

Aula 5

RNA – Perceptron Multicamadas

Sumário

1- Introdução

2- Backpropagation

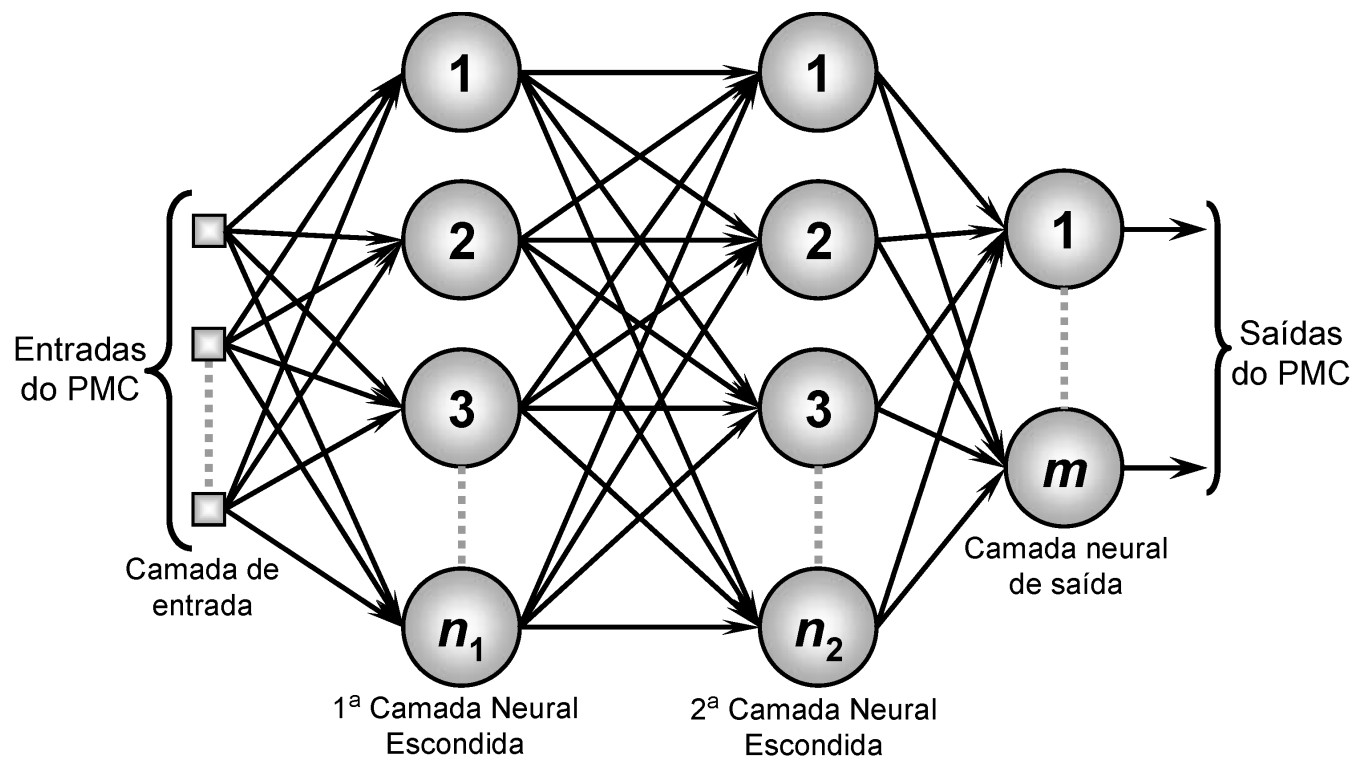
3- Algoritmo

4- Versões Aperfeiçoadas do Backpropagation

5- Aplicações da MLP

1- Introdução

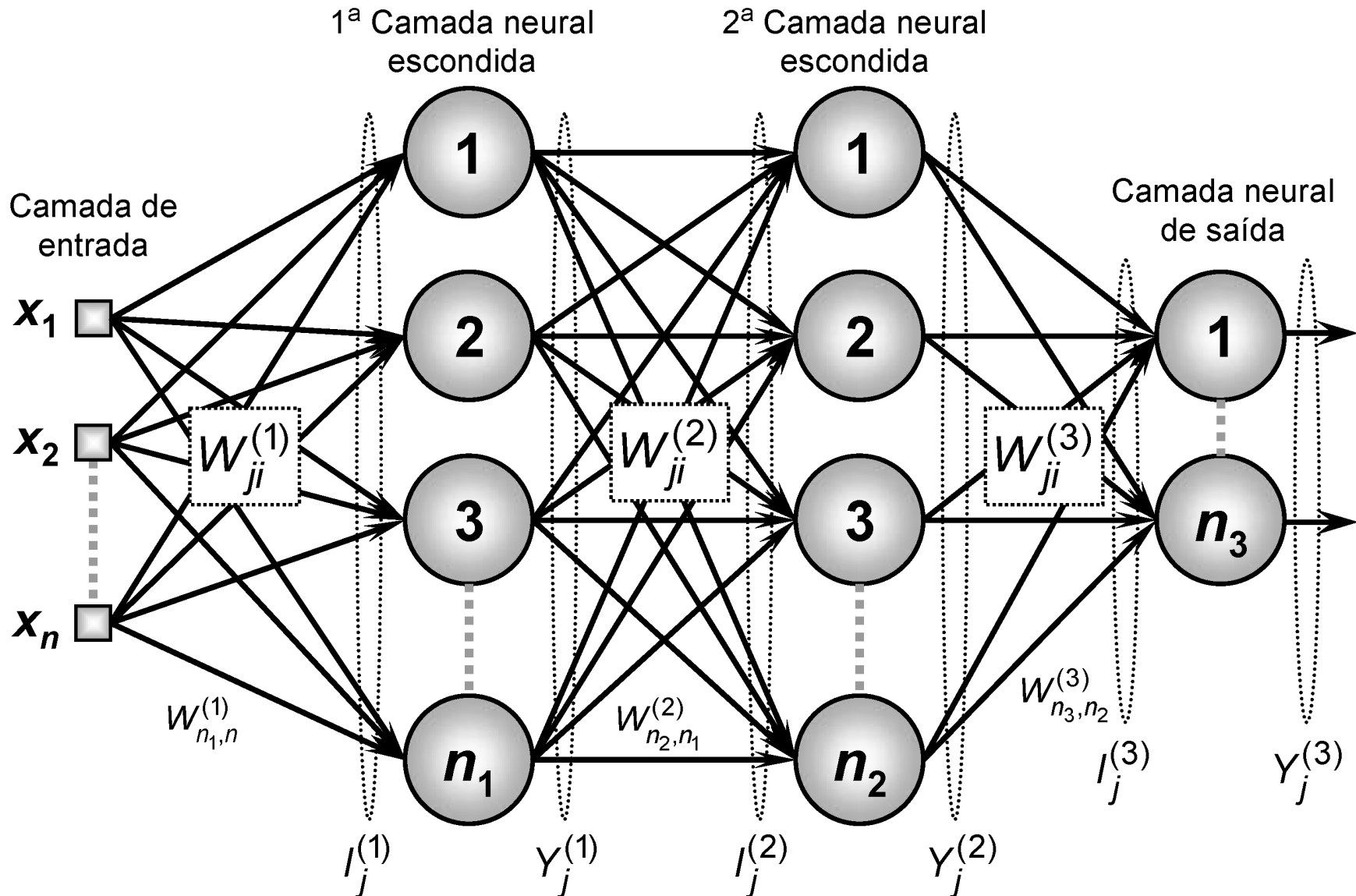
- Perceptron de múltiplas camadas (PMC) ou *Multilayer Perceptron (MLP)*;
- Apresentam no mínimo duas camadas de neurônios;
- Supervisionada e Feedforward;
- Publicada em 1986 junto ao algoritmo de aprendizado ***backpropagation***.



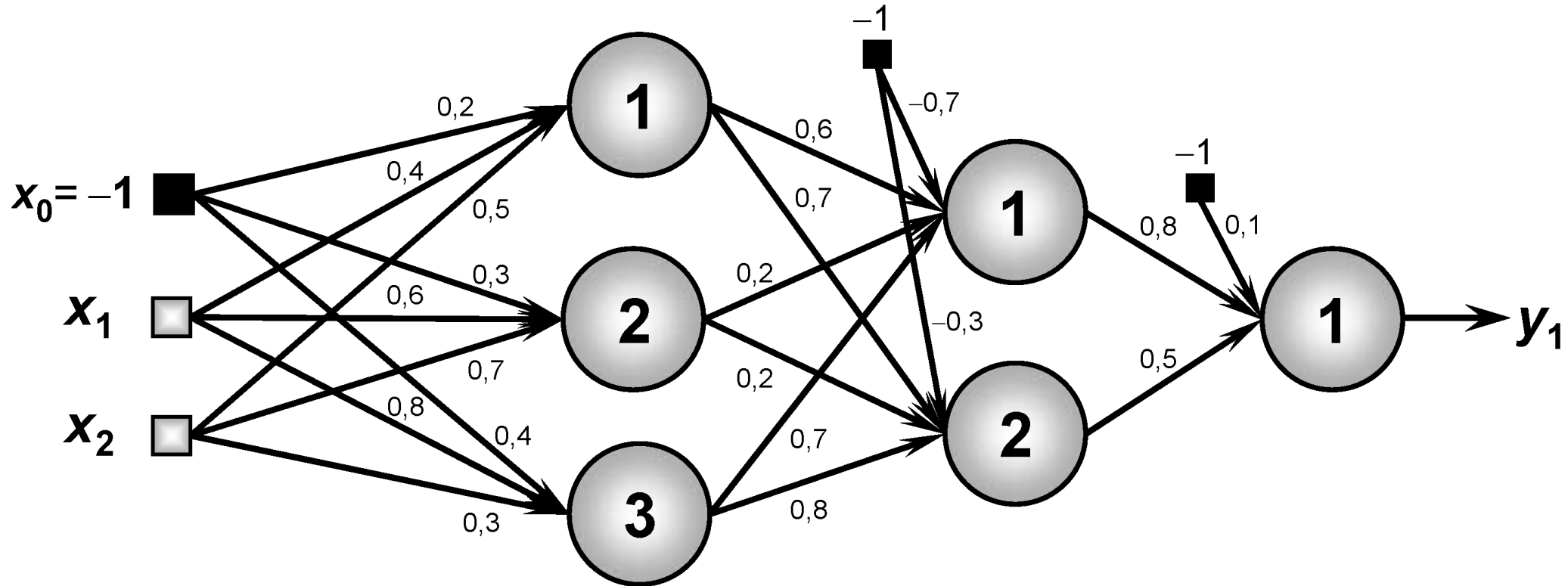
1- Introdução

- Diferentemente do Perceptron e Adaline, onde existe apenas um neurônio de saída $\{y\}$, a MLP pode relacionar o conhecimento a vários neurônios de saída.
- O algoritmo de aprendizado da MLP é chamado **backpropagation** ou **regra Delta generalizada** e é dividida em duas fases:
 - 1) Propagação adiante (forward): onde $\{x_1, x_2, \dots, x_n\}$ são propagados até a saída, mantendo os pesos e limiares inalterados. Na saída para cada neurônio são obtidos os respectivos desvios (erros) entre as saídas desejadas e obtidas.
 - 2) Propagação reversa (backward): os ajustes são realizados a cada iteração, por todos os neurônios, produzindo o ajuste gradativo frente a saída desejada.
- Cada neurônio $\{j\}$ pertence a uma camada $\{L\}$, onde $g(\cdot)$ representa a função de ativação de função logística ou hiperbólica.

2- Backpropagation



2- Backpropagation



$$W_{ji}^{(1)} = \begin{bmatrix} 0.2 & 0.4 & 0.5 \\ 0.3 & 0.6 & 0.7 \\ 0.4 & 0.8 & 0.3 \end{bmatrix}$$

$$W_{ji}^{(2)} = \begin{bmatrix} -0.7 & 0.6 & 0.2 & 0.7 \\ -0.3 & 0.7 & 0.2 & 0.8 \end{bmatrix}$$

$$W_{ji}^{(3)} = [0.1 \quad 0.8 \quad 0.5]$$

2- Backpropagation

Exemplo, $x_1 = 0,3$ e $x_2 = 0,7$:

$$l_j^{(1)} = \begin{bmatrix} l_1^{(1)} \\ l_2^{(1)} \\ l_3^{(1)} \end{bmatrix} = \begin{bmatrix} W_{1,0}^{(1)} \cdot x_0 + W_{1,1}^{(1)} \cdot x_1 + W_{1,2}^{(1)} \cdot x_2 \\ W_{2,0}^{(1)} \cdot x_0 + W_{2,1}^{(1)} \cdot x_1 + W_{2,2}^{(1)} \cdot x_2 \\ W_{3,0}^{(1)} \cdot x_0 + W_{3,1}^{(1)} \cdot x_1 + W_{3,2}^{(1)} \cdot x_2 \end{bmatrix} = \begin{bmatrix} (0,2 \cdot (-1)) + (0,4 \cdot 0,3) + (0,5 \cdot 0,7) \\ (0,3 \cdot (-1)) + (0,6 \cdot 0,3) + (0,7 \cdot 0,7) \\ (0,4 \cdot (-1)) + (0,8 \cdot 0,3) + (0,3 \cdot 0,7) \end{bmatrix} = \begin{bmatrix} 0,27 \\ 0,37 \\ 0,05 \end{bmatrix}$$

$$Y_j^{(1)} = \begin{bmatrix} Y_1^{(1)} \\ Y_2^{(1)} \\ Y_3^{(1)} \end{bmatrix} = \begin{bmatrix} g(l_1^{(1)}) \\ g(l_2^{(1)}) \\ g(l_3^{(1)}) \end{bmatrix} = \begin{bmatrix} \tanh(0,27) \\ \tanh(0,37) \\ \tanh(0,05) \end{bmatrix} = \begin{bmatrix} 0,26 \\ 0,35 \\ 0,05 \end{bmatrix} \xrightarrow{Y_0^{(1)} = -1} Y_j^{(1)} = \begin{bmatrix} Y_0^{(1)} \\ Y_1^{(1)} \\ Y_2^{(1)} \\ Y_3^{(1)} \end{bmatrix} = \begin{bmatrix} -1 \\ 0,26 \\ 0,35 \\ 0,05 \end{bmatrix}$$

tanh é a tangente hiperbólica e os argumentos estão em radianos.

2- Backpropagation

Exemplo, $x_1 = 0,3$ e $x_2 = 0,7$:

$$l_j^{(2)} = \begin{bmatrix} l_1^{(2)} \\ l_2^{(2)} \end{bmatrix} = \begin{bmatrix} W_{1,0}^{(2)} \cdot x_0 + W_{1,1}^{(2)} \cdot x_1 + W_{1,2}^{(2)} \cdot x_2 + W_{1,3}^{(2)} \cdot x_3 \\ W_{2,0}^{(2)} \cdot x_0 + W_{2,1}^{(2)} \cdot x_1 + W_{2,2}^{(2)} \cdot x_2 + W_{2,3}^{(2)} \cdot x_3 \end{bmatrix} = \begin{bmatrix} 0,96 \\ 0,59 \end{bmatrix}$$

$$Y_j^{(2)} = \begin{bmatrix} Y_1^{(2)} \\ Y_2^{(2)} \end{bmatrix} = \begin{bmatrix} g(l_1^{(2)}) \\ g(l_2^{(2)}) \end{bmatrix} = \begin{bmatrix} \tanh(0,96) \\ \tanh(0,59) \end{bmatrix} = \begin{bmatrix} 0,74 \\ 0,53 \end{bmatrix} \xrightarrow{Y_0^{(2)} = -1} Y_j^{(2)} = \begin{bmatrix} Y_0^{(2)} \\ Y_1^{(2)} \\ Y_2^{(2)} \end{bmatrix} = \begin{bmatrix} -1 \\ 0,74 \\ 0,53 \end{bmatrix}$$

tanh é a tangente hiperbólica e os argumentos estão em radianos.

2- Backpropagation

Exemplo, $x_1 = 0,3$ e $x_2 = 0,7$:

$$l_j^{(3)} = [l_1^{(3)}] = [W_{1,0}^{(3)} \cdot x_0 + W_{1,1}^{(3)} \cdot x_1 + W_{1,2}^{(3)} \cdot x_2] = [0,76]$$

$$Y_j^{(3)} = [0,76] = [g(l_1^{(3)})] = [\tanh(0,76)] = [0,64]$$

Encontrado o y_1 .

Próximo passo é medir o desvio para iniciar a derivação do **backpropagation**.

$$\delta_j^{(3)} = (d_j - Y_j^{(3)}) \cdot g'(l_j^{(3)})$$

$$W_{ji}^{(3)}(t+1) = W_{ji}^{(3)}(t) + \eta \cdot \delta_j^{(3)} Y_i^{(2)}$$

2- Backpropagation

Camada Intermediária

$$\delta_j^{(2)} = \left(\sum_{k=1}^{n_3} \delta_k^{(3)} \cdot W_{kj}^{(3)} \right) \cdot g'(l_j^{(2)})$$

$$W_{ji}^{(2)}(t+1) = W_{ji}^{(2)}(t) + \eta \cdot \delta_j^{(2)} Y_i^{(1)}$$

Camada Inicial

$$\delta_j^{(1)} = \left(\sum_{k=1}^{n_2} \delta_k^{(2)} \cdot W_{kj}^{(2)} \right) \cdot g'(l_j^{(1)})$$

$$W_{ji}^{(1)}(t+1) = W_{ji}^{(1)}(t) + \eta \cdot \delta_j^{(1)} x_i$$

3- Algoritmo

Fase de Operação e Treinamento:

Início {Algoritmo PMC – Fase de Operação}

- <1> Obter uma amostra $\{ \mathbf{x} \}$;
- <2> Assumir $W_{ji}^{(1)}$, $W_{ji}^{(2)}$ e $W_{ji}^{(3)}$ já ajustadas no treinamento;
- <3> Execute as seguintes instruções:
 - <3.1> Obter $I_j^{(1)}$ e $Y_j^{(1)}$; {conforme (5.1) e (5.4)}
 - <3.2> Obter $I_j^{(2)}$ e $Y_j^{(2)}$; {conforme (5.2) e (5.5)}
 - <3.3> Obter $I_j^{(3)}$ e $Y_j^{(3)}$; {conforme (5.3) e (5.6)}
- <4> Disponibilizar as saídas da rede, as quais são dadas pelos elementos contidos em $Y_j^{(3)}$

Passo
Forward

Fim {Algoritmo PMC – Fase de Operação}

Início {Algoritmo PMC – Fase de Treinamento}

- <1> Obter o conjunto de amostras de treinamento $\{ \mathbf{x}^{(k)} \}$;
 - <2> Associar o vetor de saída desejada $\{ \mathbf{d}^{(k)} \}$ 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 $\{\epsilon\}$;
 - <5> Iniciar o contador de número de épocas $\{época \leftarrow 0\}$;
 - <6> Repetir as instruções:
 - <6.1> $E_M^{anterior} \leftarrow E_M$; {conforme (5.8)}
 - <6.2> Para todas as amostras de treinamento $\{ \mathbf{x}^{(k)}, \mathbf{d}^{(k)} \}$, fazer:
 - <6.2.1> Obter $I_j^{(1)}$ e $Y_j^{(1)}$; {conforme (5.1) e (5.4)}
 - <6.2.2> Obter $I_j^{(2)}$ e $Y_j^{(2)}$; {conforme (5.2) e (5.5)}
 - <6.2.3> Obter $I_j^{(3)}$ e $Y_j^{(3)}$; {conforme (5.3) e (5.6)}
 - <6.2.4> Determinar $\delta_j^{(3)}$; {conforme (5.15)}
 - <6.2.5> Ajustar $W_{ji}^{(3)}$; {conforme (5.17)}
 - <6.2.6> Determinar $\delta_j^{(2)}$; {conforme (5.26)}
 - <6.2.7> Ajustar $W_{ji}^{(2)}$; {conforme (5.28)}
 - <6.2.8> Determinar $\delta_j^{(1)}$; {conforme (5.37)}
 - <6.2.9> Ajustar $W_{ji}^{(1)}$; {conforme (5.39)}
 - <6.3> Obter $Y_j^{(3)}$ ajustado; {conforme <6.2.1>, <6.2.2> e <6.2.3>}
 - <6.4> $E_M^{atual} \leftarrow E_M$; {conforme (5.8)}
 - <6.5> $época \leftarrow época + 1$;
- Até que: $|E_M^{atual} - E_M^{anterior}| \leq \epsilon$

Passo
Forward

Passo
Backward

Fim {Algoritmo PMC – Fase de Treinamento}

4- Versões Aperfeiçoadas do Backpropagation

- Método de Inserção do termo de *momentum*
- Método Resilient-Propagation
- Método de Levenberg-Marquardt

5- Aplicações da MLP

Referências:

Silva, IN da, Danilo Hernane Spatti, and Rogério Andrade Flauzino. "Redes neurais artificiais para engenharia e ciências aplicadas." São Paulo: Artliber (2010).