Universidade Presbiteriana Mackenzie

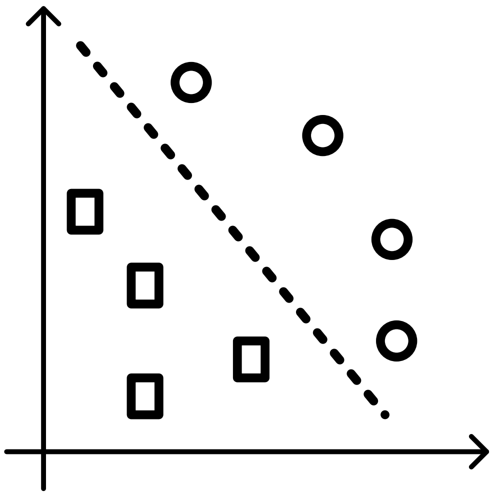
Perceptron de Múltiplas Camadas

Prof. Dr. Leandro Augusto da Silva
leandroaugusto.silva@mackenzie.br
Laboratório de Big Data e Métodos Analíticos Aplicados
Faculdade de Computação e Informática
Programa de Pós-Graduação em Engenharia Elétrica e Computação

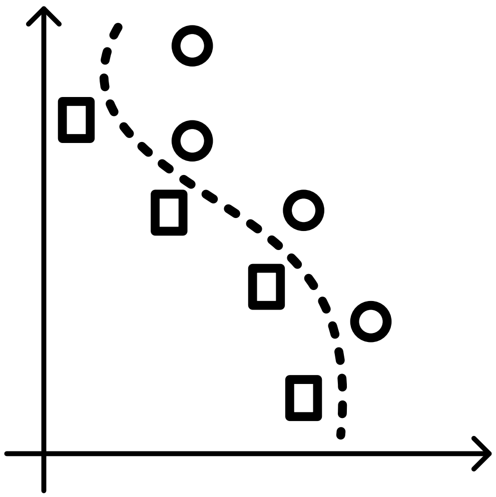
Sumário

- Problema do XOR
- Perceptron em Múltiplas Camadas
  - Arquitetura
  - Aprendizado
  - Exemplos de uso

Limitação do Perceptron

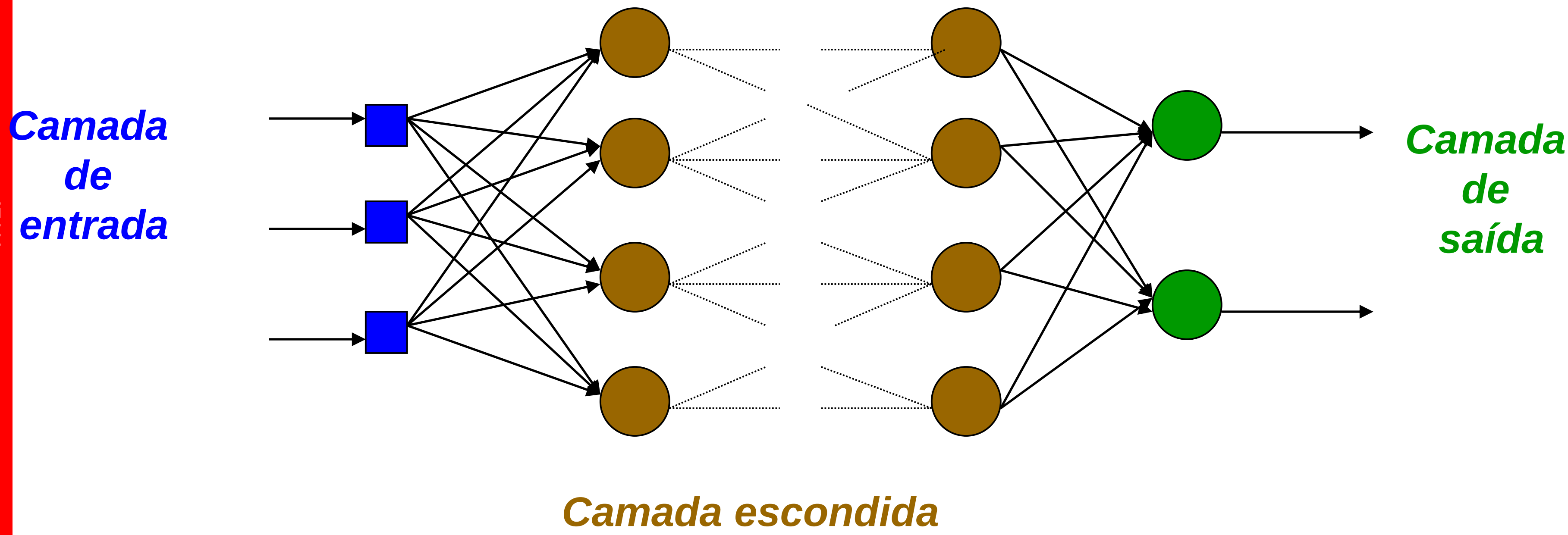


Linearmente separáveis



Não linearmente separáveis

Arquitetura Perceptrons Multicamadas

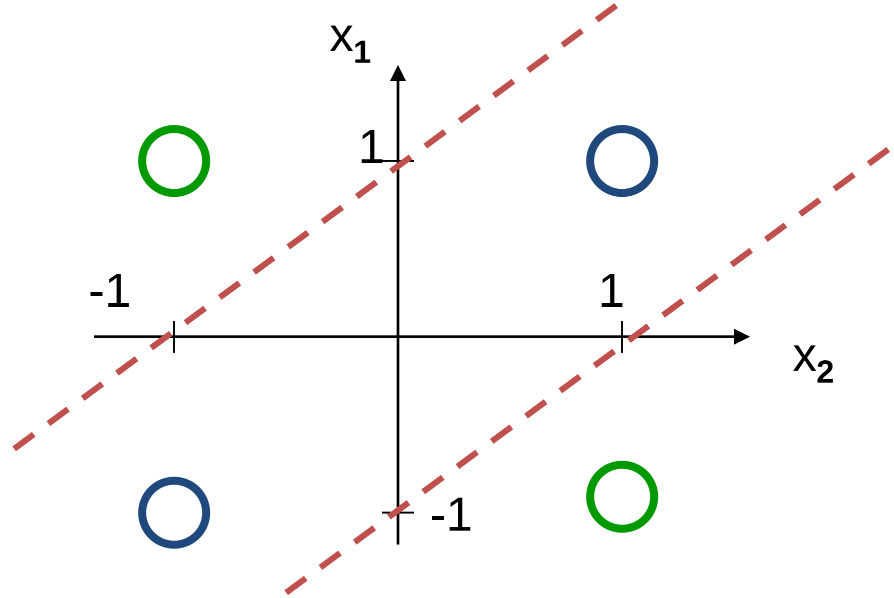


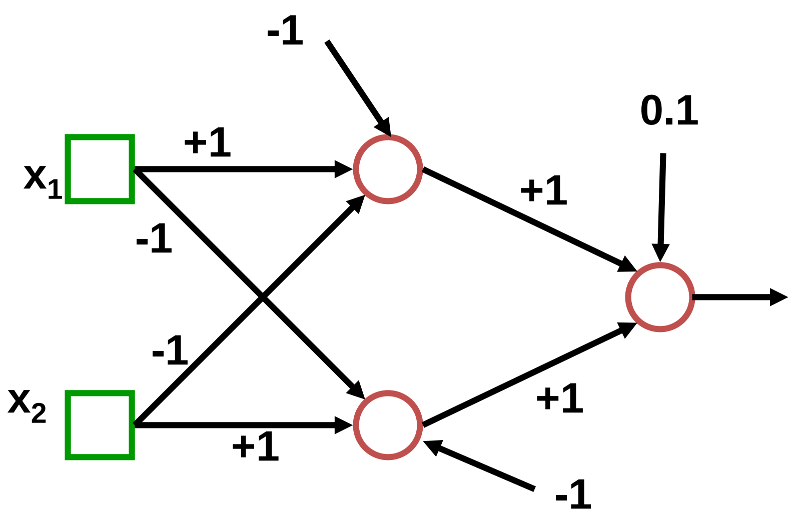
O Multi Layer Perceptron (MLP)

- Múltiplas entradas / Múltiplas saídas
- Ambas podem ser analógicas ou digitais
- Não há mais a restrição de separabilidade linear
- Aplicações das Redes Multicamadas ou MLP como são conhecidas usando o algoritmo de aprendizado *error back-propagation*

Uma solução para problema do XOR

x_1	x_2	x_1 xor x_2
-1	-1	**-1**
-1	1	**1**
1	-1	**1**
1	1	**-1**



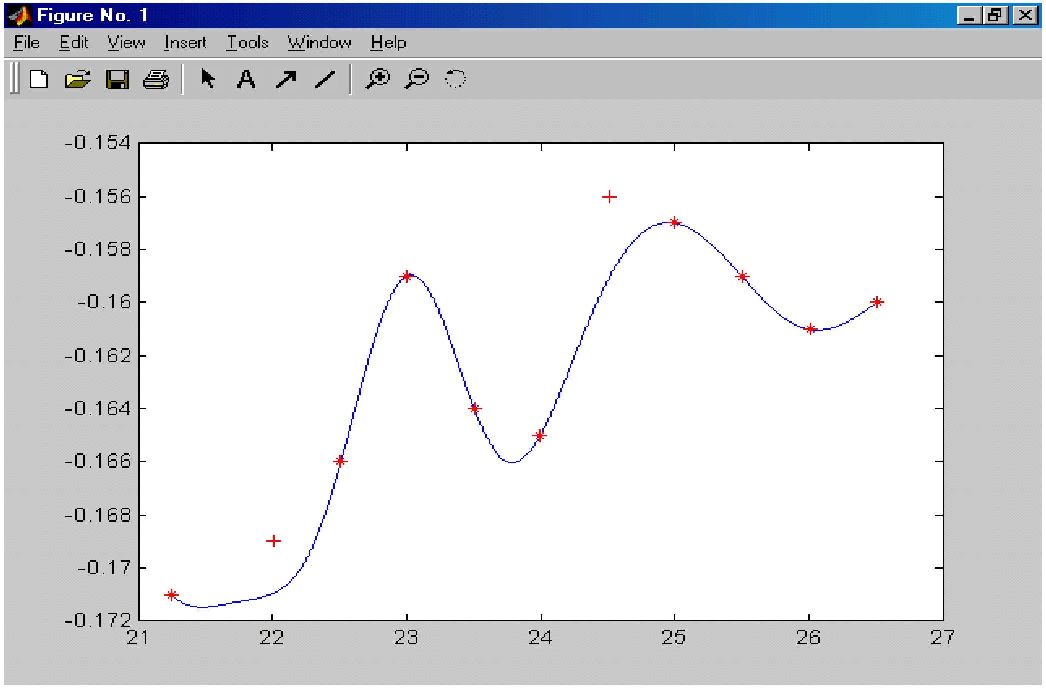


$$\varphi(v) = \begin{cases} 1 & \text{if } v > 0 \\ -1 & \text{if } v \leq 0 \end{cases}$$

φ is the sign function.

MLP

Aproximação de funções com o MLP / superposição de sigmoides deslocadas



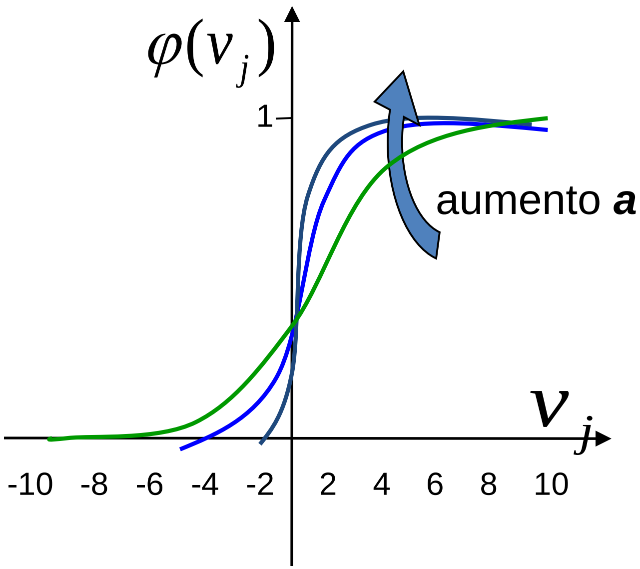
Outras aplicações importantes do MLP

Além de aproximação de funções genéricas …

- Fusão não linear de grandezas analógicas multidimensionais
- Previsão de séries temporais não lineares
- Classificação de Padrões multidimensionais sem separabilidade linear
- Note que o aprendizado por exemplos do MLP permite que ele realize as funções acima sem a necessidade de um modelo matemático conhecido / confiável

Modelando o neurônio

- Função Sigmoidal



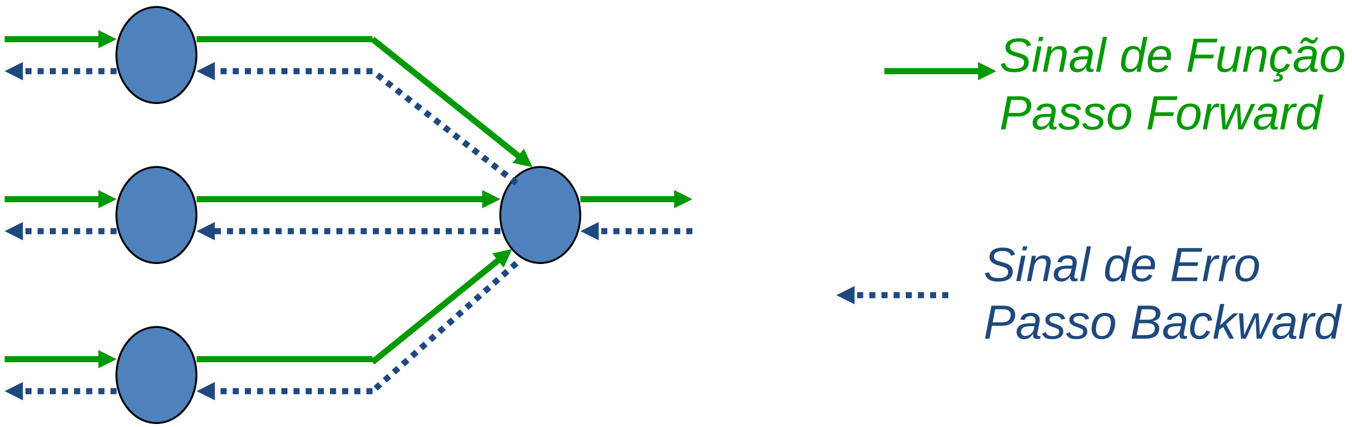
$$\boxed{\varphi(\mathrm{v}_j) = \frac{1}{1+e^{-av_j}}}$$

$$\mathrm{v}_j = \sum_{i=0,\ldots,m} w_{ji} y_i$$

- v_j campo induzido do neurônio j
- Uma das funções de ativação mais comum
- a $\rightarrow \infty \Rightarrow \varphi \rightarrow$ função threshold
- Diferenciavel

Algoritmo de Aprendizado

- Algoritmo Back-propagation



- Os pesos da rede são ajustados com o objetivo de minimizar o erro médio quadrático.

Erro Médio Quadrático

- O sinal de erro da saída do neurônio *j* na apresentação dos *n-th* exemplos de treinamento:
- Erro na saída do neurônio:

$$e_j(n) = d_j(n) - y_j(n)$$

- Energia total do erro:

$$E(n) = \frac{1}{2}\sum_{j \in C} e_j^2(n)$$

C: conjunto de neurônios na camada de saída

- Energia média do erro quadrático:

$$E_{AV} = \frac{1}{N}\sum_{n=1}^{N} E(n)$$

N: tamanho do conjunto de treinamento

- **Objetivo:** ***Ajustar os pesos da rede para diminuir*** E_{AV}

Notação

e_j Erro na saída do neurônio j

y_j Saída do neurônio j

$$v_j = \sum_{i=0,\dots,m} w_{ji} y_i$$

Campo local induzido do neurônio j

Regra de atualização dos pesos

Regra de atualização é baseada no método do gradiente descendente que caminha na direção da minimização de E

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}}$$

Passo na direção oposta ao gradiente

Com w_{ji} peso associado entre o neurônio i e o neurônio j

Definição do Gradiente Local do neurônio j

MLP

$$\boxed{\delta_j = -\frac{\partial E}{\partial v_j}}$$

Gradient Local

Obtemos

$$\boxed{\delta_j = e_j \varphi'(v_j)}$$

porque

$$-\frac{\partial E}{\partial v_j} = -\frac{\partial E}{\partial e_j}\frac{\partial e_j}{\partial y_j}\frac{\partial y_j}{\partial v_j} = -e_j(-1)\varphi'(v_j)$$

Regra de atualização

- Obtemos

$$\boxed{\Delta w_{ji} = \eta \delta_j y_i}$$

porque

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial v_j} \frac{\partial v_j}{\partial w_{ji}}$$

$$-\frac{\partial E}{\partial v_j} = \delta_j \qquad \frac{\partial v_j}{\partial w_{ji}} = y_i$$

Cálculo do gradiente local do neurônio j

- O fator chave é o cálculo de e_j
- Existem dois casos:
  - Caso 1): j é um neurônio de saída
  - Caso 2): j é um neurônio escondido

Erro e_j do neurônio de saída

- Caso 1: ***j neurônio de saída***

$$e_j = d_j - y_j$$

Então

$$\boxed{\delta_j = (d_j - y_j)\varphi'(v_j)}$$

Gradiente Local do Neurônio Escondido

- Caso 2: ***j neurônio escondido***

  o gradiente local para o neurônio j é recursivamente determinado em termos do gradiente local de todos os neurônios no qual o neurônio j está diretamente conectado

Gradiente Local do Neurônio Escondido

$$\delta_j = -\frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial v_j} \qquad \frac{\partial y_j}{\partial v_j} = \varphi'(v_j)$$

$$\boxed{E(n) = \frac{1}{2}\sum_{k \in C} e_k^2(n)}$$

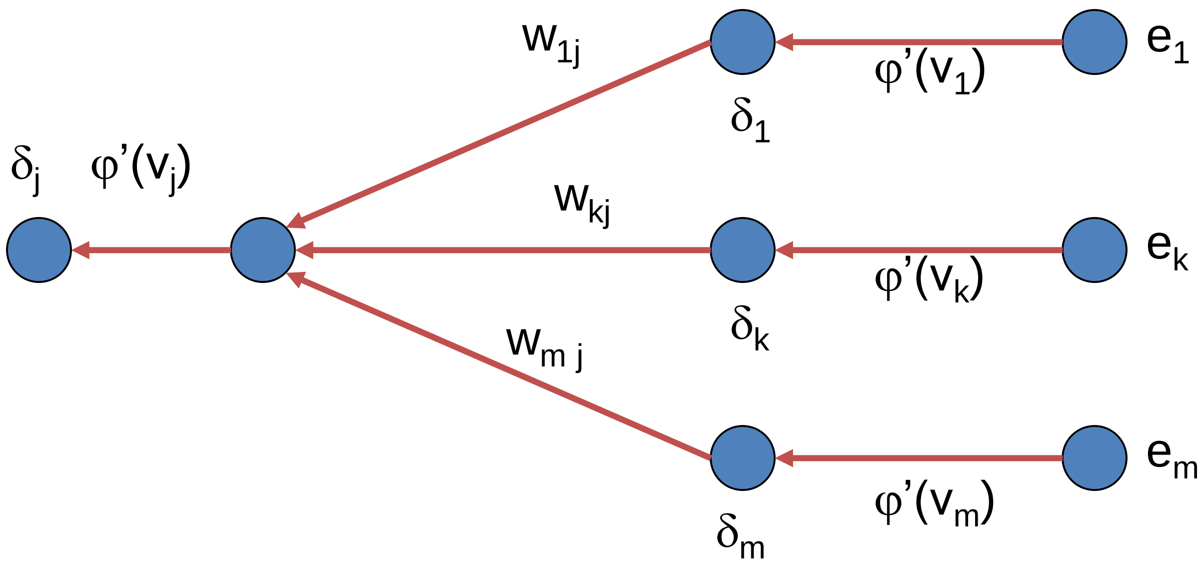
$$-\frac{\partial E}{\partial y_j} = -\sum_{k \in C} e_k \frac{\partial e_k}{\partial y_j} = \sum_{k \in C} e_k \left[\frac{-\partial e_k}{\partial v_k}\right]\frac{\partial v_k}{\partial y_j}$$

de $$-\frac{\partial e_k}{\partial v_k} = \varphi'(v_k) \qquad \frac{\partial v_k}{\partial y_j} = w_{kj}$$

Obtemos $$\boxed{-\frac{\partial E}{\partial y_j} = \sum_{k \in C} \delta_k w_{kj}}$$

Gradiente Local do Neurônio Escondido

$$\delta_j = \varphi'(v_j)\sum_{k\in C}\delta_k w_{kj}$$



Grafo orientado do sinal do erro retropropagado neurônio ***j***

Regra Delta

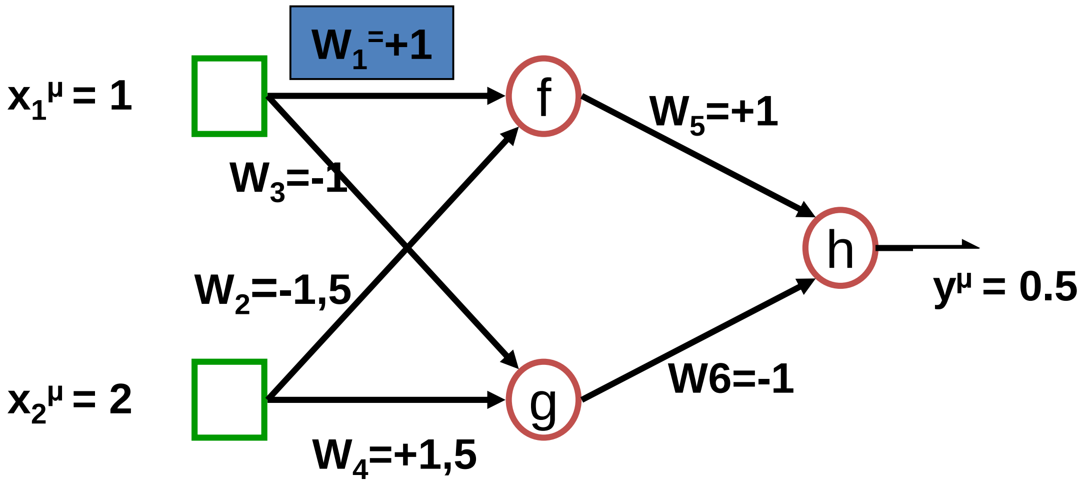
- **Regra Delta** $\Delta w_{ji} = \eta \delta_j y_i$

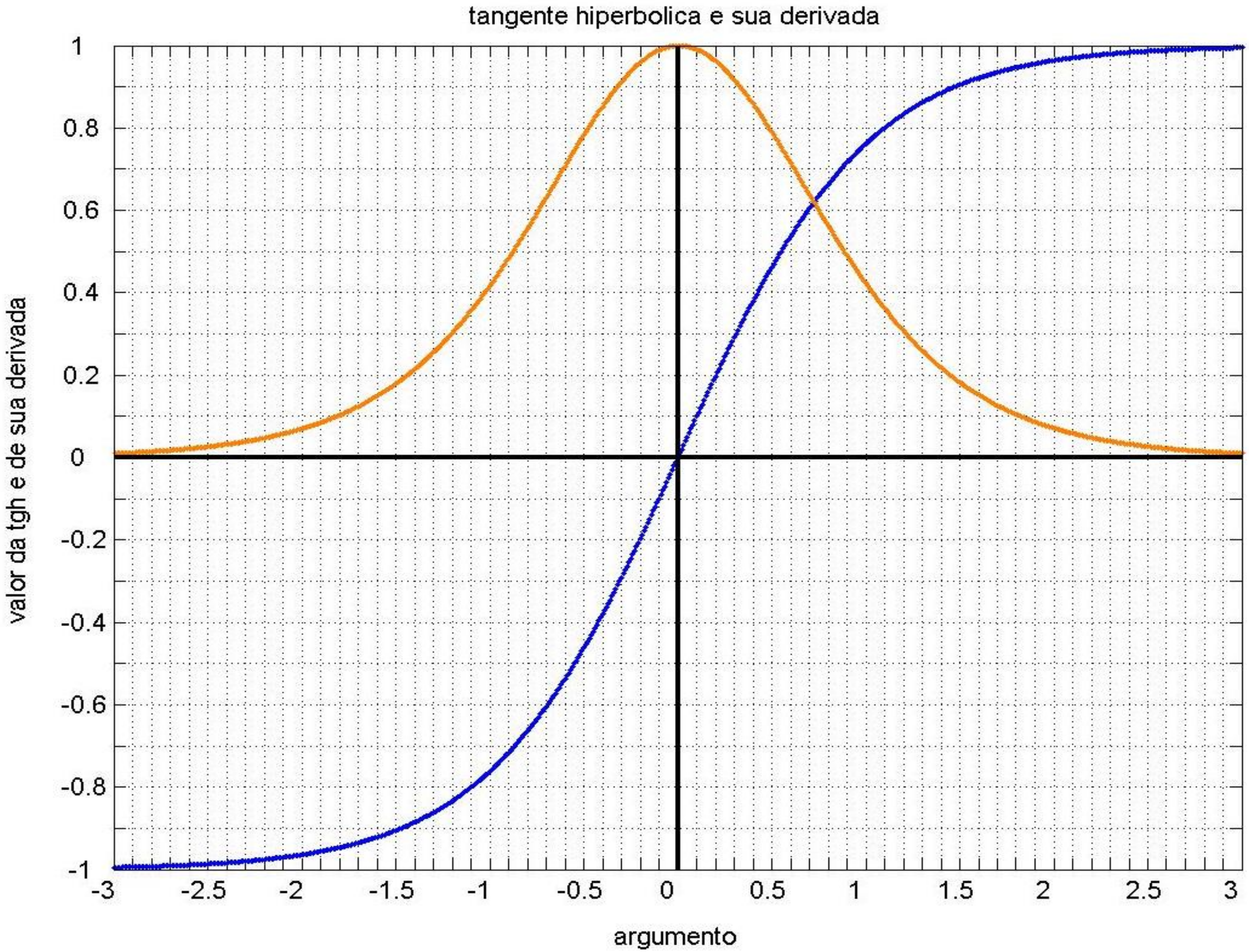
$$\delta_j = \begin{cases} \varphi'(v_j)(d_j - y_j) & \text{SE j nó de saída} \\ \varphi'(v_j)\sum_{k \in C} \delta_k w_{kj} & \text{SE j nó escondido} \end{cases}$$

C: Conjunto de neurônios na camada que contém um ***j***

Exemplo Aprendizado

Na figura a seguir, temos uma arquitetura neural do tipo MLP. Os valores dos pesos iniciais e do padrão de treinamento está representado na figura. Calcule a adaptação **Δw1** no peso superior do nó escondido superior, considerando apenas o primeiro passo de adaptação / aprendizado.Considere a taxa de aprendizado η = 0.01.




tangente hiperbolica e sua derivada
valor da tgh e de sua derivada
argumento
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-3
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
3

MLP

BigMAAp

Laborátorio de BIG DATA e Métodos Analíticos Aplicados

Prof. Dr. Leandro Augusto da Silva

leandroaugusto.silva@mackenzie.br

Faculdade de Computação e Informática

Programa de Pós-Graduação em Engenharia Elétrica e Computação