que separe os pontos corretamente, no caso do ou exclusivo:

x_1	x_2	$x_1 \otimes x_2$
0	0	0
0	1	1
1	0	1
1	1	0



$$s = \begin{cases} 1 & \text{se } net > 0 \\ 0 & \text{se } net \leq 0 \end{cases}$$

4 – Rede Perceptron

4.4 – Exemplo de Rede Perceptron:

Livro Artero, pag 131

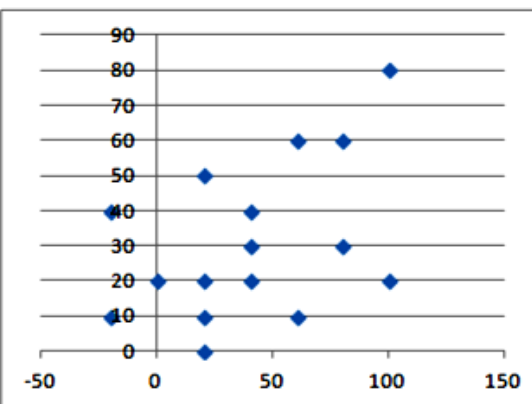
Taxa de aprendizagem = 0,2

	Entradas						Valor	Saída	Diferença				X	Y	
	x1	x2	Bias	w1	w2	wBias	esperado			f(net)	$\Delta w1$	$\Delta w2$			$\Delta wBias$
1ª	20	10	0	0,1	0,1	0,1	0	3	1	-1	-4,0	-2,0	0,0	20	10
	80	60	0	-3,9	-1,9	0,1	1	-426	0	1	16,0	12,0	0,0	80	60
	-20	10	1	12,1	10,1	0,1	1	-140,9	0	1	-4,0	2,0	0,2	-20	10
2ª	20	10	1	8,1	12,1	0,3	0	283,3	1	-1	-4,0	-2,0	-0,2	20	0
	80	60	1	4,1	10,1	0,1	1	934,1	1	0	0,0	0,0	0,0	20	20
	-20	10	1	4,1	10,1	0,1	1	19,1	1	0	0,0	0,0	0,0	40	30
3ª	20	10	1	4,1	10,1	0,1	0	183,1	1	-1	-4,0	-2,0	-0,2	40	40
	80	60	1	0,1	8,1	-0,1	1	493,9	1	0	0,0	0,0	0,0	40	20
	-20	10	1	0,1	8,1	-0,1	1	78,9	1	0	0,0	0,0	0,0	60	10
4ª	20	10	1	0,1	8,1	-0,1	0	82,9	1	-1	-4,0	-2,0	-0,2	100	20
	80	60	1	-3,9	6,1	-0,3	1	53,7	1	0	0,0	0,0	0,0	100	80
	-20	10	1	-3,9	6,1	-0,3	1	138,7	1	0	0,0	0,0	0,0	0	20
5ª	20	10	1	-3,9	6,1	-0,3	0	-17,3	0	0	0,0	0,0	0,0	-20	40
	80	60	1	-3,9	6,1	-0,3	1	53,7	1	0	0,0	0,0	0,0	60	60
	-20	10	1	-3,9	6,1	-0,3	1	138,7	1	0	0,0	0,0	0,0	80	30

Treinamento encerrado, pois os pesos se estabilizaram, ou seja, a rede convergiu.

A partir de agora, utilize os pesos encontrados no treinamento:

Reais	20	50	1	-3,9	6,1	-0,3		226,7	1
	20	0	1	-3,9	6,1	-0,3		-78,3	0
	20	20	1	-3,9	6,1	-0,3		43,7	1
	40	30	1	-3,9	6,1	-0,3		26,7	1
	40	40	1	-3,9	6,1	-0,3		87,7	1
	40	20	1	-3,9	6,1	-0,3		-34,3	0
	80	30	1	-3,9	6,1	-0,3		-129,3	0
	100	20	1	-3,9	6,1	-0,3		-268,3	0
	100	80	1	-3,9	6,1	-0,3		97,7	1



5

Rede *Perceptron Backpropagation*

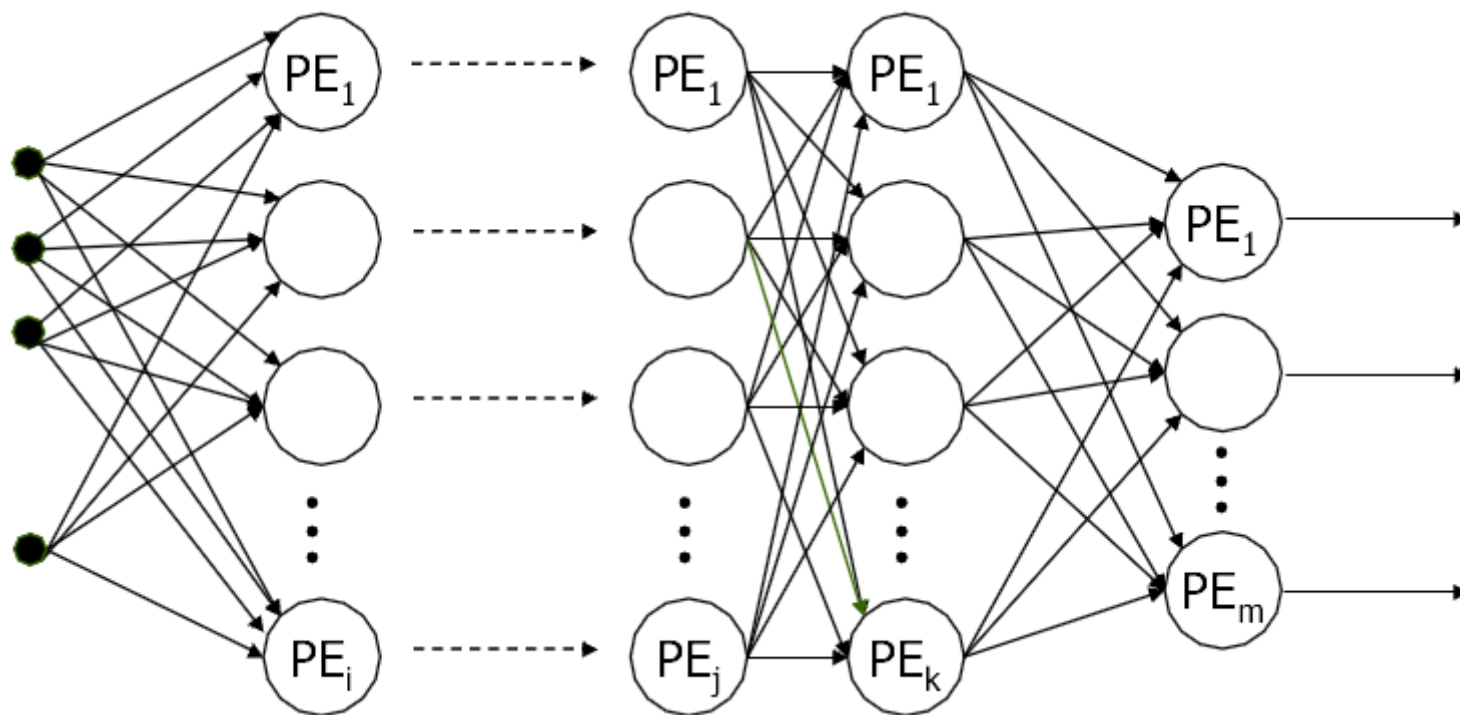
4 – Rede *Perceptron*

4.4 – A Rede *Perceptron Backpropagation*:

- Aprendizado supervisionado;
- Valores de entrada/saída binários ou **contínuos**;
- Topologia: **múltiplas camadas**;
- Função de ativação: **sigmóide** ou **linear**;
- Executa 2 estágios de processamento para cada padrão apresentado:
 - ***Feed-Forward***: as entradas se propagam pela rede, indo da camada de entrada até a camada de saída;
 - ***Feed-Backward***: os erros se propagam na direção contrária ao fluxo de dados, indo da camada de saída até a primeira camada escondida.

4 – Rede *Perceptron*

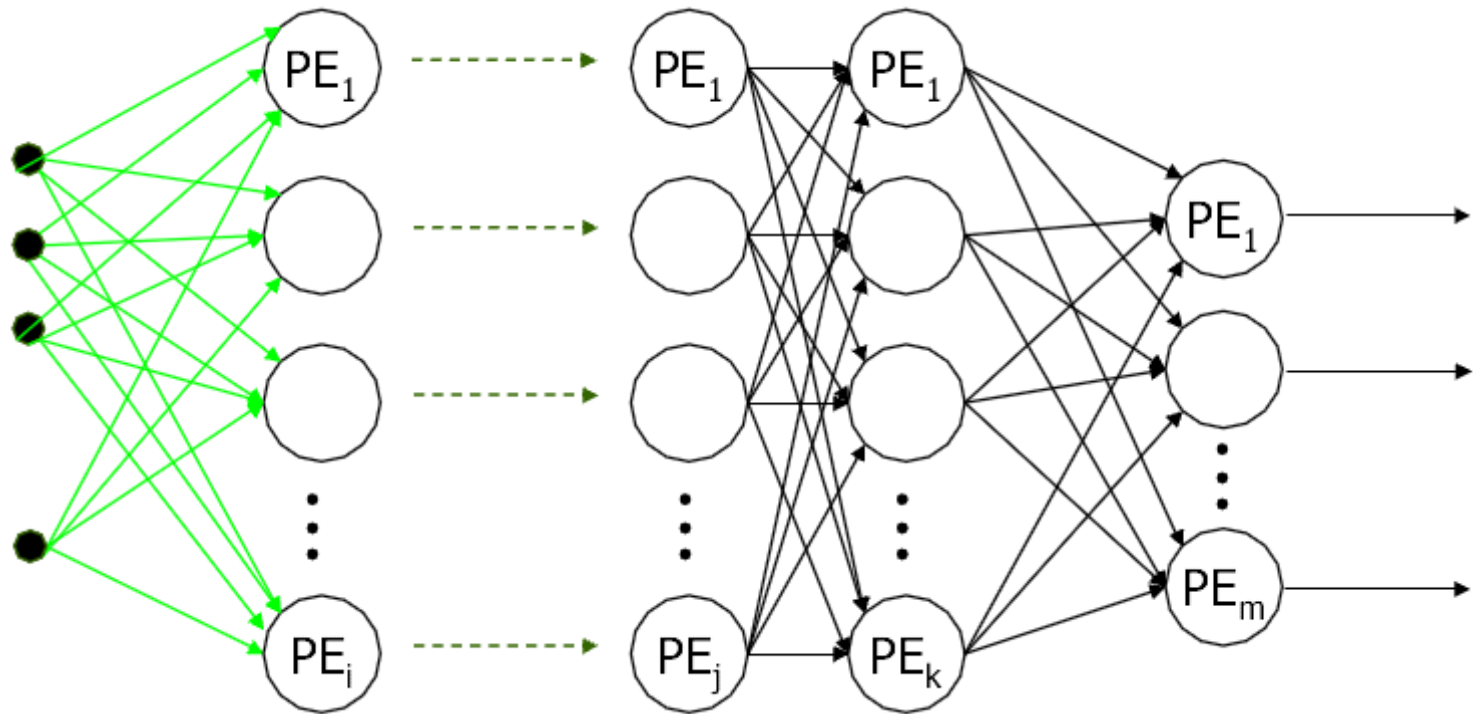
4.4 – A Rede *Perceptron* Backpropagation:



4 – Rede *Perceptron*

4.4 – A Rede *Perceptron Backpropagation*:

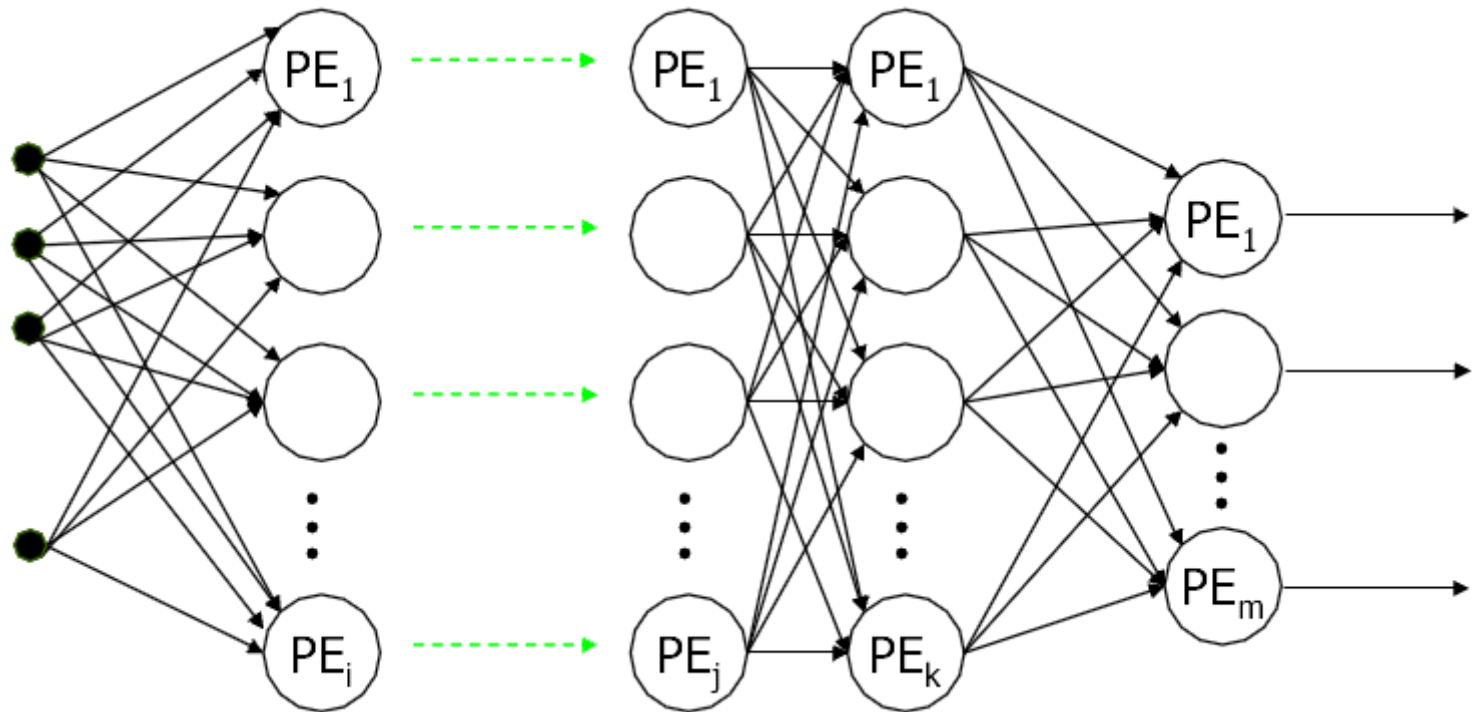
Estágio 1: *Feed-Forward* das entradas (\rightarrow)



4 – Rede *Perceptron*

4.4 – A Rede *Perceptron Backpropagation*:

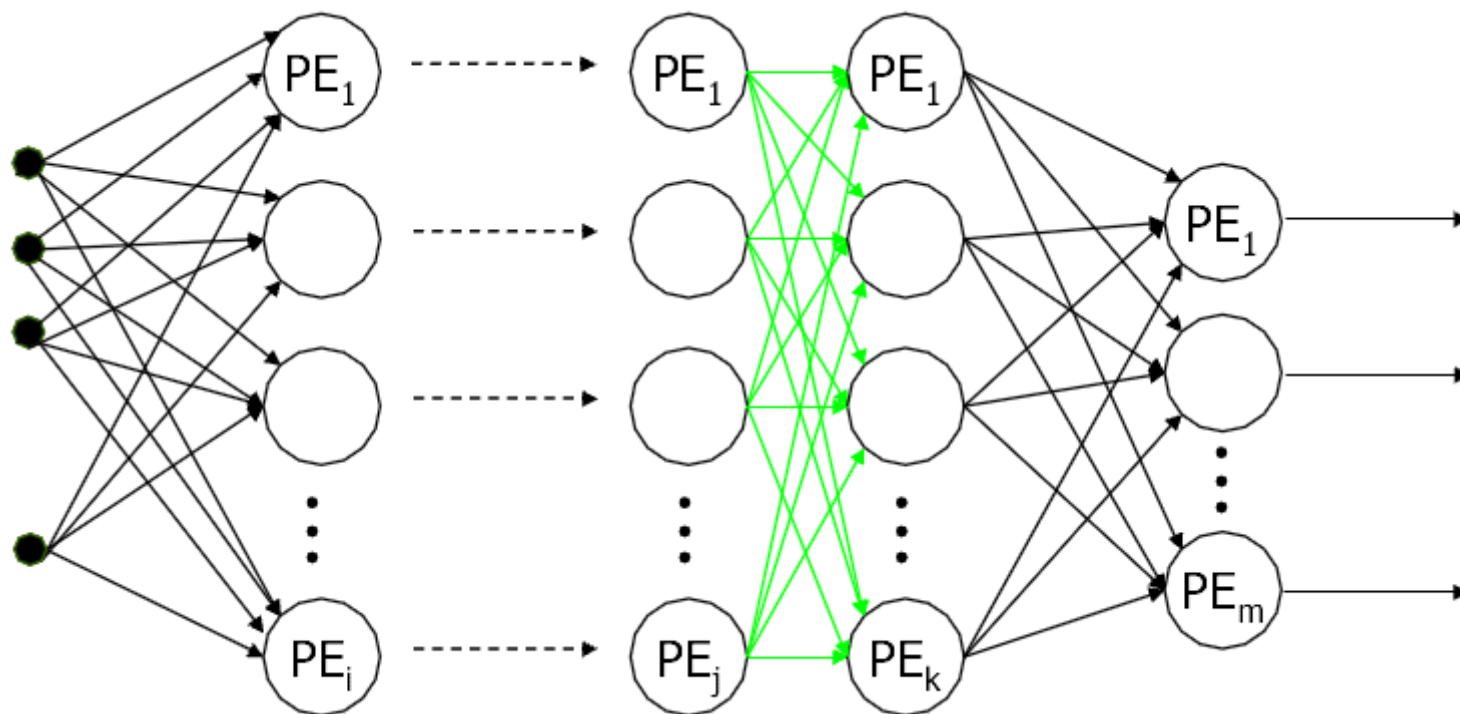
Estágio 1: *Feed-Forward* das entradas (\rightarrow)



4 – Rede *Perceptron*

4.4 – A Rede *Perceptron Backpropagation*:

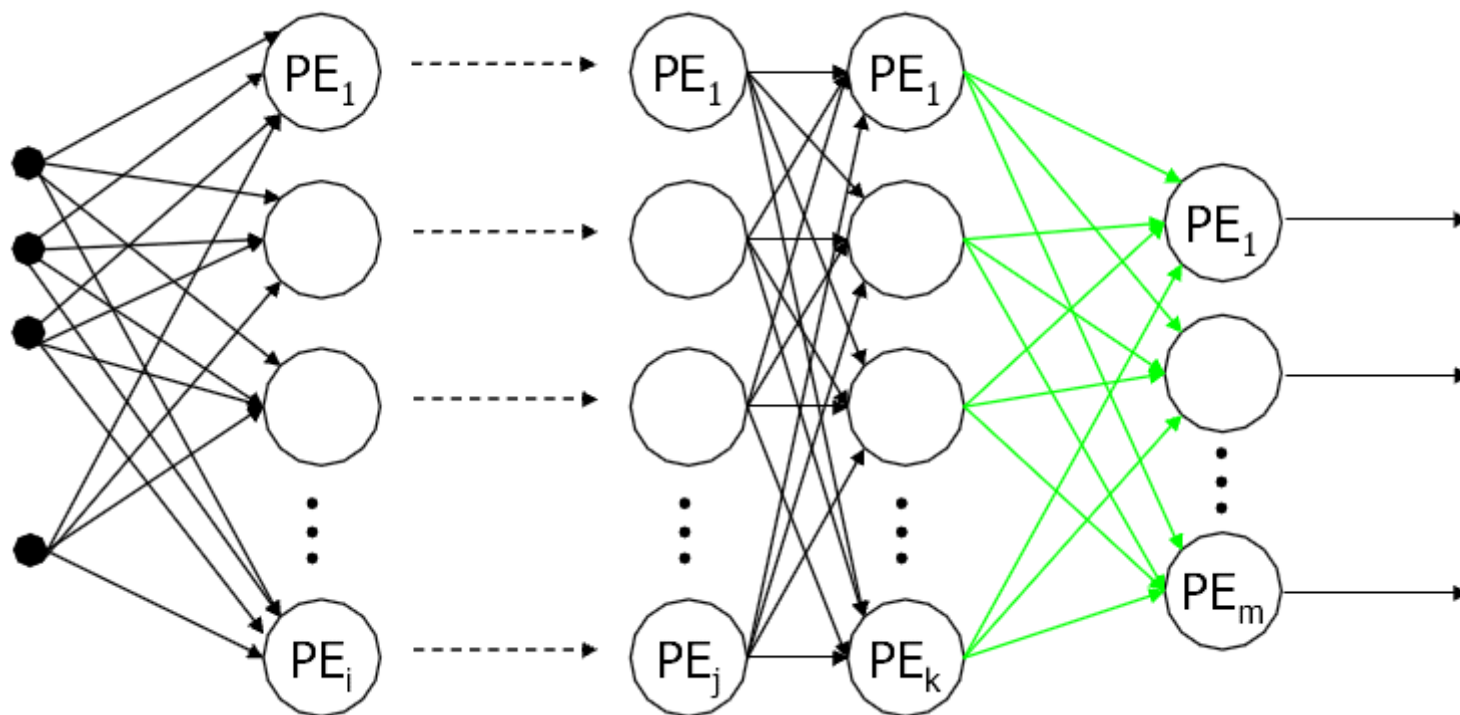
Estágio 1: *Feed-Forward* das entradas (\rightarrow)



4 – Rede *Perceptron*

4.4 – A Rede *Perceptron Backpropagation*:

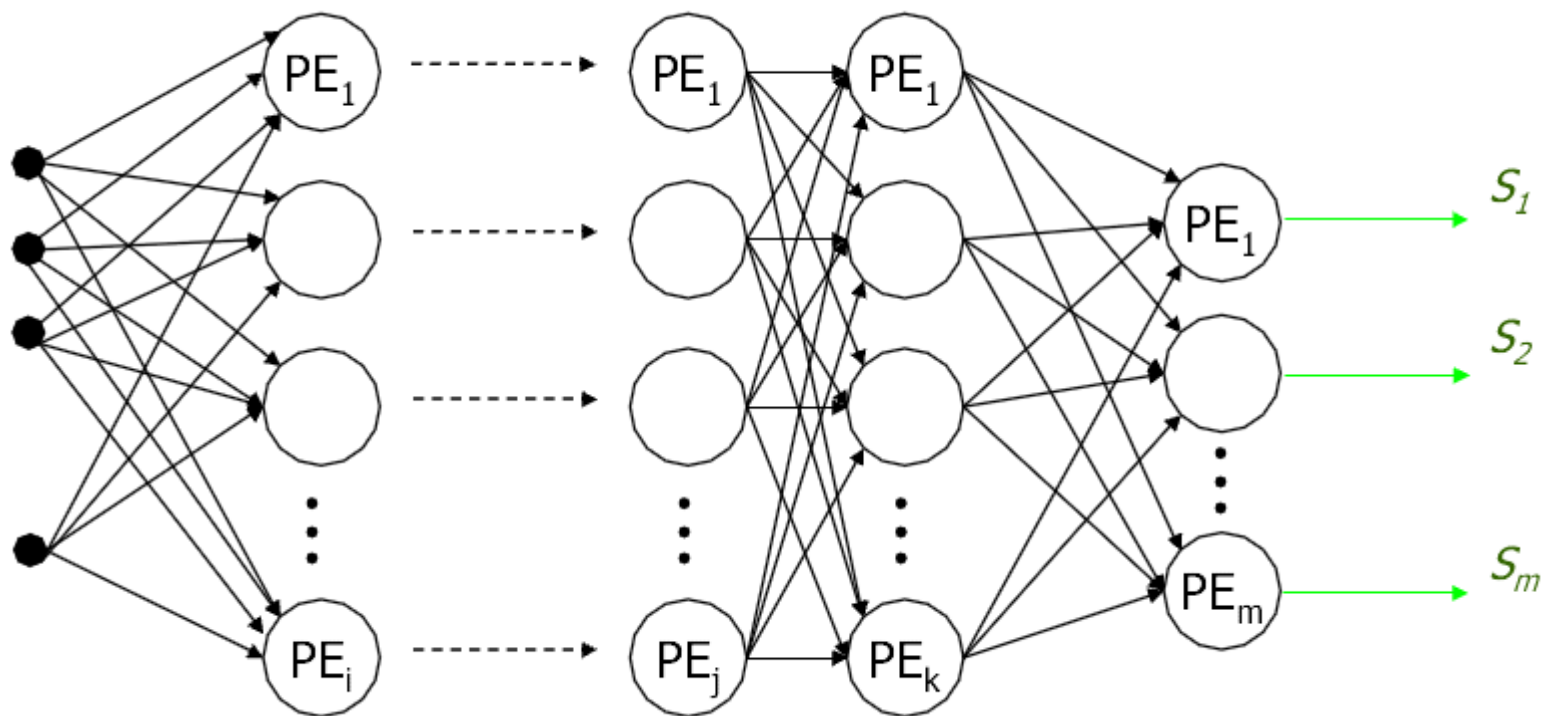
Estágio 1: *Feed-Forward* das entradas (\rightarrow)



4 – Rede *Perceptron*

4.4 – A Rede *Perceptron Backpropagation*:

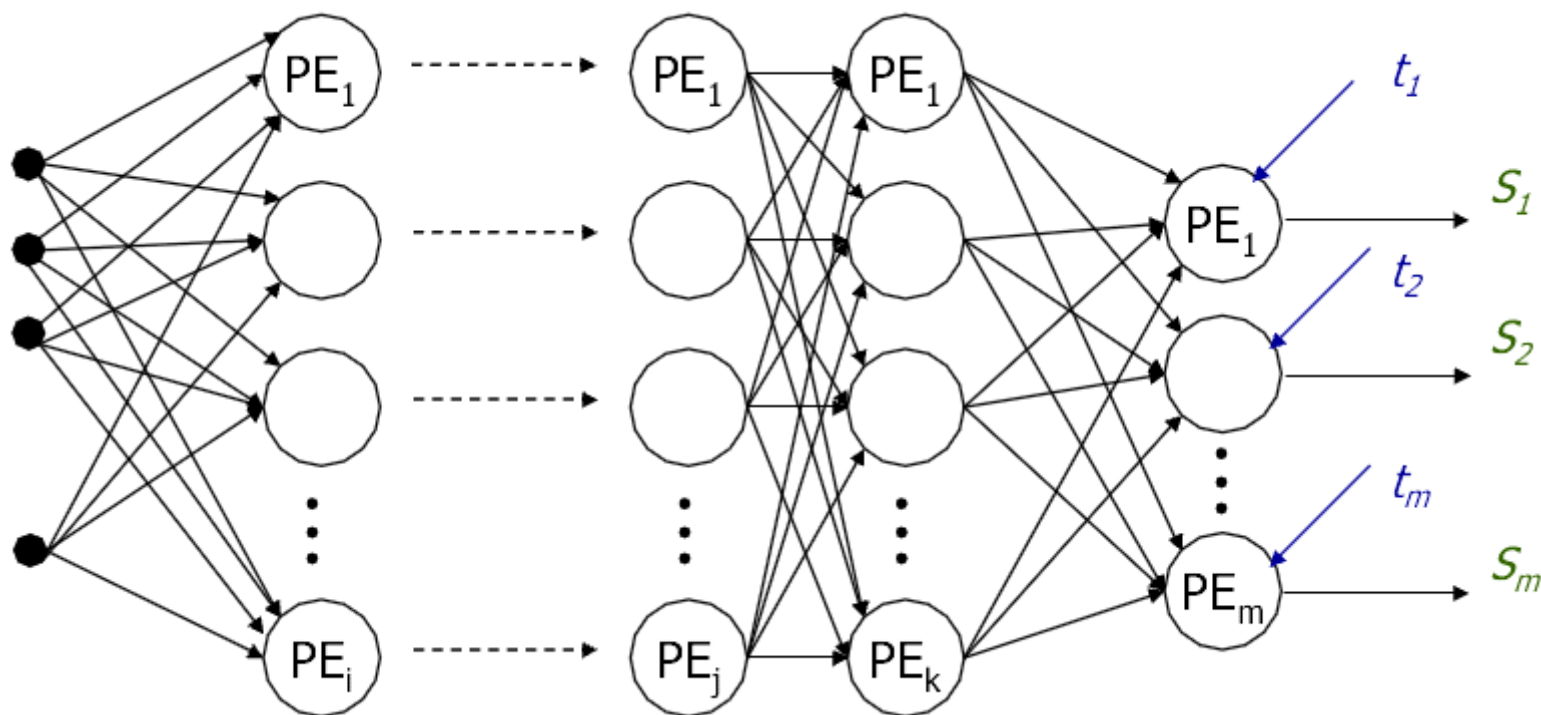
Estágio 1: *Feed-Forward* das entradas (\rightarrow)



4 – Rede *Perceptron*

4.4 – A Rede *Perceptron* Backpropagation:

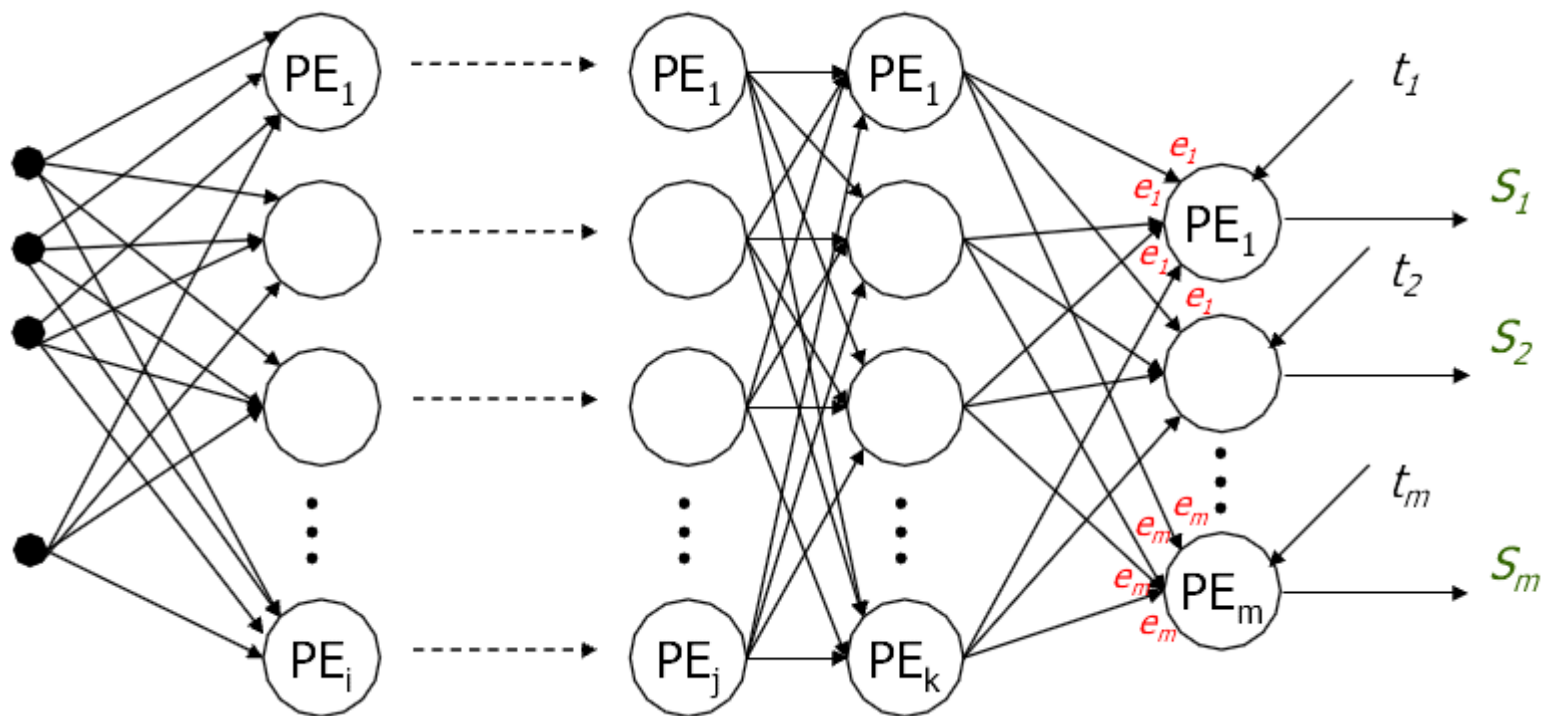
Estágio 1: *Feed-Forward* das entradas (\rightarrow)



4 – Rede *Perceptron*

4.4 – A Rede *Perceptron* Backpropagation:

Estágio 2: *Feed-Backward* dos erros (\leftarrow)

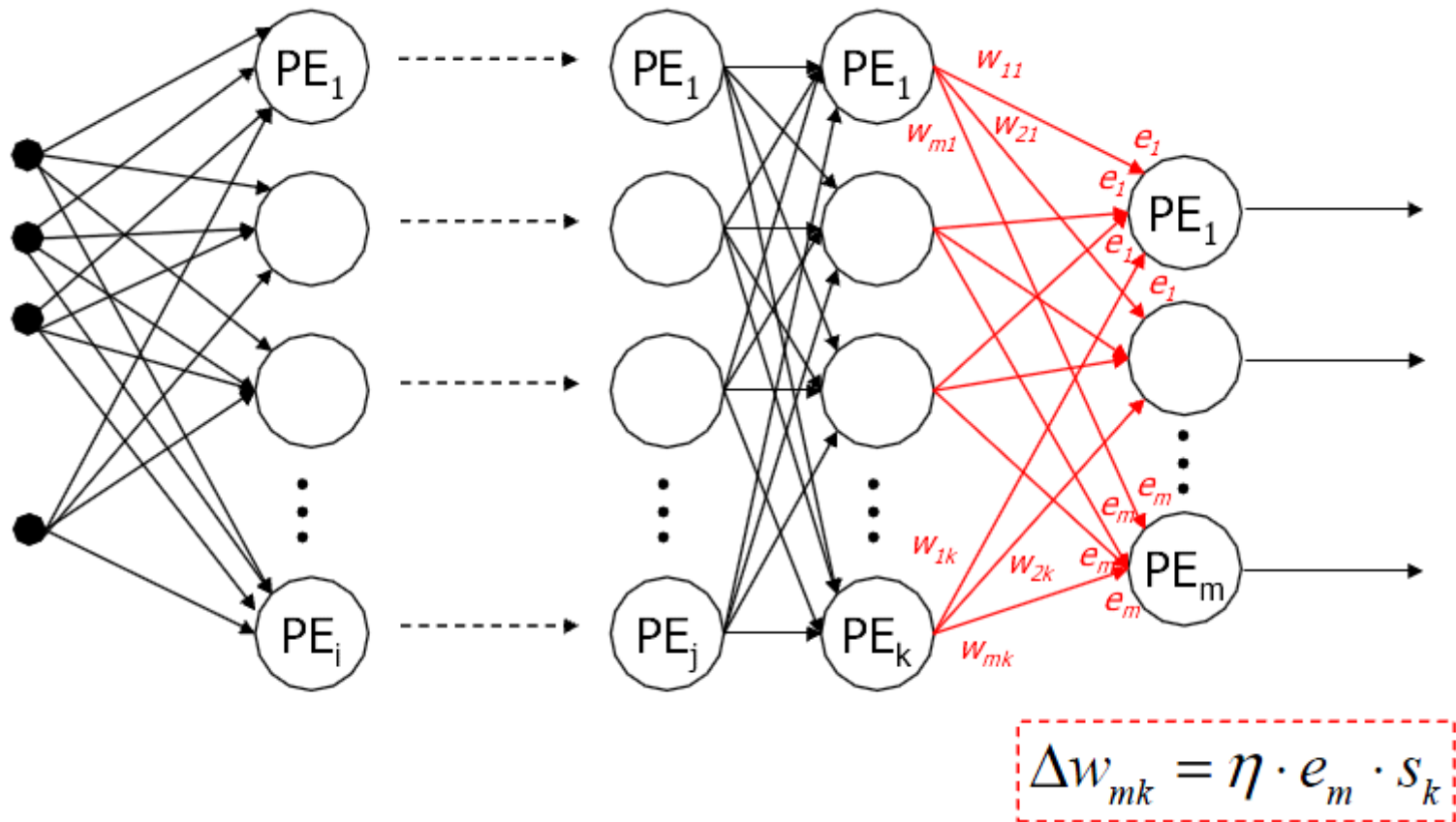


$$e_m = (t_m - s_m) f'(net_m)$$

4 – Rede *Perceptron*

4.4 – A Rede *Perceptron* Backpropagation:

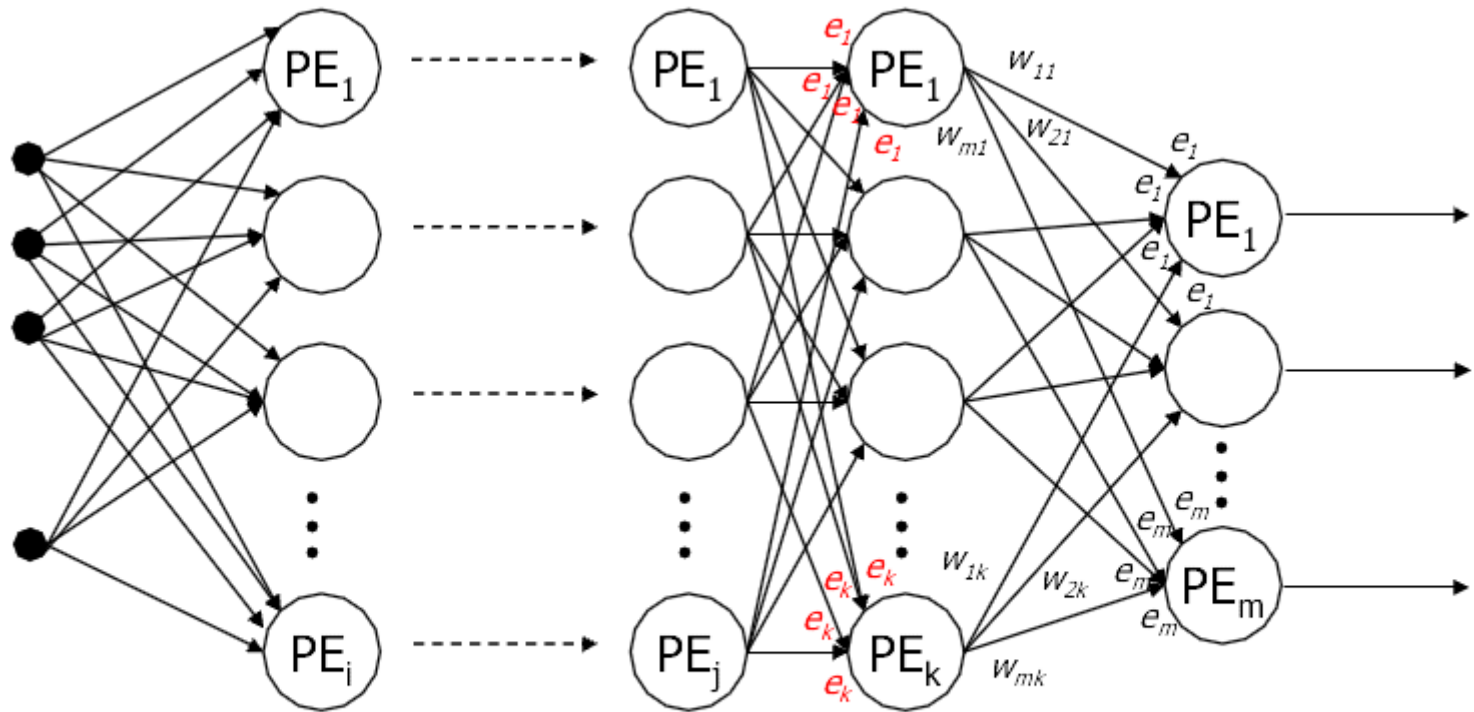
Estágio 2: *Feed-Backward* dos erros (\leftarrow)



4 – Rede *Perceptron*

4.4 – A Rede *Perceptron* Backpropagation:

Estágio 2: *Feed-Backward* dos erros (\leftarrow)

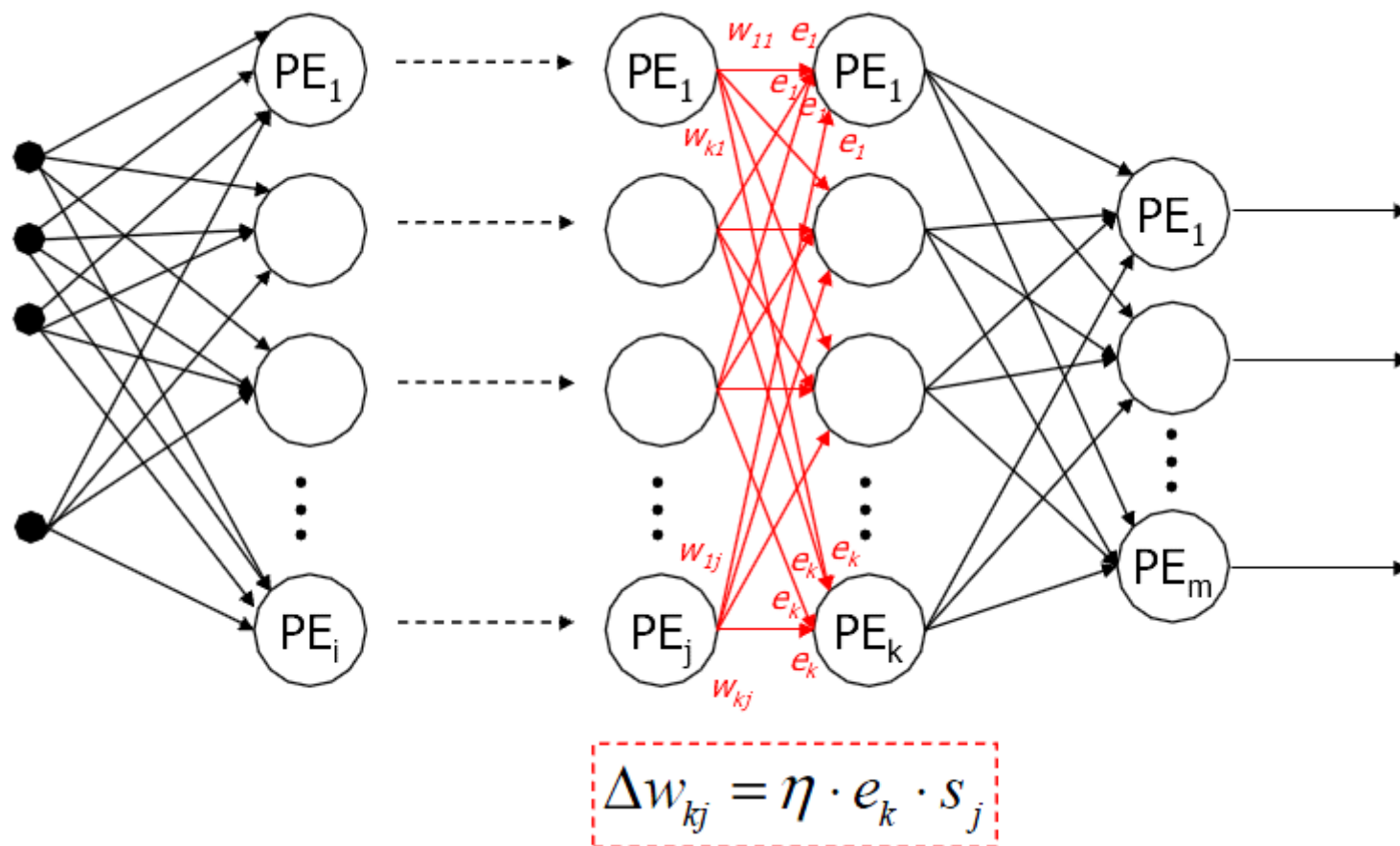


$$e_k = \left(\sum e_m w_{mk} \right) f'(net_k)$$

4 – Rede *Perceptron*

4.4 – A Rede *Perceptron Backpropagation*:

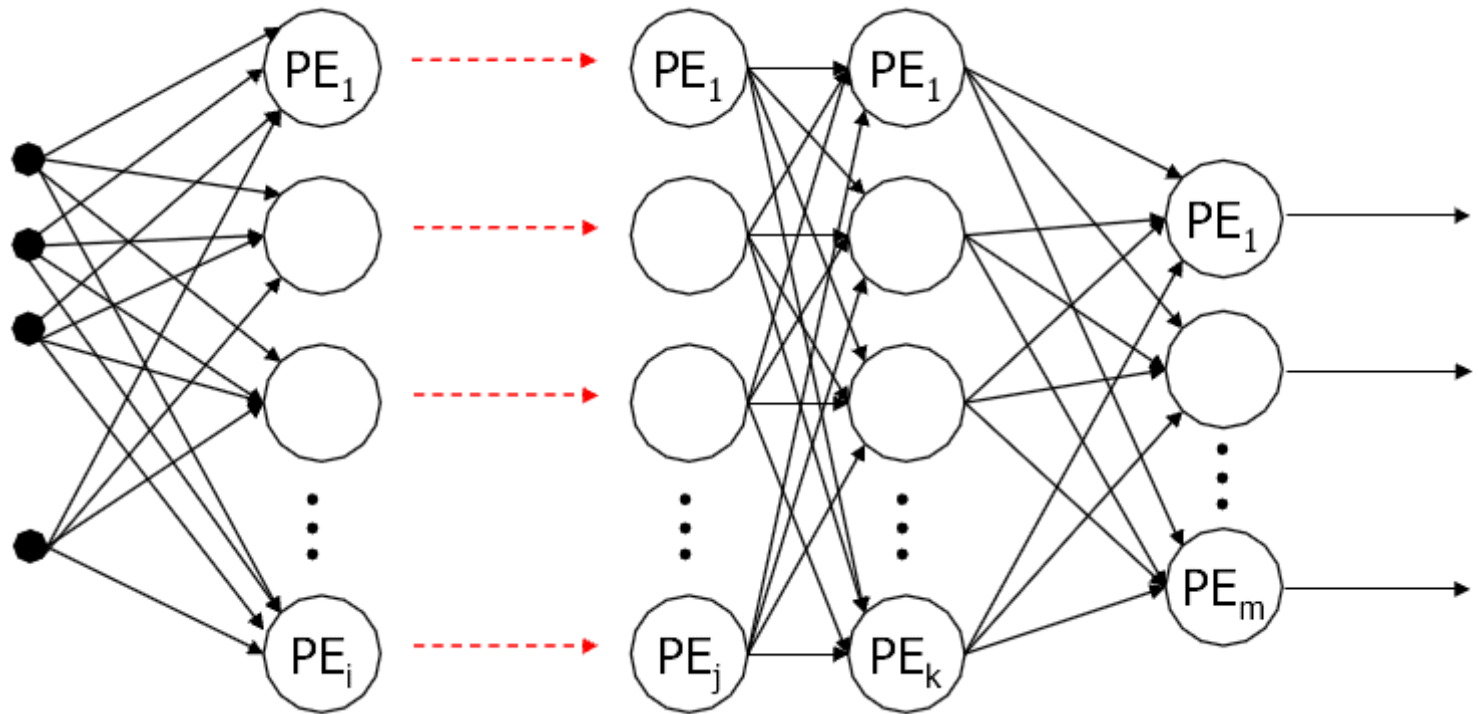
Estágio 2: *Feed-Backward* dos erros (\leftarrow)



4 – Rede *Perceptron*

4.4 – A Rede *Perceptron Backpropagation*:

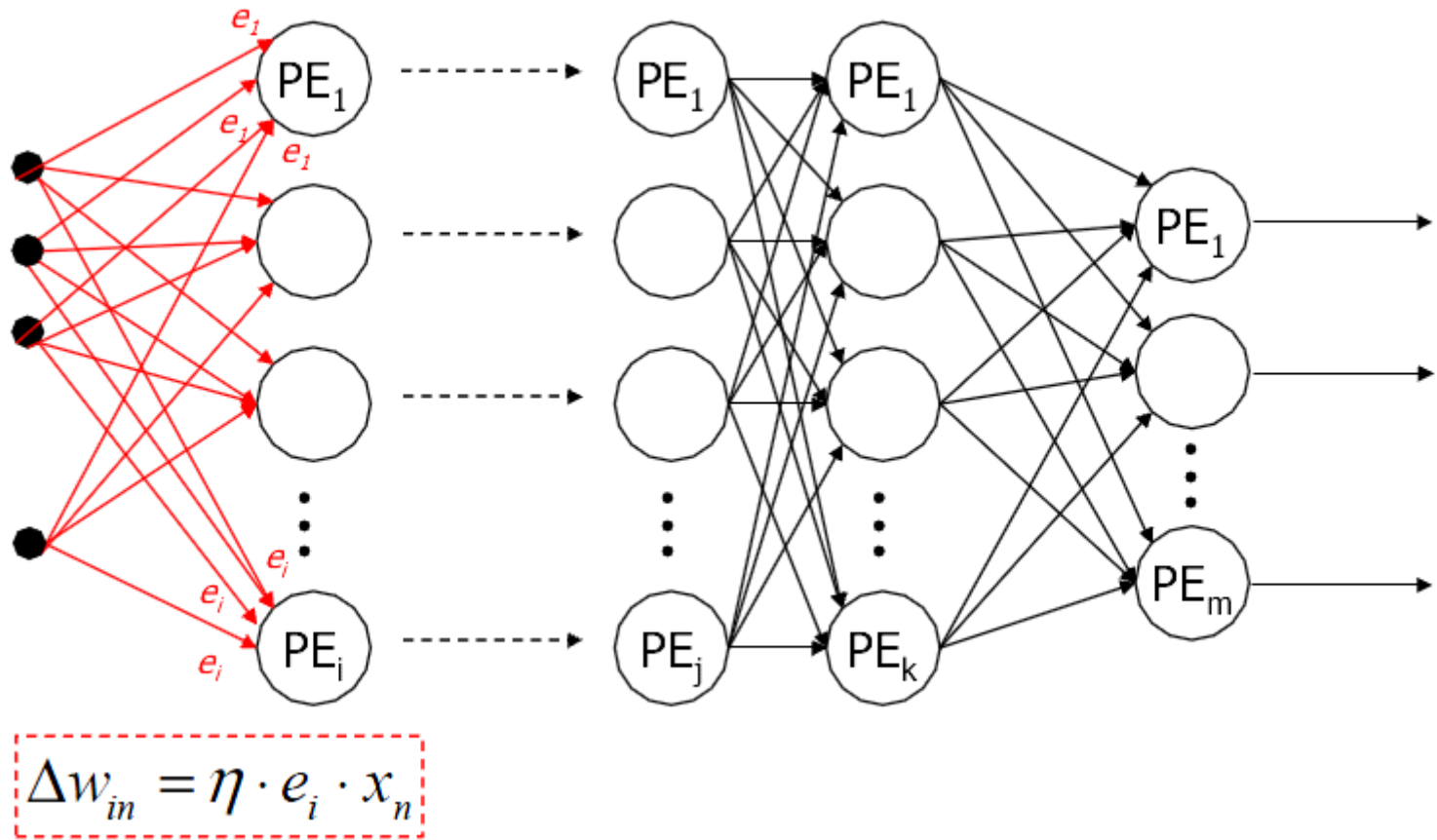
Estágio 2: *Feed-Backward* dos erros (\leftarrow)



4 – Rede *Perceptron*

4.4 – A Rede *Perceptron Backpropagation*:

Estágio 2: *Feed-Backward* dos erros (←)



4 – Rede *Perceptron*

4.4 – A Rede *Perceptron Backpropagation*:

- Este procedimento de aprendizado é repetido diversas vezes até que, para todos os processadores da camada de saída e para todos os padrões de treinamento, o **erro seja menor do que o especificado** como **tolerância**.
- Originalmente, esta rede é **completamente conectada** e a utilização de *bias* é opcional.
- Resolveu o problema do *Xor*.

Bibliografias

1. HAYKIN, Simon. **Redes Neurais: Princípios e Prática**. 2ª ed. traduzida, Porto Alegre: Bookman, 2001.
2. ROSA, João Luis Garcia. **Fundamentos da Inteligência Artificial**. 1ª ed. Rio de Janeiro: LTC, 2011, *Capítulo 9 - Redes Neurais Artificiais*.
3. ARTERO, Almir Olivette. **Inteligência Artificial: Teórica e Prática**. 1ª ed. São Paulo: Livraria da Física, 2009, *Capítulo 7 - Redes Neurais*.
4. SIMÕES, Marcelo Godoy; SHAW, Ian S. **Controle e Modelagem Fuzzy**. 2ª ed. São Paulo: Blusher Fapesp, 2007, *Capítulo 14 - Controladores Neurofuzzy*.

Outras Referências

1. THOMAZ, Carlos E. **Inteligência Computacional**. Departamento de Engenharia Elétrica do Centro Universitário da FEI, 2010.
2. BARRETO, Guilherme de Alencar. **Redes Neurais Artificiais: Conjuntos Fuzzy e Redes RBF**. Departamento de Engenharia de Teleinformática da Universidade Federal do Ceará, 2008.
3. <http://timedicina.blogspot.com.br/2009/10/redes-neurais-neurociencia-e.html>