Trabalho 3: Aprendizagem Widrow-Hoff

Pedro Victor Dos Santos Matias, 21601225

pvsm@icomp.ufam.edu.br

I. Introdução

O presente trabalho é sobre a rede ADALINE, mais especificadamente em uma de suas aplicações que é filtro preditivo. Na qual é informado a as característica do filtro, como por exemplo, a função de autocorrelação. A partir disso é respondido uma serie de perguntas que envolvem desde o cálculo do erro médio quadrático e até implementação do algoritmo LMS.

Este documento está organizado em 6 seções. Na segunda, será realizado uma revisão bibliográfica sobre as técnicas de reconhecimento de padrões . No capítulo 3 é descrita a arquitetura geral da rede e as perguntas a serem respondidas. Os resultados, como gráficos e valores obtidos são apresentados na seção 4. Na seção de conclusão é realizado uma as considerações finais entre o problema e os resultados obtidos. E por último um apêndice com a listagem dos códigos em MATLAB.

II. Revisão Bibliográfica

Como mencionamos na seç ao anterior, o trabalho está diretamente relacionado com a rede ADALINE. Esse modelo possui a mesma limitação principal que a rede perceptron, a incapacidade de solucionar problemas não separáveis linearmente. Contudo é mais utilizada que ele, principalmente em aplicações de processamento digital de sinais.

Uma das principais áreas de aplicação do ADALINE foi a filtragem adaptativa,como exemplo temos no artigo de Bhattacharya and Chakraborty (2011) que tenta melhorar o desempenho dinâmico de um filtro de energia ativa do tipo shunt. As propriedades preditivas e adaptativas das redes neurais artificiais (RNAs) são utilizadas para estimar rapidamente a corrente de compensação. A dinâmica da tensão dc-link é utilizada em um controlador preditivo para gerar a primeira estimativa seguida pela convergência do algoritmo por uma rede adaptativa baseada em RNA (adaline). Os pesos da rede são ajustados para minimizar a distorção harmônica total da corrente da fonte. A partir de experimentos extensivos eles confirmam a validade do esquema proposto para todos os tipos de carga (balanceada e não balanceada) para um sistema trifásico de três fios.

Outro exemplo é apresentado por Kavak et al. (2005) em aplicação para área de comunicações sem fio no modo de divisão de tempo e duplex (TDD), o desempenho do beamforming de downlink de um sistema de antena inteligente na estação base pode ser degradado devido à variação de vetores de assinatura espacial correspondentes a usuários móveis. Para atenuar isso, os feixes de downlink devem ser controlados pelo ajuste adequado de seus vetores de peso em resposta à alteração da dinâmica de propagação. Isto pode ser conseguido modelando os vetores de assinatura espacial no período de ligação ascendente e, em seguida, prevendo que eles sejam usados como vetores de peso de forma de feixe para a nova posição móvel no período de transmissão de downlink. A predição de assinaturas espaciais baseada em modelagem de redes ADALINE forneceu certo nível de melhoria de desempenho em comparação com o método de formação de feixe convencional que emprega assinatura espacial obtida no intervalo de uplink anterior. A modelagem ADALINE supera a modelagem autorregressiva (AR) em termos de melhoria de SNR de downlink e melhoria de erro relativo, especialmente sob altas velocidades móveis.

III. Metodologia

Considerando a rede abaixo E assumindo que $y(k)$ é um processo estacionário com função de

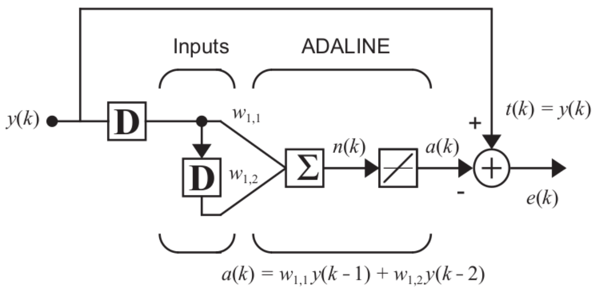

Figura 1: Preditor Adaptativo

autocorrelação $C_y(n) = E[y(k)(y(k+n))]$.

i Escrever uma expressão para o erro quadrático médio em termos de $C_y(n)$

ii Encontre uma expressão para o erro quadrático médio quando $y(k) = sen(\frac{k\pi}{5})$

iii Encontre os autovalores e autovetores da matriz Hessiana para o erro quadrático médio. Localize o ponto mínimo e trace um contorno grosseiro para as curvas de nível.

iv Encontre a máxima taxa de aprendizado estável para o algoritmo LMS.

v Implemente o algoritmo LMS para esse problema. Simule 40 passos do mesmo usando uma taxa de aprendizado estável e trace um gráfico do erro versus o passo da simulação. Use o vetor nulo como o estado inicial da simulação. Verifique o que acontece com o erro.

Para resolver essas questões será usado os conceitos aprendidos sobre a arquitetura da rede ADALINE e o algoritmo de aprendizagem LMS. No próximo capítulo as equações e resultados serão apresentados.

IV. Resultados

1 **Escrever uma expressão para o erro quadrático médio em termos de $C_y(n)$.**

A função de autocorrelação é uma medida da correlação entre as amostras de uma série temporal que são separadas por k unidades de tempo. De acordo com Hagan et al. (1997) o erro quadrático médio para uma rede ADALINE é:

$$F(x) = E[e^2] = E[(t - \mathbf{x}^T\mathbf{z})^2]$$

$$F(x) = E[t^2] - 2.\mathbf{x}^T.h + \mathbf{x}^T.\mathbf{R}.\mathbf{x}$$

O vetor de entrada $\mathbf{z}$ é dado por

$$\mathbf{z}(k) = \begin{bmatrix} y(k-1) \\ y(k-2) \end{bmatrix}.$$

E a saída desejada por

$$t(k) = y(k).$$

De modo que

$$E[t^2] = E[y^2(k)] = C_y(0)$$

Nesse caso a matriz de correlação será:

$$R = [\mathbf{z}.\mathbf{z}^T] \quad \mathbf{h} = E[t.\mathbf{z}]$$

$$\boldsymbol{R} = \begin{bmatrix} E[y^2(k-1)] & E[y(k-1)y(k-2)] \\ E[y(k-2)y(k-1)] & E[y^2(k-2)] \end{bmatrix}$$

como a função de correlação é $C_y(n) = E[y(k)y(k+n)]$. Logo:

$$\boldsymbol{R} = \begin{bmatrix} C_y(0) & C_y(1) \\ C_y(1) & C_y(0) \end{bmatrix}.$$

Já o vetor $\mathbf{h}$ que fornece a correlação cruzada entre o vetor de entrada e o valor desejado é dado por

$$\mathbf{h} = E[t\mathbf{z}].$$

Assim o $\mathbf{h}$ será:

$$\mathbf{h} = \begin{bmatrix} y(k)y(k-1) \\ y(k)y(k-2) \end{bmatrix} = \begin{bmatrix} C_y(1) \\ C_y(2) \end{bmatrix}.$$

Por fim, a expressão do erro fica escrita da seguinte forma:

$$F(x) = C_y(0) - 2.\mathbf{x}^T.\begin{bmatrix} C_y(1) \\ C_y(2) \end{bmatrix} + \mathbf{x}^T.\begin{bmatrix} E[y^2(k-1)] & E[y(k-1)y(k-2)] \\ E[y(k-2)y(k-1)] & E[y^2(k-2)] \end{bmatrix}.\mathbf{x}.$$

2 **Encontre uma expressão para o erro quadrático médio quando $y(k) = sen(\frac{k\pi}{5})$.**

$$C_y(0) = E[y^2(k-1)] = E\left[sen^2\left(\frac{(k)\pi}{5}\right)\right] = \frac{1}{10}\sum_{k=1}^{10} sen^2\left(\frac{(k)\pi}{5}\right) = 0.5$$

$$C_y(1) = E\left[sen\left(\frac{(k)\pi}{5}\right)\cdot sen\left(\frac{(k)\pi}{5}\right)\right] = 0.4045$$

Assim, a matriz **R**:

$$\boldsymbol{R} = \begin{bmatrix} 0.5 & 0.4045 \\ 0.4045 & 0.5 \end{bmatrix}.$$

E para o vetor **h** igual

$$\boldsymbol{h} = \begin{bmatrix} C_y(1) \\ C_y(2) \end{bmatrix}.$$

Calculando $C_y(2)$

$$C_y(2) = E\left[sen\left(\frac{(k)\pi}{5}\right)\cdot sen\left(\frac{(k-2)\pi}{5}\right)\right] = 0.1545.$$

Assim, o vetor **h** será:

$$\mathbf{h} = \begin{bmatrix} 0.4045 \\ 0.1545 \end{bmatrix}.$$

Os pesos otimizados são

$$\mathbf{x}^* = \boldsymbol{R}^{-1}\boldsymbol{h} = \begin{bmatrix} 0.5 & 0.4045 \\ 0.4045 & 0.5 \end{bmatrix}^{-1}\begin{bmatrix} 0.4045 \\ 0.1545 \end{bmatrix} = \begin{bmatrix} 5.7884 & -4.6828 \\ -4.6828 & 5.7884 \end{bmatrix}^{-1}\begin{bmatrix} 0.4045 \\ 0.1545 \end{bmatrix} = \begin{bmatrix} 1.6179 \\ -0.9999 \end{bmatrix}.$$

Assim o erro médio quadrático é:

$$F(x) = C_y(0) - 2.\mathbf{x}^T.\begin{bmatrix} C_y(1) \\ C_y(2) \end{bmatrix} + \mathbf{x}^T.\begin{bmatrix} E[y^2(k-1)] & E[y(k-1)y(k-2)] \\ E[y(k-2)y(k-1)] & E[y^2(k-2)] \end{bmatrix}.\mathbf{x}.$$

$$F(\mathbf{x}^*) = 0.500 - 2\begin{bmatrix} 0.4045 \\ 0.1545 \end{bmatrix} + \begin{bmatrix} 1.6179 & -0.9999 \end{bmatrix}\begin{bmatrix} 0.5 & 0.4045 \\ 0.4045 & 0.5 \end{bmatrix}\begin{bmatrix} 1.6179 \\ -0.9999 \end{bmatrix} = 3.7998.10^{-05}.$$

3 **Encontre os autovalores e autovetores da matriz Hessiana para o erro quadrático médio. Localize o ponto mínimo e trace um contorno grosseiro para as curvas de nível.**

A matriz Hessiana é

$$\nabla^2 F(x) = A = 2R = \begin{bmatrix} 1 & 0.8090 \\ 0.8090 & 1 \end{bmatrix}.$$

Calculando os autovalores e autovetores da matriz **A**:

$$|\mathbf{A} - \lambda\mathbf{I}| = \begin{bmatrix} 1-\lambda & 0.8090 \\ 0.8090-\lambda & 1 \end{bmatrix}.$$

Utilizamos o MATLAB para realizar esse cálculo e obtivemos,

$$\lambda_1 = 0.1910 \quad \boldsymbol{v_1} = \begin{bmatrix} -0.7071) \\ 0.7071 \end{bmatrix}$$

$$\lambda_2 = 1.8090 \quad \boldsymbol{v_2} = \begin{bmatrix} 0.7071) \\ 0.7071 \end{bmatrix}.$$

Traçando um contorno da função erro F(x) centrada no valor do mínimo **x***, usando o MATLAB foi obtido a imagem seguinte.

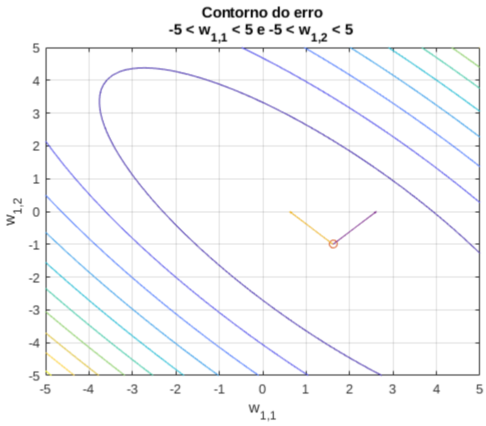

Figura 2: Contorno do Erro

4 **Encontre a máxima taxa de aprendizado estável para o algoritmo LMS.**

$$\alpha < \frac{1}{\lambda_{max}}$$

Assim, a taxa máxima de aprendizado estável é

$$\alpha < \frac{1}{1.8090} = 0.5527$$

5 **Implemente o algoritmo LMS para esse problema. Simule 40 passos do mesmo usando uma taxa de aprendizado estável e trace um gráfico do erro versus o passo da simulação. Use o vetor nulo como o estado inicial da simulação. Verifique o que acontece com o erro.**

O algoritmo do LMS desenvolvido realiza os seguintes passos:

1 inicializa a matriz de pesos em zero;

2 inicializa um vetor de erros;

3 calcula a saída da rede

$$a(k) = \mathbf{W.p};$$

4 calcula o erro da saída em relação ao desejado

$$e = t(k) - a(k)$$

5 atualiza os pesos seguindo a regra:

$$W(k+1) = W(k) + 2.\alpha.e(k).\mathbf{p}^T$$

6 repetir desde o passo 3 até ter executado todos as iterações de treinamento pré-definida.

Os resultados obtidos são mostrados no gráficos, sendo eles respectivamente o de curvas de nível da função **F(x)** e da variação do erro *versus* os passos de simulação.

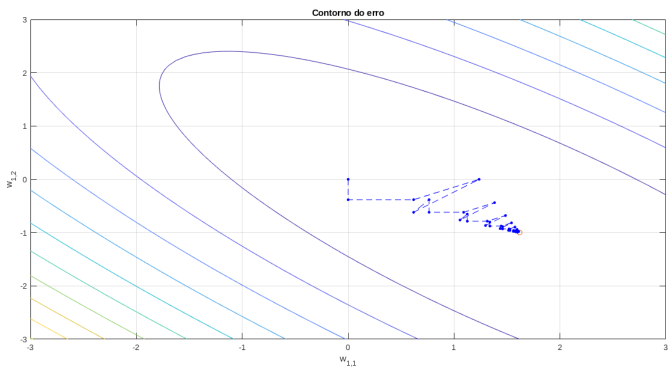

Figura 3: Trajetória dos pesos em F(x)

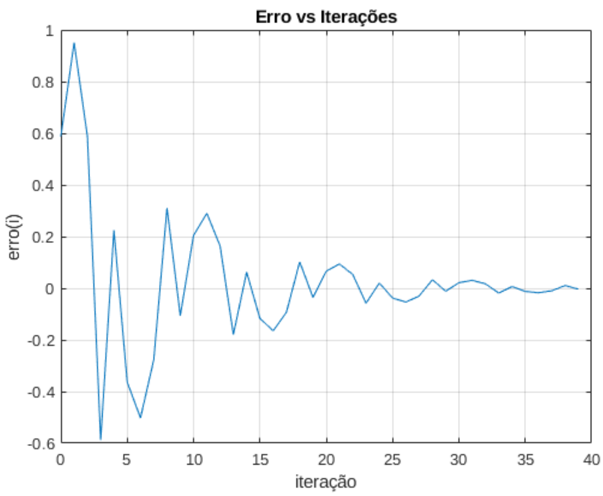

Figura 4: Erro *versus* iterações, com taxa de aprendizagem de $\alpha = 0.5527$

V. Conclusão

Neste trabalho realizamos o estudo e implementação de uma rede neural ADALINE como um filtro preditivo, para esse caso a saída da rede é uma função contínua, isto é, o objetivo foi realizar a tarefa regressão. Foi desenvolvido o algoritmo iterativo LMS em matlab, baseado no principio da pertubação mínima, para realizar a simulação da rede com uma função senoidal de entrada. Os resultados obtidos da simulação foram satisfatórios pois foram após 40 iterações os pesos da rede já haviam estabilizados e o erro convergido para o valor do mínimo global obtido pela equação de Wiener-Hopf.

Referências

A. Bhattacharya and C. Chakraborty. A shunt active power filter with enhanced performance using ann-based predictive and adaptive controllers. *IEEE Transactions on Industrial Electronics*, 58(2):421–428, Feb 2011. ISSN 0278-0046. doi: 10.1109/TIE.2010.2070770.

M. T. Hagan, H. B. Demuth, M. Hudson, et al. Neural network design. 1997.

A. Kavak, H. Yigit, and H. M. Ertunc. Using adaline neural network for performance improvement of smart antennas in tdd wireless communications. *IEEE Transactions on Neural Networks*, 16(6):1616–1625, Nov 2005. ISSN 1045-9227. doi: 10.1109/TNN.2005.857947.

VI. Apêndice

Neste Apêndice encontram-se os códigos fonte do script implementado em MATLAB utilizado neste trabalho para realizar a especificação e treinamento da rede, desenvolvido para realizar a função de filtro preditivo.

I. Script

```
close all,clear all,clc;

C0 = 0.5;
C1 = 0.4045;
C2 = 0.1545;
R = [C0 C1; C1 C0];
h = [C1;C2];
x = inv(R)*h; % Ponto minimo da função
F = C0 - 2*x'*h + x'*R*x;
A = 2*R;  % Matriz Hessiana
[v, lambda] = eig(A); % Auto valores e auto vetores


f = @(W1,W2) C0 - 2*[W1 , W2]*h + [W1 ,W2]*R*[W1 ; W2];
figure
fcontour(f);
hold on;
scatter(x(1),x(2),'o');
quiver(x(1),x(2),v(1,1),v(2,1),sqrt(2));
quiver(x(1),x(2),v(1,2),v(2,2),sqrt(2));
hold off
grid on
title({'Contorno do erro','-5 < w_{1,1} < 5 e -5 < w_{1,2} < 5 '});
xlabel('w_{1,1}');
ylabel('w_{1,2}');

%% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %%

% Padrões de treinamento
k = 0:9; % 10 amostras
y = sin(k*pi/5);

% Prototipo de entrada P
P = [sin((k-1)*pi/5);sin((k-2)*pi/5)];
t = y;

```

```
% Taxa máxima de aprendizagem
alpha = 1/max(max(lambda));

%% Algortimos LMS
W = [0 , 0];
trajetoria = [0,0];
E = [];
for epoca=1:40
    i = mod(epoca,length(k))+1;
    p = P(:,i); % extrai um padrão de entrada
    a = W*p;
    e = t(i) - a; % calculo do erro
    E(end+1) = e;
    W = W + 2 * alpha * e * p'; %LMS Algoritmo
    trajetoria(end+1,:) = W;  % Armazena a trajetoria dos pesos
end

figure
fcontour(f,[-3 3]);
hold on;
scatter(x(1),x(2),'o');
line(trajetoria(:,1)',trajetoria(:,2)','Color','b','LineStyle','--','Marker','.','LineWidth',0.7,
hold off
grid on
title({'Contorno do erro'});
xlabel('w_{1,1}');
ylabel('w_{1,2}');

figure
plot(0:39,E);
grid on
title({'Erro vs Iterações'});
xlabel('iteração');
ylabel('erro(i)');
```