



Universidade Federal
de Santa Catarina

Inteligência Artificial II

Engenharia de Computação

Multilayer Perceptron

Perceptron de Múltiplas Camadas

Prof. Anderson Luiz Fernandes Perez

Email: anderson.perez@ufsc.br

Sumário

- Introdução
- Características
- Número de Camadas
- Funções de Ativação
- Número de Neurônios
- Algoritmo de treinamento da rede MLP

Introdução

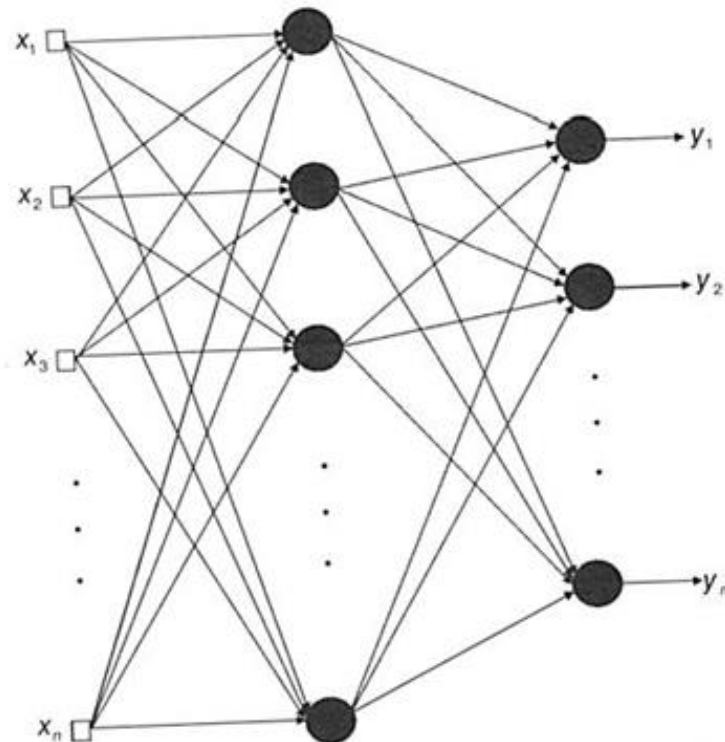
- Uso da rede neural Perceptron de uma única camada tornou-se inviável devido ao problema do XOR.
- A solução para o problema do Perceptron foi introduzir camadas de neurônios internas entre as entradas e as saídas da rede.

Introdução

- Redes MLP apresentam uma capacidade computacional maior do que aquele apresentado por redes de uma única camada.
- MLPs podem lidar com conjuntos de dados que não sejam linearmente separáveis.

Introdução

- Visão geral de uma rede MLP



Características

- A qualidade de aproximação obtida dependerá da complexidade da rede.
- O número de neurônios utilizados nas camadas intermediárias é fator preponderante para um bom desempenho da rede.

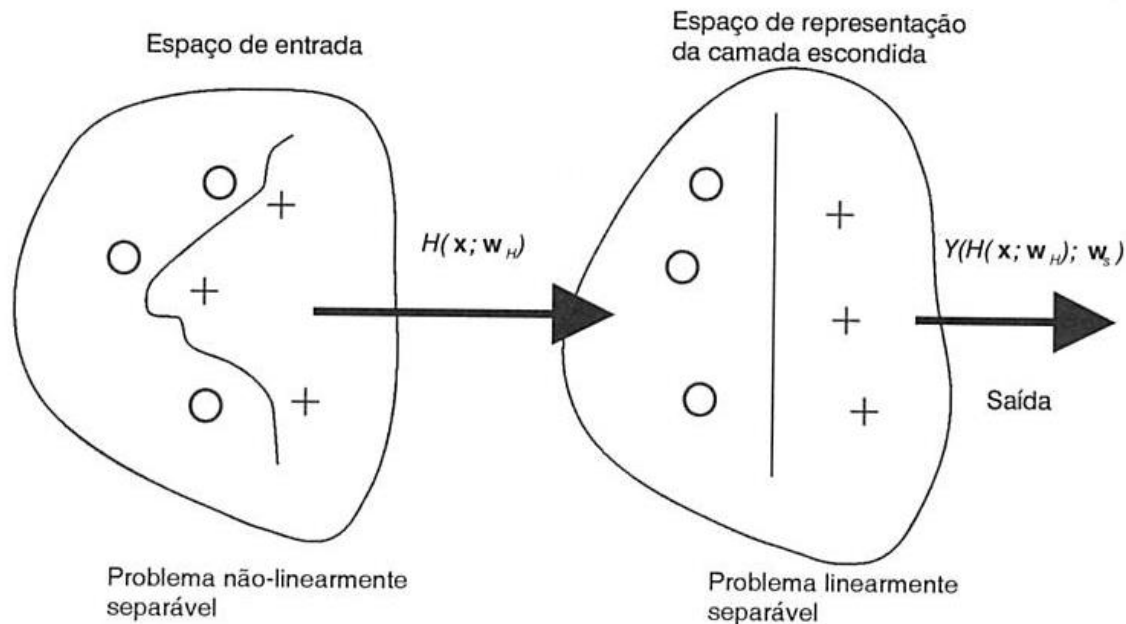
Características

- O Perceptron de Múltiplas (MLP) Camadas é uma rede neural com uma camada de entrada (sensorial), uma ou mais camadas ocultas de neurônios e uma camada de saída.
- O sinal de entrada se propaga para a frente através das camadas ocultas até a camada de saída (rede *feedforward* – alimentada para frente).



Características

- Mapeamento sucessivo em camadas realizado por uma rede MLP.



Características

- O treinamento em redes de uma única camada é feito de maneira supervisionada.
- A correção de erros é realizada através da equação (regra delta):
 - $\Delta w = \eta ex$
 - onde:
 - w = vetor de pesos
 - η = taxa de treinamento
 - e = erro obtido através da diferença entre a saída desejada e a saída corrente (obtida).
 - x = entrada do neurônio x .

Características

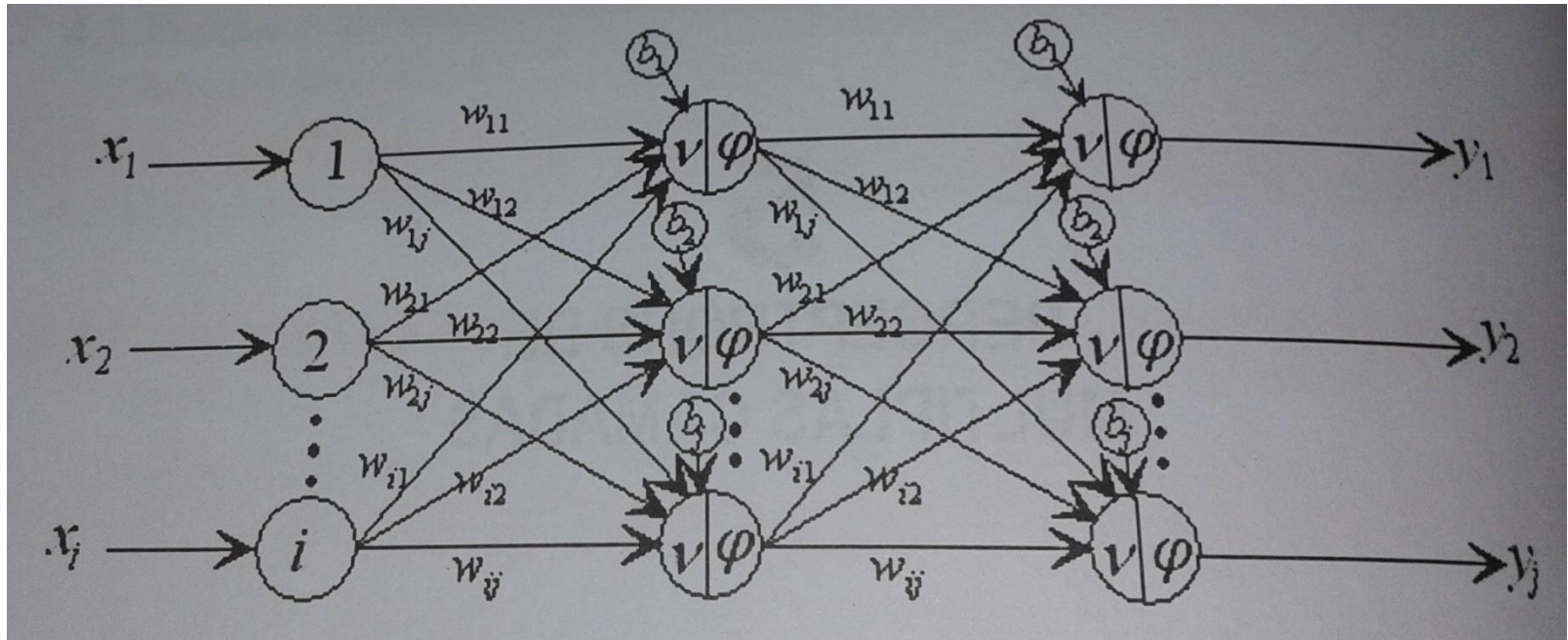
- Para redes MLP o procedimento de treinamento baseado na regra Δ só pode ser aplicado na camada de saída.
- Não existem saídas desejadas definidas para as camadas intermediárias.
- A questão que se coloca é como calcular ou estimar o erro das camadas intermediárias?

Características

- O MLP é uma generalização da rede Perceptron e é treinado de forma supervisionada.
- O algoritmo de aprendizado do MPL é conhecido como retropropagação do erro (*backpropagation*).

Características

- Visão Geral



Características

- Quando se é aplicado à camada de entrada da rede o vetor de sinais de entrada, calcula-se o campo local induzido v e os sinais de saída para todos os neurônios desta camada.

- $x(n) = [x_1, x_2, \dots, x_n]^T$

- Os sinais de saída são distribuídos, como sinal de entrada, para todos os neurônios da camada seguinte.
- Com a distribuição dos sinais entre os neurônios das camadas da rede se obtém o vetor de sinais funcionais da rede:

- $y(n) = [y_1, y_2, \dots, y_n]^T$

Características

Nota

Uma rede MLP funciona como uma sequência de Perceptrons simples interconectados, de forma a gerar a propagação dos sinais de entrada atravessando todas as camadas até a saída da rede.

Número de Camadas

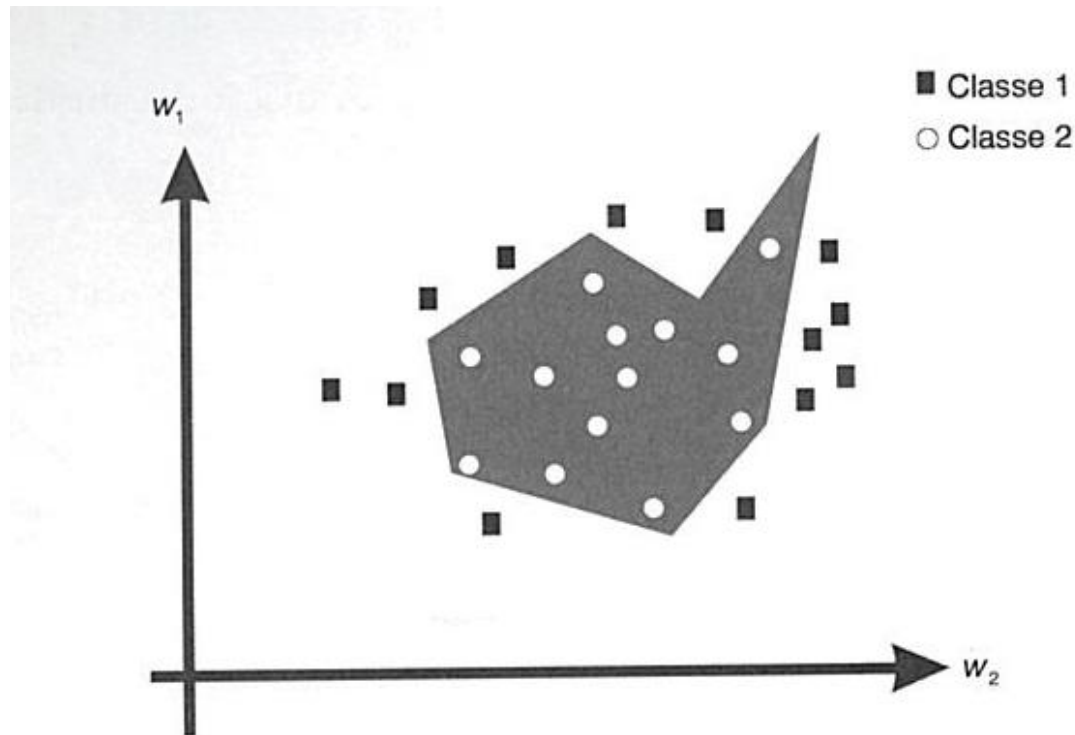
- O processamento realizado por cada neurônio de uma determinada camada é definido pela combinação dos processamento realizados pelos neurônios da camada anterior que estão conectados a ele.
- Para uma rede com pelo menos duas camadas intermediárias, pode-se afirmar que o seguinte processamento ocorre em cada uma das camadas:

Número de Camadas

- Primeira camada intermediária: cada neurônio contribui com retas para a formação da superfície no espaço de entrada;
- Segunda camada intermediária: cada neurônio combina as retas descritas pelos neurônios da camada anterior conectados a ele, formando regiões convexas, onde o número de lados é definido pelo número de unidades a ele conectadas.
- Camada de saída: cada neurônio forma regiões que são combinações das regiões convexas definidas pelos neurônios a ele conectados da camada anterior.

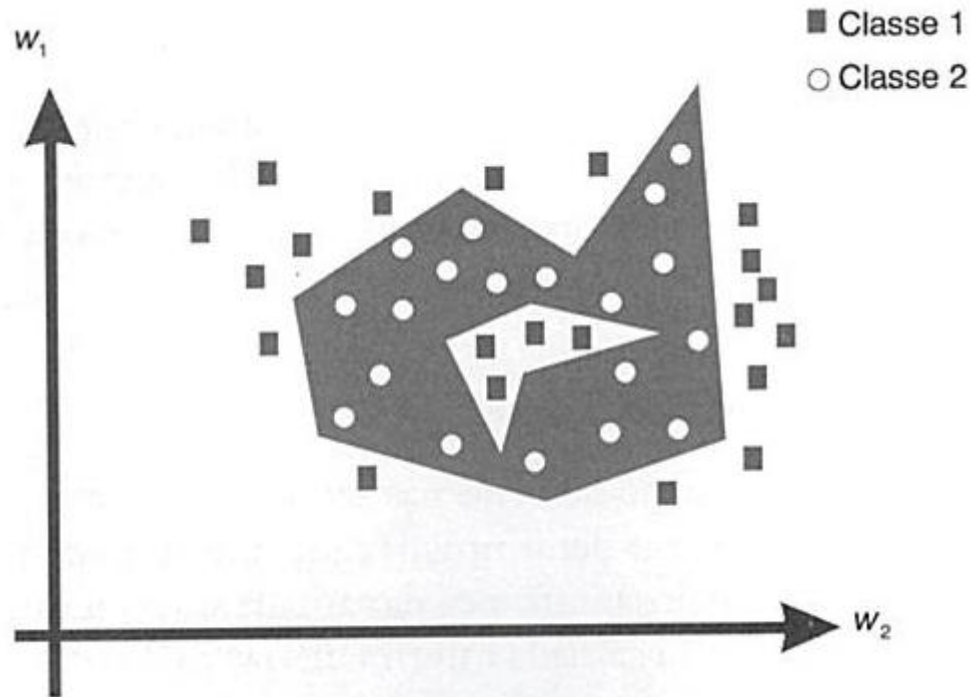
Número de Camadas

- Região definida pela processamento realizado pela segunda camada intermediária.



Número de Camadas

- Região definida pela processamento realizado pela camada de saída.



Número de Camadas

- Nota:
 - A utilização de um número maior de camadas do que o necessário pode levar a problemas de convergência da rede.
 - Camadas intermediárias trabalham com estimativas de erro, as quais, quando em maior quantidade, podem dificultar a convergência global da rede.

Funções de Ativação

- A função de ativação na camada intermediária deve ser não linear, para que a composição das funções nas camadas sucessivas seja capaz de resolver os problemas de maior ordem no espaço de entrada.
- As redes MLP se caracterizam pela utilização de funções **sigmoidais** nas **camadas intermediárias** e **sigmoidas ou lineares na saída**.

Funções de Ativação

- **Aproximação de Funções**
 - Utiliza-se funções sigmoidas em todas as camadas da rede.
- **Problemas de Classificação**
 - Funções sigmoidas nas camadas intermediárias e linear na camada de saída.
- **Redes Neurais de Base Radial (*Radial Basis Functions*)**
 - Utiliza funções gaussianas na camada escondida (intermediária) e linear na saída.

Número de Neurônios

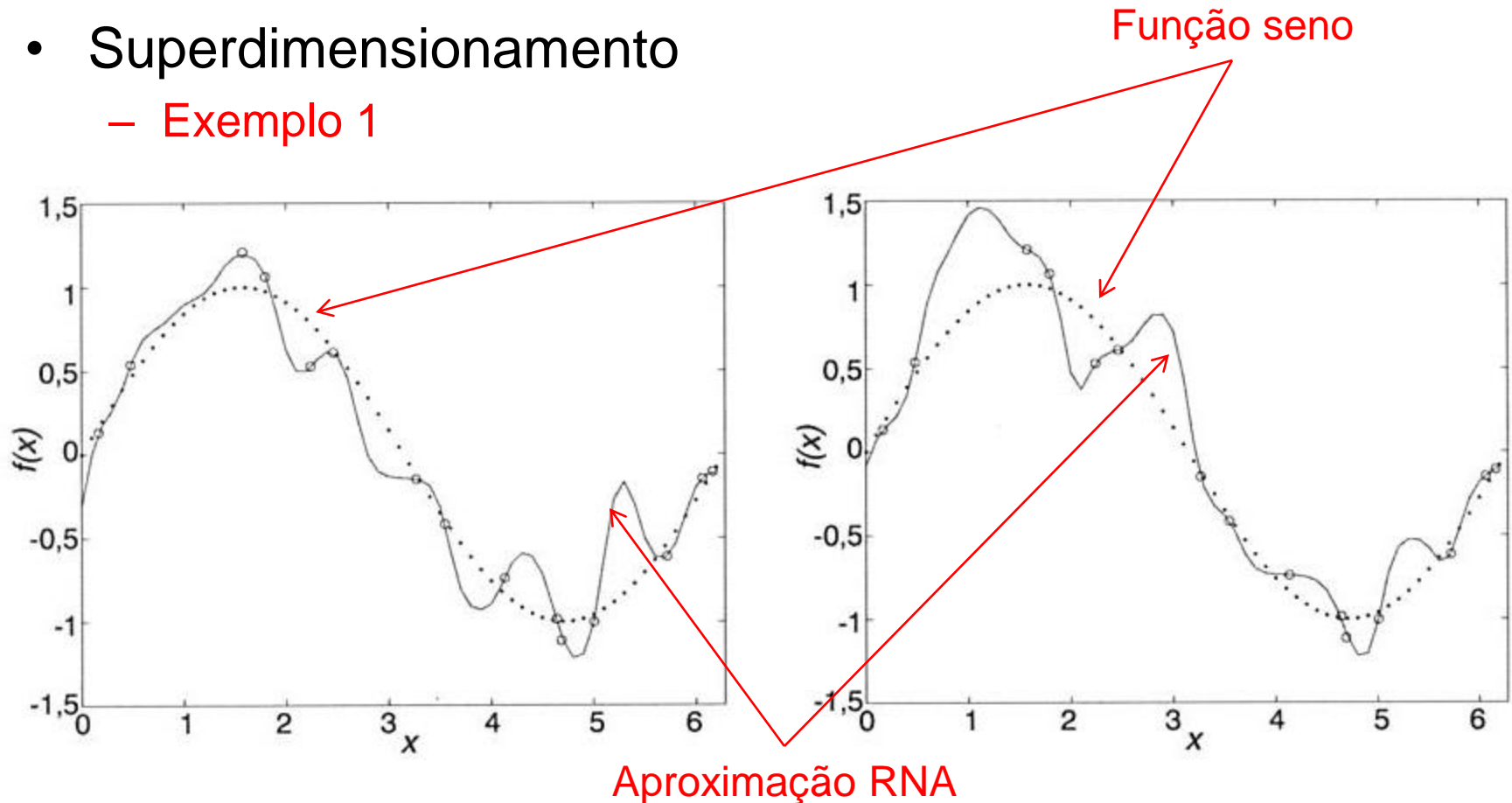
- O número de neurônio determina a capacidade da rede em resolver problemas de determinada complexidade.
- Quanto maior o número de neurônio maior a complexidade da rede e maior a sua abrangência em termos de soluções possíveis.

Número de Neurônios

- Superdimensionamento
 - Exemplo 1
 - Soluções diferentes para a aproximação da função $\text{Seno}(x)$ encontradas por uma rede MLP com 15 neurônios na camada intermediária.

Número de Neurônios

- Superdimensionamento
 - Exemplo 1

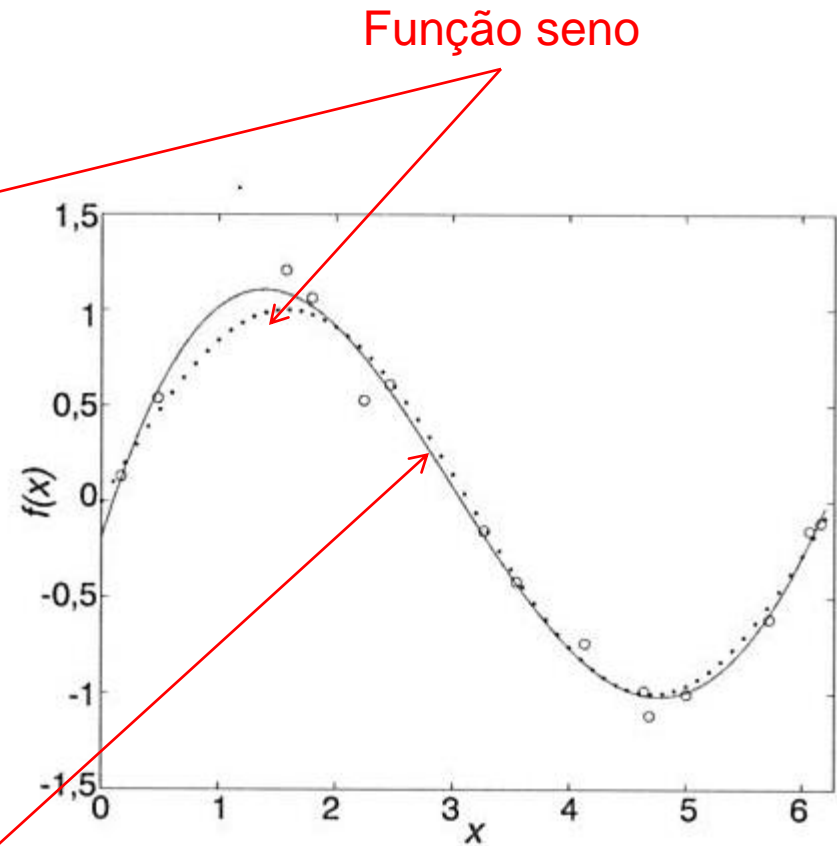
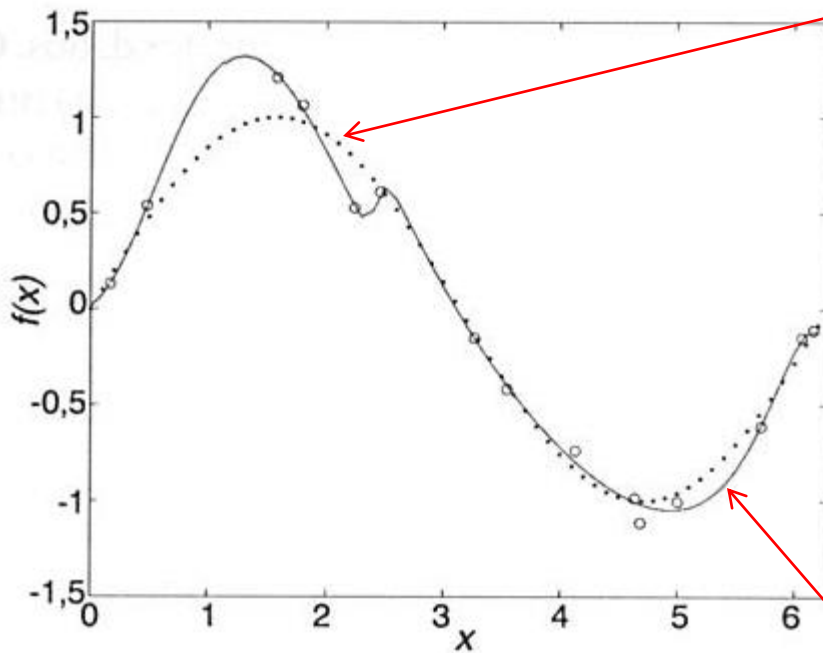


Número de Neurônios

- Superdimensionamento
 - Exemplo 2
 - Soluções diferentes para a aproximação da função $\text{Seno}(x)$ encontradas por uma rede MLP com 5 e 3 neurônios na camada intermediária.

Número de Neurônios

- Superdimensionamento
 - Exemplo 2



Aproximação RNA

Algoritmo de treinamento da rede MLP

- A rede MLP, assim como o Perceptron, precisa da definição do sinal de erro e da taxa de aprendizagem.
- Todo o procedimento de treinamento do MLP é baseado no sinal de erro.

Algoritmo de treinamento da rede MLP

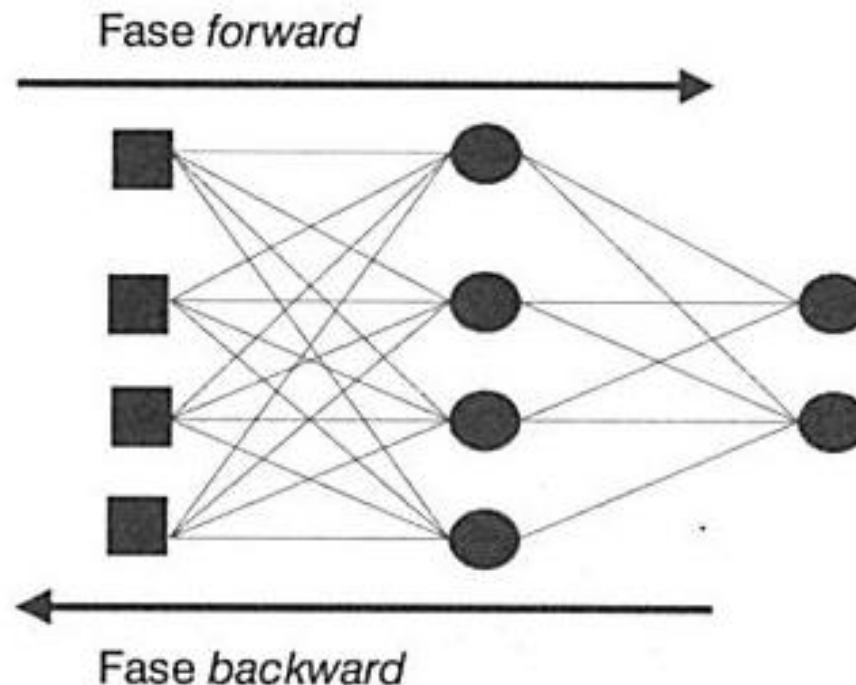
- O algoritmo mais popular para o treinamento de uma rede MLP é o backpropagation, por ser supervisionado, utiliza pares de entrada e saída:
 - (x, y_d)
 - por meio de um mecanismo de correção de erros, ajusta os pesos da rede.

Algoritmo de treinamento da rede MLP

- O treinamento ocorre em duas fases, onde cada fase percorre um sentido na rede.
- A fase *forward* é utilizada para definir a saída da rede para um dado padrão de entrada.
- A fase *backward* utiliza a saída desejada e a saída fornecida pela rede para atualizar os pesos de suas conexões.

Algoritmo de treinamento da rede MLP

- Visão geral do algoritmo *backpropagation*



Algoritmo de treinamento da rede MLP

- O sinal de erro do neurônio j , na iteração n , é definido por:
 - $e_j(n) = d_j(n) - y_j(n)$
- Onde:
 - $y_j(n)$ é a resposta calculada para o neurônio j ;
 - $d_j(n)$ é a resposta desejada para o neurônio j .

Algoritmo de treinamento da rede MLP

- O sinal de erro global e instantâneo da rede, para o j neurônios da camada de saída, na iteração n , é definido por:

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

- O erro de cada neurônio é elevado ao quadrado para evitar que um erro negativo oriundo de um neurônio compense um erro positivo de outro.

Algoritmo de treinamento da rede MLP

- O erro global médio da rede para todo o conjunto de treinamento é definido pela média aritmética dos erros globais instantâneos.

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

- O erro global instantâneo médio e o erro global instantâneo são funções de todos os parâmetros livres (pesos sinápticos e bias) da rede.

Algoritmo de treinamento da rede MLP

- Resumo do algoritmo de Retropropagação
 1. Inicialmente arbitram-se os valores aleatórios aos pesos sinápticos e níveis de bias, em uma distribuição uniforme, cuja média deverá ser zero.
 2. Apresenta-se uma época de exemplos à rede. Para cada exemplo, realiza-se a propagação dos sinais e a retropropagação dos erros com a correção dos pesos e níveis de bias.

Algoritmo de treinamento da rede MLP

- Resumo do algoritmo de Retropropagação
 3. Aplica-se à camada de entrada da rede o vetor de sinais de entrada $x(n)$ e calcula-se o campo local induzido e o sinal de saída para todos os neurônios, começando da camada de entrada até a camada de saída, onde se obtém o vetor de sinais funcionais da rede $y(n)$. Em seguida, calcula-se o sinal de erro $e_j(n)$ para cada neurônio da camada de saída, pela comparação de $y(n)$ com o vetor de sinais de saída desejados $d(n)$. Então, calculam-se o erro instantâneo e com este, o erro médio global, para teste de finalização.

Algoritmo de treinamento da rede MLP

- Resumo do algoritmo de Retropropagação
 4. Calculam-se os gradientes locais para todos os neurônios da camada de saída:

$$\delta_j(n) = -e_j(n)\varphi'_j(v_j(n))$$

Em seguida, calculam-se os ajustes para os pesos daquela camada, bem como dos bias, os quais devem ser somados aos valores atuais:

$$\Delta w_{ij} = \eta e_j(n)\varphi'_j(v_j(n))y_i(n) = -\eta\delta_j(n)y_i(n)$$

$$\Delta b_j = \eta e_j(n)\varphi'_j(v_j(n))y_i(n) = -\eta\delta_j(n)y_i(n).$$

Algoritmo de treinamento da rede MLP

- Resumo do algoritmo de Retropropagação
 4. O próximo passo é o cálculo do gradiente local para os neurônios da penúltima camada (camada oculta):

$$\delta_i(n) = \varphi_i'(v_i(n)) \sum_{j=1}^J (\delta_j(n) w_{ij}(n)).$$

Então, calcula-se o ajuste para todos os pesos desta camada, bem como dos bias, os quais devem ser somados aos valores atuais:

$$\Delta w_{ki} = -\eta \varphi_i'(v_i(n)) y_k(n) \sum_{j=1}^J (\delta_j(n) w_{ij}(n)) = -\eta \delta_i(n) y_k(n),$$

$$\Delta b_i = -\eta \varphi_i'(v_i(n)) y_k(n) \sum_{j=1}^J (\delta_j(n) w_{ij}(n)) = -\eta \delta_i(n) y_k(n).$$

Algoritmo de treinamento da rede MLP

- Resumo do algoritmo de Retropropagação
 4. O processo prossegue de forma idêntica para as demais camadas oculta, assim para como para a camada de entrada, em que os valores dos ajustes na primeira camada oculta (após a entrada da rede) deve ter o valor $y_k(n)$ substituído pelo valor $x_k(n)$.
 5. Iteram-se as computações apresentando novas épocas de exemplos de treinamento para a rede de forma aleatória de época para época, até que seja satisfeito o critério de parada, que pode ser o número máximo de iterações ou um valor limite para o erro global médio da rede.