

Segundo trabalho computacional

Disciplina ACH2016 – Inteligência Artificial

1 Introdução

Este trabalho consiste na tarefa de predição de séries temporais, especificadamente série temporal do ibovespa, usando Rede MLP feedforward e Redes de Funções de Base Radial. O trabalho poderá ser realizado em grupos de até quatro alunos. Cabe aos alunos se organizarem na formação dos grupos. O aluno deverá implementar seu modelo em python, java, C ou C++. Abaixo de cada passo consta um código em matlab que pode ser utilizado para ajudar nas implementações.

O grupo deverá entregar, na forma digital, o relatório, o qual deve conter a descrição e análise do processo de desenvolvimento dos algoritmos de predição. Este relatório deverá ser entregue, impreterivelmente, em 02/07/2023 e não serão admitidos atrasos, sendo que o aluno recebe nota zero se não entregar o trabalho até esta data.

Todo o material digital gerado e utilizado no trabalho (códigos, bases de dados de testes, bem como um arquivo.pdf referente ao relatório entregue impresso) deverão ser depositados no edisciplina, em área específica para o grupo, até 23:59 do dia de entrega do trabalho. No caso dos arquivos ultrapassem a capacidade de armazenamento do edisciplina, o aluno poderá utilizar github ou google drive e compartilhar o link no edisciplinas.

Além disso, todos os alunos, individualmente, deverão gravar um vídeo de 15 minutos descrevendo qual foi o trabalho que realizou. A nota que cada aluno receberá pelo trabalho está condicionada ao bom desempenho nesta defesa.

2 Motivação

Em qualquer ramo de atuação profissional, percebe-se um crescimento acentuado na demanda por predições e detecção de tendências junto a variáveis de interesse. Ferramentas computacionais capazes de fornecer predições automáticas acerca de valores futuros de certas variáveis que estão sendo monitoradas já fazem parte do dia-a-dia de muitas empresas privadas e órgãos governamentais, e têm contribuído para o sucesso na definição de políticas estratégicas, em processos de tomada de decisão e em todo tipo de planejamento de curto e médio prazo. Uma série temporal é dada pelos valores ao longo do tempo de uma variável de interesse, como em:

- atividades vitais ou funções orgânicas de um indivíduo;
- índices econômicos;
- índices sociais;
- variáveis climáticas;
- variáveis de ecossistemas;
- monitoramento de operação de plantas industriais;
- monitoramento de processos químicos e físicos.

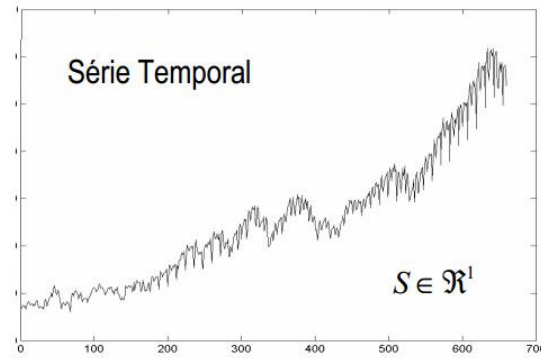
Sendo a predição uma estimativa de valores futuros a partir do conhecimento do histórico de uma variável até o presente, surgem algumas questões:

- o histórico de uma variável até o presente é capaz de auxiliar na predição do comportamento futuro
- como geralmente a variável de interesse tem seu comportamento atrelado a uma grande quantidade de fatores e a uma complexa rede de inter-relações, como é possível prever seu comportamento sem modelar os fenômenos complexos que regem o comportamento da variável e sem monitorar outras variáveis que influenciam nesse comportamento? Exemplo: como prever a vazão de um rio com base apenas em seu histórico, sem levar em conta a vazão dos afluentes, o nível de chuvas na cabeceira do rio, o consumo de água para irrigação agrícola e o seu grau de assoreamento, dentre outros fatores?

3 Metodologia

Para o emprego de uma rede neural artificial feedforward como preditor de um passo à frente, é necessário definir quais valores passados da série serão utilizados na definição da entrada da rede neural. Feito isso, o problema de síntese do preditor se transforma em um problema de treinamento supervisionado, onde o que se deseja é obter um mapeamento multidimensional não-linear de entrada-saída, como indicado na sequência de passos abaixo.

Passo 1: Obter as séries temporais, ou seja, os valores históricos da variável a ser predita um passo à frente e cinco passos a frente. Se necessário, normalize os dados (média zero e variância unitária), evitando que o intervalo de excursão dos valores seja qualquer. A série temporal corresponde a coluna Open da planilha ibov.csv.



$$S = s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8 \dots, s_N$$

Passo 2: Definir quais valores passados da série serão considerados na predição. Suponha aqui que L valores passados consecutivos sejam considerados. Com isso, monte a tabela a seguir, a qual retrata o comportamento desejado do preditor.

s_{t-L+1}	s_{t-L+2}	...	s_t	s_{t+1}
s_1	s_2	...	s_L	s_{L+1}
s_2	s_3	...	s_{L+1}	s_{L+2}
\vdots	\vdots	\vdots	\vdots	\vdots
s_{N-L}	s_{N-L-1}	...	s_{N-1}	s_N
\Downarrow	\Downarrow		\Downarrow	\Downarrow
X_1	X_2		X_L	Y

```
function [Xtr,Xval,Xtest,Ytr,Yval,Ytest]=geradados(open,lag,passo, Ntr, Nval, Ntest)
x=[];
y=[];
N=length(open);
for i=1:N-lag-passo
    x=[x;open(i:lag+i)'];
    y=[y;open(lag+i+passo)];
end
Xtr=x(1:Ntr,:);
Xval=x(Ntr+1:Ntr+Nval,:);
Xtest=x(Ntr+Nval+1:end,:);
Ytr=y(1:Ntr,:);
Yval=y(Ntr+1:Ntr+Nval,:);
Ytest=y(Ntr+Nval+1:end,:);
```

O valor de L pode ser estimado usando correlação entre a série original e a série adiantada.

A função de autocorrelação r_k mede a correlação entre series temporais invariáveis y_t e y_{t+k}

onde $k = 0, \dots, K$ e y_t é um processo estocástico.

$$r_k = \frac{c_k}{c_0}$$

Onde $c_k = \frac{1}{N} \sum_{t=1}^{N-k} (y_t - \bar{y})(y_{t+k} - \bar{y})$

c_0 é a variância amostral da série temporal

Se a série temporal é completamente randômica, então o erro padrão reduz para $1/\sqrt{N}$

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
% X – série temporal  
% numlag – número máximo de lag  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
function lag=calc_autocorr(X,numlag)  
X = X - mean(X);  
N = length(X);  
co = X'*X;  
rk(1,1)=1;  
for i=1:numlag  
    rk(i+1,1)= X(1:end-i)*X(i+1:end)/co;  
end  
figure(1)  
clf  
stem(0:numlag,rk)  
hold on  
plot(0:numlag,ones(numlag+1,1)*2/sqrt(N),'r')  
pos=find(rk>(2/sqrt(N)))  
lag=pos(end)-1;
```

Passo 3: Separe os dados da tabela acima em 3 conjuntos: conjunto de treinamento, conjunto de validação e conjunto de teste. O conjunto de treinamento e validação vai até 29/12/2022. O conjunto de teste compreende entre 02/01/2023 à 15/05/2023. O tamanho conjunto de validação deve ser determinado pelo grupo.

Passo 4: Treine a rede neural com o conjunto de treinamento (ela vai produzir um mapeamento do $\mathfrak{R}_L \rightarrow \mathfrak{R}_1$) e pare o treinamento quando for atingido o valor mínimo do erro quadrático médio para os dados de validação.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Rede Neural MLP %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
function [Aval,Bval]=treina_mlp(X,Yd,Xval,Ydval,h,nepocasmax)  
[N,ne]= size(X);  
Nval = size(Xval,1);  
X = [X,ones(N,1)];  
Xval=[Xval,ones(Nval,1)];  
ns= size(Yd,2);  
A = rands(h,ne+1);  
B = rands(ns,h+1);  
Yr = calc_saida(X,A,B,h,N);  
[dJdA,dJdB]=calc_grad(X,Yd,A,B,N);  
grad = [dJdA(:);dJdB(:)];  
erro = Yr-Yd;  
EQM = 1/N*sum(sum(erro.*erro));  
Yrval = calc_saida(Xval,A,B,h,Nval);  
erroval = Yrval-Ydval;  
EQMval = 1/N*sum(sum(erroval.*erroval));  
vetEQM =[];  
vetEQM = [vetEQM,EQM];  
nep = 0;
```

```

while norm(grad)>1e-5 & nep<nepocasmax
    nep = nep+1;
    alfa=calc_alfa(X,Yd,A,B,dJdA,dJdB,N);
    A = A - alfa*dJdA;
    B = B - alfa*dJdB;
    Yr = calc_saida(X,A,B,h,N);
    [dJdA,dJdB]=calc_grad(X,Yd,A,B,N);
    grad = [dJdA(:);dJdB(:)];
    erro = Yr-Yd;
    EQM = 1/N*sum(sum(erro.*erro));
    vetEQM = [vetEQM,EQM];
    Yrval = calc_saida(Xval,A,B,h,Nval);
    erroval = Yrval-Ydval;
    EQMval_new = 1/N*sum(sum(erroval.*erroval));
    if EQMval_new < EQMval
        EQMval = EQMval_new;
        Aval = A;
        Bval = B;
    end

end

plot(vetEQM)
vetEQM(end)
end

```

```

function Y = calc_saida(X,A,B,h,N)
Zin = X*A';
Z = 1./(1+exp(-Zin));
Zb = [Z,ones(N,1)];
Yin = Zb*B';
Y = 1./(1+exp(-Yin));
end

```

```

function [dJdA,dJdB]=calc_grad(X,Yd,A,B,N)
Zin = X*A';
Z = 1./(1+exp(-Zin));
Zb = [Z,ones(N,1)];
Yin = Zb*B';
Y = 1./(1+exp(-Yin));
erro = Y-Yd;
fl = (1-Z).*Z;
gl = (1-Y).*Y;
dJdB = 1/N*(erro.*gl)*Zb;
dJdZ = (erro.*gl)*B(:,1:end-1);
dJdA = 1/N*(dJdZ.*fl)*X;
end

```

```

function alfa_m=calc_alfa(X,Yd,A,B,dJdA,dJdB,N)
d = [-dJdA(:);-dJdB(:)];
alfa_l =0;
alfa_u =rand(1,1);
Aaux = A - alfa_u*dJdA;
Baux = B - alfa_u*dJdB;
[dJdAaux,dJdBaux]=calc_grad(X,Yd,Aaux,Baux,N);
grad = [dJdAaux(:);dJdBaux(:)];
hl = grad'*d;
while hl<0
    alfa_l = alfa_u;
    alfa_u = 2*alfa_u;

```

```

Aaux = A - alfa_u*dJdA;
Baux = B - alfa_u*dJdB;
[dJdAaux,dJdBaux]=calc_grad(X,Yd,Aaux,Baux,N);
grad = [dJdAaux(:);dJdBaux(:)];
hl = grad'*d;
end
alfa_m = (alfa_l+alfa_u)/2;
Aaux = A - alfa_m*dJdA;
Baux = B - alfa_m*dJdB;
[dJdAaux,dJdBaux]=calc_grad(X,Yd,Aaux,Baux,N);
grad = [dJdAaux(:);dJdBaux(:)];
hl = grad'*d;
k =0;
kmax = ceil(log((alfa_u - alfa_l)/1e-4));
while k<kmax & abs(hl)>1e-4
    if hl>0
        alfa_u = alfa_m;
    elseif hl<0
        alfa_l = alfa_m;
    else
        break;
    end
    alfa_m = (alfa_l+alfa_u)/2;
    Aaux = A - alfa_m*dJdA;
    Baux = B - alfa_m*dJdB;
    [dJdAaux,dJdBaux]=calc_grad(X,Yd,Aaux,Baux,N);
    grad = [dJdAaux(:);dJdBaux(:)];
    hl = grad'*d;
end
end

```

O código da RBF encontra-se nos slides.

Passo 5: Avalie o preditor recém-obtido junto aos dados de teste.

Passo 6: Calcule as métricas de avaliação vista em sala de aula.

O aluno deve avaliar qual a melhor configuração de rede neural para o problema de predição um passo à frente e cinco passos a frente.