**PROGRAMAS**

**MANIPULAÇÃO DE DADOS DO CAMPO DE OBSERVAÇÃO**

function [entrada2,saida2] = prepara\_data2(entrada,saida)

periodos=4;

[entrada, saida] = remove\_linha2(entrada,saida);

entrada2 = separa\_periodos\_passado(entrada,periodos);

saida2 = separa\_periodos\_futuro(saida,periodos);

end

function [y] = separa\_periodos\_passado(x,periodos)

[m,n] = size(x);

a=0;

for i = 1:1:m-(2\*(periodos-1))

for f = 1:1:n

for s = 0:1:(periodos-1)

z(s+1+(f-1)\*4)=x(i+s,f);

end

end

if a==0

y=z;

a=1;

else

y= [y;z];

end

end

end

function [y] = separa\_periodos\_futuro(x,periodos)

[m,n] = size(x);

a=0;

for i = periodos:1:m-(periodos-1)

for f = 1:1:n

for s = 0:1:(periodos-1)

z(s+1+(f-1)\*4)=x(i+s,f);

end

end

if a==0

y=z;

a=1;

else

y= [y;z];

end

function [entrada2, saida2]=remove\_linha2(entrada,saida)

M=[entrada saida];

[m,n] = size(M);

for i=1:1:m

for j=1:1:n

if abs(M(i,j))> 80

M(i,j)=M(i-1,j)

end

end

end

entrada2 = M(:,(1:end-1));

saida2=M(:,end);

end

**MANIPULAÇÃO DADOS FORECAST NAM**

function data = open\_files()

data\_inicial=datenum(2014,6,2);

data\_final=datenum(2014,7,2);

MANIPULAÇÃO DE DADOS DE PREVISÃO DO TEMPO

delta=datenum(0,0,1);

i=0;

for nw=data\_inicial:delta:data\_final

i=i+1;

completa = str2double(datestr(nw,'yyyymmdd'));

ano\_mes= str2double(datestr(nw,'yyyymm'));

url=sprintf('http://nomads.ncdc.noaa.gov/dods/NCEP\_NAM/%i/%i/nam\_218\_%i\_0000\_fff',ano\_mes,completa,completa)

data(i,:) = ncread(url,'apcpsfc',[678,216,21],[1,1,8]);

end

data= sum(data,2);

end

%cat(2,'redes\_treinadas\',(sprintf('network\_%i\_%i\_%i\_%i.mat',topologia,t,comb\_input,fcn))

function data = open\_files\_NAM()

data\_inicial=busca\_data\_NAM(2012,7,2);

data\_final=busca\_data\_NAM(2013,2,2);

delta = data\_final-data\_inicial;

url=sprintf('http://nomads.ncdc.noaa.gov/dods/NCEP\_NAM\_ANALYSIS/6hr\_Pcp');

data(:,:) = ncread(url,'apcpsfc',[678,216,data\_inicial],[1,1,delta]);

%data= sum(data,2);

end

function [completa, ano\_mes]= loop\_de\_datas

data\_inicial=datenum(2012,1,2);

data\_final=datenum(2014,3,5);

delta=datenum(0,0,1);

for nw=data\_inicial:delta:data\_final

completa = datestr(nw,'yyyymmdd');

ano\_mes= datestr(nw,'yyyymm');

end

en

**RFDE NEURAL**

mat\_acertos=[];

for comb\_input=1:16

for topologia = 1:number\_max\_neuron

for t = 1:number\_max\_train

for fcn = 1:4

if exist(cat(2,'redes\_treinadas\',(sprintf('network\_%i\_%i\_%i\_%i.mat',topologia,t,comb\_input,fcn))),'file')

load (cat(2,'redes\_treinadas\',(sprintf('network\_%i\_%i\_%i\_%i.mat',topologia,t,comb\_input,fcn))));

train\_set\_mod=cell2mat(train\_set(comb\_input));

test\_set=[];

target\_test=[];

for i=tr.testInd

test\_set=horzcat(test\_set,train\_set\_mod(:,i));

target\_test=horzcat(target\_test,target(:,i));

end

test\_out = net(test\_set);

for i=1:size(test\_out,2)

if test\_out(i)>=0.5

test\_out(i)=1;

else

test\_out(i)=0;

end

end

acerto=0;

for i=1:size(test\_out,2)

if target\_test(i)==test\_out(i)

acerto=acerto+1;

end

end

acerto=acerto/i;

all\_out = net(train\_set\_mod);

for i=1:size(all\_out,2)

if all\_out(i)>=0.5

all\_out(i)=1;

else

all\_out(i)=0;

end

end

acerto2=0;

for i=1:size(target,2)

if target(i)==all\_out(i)

acerto2=acerto2+1;

end

end

acerto2=acerto2/i;

nova\_linha = [{'network\_%i\_%i\_%i\_%i.mat'}; acerto; acerto2];

mat\_acertos=horzcat(mat\_acertos, nova\_linha);

end

end

end

end

end

end

close all

clear all

%adicionando pasta de funcoes

addpath('functions')

% Carregando dados

load('data.mat');

%Definindo constantes

number\_max\_neuron=20;

number\_max\_train=10;

%Criando a rede neural

for comb\_input=1:16

for topologia = 1:number\_max\_neuron

for t = 1:number\_max\_train

%treinando com 'trainlm'

net=set\_net(topologia,'trainlm',size(cell2mat(train\_set(comb\_input)),1));

[net tr]=train(net,cell2mat(train\_set(comb\_input)),target);

sprintf('rede treinada : treinamento - trainlm; number of neuron - %i; iteração %i',topologia,t)

save (cat(2,'redes\_treinadas\',(sprintf('network\_%i\_%i\_%i\_1',topologia,t,comb\_input))));

%treinando com 'trainrp'

net=set\_net(topologia,'trainrp',size(cell2mat(train\_set(comb\_input)),1));

[net tr]=train(net,cell2mat(train\_set(comb\_input)),target);

sprintf('rede treinada : treinamento - trainrp; number of neuron - %i; iteração %i',topologia,t)

save (cat(2,'redes\_treinadas\',(sprintf('network\_%i\_%i\_%i\_2',topologia,t,comb\_input))));

%treinando com 'traingdx'

net=set\_net(topologia,'traingdx',size(cell2mat(train\_set(comb\_input)),1));

[net tr]=train(net,cell2mat(train\_set(comb\_input)),target);

sprintf('rede treinada : treinamento - traingdx; number of neuron - %i; iteração %i',topologia,t)

save (cat(2,'redes\_treinadas\',(sprintf('network\_%i\_%i\_%i\_3',topologia,t,comb\_input))));

%treinando com 'trainbfg'

net=set\_net(topologia,'trainbfg',size(cell2mat(train\_set(comb\_input)),1));

[net tr]=train(net,cell2mat(train\_set(comb\_input)),target);

sprintf('rede treinada : treinamento - trainbfg; number of neuron - %i; iteração %i',topologia,t)

save (cat(2,'redes\_treinadas\',(sprintf('network\_%i\_%i\_%i\_4',topologia,t,comb\_input))));

end

end

end

function [ net ] = set\_net( neurons, net\_trainFcn, num\_input )

%[ net ] = new\_network( neurons, train, net\_delays )

% net - Rede criada

% neurons - Número de neurónis na camada escondida

% net\_trainFcn - Função de treinamento

% Função para criar uma Rede Neural para previsão da cotação da Petrobras

% Criando a rede

net = feedforwardnet(neurons,net\_trainFcn);

% Configurando funções de transferência e de treinamento

net.layers{1}.transferFcn='tansig';

net.layers{2}.transferFcn='tansig';

net.performFcn='mse';

% Configurando a inicialização da rede

net.initFcn='initlay';

net.layers{1}.initFcn='initnw';

net.layers{2}.initFcn='initnw';

net.inputs{1}.size=num\_input;

net.layers{2}.size=1;

%Inicializando rede - pesos e biases

net=init(net);

%Definindo parâmetros da rede

net.trainParam.lr = 0.01;

net.trainParam.max\_fail=100;

net.trainParam.mc = 0.99;

net.trainParam.min\_grad=1e-7;

net.trainParam.goal = 0;

net.trainParam.epochs = 400;

net.trainParam.show= 1;

end