Adaptive Linear Element (ADALINE)

Prof. Dr. Ajalmar Rocha

Disciplina: Inteligência Computacional Aplicada (ICA)
Programa de Pós-Graduação em Eng. de Telecomunicações (PPGET)
Instituto Federal do Ceará (IFCE)

Agosto/2013

Resumo

- Introdução
- ADALINE
- Função Quadrática de Erro
- Minimização da Função Custo
- Gradiente da Função Custo
- Regra de Aprendizagem

Introdução

- O objetivo do processo de aprendizado de uma rede neural é estimar uma função $F : \mathbb{R}^{p+1} \rightarrow \mathbb{R}$ usando um conjunto:

$$(\mathbf{x}_1, d_1), \ldots, (\mathbf{x}_n, d_n) \in \mathbb{R}^{p+1} \times \mathbb{R} \tag{1}$$

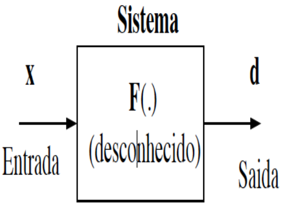
  em que $\mathbf{x}_i$ é o i-ésimo padrão de entrada (ou vetor de entrada) e d_i é a saída desejada.
- Este conjunto pode ser apresentado de forma concisa, tal como:

$$\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}. \tag{2}$$

- Ressalta-se que o mapeamento $F(.)$ para um vetor de entrada não especificado é completamente desconhecido, sendo a sua determinação o objetivo do treinamento das redes neurais.

Introdução

- Nesse contexto, um mapeamento $F(.)$ pode ser representado de forma simplificada como:



- E, portanto, uma rede neural pode ser utilizada para a obtenção de um mapeamento aproximado para o conjunto de dados $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$, i.e., $\hat{F}(.)$. Portanto, a saída estimada $\hat{d}$, pode ser descrita por $y = \hat{d} = \hat{F}(\mathbf{x})$.

Introdução

- E, portanto, uma rede neural pode ser utilizada para a obtenção de um mapeamento aproximado para o conjunto de dados $\{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$, i.e., $\hat{F}(.)$. Assim, a saída estimada $\hat{d} = y$ pela rede neural, tal que:

$$y = \hat{F}(\mathbf{x}). \tag{3}$$

- O processo que permite a descoberta de leis gerais pela observação e combinação de exemplos particulares é chamado **aprendizado indutivo**.

ADALINE

- Em 1960, o Elemento Linear Adaptativo (*Adaptive Linear Element* - ADALINE) surgiu na literatura quase que simultaneamente com o Perceptron, no trabalho:

Widrow and Hoff (1960)

B. Widrow and M. E. Hoff. Adaptive switching circuits. In IRE WESCON Convention Record-Part 4, pages 96-104, 1960.

- Ambos os modelos são baseados em elementos de processamento que executam operações sobre soma ponderada de suas entradas.
- A operação é não-linear do tipo degrau para o Perceptron.
- Enquanto, a operação é puramente linear para o ADALINE.

ADALINE

- Outros trabalhos relacionados ao ADALINE são apresentados a seguir.

B. Widrow and S. D. Stearns. Adaptive Signal Processing. Prentice-Hall, Upper Saddle River, NJ, 1985.

B.Widrow and R. Winter. Neural nets for adaptive filtering and adaptive pattern recognition. IEEE Computer, 21(3):25-39, 1988.

B. Widrow and M.A. Lehr. 30 years of adaptive neural networks: Perceptron, madaline and backpropagation. In Proceedings of the IEEE, volume 78, pages 1415-1442, 1990.

Neurônio Artificial para ADALINE

- Como dito, um neurônio artificial possui p entradas $\{x_i\}_{i=1}^{p}$ e possui p pesos sinápticos $\{w_i\}_{i=1}^{p}$, bem como um limiar (thresold) de ativação θ.
- O neurônio possui uma variável de ativação u, tal que

$$u = x_1 w_1 + x_2 w_2 + \cdots - \theta = \sum_{i=1}^{p} x_i w_i - \theta. \tag{4}$$

- Enquanto, a variável de saída para o ADALINE é descrita como segue:

$$y = f(u) = u. \tag{5}$$

Neurônio Artificial para ADALINE

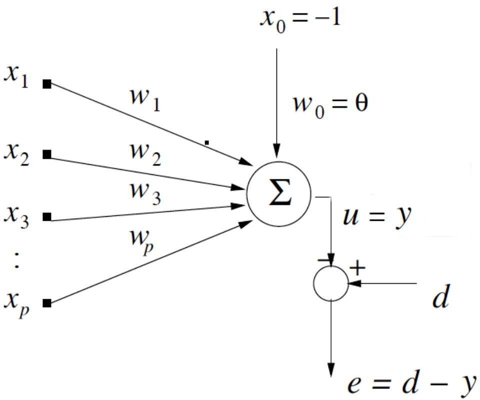
- Na equação para o cálculo da variável de saídas

$$y = f(u) = u = \mathbf{w}^T \mathbf{x} = \mathbf{x}^T \mathbf{w} = \sum_{i=0}^{p} w_i x_i. \tag{6}$$

o vetor de pesos **w** e de entrada **x** é descrito por

$$\mathbf{w} = \begin{bmatrix} \theta \\ w_1 \\ w_2 \\ \dots \\ w_p \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \dots \\ x_p \end{bmatrix}$$

Arquitetura para ADALINE



Função Quadrática de Erro

- O erro obtido com base na variável de saída e na saída desejada para um dado padrão $(\mathbf{x}_i, d_i)$ é descrito por:

$$e = d_i - y = d_i - u. \tag{7}$$

- O erro quadrático instantâneo para um dado vetor de entrada $(\mathbf{x}_i, d_i)$ é dado pela Equação (8)

$$\begin{aligned} e^2 &= (d_i - y)^2 \\ e^2 &= (d_i - \mathbf{w}^T\mathbf{x}_i)^2 \\ e^2 &= d_i^2 - 2d_i\mathbf{w}^T\mathbf{x}_i + (\mathbf{w}^T\mathbf{x}_i)^2, \end{aligned} \tag{8}$$

  a qual contém um termo quadrático em **w** e, por conseguinte, resulta em uma superfície em forma de parábola.

Minimização da Função Custo

- Considerando o erro quadrático como critério de desempenho, a partir de um valor inicial para o vetor de pesos, objetiva-se que o vetor de pesos se aproxime gradativamente do mínimo global.
- No entanto, busca-se atingir o mínimo da superfície J resultante, a saber:

$$
\begin{aligned}
J &= \frac{1}{2}e^2 \\
J &= \frac{1}{2}(d_i - y)^2 \\
J &= \frac{1}{2}(d_i - u)^2 \\
J &= \frac{1}{2}(d_i - \mathbf{w}^T\mathbf{x}_i)^2 \qquad (9)
\end{aligned}
$$

Minimização da Função Custo

- Com o intuito de minimizar a função custo

$$J = \frac{1}{2}(d_i - \mathbf{w}^T \mathbf{x}_i)^2$$

  é necessário que seja obtido a direção do ajuste a ser aplicado no vetor de pesos para aproximar a solução do mínimo de J.
- Para tal, usa-se o gradiente da função custo J no ponto $\mathbf{w}(t)$.
- Sabe-se que o gradiente possui a mesma direção da maior variação do erro (aponta na direção de crescimento da função), portanto o ajuste deve ocorrer na direção contrária ao gradiente. Logo, a variação dos pesos pode ser descrita por:

$$\Delta \mathbf{w}(t) \propto -\nabla J. \tag{10}$$

Gradiente da Função Custo

Consideração

Sabe-se que a função custo J depende de e, bem como sabe-se que e depende de y, e ainda sabe-se que y depende de $\mathbf{w}$.

- Cada componente do gradiente da função custo ∇J pode ser obtida com base na regra da cadeia, a saber:

$$\frac{\partial J}{\partial w_i} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial u} \frac{\partial u}{\partial w_i} \tag{11}$$

- Sabendo ainda que

$$\frac{\partial J}{\partial e} = e, \qquad \frac{\partial e}{\partial u} = -1 \qquad \text{and} \qquad \frac{\partial u}{\partial w_i} = x_i,$$

Gradiente da Função Custo

- Logo a *i*-ésima componente do gradiente da função custo

$$\frac{\partial J}{\partial w_i} = \frac{\partial J}{\partial e}\frac{\partial e}{\partial u}\frac{\partial u}{\partial w_i} \tag{12}$$

considerando que

$$\frac{\partial J}{\partial e} = e, \qquad \frac{\partial e}{\partial u} = -1 \qquad \text{and} \qquad \frac{\partial u}{\partial w_i} = x_i,$$

é igual a

$$\begin{aligned} \frac{\partial J}{\partial w_i} &= -x_i e \\ \frac{\partial J}{\partial \mathbf{w}} &= -\mathbf{x} e \end{aligned} \tag{13}$$

Regra de Aprendizagem

- A regra de aprendizagem pode então ser obtida como segue.

$$\begin{aligned} \Delta \mathbf{w} &= \mathbf{w}(t+1) - \mathbf{w}(t) \\ \mathbf{w}(t+1) &= \mathbf{w}(t) + \Delta \mathbf{w} \end{aligned} \tag{14}$$

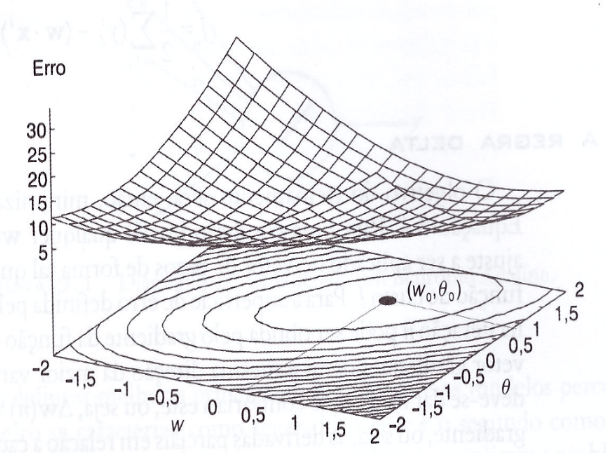
- Considerando que $\Delta \mathbf{w}(t) \propto -\nabla J$, pode se escrever

$$\begin{aligned} \Delta \mathbf{w} &= \mathbf{w}(t+1) - \mathbf{w}(t) \\ \mathbf{w}(t+1) &= \mathbf{w}(t) + \Delta \mathbf{w} \\ \mathbf{w}(t+1) &= \mathbf{w}(t) - \eta \nabla J \end{aligned} \tag{15}$$

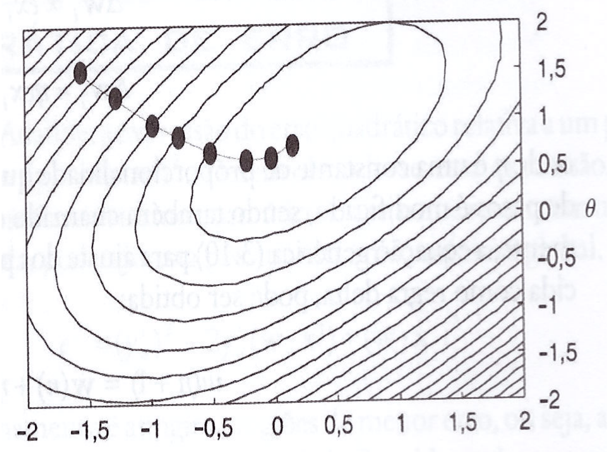
- Além disto, sabendo que $\frac{\partial J}{\partial \mathbf{w}} = -\mathbf{x}e$ temos que

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta e(t) \mathbf{x}(t) \tag{16}$$

Superfície de Erro



Curvas de Nível para Otimização dos Pesos



Diferença entre o Perceptron e o ADALINE

- **Qual a principal diferença entre os modelos Perceptron e ADALINE?**
- O Perceptron é um separador linear, enquanto o ADALINE é um aproximador linear de funções.
- Assim, os modelos se aplicam a problemas de natureza diferente.
- O Perceptron é aplicado em problemas de classificação de padrões; enquanto o ADALINE em problemas de aproximação de funções.
- **Seria possível usar o ADALINE para classificação de padrões? ou o Perceptron para aproximação de funções?**

Obrigado