

Métodos de Inteligencia Artificial

Reporte 12

IF698972

Josefina Esmeralda Arriaga Hernández

4 de mayo del 2017 Guadalajara, Jalisco

**Objetivo**

Comprender el funcionamiento de una red neuronal dinámica como predictor.

**Problema a resolver**

La diferencia entre una red neuronal y una red neuronal dinámica es el momento de recibir nueva información al finalizar la red (valor de salida), se ve alimentando de esto haciendo una mejor estimación siempre y cuando se logre una convergencia en el modelo, los pasos para crear éste modelo son los siguientes:

1. Descargar la serie de datos
2. Seleccionar los precios de cierre de los datos
3. Se declara las variables a entrenar
4. Se realiza la red neuronal
5. Se corre una simulación con los resultados
6. Se cierra el lazo de salida con el de entrada (nuevo valor que se va a introducir como dato de entrada)
7. Se remueve el rezago o se mueve el modelo a futuro (predicción)

**Código desarrollado**

%LIMPIEZA

clear all;

close all;

clc;

%modelo dinamico

%red neuronal dinamico es un modelo en el cual se tienen diferentes

%neuronas y en la salida se retroalimenta a la entrada

%tambien puede recibir otras entradas rezagadas

%% Cargar datos

load datos4.mat;

Y = IPCfinal(:,5)'; %valor del ipc final precio de cierre

U = BMV\_17final(:,5)';%valor final de una accion precio de cierre

%MATLAB requiere matrices de celdas para el paquete de redes din?micas

Yc = num2cell(Y);%convierte de matriz a matriz de celdas

Uc = num2cell(U);

%% Conjuntos de entrenamiento

%Separar el conjunto de entrenamiento y el de prueba

ndatos = round(0.9\*size(Y,2));

%datos de entrenamiento

%Para entrenar el modelo

Ytrain = Yc(:,1:ndatos);

Utrain = Uc(:,1:ndatos);

%datos de prueba

%Para probar el modelo qu? tan bueno fue

Ytest = Yc(:,ndatos+1:end);

Utest = Uc(:,ndatos+1:end);

%% Crear la red neuronal

nrez = 10; %N?mero de rezagos

noc = 10; %N?mero de neuronas en la capa oculta

red = narxnet(1:nrez,1:nrez,noc) %para redes din?micas

%El primer par?metro es el n?mero de rezagos de Y

%El segundo par?metro es el n?mero de rezagos de U

%1:nrez son las derivadas

[Us,Ui,Yi,Ys] = preparets(red,Utrain,{},Ytrain);

red = train(red,Us,Ys,Ui,Yi);

Yg\_train = red(Us,Ui,Yi);

figure(1);

plotresponse(Ys,Yg\_train);

%% Simulaci?n

[Us,Ui,Yi,Ys] = preparets(red,Utest,{},Ytest);

Yg\_test = red(Us,Ui,Yi);

figure(2);

plotresponse(Ys,Yg\_test);

%% Cerrar el lazo de salida a la entrada

redc = closeloop(red);

[Us,Ui,Yi,Ys] = preparets(redc,Utest,{},Ytest); %Preparar rezagos

Ygc\_test = redc(Us,Ui,Yi);%el de lazo cerrado

figure(3);

plotresponse(Ys,Ygc\_test);

%% Removiendo el rezago o moviendo el modelo a futuro

redcnr = removedelay(redc); %red de lazo cerrado sin rezago

[Us,Ui,Yi,Ys] = preparets(redcnr,Utest,{},Ytest); %Preparar rezagos

Ygcnr\_test = redcnr(Us,Ui,Yi);

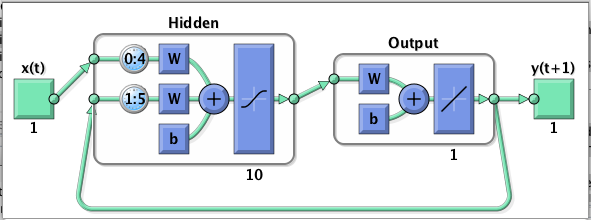
figure(4);

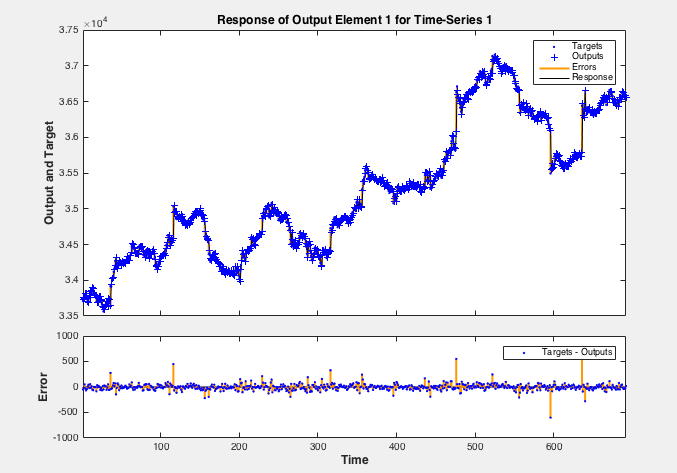
plotresponse(Ys,Ygcnr\_test);

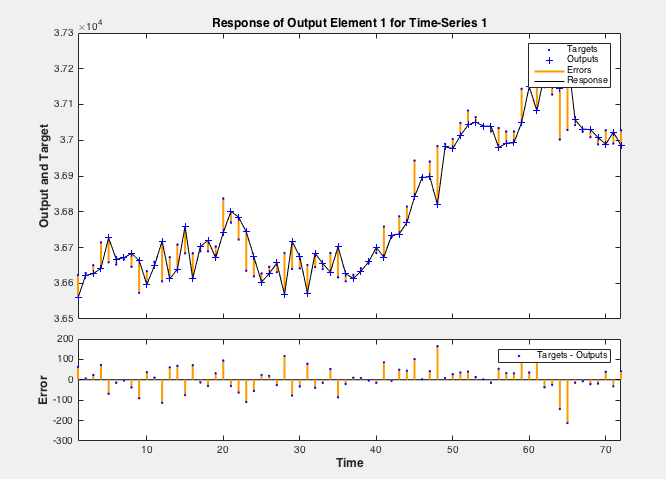
view(redcnr)

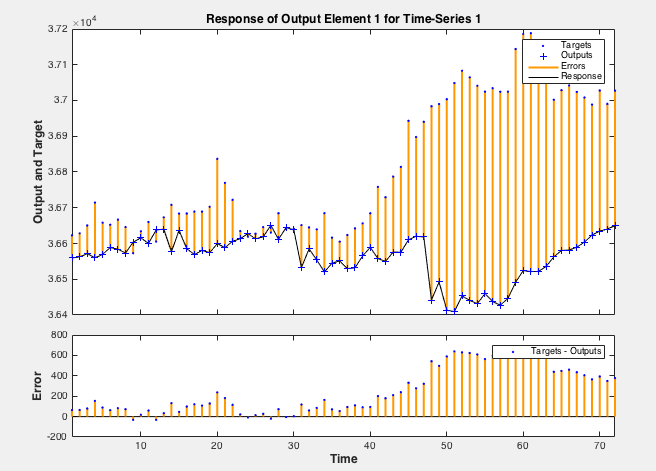
J = perform(redc,cell2mat(Ys),cell2mat(Ygc\_test))

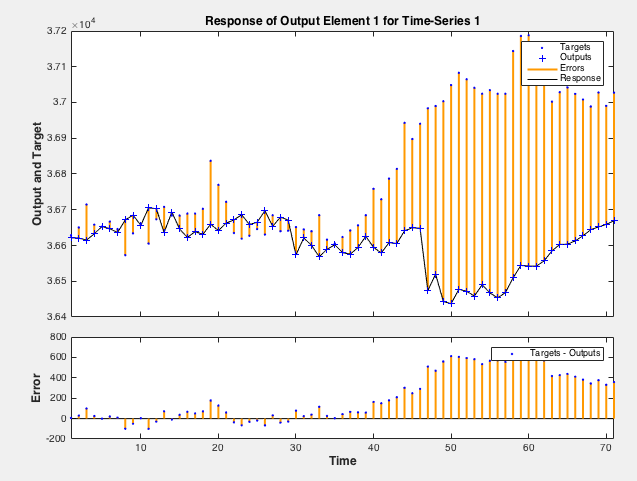
**Gráficos**

****

****

****

****

****

**Interpretación de gráficos**

El primero gráfico muestra la red neuronal con sus entradas, salidas, capas ocultas y en este caso la relación que se hace con el valor de salida con el valor de entrada. Las siguientes gráficas muestran los datos reales comparados con la estimación de la red neuronal, siendo el error o diferencia las líneas naranjas.

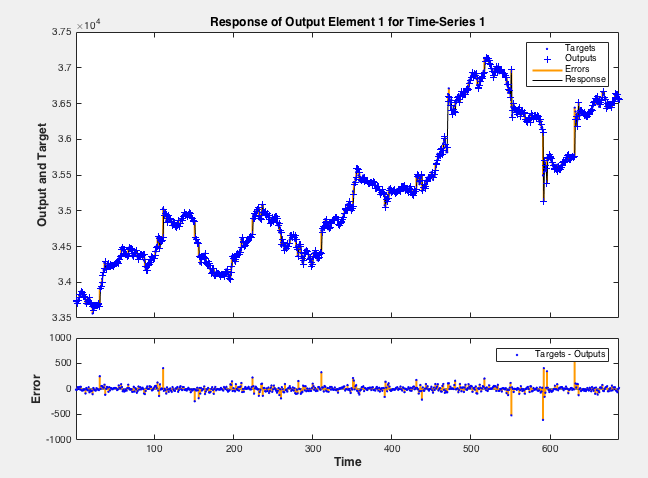
**Resultados**

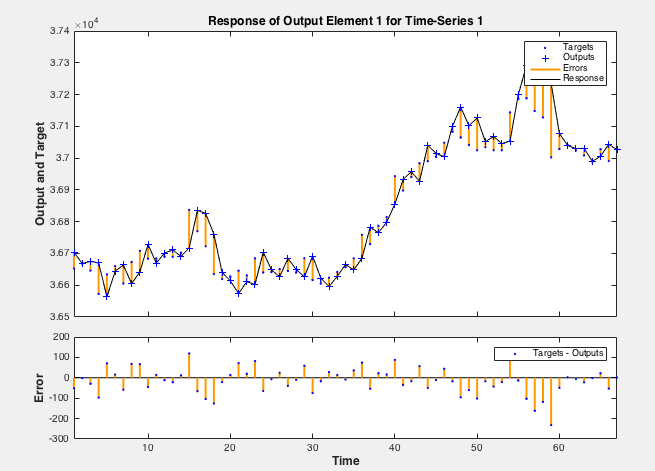
La red neuronal dinámico arroja una predicción de t+1, se observa una buena convergencia en el modelo ya que hay una variación mínima entre cada Y en las diferentes etapas de entrenamiento.

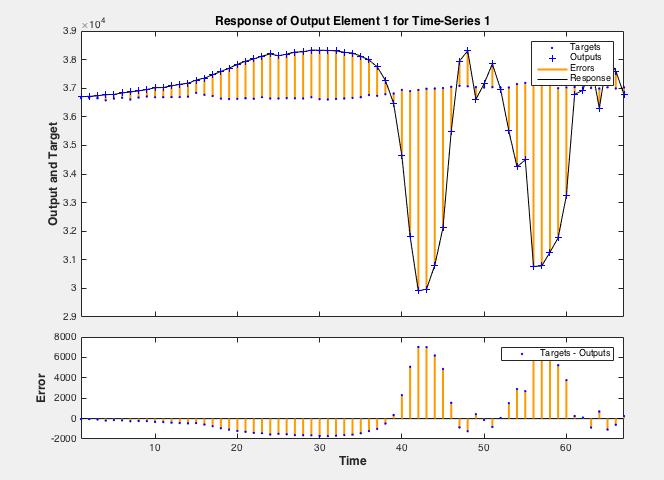
**Conclusiones**

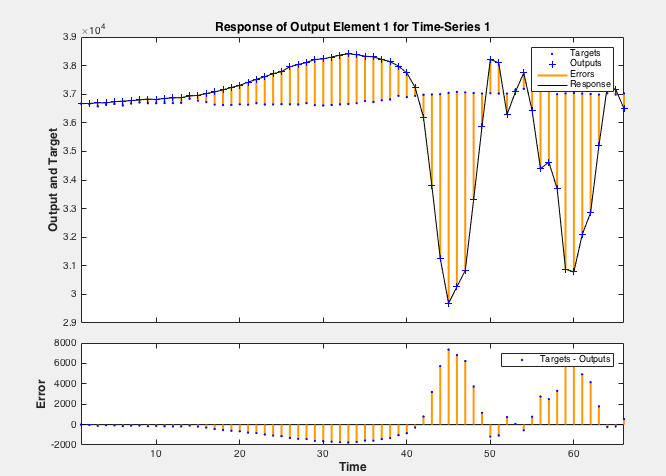
En conclusión, si se compara con las redes neuronales anteriores, se podría observar menos error en la red neuronal dinámica, es importante mencionar que la red se hizo con los precios de cierra del IPC con un activo de la BMV, siendo Y el IPC y U el activo, el entrenamiento se realizó con funciones de Matlab.

**Funcionamiento**

****

****

****

****