# REDES NEURAIS ARTIFICIAIS

## AULA 4 – REDE *PERCEPTRON MULTICAMADAS*

Prof. Rodrigo Palácios

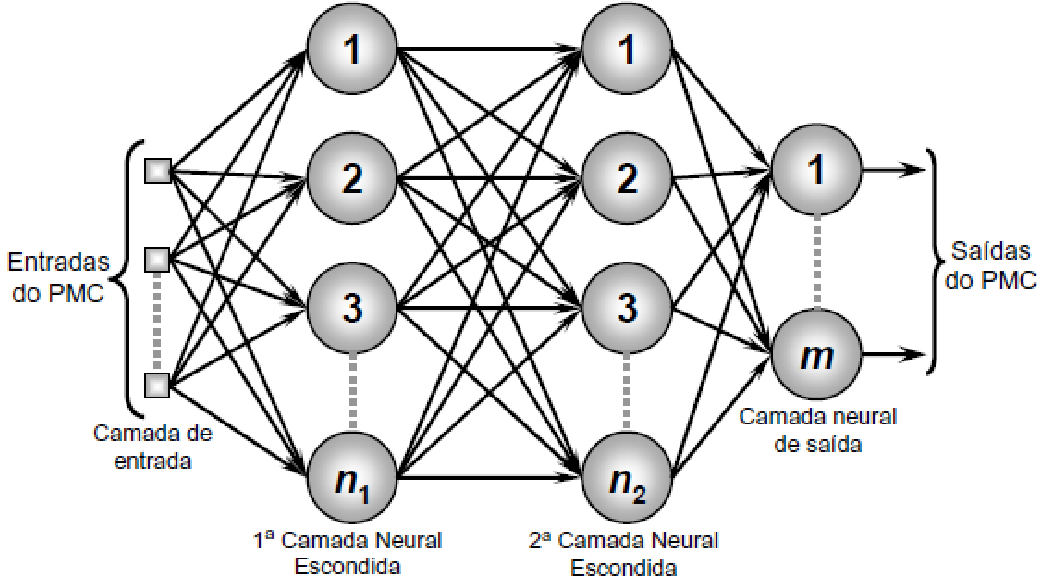
rodrigopalacios@utfpr.edu.br

# REDE *PERCEPTRON MULTICAMADAS*

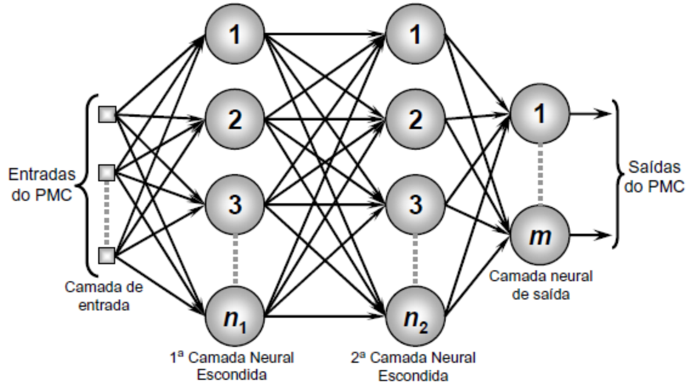
- **Aspectos da arquitetura**
  - Redes ***Perceptron de Múltiplas Camadas*** (***PMC***), também conhecidas como redes ***MLP*** (***Multi Layer Perceptron***), são caracterizadas pela presença de pelo menos uma camada intermediária (escondida) de neurônios.
  - As camadas intermediárias são aquelas situadas entre a camada de entrada e a respectiva camada neural de saída.
  - Consequentemente, as redes ***PMC*** possuem no mínimo duas camadas de neurônios, os quais estarão distribuídos entre as camadas intermediárias e a camada de saída.
  - Redes ***PMC*** é uma das mais versáteis quanto às suas aplicações, podendo ser utilizadas nos seguintes tipos de problemas:
    - Aproximação universal de funções.
    - Classificação de padrões.
    - Identificação e controle de processos.
    - Previsão de séries temporais.
    - Otimização de sistemas.
  - O ***PMC*** pertence à arquitetura *feedforward* de camadas múltiplas.
  - O treinamento do ***PMC*** é executado de forma **SUPERVISIONADA**.

# REDE *PERCEPTRON MULTICAMADAS*

- **Fluxo de Informações**
  - Síntese do fluxo de informações na estrutura da rede ***PMC***:
    1. Inicia-se na camada de entrada;
    2. Percorre, em seguida, as camadas intermediárias;
    3. Finaliza-se na camada neural de saída.
  - No ***PMC*** convencional inexiste qualquer tipo de realimentação de valores produzidos pela camada neural de saída ou pelas próprias camadas neurais intermediárias.



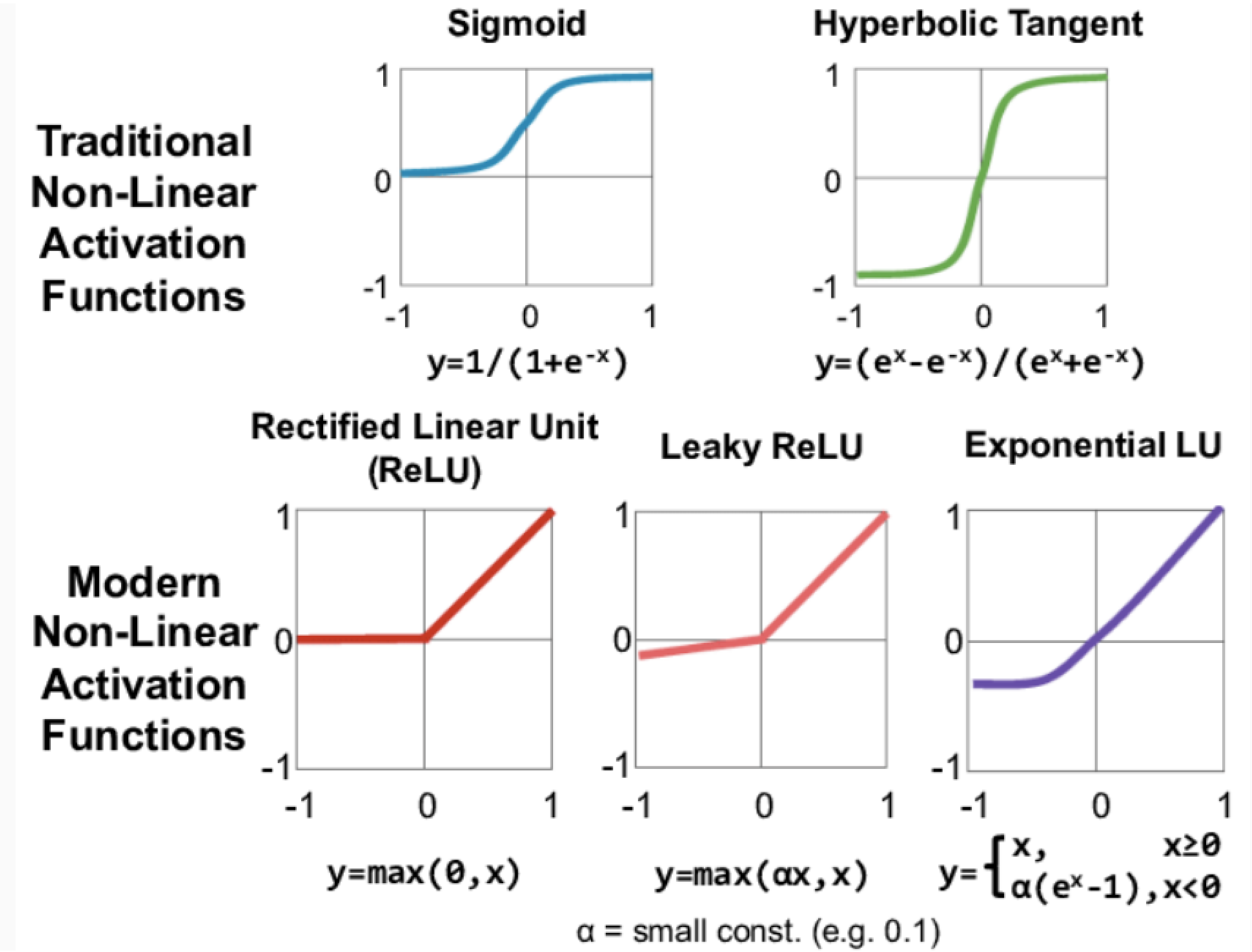
# REDE *PERCEPTRON MULTICAMADAS*

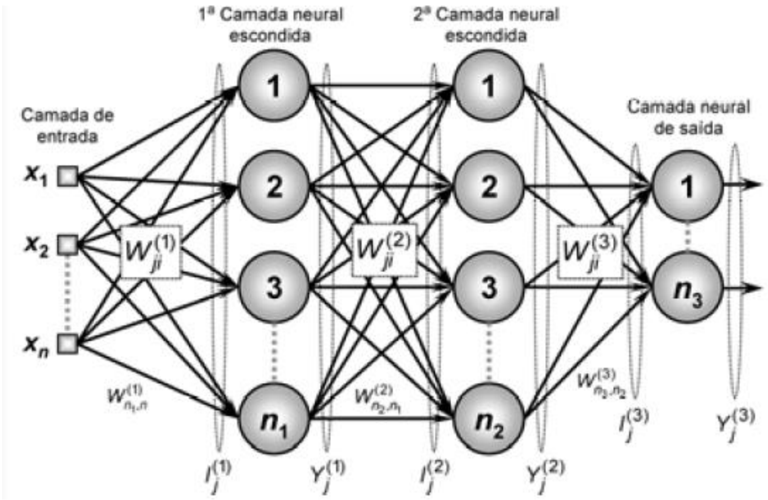



- **Princípios de Funcionamento**
  - Síntese do funcionamento da rede ***PMC***:
    1. As entradas do ***PMC***, representando os sinais advindos de determinada aplicação, será propagada camada-a-camada em direção à sua camada neural de saída.
    2. As saídas dos neurônios da primeira camada neural de saída serão as próprias entradas daqueles neurônios pertencentes à segunda camada neural escondida.
    3. As saídas dos neurônios da segunda camada neural escondida serão as respectivas entradas dos neurônios pertencentes à sua camada neural de saída.
  - Diferentemente do Perceptron e ADALINE, além da presença de camadas escondidas, a camada neural de saída do PMC pode ser composta por diversos neurônios:
    - Cada um destes neurônios de saída representaria uma das saídas do processo a ser mapeado.
    - As camadas intermediárias, por sua vez, extraem a maioria das informações referentes ao seu comportamento e as codificam por meio dos pesos sinápticos e limiares de seus neurônios.
  - O projeto de um PMC depende dos seguintes aspectos:
    - Classe de problema a ser tratado.
    - Disposição espacial das amostras de treinamento.
    - Valores iniciais atribuídos tanto aos parâmetros de treinamento como para as matrizes de pesos.
    - Nível de ruídos presentes nas amostras de treinamento.

# REDE *PERCEPTRON MULTICAMADAS*

- **Funções de ativação**

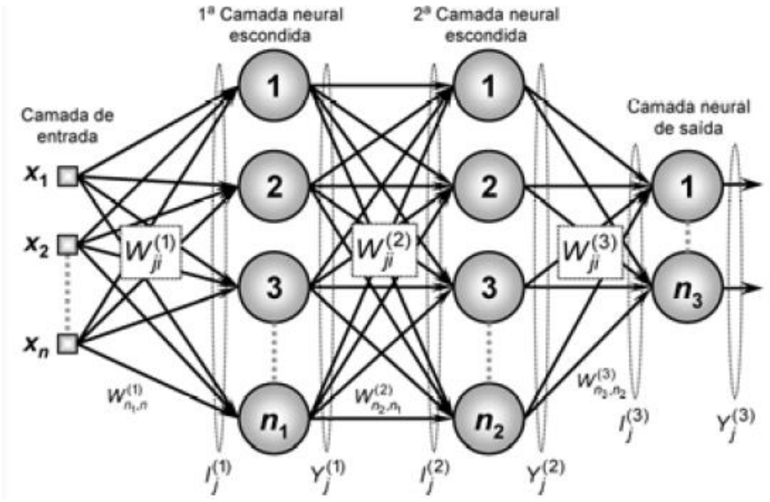



https://iaexpert.academy/2020/05/25/funcoes-de-ativacao-definicao-caracteristicas-e-quando-usar-cada-uma/

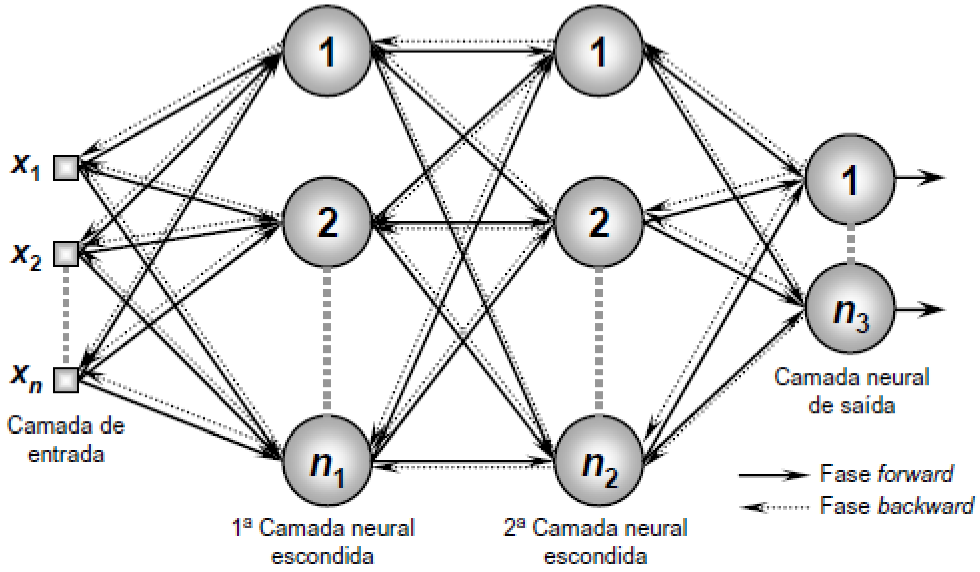
# REDE *PERCEPTRON MULTICAMADAS*

- **Funções de ativação**
  - Sigmóides e tangente hiperbólica não são tão eficientes quando a RNA possuem muitas camadas na rede.
  - Neste caso, em geral, aplica-se sigmóide ou tangente hiperbólica na última camada (na saída) quando o problema é de classificação de padrões.
  - Em redes com muitas camadas, nas internas adota-se outras funções de ativação como a ReLU ou outras mais especializadas para imagens (redes convolucionais).

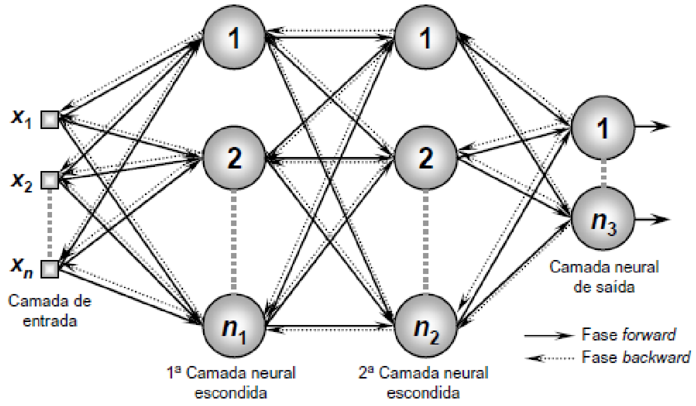


# REDE *PERCEPTRON MULTICAMADAS*

- **Introdução ao Algoritmo *backpropagation***
  - O processo de treinamento do ***PMC*** é feito mediante o algoritmo ***backpropagation***, conhecido também como *regra delta generalizada.*
    - O processo é realizado por meio das aplicações sucessivas de **duas fases** bem específicas.
  - Como ilustração, considera-se um ***PMC*** constituído de duas camadas escondidas, tendo-se a seguinte composição:
    - $n$ sinais em sua camada de entrada.
    - $n_1$ neurônios na primeira camada neural escondida.
    - $n_2$ neurônios na segunda camada neural escondida.
    - $n_3$ sinais associados à camada neural de saída (terceira camada neural).



# REDE *PERCEPTRON MULTICAMADAS*



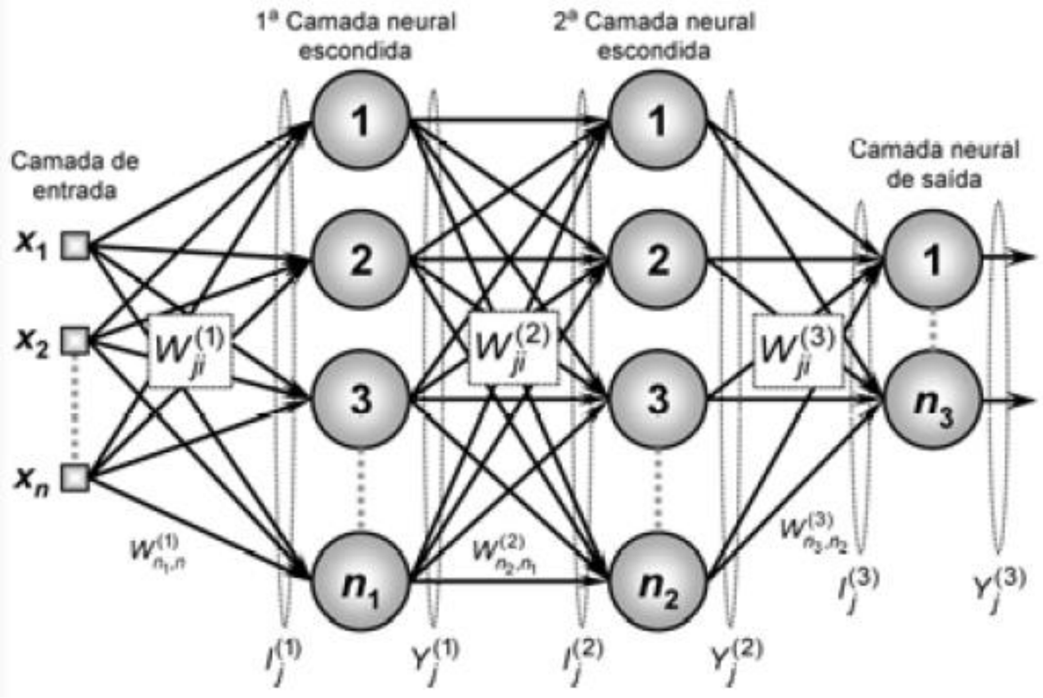

- **Fases do Algoritmo *backpropagation***
  - **Primeira Fase → *Forward* (propagação adiante)**
    - Os sinais $\{x_1, x_2, \ldots, x_n\}$ de uma amostra de treinamento são inseridos nas entradas da rede.
    - Estes são propagados camada-a-camada até a produção das respectivas saídas.
    - Leva-se em consideração apenas valores atuais de pesos sinápticos e limiares de seus neurônios, os quais permanecerão inalterados durante cada execução desta fase.
    - **CONCLUSÃO:** A aplicação desta fase visa tão somente obter as respostas da rede.
  - As respostas produzidas pelas saídas do ***PMC*** são comparadas com as respectivas respostas desejadas (aprendizado supervisionado).
  - **Segunda Fase → *Backward* (propagação reversa)**
    - Baseados nos desvios (erros) entre às respostas desejadas e àquelas produzidas pelos neurônios de saída, ajustam-se os pesos e limiares dos neurônio do ***PMC***.
    - **CONCLUSÃO:** A aplicação desta fase visa então ajustar pesos e limiares de todos os neurônios.
  - Em suma, tem-se:
    - As aplicações sucessivas de ambas as fazem com que os pesos sinápticos e limiares dos neurônios se ajustem automaticamente em cada iteração.
    - Consequentemente, ter-se-á então uma gradativa diminuição da soma dos erros produzidos pelas respostas da rede frente àquelas desejadas.
    - O processo cessa quando essa soma dos erros já estiver dentro de valores aceitáveis.

# REDE *PERCEPTRON MULTICAMADAS*

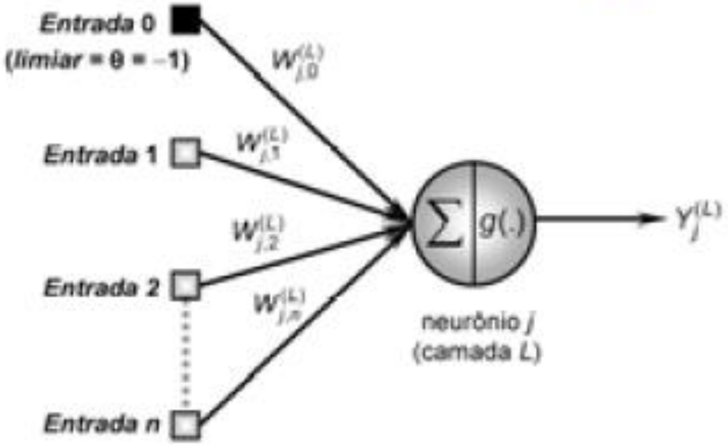
- **Derivação do Algoritmo *backpropagation***

$W_{ji}^{(L)}$ são matrizes de pesos cujos elementos denotam o valor do peso conectando o *j*-ésimo neurônio da camada (*L*) ao *i*-ésimo neurônio da camada (*L*-1). Para a topologia ilustrada, tem-se:

- $W_{ji}^{(3)}$ é o peso sináptico conectando o *j*-ésimo neurônio da camada de saída ao *i*-ésimo neurônio da camada 2.
- $W_{ji}^{(2)}$ é o peso sináptico conectando o *j*-ésimo neurônio da camada escondida 2 ao *i*-ésimo neurônio da camada 1.
- $W_{ji}^{(1)}$ é o peso sináptico conectando o *j*-ésimo neurônio da camada 1 ao *i*-ésimo sinal da camada de entrada.




Cada neurônio tem a seguinte configuração:

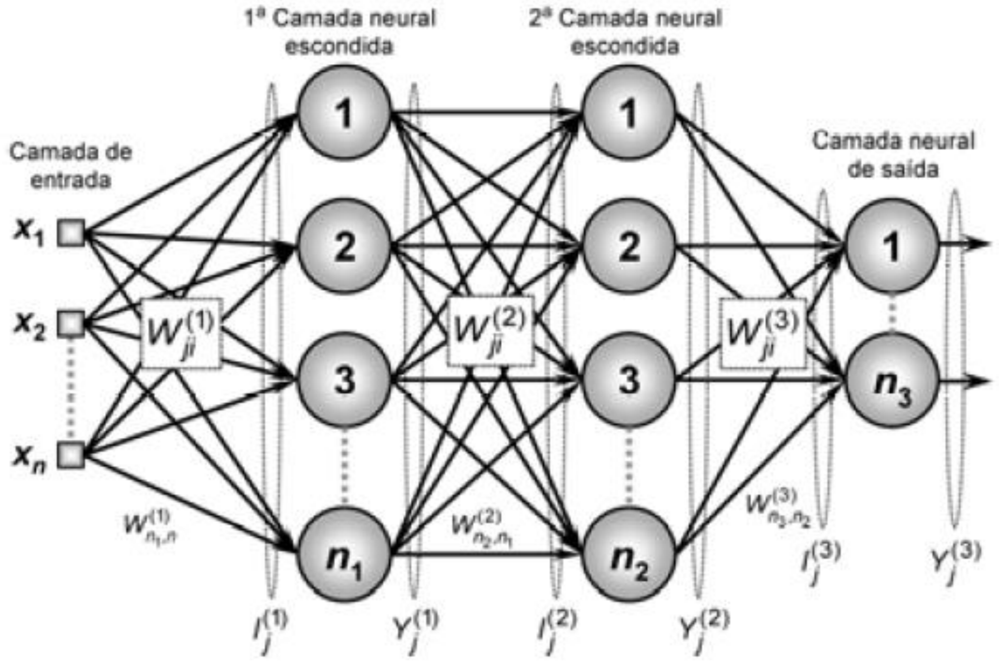


# REDE *PERCEPTRON MULTICAMADAS*

- **Definindo variáveis e parâmetros (Vetores de Entradas) (Algoritmo *backpropagation*)**

$I_j^{(L)}$ são vetores cujos elementos denotam a entrada ponderada em relação ao $j$-ésimo neurônio da camada $L$, os quais são definidos por:

- $$I_j^{(1)} = \sum_{i=0}^{n} W_{ji}^{(1)} \cdot x_i \Leftrightarrow I_j^{(1)} = W_{1,0}^{(1)} \cdot x_0 + W_{1,1}^{(1)} \cdot x_1 + \cdots + W_{1,n}^{(1)} \cdot x_n \qquad (1)$$
- $$I_j^{(2)} = \sum_{i=0}^{n_1} W_{ji}^{(2)} \cdot Y_i^{(1)} \Leftrightarrow I_j^{(2)} = W_{1,0}^{(2)} \cdot Y_0^{(1)} + W_{1,1}^{(2)} \cdot Y_1^{(1)} + \cdots + W_{1,n_1}^{(2)} \cdot Y_{n_1}^{(1)} \qquad (2)$$
- $$I_j^{(3)} = \sum_{i=0}^{n_2} W_{ji}^{(3)} \cdot Y_i^{(2)} \Leftrightarrow I_j^{(3)} = W_{1,0}^{(3)} \cdot Y_0^{(2)} + W_{1,1}^{(3)} \cdot Y_1^{(2)} + \cdots + W_{1,n_2}^{(3)} \cdot Y_{n_2}^{(2)} \qquad (3)$$



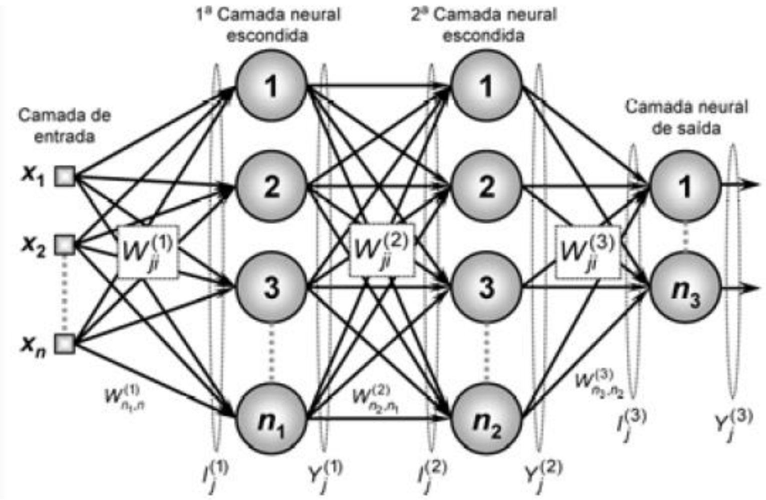
# REDE *PERCEPTRON MULTICAMADAS*

- **Definindo variáveis e parâmetros (Vetores de Saída) (Algoritmo *backpropagation*)**

$Y_j^{(L)}$ são vetores cujos elementos denotam a saída do *j*-ésimo neurônio em relação à camada *L*, os quais são definidos por:

- $Y_j^{(1)} = g(I_j^{(1)})$ **(4)**
- $Y_j^{(2)} = g(I_j^{(2)})$ **(5)**
- $Y_j^{(3)} = g(I_j^{(3)})$ **(6)**

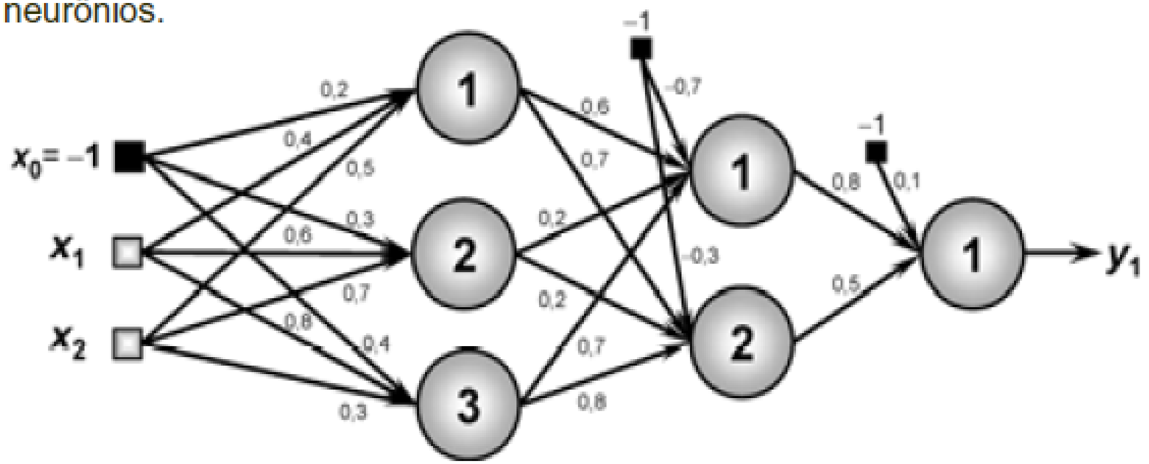
O funcional *g*(.) representa uma função de ativação que deve ser contínua e diferenciável em todo o seu domínio, tais como a função de ativação logística ou tangente hiperbólica.



# REDE *PERCEPTRON MULTICAMADAS*

- **Definindo variáveis e parâmetros (Exemplos) (Algoritmo *backpropagation*)**

Considera-se um *PMC* composto de duas entradas $x_1$ e $x_2$ ($n$ = 2), 3 neurônios na primeira camada escondida ($n_1$ = 3), 2 neurônios na segunda camada escondida ($n_2$ = 2) e um neurônio de saída ($n_3$ = 1). Considera-se também que a tangente hiperbólica é ativação para todos os neurônios.




$$W_{ji}^{(1)} = \begin{bmatrix} 0{,}2 & 0{,}4 & 0{,}5 \\ 0{,}3 & 0{,}6 & 0{,}7 \\ 0{,}4 & 0{,}8 & 0{,}3 \end{bmatrix}$$

$$W_{ji}^{(2)} = \begin{bmatrix} -0{,}7 & 0{,}6 & 0{,}2 & 0{,}7 \\ -0{,}3 & 0{,}7 & 0{,}2 & 0{,}8 \end{bmatrix}$$

$$W_{ji}^{(3)} = \begin{bmatrix} 0{,}1 & 0{,}8 & 0{,}5 \end{bmatrix}$$

## Cálculo de $I_j^{(1)}$ e $Y_j^{(1)}$ para $x_1$=0,3 e $x_2$ = 0,7:

$$I_j^{(1)} = \begin{bmatrix} I_1^{(1)} \\ I_2^{(1)} \\ I_3^{(1)} \end{bmatrix} = \begin{bmatrix} W_{1,0}^{(1)} \cdot x_0 + W_{1,1}^{(1)} \cdot x_1 + W_{1,2}^{(1)} \cdot x_2 \\ W_{2,0}^{(1)} \cdot x_0 + W_{2,1}^{(1)} \cdot x_1 + W_{2,2}^{(1)} \cdot x_2 \\ W_{3,0}^{(1)} \cdot x_0 + W_{3,1}^{(1)} \cdot x_1 + W_{3,2}^{(1)} \cdot x_2 \end{bmatrix} = \begin{bmatrix} 0{,}2 \cdot (-1) + 0{,}4 \cdot 0{,}3 + 0{,}5 \cdot 0{,}7 \\ 0{,}3 \cdot (-1) + 0{,}6 \cdot 0{,}3 + 0{,}7 \cdot 0{,}7 \\ 0{,}4 \cdot (-1) + 0{,}8 \cdot 0{,}3 + 0{,}3 \cdot 0{,}7 \end{bmatrix} = \begin{bmatrix} 0{,}27 \\ 0{,}37 \\ 0{,}05 \end{bmatrix}$$

$$Y_j^{(1)} = \begin{bmatrix} Y_1^{(1)} \\ Y_2^{(1)} \\ Y_3^{(1)} \end{bmatrix} = \begin{bmatrix} g(I_1^{(1)}) \\ g(I_2^{(1)}) \\ g(I_3^{(1)}) \end{bmatrix} = \begin{bmatrix} \tanh(0{,}27) \\ \tanh(0{,}37) \\ \tanh(0{,}05) \end{bmatrix} = \begin{bmatrix} 0{,}26 \\ 0{,}35 \\ 0{,}05 \end{bmatrix} \xrightarrow{Y_0^{(1)} = -1} Y_j^{(1)} = \begin{bmatrix} Y_0^{(1)} \\ Y_1^{(1)} \\ Y_2^{(1)} \\ Y_3^{(1)} \end{bmatrix} = \begin{bmatrix} -1 \\ 0{,}26 \\ 0{,}35 \\ 0{,}05 \end{bmatrix}$$

## Cálculo de $I_j^{(2)}$ e $Y_j^{(2)}$:

$$I_j^{(2)} = \begin{bmatrix} I_1^{(2)} \\ I_2^{(2)} \end{bmatrix} = \begin{bmatrix} W_{1,0}^{(2)} \cdot Y_0^{(1)} + W_{1,1}^{(2)} \cdot Y_1^{(1)} + W_{1,2}^{(2)} \cdot Y_2^{(1)} + W_{1,3}^{(2)} \cdot Y_3^{(1)} \\ W_{2,0}^{(2)} \cdot Y_0^{(1)} + W_{2,1}^{(2)} \cdot Y_1^{(1)} + W_{2,2}^{(2)} \cdot Y_2^{(1)} + W_{2,3}^{(2)} \cdot Y_3^{(1)} \end{bmatrix} = \begin{bmatrix} 0{,}96 \\ 0{,}59 \end{bmatrix}$$

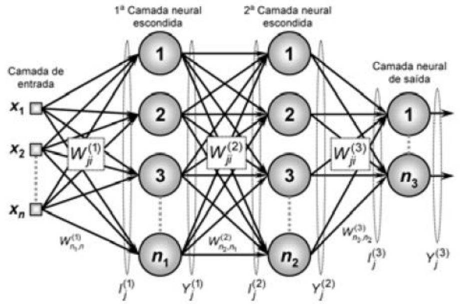
$$Y_j^{(2)} = \begin{bmatrix} Y_1^{(2)} \\ Y_2^{(2)} \end{bmatrix} = \begin{bmatrix} g(I_1^{(2)}) \\ g(I_2^{(2)}) \end{bmatrix} = \begin{bmatrix} \tanh(0{,}96) \\ \tanh(0{,}59) \end{bmatrix} = \begin{bmatrix} 0{,}74 \\ 0{,}53 \end{bmatrix} \xrightarrow{Y_0^{(2)} = -1} Y_j^{(2)} = \begin{bmatrix} Y_0^{(2)} \\ Y_1^{(2)} \\ Y_2^{(2)} \end{bmatrix} = \begin{bmatrix} -1 \\ 0{,}74 \\ 0{,}53 \end{bmatrix}$$

## Cálculo de $I_j^{(3)}$ e $Y_j^{(3)}$ (saída da rede):

$$I_j^{(3)} = \begin{bmatrix} I_1^{(3)} \end{bmatrix} = \begin{bmatrix} W_{1,0}^{(3)} \cdot Y_0^{(2)} + W_{1,1}^{(3)} \cdot Y_1^{(2)} + W_{1,2}^{(3)} \cdot Y_2^{(2)} \end{bmatrix} = \begin{bmatrix} 0{,}76 \end{bmatrix}$$

$$Y_j^{(3)} = \begin{bmatrix} Y_1^{(3)} \end{bmatrix} = \begin{bmatrix} g(I_1^{(3)}) \end{bmatrix} = \begin{bmatrix} \tanh(0{,}76) \end{bmatrix} = \begin{bmatrix} 0{,}64 \end{bmatrix}$$

# REDE *PERCEPTRON MULTICAMADAS*




- **Definindo a função representativa dos erros (desvios) (*Backpropagation*)**
  - A sua incumbência será medir o desvio entre as respostas produzidas pelos neurônios de saída da rede em relação aos respectivos valores desejados.
  - Considerando a *k*-ésima amostra de treinamento para a topologia ilustrada abaixo, assume-se a função erro quadrático como aquela a ser utilizada para medir o desempenho local associado aos resultados produzidos pelos neurônios de saída frente à referida amostra, ou seja:

$$E(k) = \frac{1}{2}\sum_{j=1}^{n_3}\left(d_j(k) - Y_j^{(3)}(k)\right)^2 \qquad (7)$$

onde $dj(k)$ é o respectivo valor desejado para a *k*-ésima amostra.

  - Consequentemente, para um conjunto de treinamento composto por *p* amostras, a evolução do desempenho global do aprendizado pode ser feito por meio da avaliação do "erro quadrático médio", isto é:

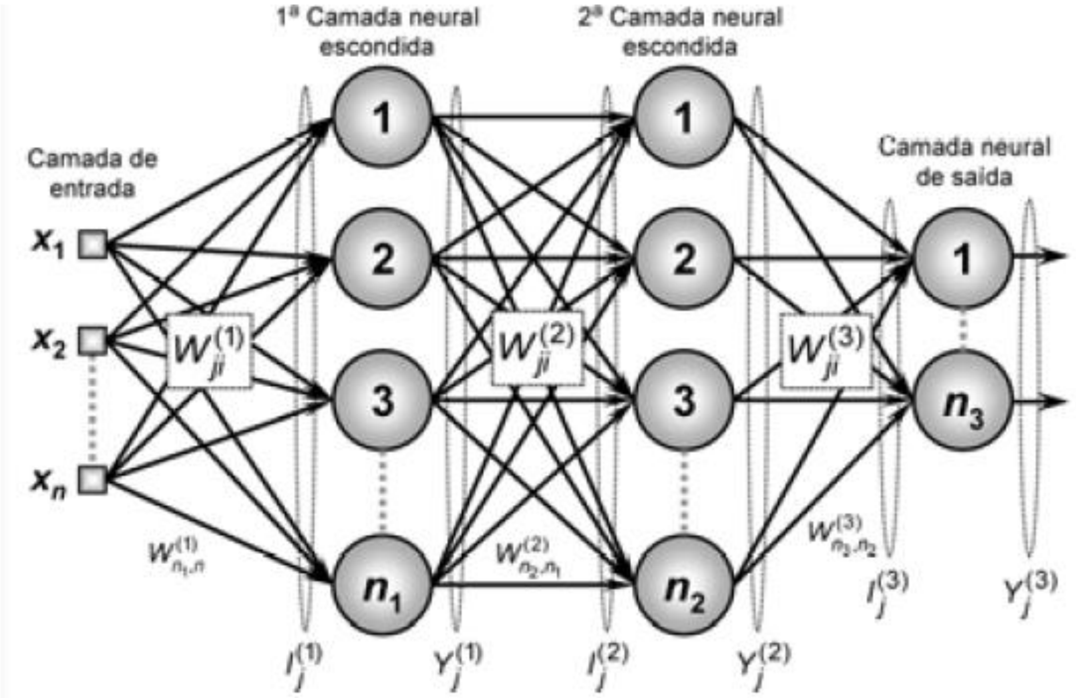
$$E_M = \frac{1}{p}\sum_{k=1}^{p} E(k) \qquad (8)$$

onde $E(k)$ é o erro quadrático obtido em (7).

  - Para melhor entendimento, divide-se o algoritmo em duas partes:
    - Parte I: destinada ao ajuste da matriz de pesos sinápticos referente à camada neural de saída.
    - Parte II: destinada ao ajuste das matrizes de pesos associadas às camadas intermediárias.

# REDE *PERCEPTRON MULTICAMADAS*

- **Parte I → Ajuste da matriz de pesos da camada de saída (*Backpropagation*)**
  - Consiste de ajustar a matriz $W_{ji}^{(3)}$ a fim de minimizar o erro entre a saída da rede frente à saída desejada.
- **Parte II(a) → Ajuste da matriz de pesos da 2ª camada escondida (*Backpropagation*)**
  - Consiste de ajustar a matriz $W_{ji}^{(2)}$.
- **Parte II(b) → Ajuste da matriz de pesos da 1ª camada escondida (*Backpropagation*)**
  - Consiste de ajustar a matriz $W_{ji}^{(1)}$.




- **Para conhecimento e estudo da matemática de ajuste dos pesos** $W_{ji}^{(1)}$, $W_{ji}^{(2)}$ e $W_{ji}^{(3)}$ **consultar o livro:** SILVA, Ivan Nunes da e SPATTI, Danilo Hernane e FLAUZINO, Rogério Andrade. Redes neurais artificiais para engenharia e ciências aplicadas. . São Paulo: Artliber Editora, Capítulo 5.

# REDE *PERCEPTRON MULTICAMADAS*

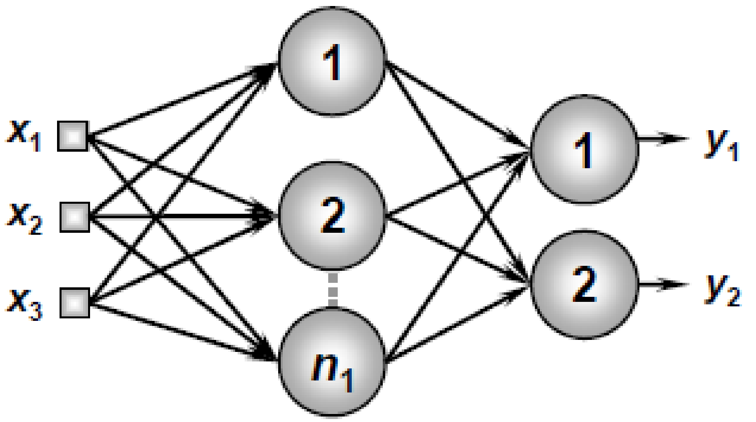
- **Aspectos de Preparação de Dados: Montagem do Conjunto de Treinamentos**
  - Supõe-se que um problema a ser mapeado pelo ***PMC*** tenha três entradas $\{x_1, x_2, x_3\}$, e duas saídas $\{y_1, y_2\}$ conforme a figura ao lado (abaixo).
  - Assume-se que se tem quatro amostras, constituída dos seguintes valores de entrada:

**Amostra 1** ➔ Entrada: [0,2  0,9  0,4] → Saída desejada: [0,7  0,3]

**Amostra 2** ➔ Entrada: [0,1  0,3  0,5] → Saída desejada: [0,6  0,4]

**Amostra 3** ➔ Entrada: [0,9  0,7  0,8] → Saída desejada: [0,9  0,5]

**Amostra 4** ➔ Entrada: [0,6  0,4  0,3] → Saída desejada: [0,2  0,8]




Então, de forma similar ao ***Perceptron*** e ***ADALINE***, pode-se converter tais sinais para que estes possam ser usados no treinamento do ***PMC***:

Conjunto de treinamento

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

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

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

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

**forma matricial**

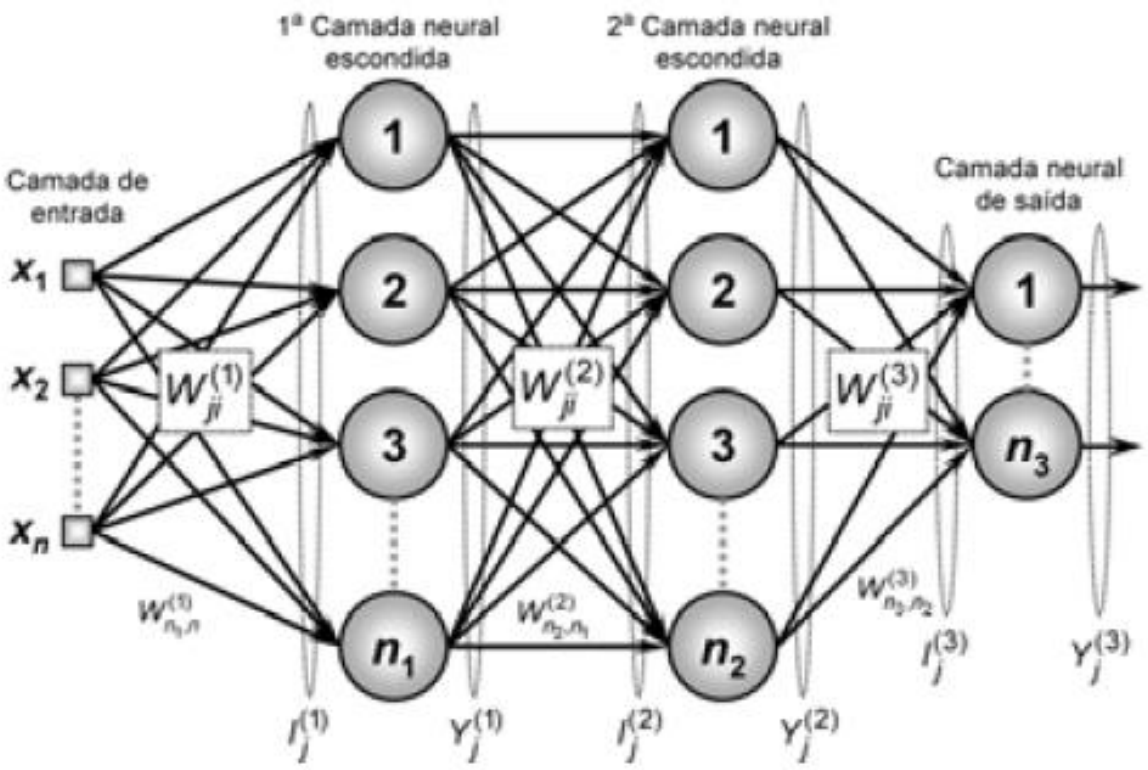
$$\Omega^{(x)} = \begin{array}{c} \\ x_0 \\ x_1 \\ x_2 \\ x_3 \end{array} \begin{array}{cccc} x^{(1)} & x^{(2)} & x^{(3)} & x^{(4)} \\ \end{array} \begin{bmatrix} -1 & -1 & -1 & -1 \\ 0{,}2 & 0{,}1 & 0{,}9 & 0{,}6 \\ 0{,}9 & 0{,}3 & 0{,}7 & 0{,}4 \\ 0{,}4 & 0{,}5 & 0{,}8 & 0{,}3 \end{bmatrix};$$

$$\Omega^{(d)} = \begin{array}{c} \\ d_1 \\ d_2 \end{array} \begin{array}{cccc} d^{(1)} & d^{(2)} & d^{(3)} & d^{(4)} \end{array} \begin{bmatrix} 0{,}7 & 0{,}6 & 0{,}9 & 0{,}2 \\ 0{,}3 & 0{,}4 & 0{,}5 & 0{,}8 \end{bmatrix}$$

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

# REDE *PERCEPTRON MULTICAMADAS*

- **Pseudocódigo para Fase de Treinamento**




**Início {Algoritmo PMC – Fase de Treinamento}**

<1> Obter o conjunto de amostras de treinamento $\{x^{(k)}\}$;

<2> Associar o vetor de saída desejada $\{d^{(k)}\}$ para cada amostra;

<3> Iniciar $W_{ji}^{(1)}$, $W_{ji}^{(2)}$ e $W_{ji}^{(3)}$ com valores aleatórios pequenos;

<4> Especificar taxa de aprendizagem $\{\eta\}$ e precisão requerida $\{\varepsilon\}$;

<5> Iniciar o contador de número de épocas $\{época \leftarrow 0\}$;

<6> Repetir as instruções:

- <6.1> $E_M^{anterior} \leftarrow E_M$;
- <6.2> Para todas as amostras de treinamento $\{x^{(k)}, d^{(k)}\}$, fazer:
  - <6.2.1> Obter $I_j^{(1)}$ e $Y_j^{(1)}$;
  - <6.2.2> Obter $I_j^{(2)}$ e $Y_j^{(2)}$;
  - <6.2.3> Obter $I_j^{(3)}$ e $Y_j^{(3)}$;

    **Passo *Forward***
  - <6.2.4> Determinar $\delta_j^{(3)}$;
  - <6.2.5> Ajustar $W_{ji}^{(3)}$;
  - <6.2.6> Determinar $\delta_j^{(2)}$;
  - <6.2.7> Ajustar $W_{ji}^{(2)}$;
  - <6.2.8> Determinar $\delta_j^{(1)}$;
  - <6.2.9> Ajustar $W_{ji}^{(1)}$;

    **Passo *Backward***
- <6.3> Obter $Y_j^{(3)}$ ajustado; {conforme <6.2.1>, <6.2.2> e <6.2.3>}
- <6.4> $E_M^{atual} \leftarrow E_M$;
- <6.5> $época \leftarrow época + 1$;

Até que: $|E_M^{atual} - E_M^{anterior}| \le \varepsilon$

**Fim {Algoritmo PMC – Fase de Treinamento}**

# REDE *PERCEPTRON MULTICAMADAS*

- **Pseudocódigo para Fase de Operação**

**Início {Algoritmo PMC – Fase de Operação}**

<1> Obter uma amostra { $\boldsymbol{x}$ };

<2> Assumir $W_{ji}^{(1)}$, $W_{ji}^{(2)}$ e $W_{ji}^{(3)}$ já ajustadas no treinamento;

<3> Execute as seguintes instruções:

$$\left.\begin{array}{l} \text{<3.1> Obter } I_j^{(1)} \text{ e } Y_j^{(1)}; \\ \text{<3.2> Obter } I_j^{(2)} \text{ e } Y_j^{(2)}; \\ \text{<3.3> Obter } I_j^{(3)} \text{ e } Y_j^{(3)}; \end{array}\right\} \textbf{Passo } \textbf{\textit{Forward}}$$

<4> Disponibilizar as saídas da rede, as quais são dadas pelos elementos contidos em $Y_j^{(3)}$

**Fim {Algoritmo PMC – Fase de Operação}**

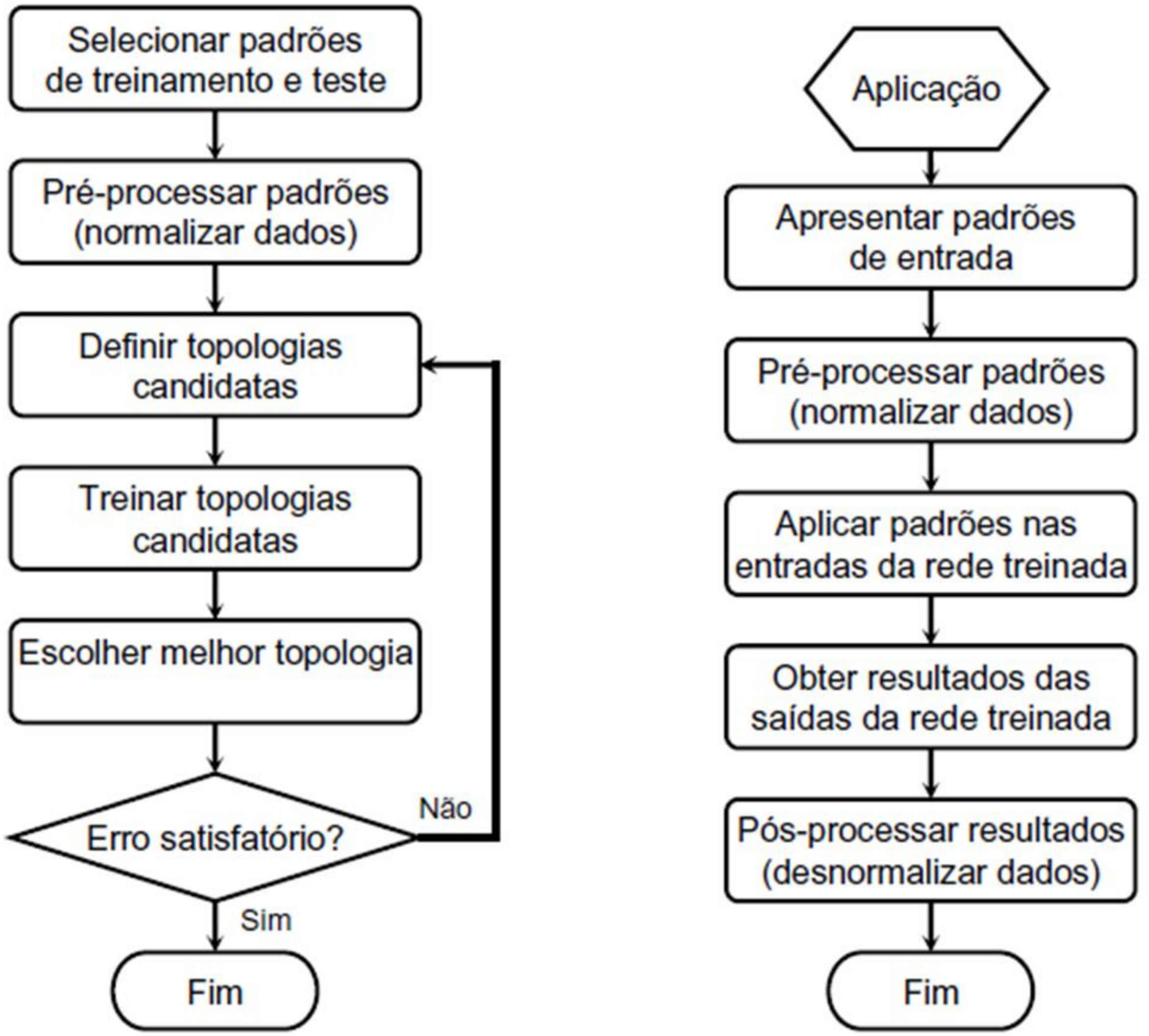
**Obs 1:** A "fase de operação" é usada somente após a "fase de treinamento", pois aqui a rede já está apta para ser usada no processo.

**Obs 2:** Lembrar de incluir o valor -1 dentro do vetor *x*.

$$\boldsymbol{x} = [-1 \quad x_1 \quad x_2 \ldots \; x_n]^T$$

# ASPECTOS DE PROJETO DE PMC

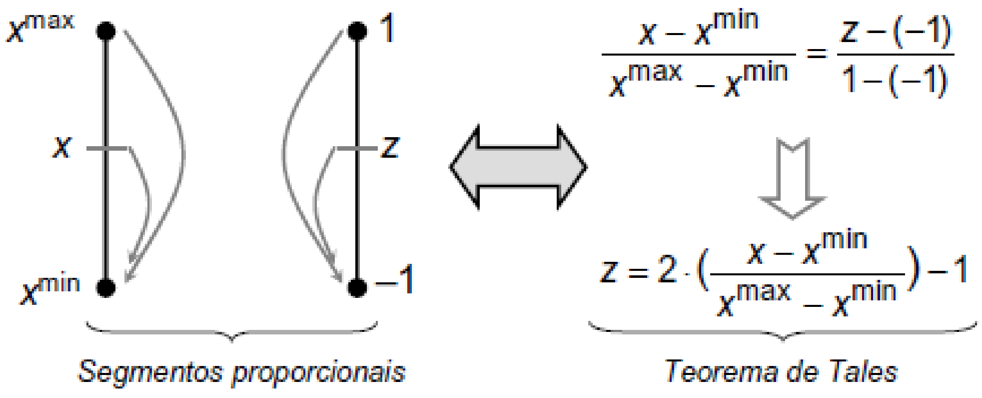
- **Principais etapas de projeto**



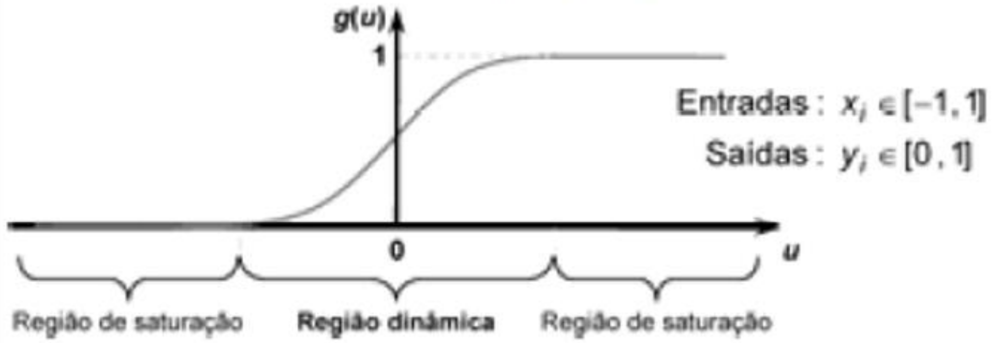

(a) Fase de Treinamento (b) Fase de Operação

# ASPECTOS DE PROJETO DE PMC

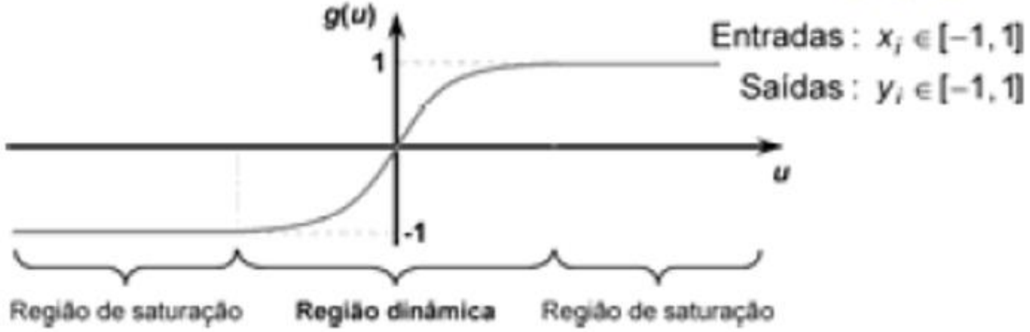
- **Pré-processamento de dados (normalização)**
  - Há necessidade de pré-processamento dos padrões de treinamento/teste visando aspectos de melhoria do desempenho de treinamento.
  - Isto implica geralmente em escalar as respectivas amostras p/ a faixa de variação dinâmica das funções de ativação dos neurônios, evitando-se assim a saturação de suas saídas.
  - Uma das técnicas de escalamento mais utilizada é aquela baseada no princípio dos segmentos proporcionais (Teorema de Tales) ilustrado na figura seguinte, isto é:
    - **Antes de Normalizar →** valores inicialmente compreendidos entre a faixa delimitada por $\boldsymbol{x}^{\min}$ e $\boldsymbol{x}^{\max}$, ou seja, $\boldsymbol{x} \in [\boldsymbol{x}^{\min}, \boldsymbol{x}^{\max}]$.
    - **Depois de Normalizar →** valores estarão convertidos para um domínio proporcional entre –1 e 1, o qual representa as faixas de variações dinâmicas das funções de ativação.




- Domínios de normalização (Logística)




- Domínios de normalização (Tangente Hiperbólica)

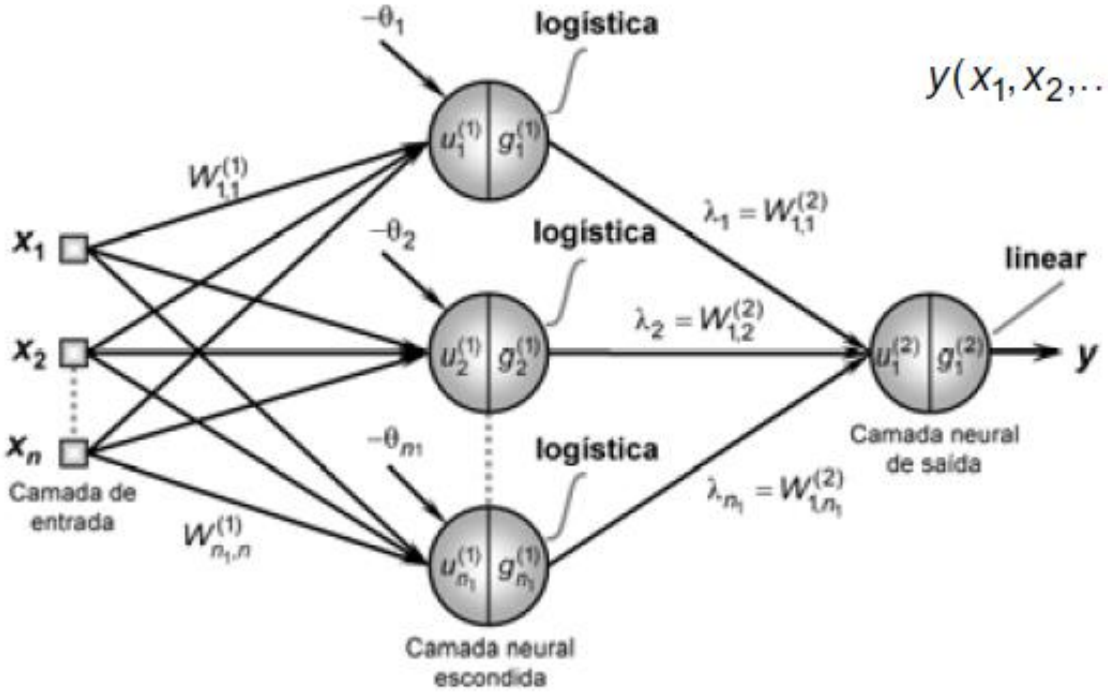


# APLICABILIDADE DO PMC

- **Caracterização de problemas de aproximação funcional**
  - É a classe de problemas em que as redes ***PMC*** podem usufruir de maior destaque.
  - Consiste de mapear o comportamento de um processo se baseando somente em diversas medições efetivadas em suas entradas e saídas (sem conhecer a modelagem matemática).
  - Observa-se aqui uma das principais características intrínsecas das redes neurais artificiais, ou seja, o aprendizado a partir de exemplos.
  - No caso de aproximação de funções, traduz-se na disponibilização de um conjunto de entradas/saídas que reproduzem o comportamento do sistema a ser tratado.
  - De fato, há muitas aplicações em que as únicas informações disponíveis se resumem a uma coleção de dados de entradas/saídas.
  - Nesta direção, constata-se que as ***RNA*** têm sido extensivamente aplicados nas seguintes situações:
    - O processo a ser modelado é de certa forma complexo.
    - Naqueles casos em que as utilizações de métodos convencionais produzem resultados insatisfatórios.
    - Naqueles casos em que os sistemas convencionais exigem requisitos computacionais bem sofisticados.

# APLICABILIDADE DO PMC

- **Aspectos do teorema da aproximação universal (Regressão)**
  - Baseado nas demonstrações de Kolmogorov, estas fornecem as bases para se definir as configurações de redes ***PMC*** para finalidade de mapear funções algébricas.
  - Assumindo que *g*(.) a ser adotada nas redes ***PMC*** sejam contínuas e limitadas em suas imagens, tais como a logística e tangente hiperbólica, demonstra-se então que:
    - Um ***PMC***, **composto de apenas uma camada escondida**, é capaz de mapear qualquer função contínua no espaço real. Em termos matemáticos, tem-se:




$$y(x_1,x_2,\ldots,x_n)=\sum_{i=1}^{n_1}\underbrace{\lambda_i}_{\text{parcela (i)}}\cdot\underbrace{g_i^{(1)}(u_i^{(1)})}_{\text{parcela (ii)}}$$

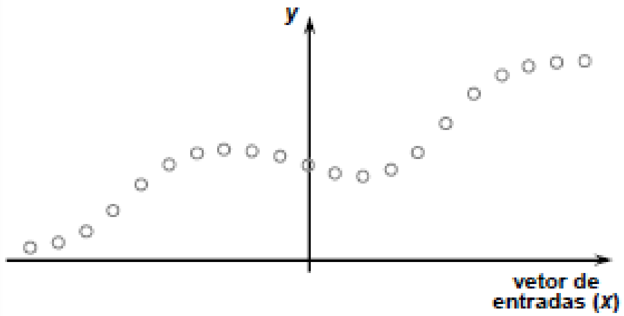
$$u_i^{(1)}=\sum_{j=1}^{n}W_{ji}^{(1)}\cdot x_j-\theta_i$$

- O neurônio de saída (ativação linear) realiza tão somente a combinação linear das funções de ativação logística implementadas pelos neurônios da camada intermediária.
- A função *y* a ser mapeada será constituída por superposição de logísticas {parcela (ii)}, representadas pelos termos $g_i^{(1)}(u_i^{(1)})$ , que são ponderadas por fatores $\lambda_i$ {parcela (i)}.

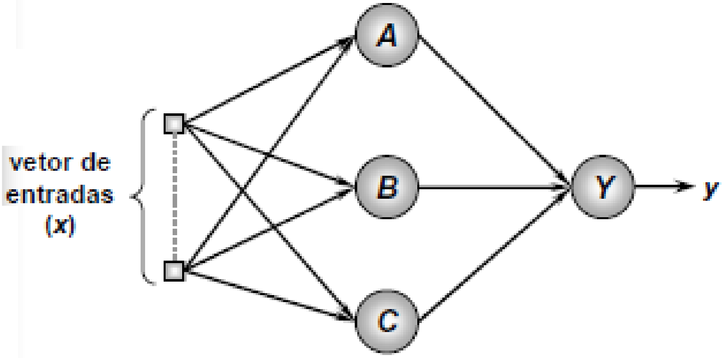
# APLICABILIDADE DO PMC

- **Teorema da aproximação universal (Ilustração)**

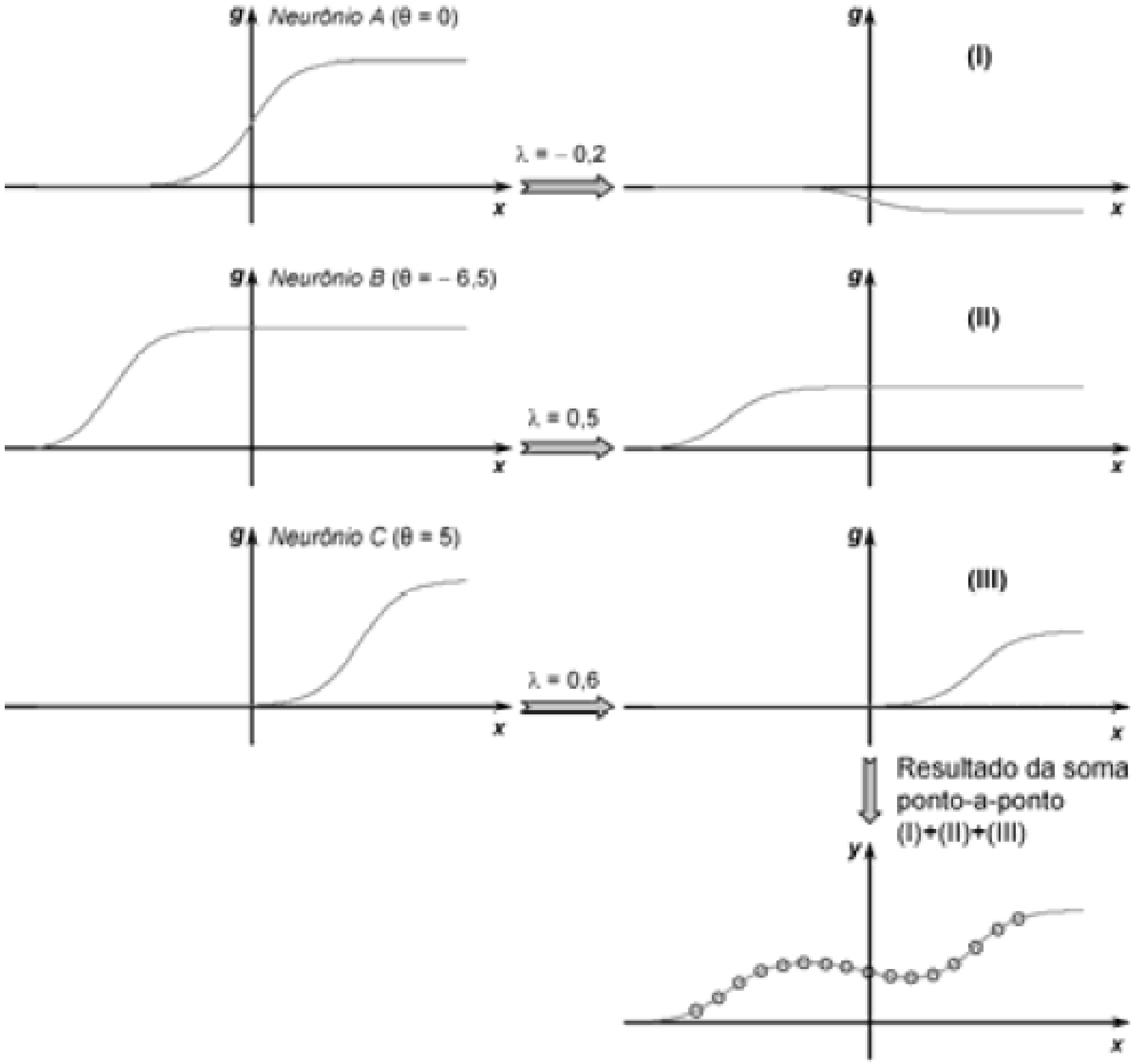
  - Conjunto de amostras relacionando entradas/saídas referente ao processo (função) a ser mapeado.




  - *PMC* aplicado para mapear a função representada pelas amostras de treinamento acima:




- Configuração de *PMC* após o ajuste de seus pesos:
  - Parâmetro θ → responsável pela translação das funções de ativação.
  - Parâmetro λ → responsável pelo escalamento das funções de ativação.



# AINDA SOBRE O PMC

- **Aspectos Práticos**
  - Embora um ***PMC*** com apenas uma camada escondida seja suficiente para mapear qualquer função não-linear contínua definida num domínio compacto (fechado), há situações em que se utilizam mais de duas camadas delas.
  - A adoção de mais camadas escondidas podem ser apropriadas tanto para o propósito de incrementar o desempenho do treinamento como de reduzir a topologia estrutural da rede.

# REFERÊNCIA


- SILVA, Ivan Nunes da e SPATTI, Danilo Hernane e FLAUZINO, Rogério Andrade. Redes neurais artificiais para engenharia e ciências aplicadas. . São Paulo: Artliber Editora. . Acesso em: 23 dez. 2023. , 2010