

Métodos de Inteligencia Artificial

Reporte 11

IF698972

Josefina Esmeralda Arriaga Hernández

18 de Abril del 2017 Guadalajara, Jalisco

**Objetivo**

Comprender el funcionamiento de perceptrón multicapa y sus diversos usos (regresión lineal, clasificación o predictor).

**Problema a resolver**

Con el perceptrón multicapa se realizarán dos ejemplos, un predictor y un clasificador. En el predictor se realizan los siguientes pasos:

1. Descargar la serie de datos
2. Seleccionar los precios de cierre de los datos
3. Se genera la matriz con los rezagos deseados
4. Se realiza el entrenamiento de la red neuronal con Levenberg-Marquart
5. Se grafica las salidas neuronales, siendo estas las gráficas predictores

Para el segundo ejemplo en donde se hace una clasificación de datos se realizan los siguientes pasos:

1. Descargar la serie de datos
2. Seleccionar los datos a entrenar
3. Se realiza el entrenamiento de la red neuronal con ‘trainrp’
4. Por último, se almacenan en vectores los datos que entraron en el grupo uno o grupo cero de acuerdo al entrenamiento

**Código desarrollado**

**Perceptrón multicapa predictor**

%% Limpieza

clear all;

close all;

clc;

%perceptron multicapa por cada neurona de salida un grupo y se puede tener

%muchas clases es supervisada

%% Cargar datos

%Se har? un predictor a n pasos

load datos4.mat;

data=IPCfinal(:,5);%precio de cierre en columna 5

nrez=6;%rezagos

temp=[];

for k=0:nrez

    temp(:,k+1)=data(nrez+1-k:end-k);%va generar 4 columnas 3 de rezago y el vector

end

Y=temp(:,1:4)';%aumentar numero de columnas en Y para tener mas capas

X=temp(:,5:end)';

ndatos=round(0.9\*size(Y,2));%seleccionar los datos de entrenamiento

Xtrain=X(:,1:ndatos);

Ytrain=Y(:,1:ndatos);

%% Creaci?n de red neuronal

red=feedforwardnet(10);

red.trainFcn='trainlm'; %Entrenamiento con Levenberg-Marquart

red=train(red,Xtrain,Ytrain);

Yg=red(X);%estimaci?n

%% Graficar las salidas

nsal=size(Y,1);%salida neuronal

for k=1:nsal

    subplot(nsal,1,k)

    plot(1:size(Y,2),Y(k,:),'-b',1:size(Yg,2),Yg(k,:),'r--')

    grid;

    title(['IPC(t+' num2str(nrez-k-1) ')' ])

end

**Perceptrón multicapa clasificador**

%% Limpieza

clear all;

close all;

clc;

%perceptron multicapa por cada neurona de salida un grupo y se puede tener

%muchas clases es supervisada

%% Cargar datos

%Se har? un clasficador

load data1.txt;

data=data1

Y=data(:,3)';%aumentar numero de columnas en Y para tener mas capas

X=data(:,1:2)';

ndatos=round(0.9\*size(Y,2));%seleccionar los datos de entrenamiento

Xtrain=X(:,1:ndatos);

Ytrain=Y(:,1:ndatos);

%% Creaci?n de red neuronal

red=feedforwardnet(10);

%red.trainFcn='trainscg';

red.trainFcn='trainrp';

red=train(red,Xtrain,Ytrain);

Yg=red(X);%estimaci?n

%% Grupos

G0=data(data(:,3)== 0,1:2); %Grupo cero

G1=data(data(:,3)== 1,1:2); %Grupo uno

Ygr=(Yg>.5)',1:2;

G0p=data((Yg<=.5)',1:2);

G1p=data((Yg>.5)',1:2);

plot(G0(:,1),G0(:,2),'bo',G1(:,1),G1(:,2),'rx');

hold on

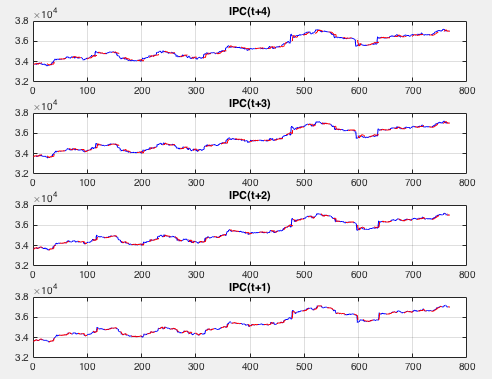
plot(G0p(:,1),G0p(:,2),'go',G1p(:,1),G1p(:,2),'kx');

hold off

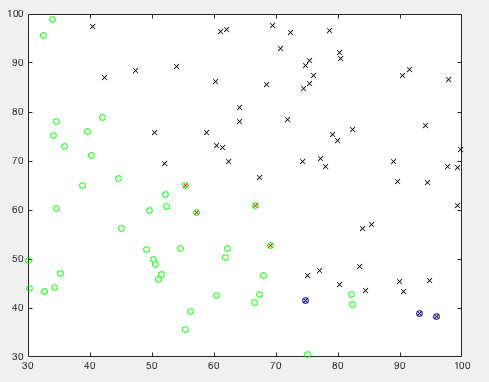
%red([1;10])

**Gráficos**

**Perceptrón multicapa predictor**

****

**Perceptrón multicapa clasificador**

****

**Interpretación de gráficos**

**Perceptrón multicapa predictor**

Se observa 4 gráficas que equivalen a la predicción, en cada uno se predice un tiempo extra (t+1) siendo esto del precio de cierre de IPC, la línea azul son los datos y la línea roja arroja la última predicción.

**Perceptrón multicapa clasificador**

Se observa la serie de datos del grupo uno y los datos del grupo cero y marcado en X son los datos que se salieron de su respectivo grupo en este caso son 7 datos que no se clasificaron de manera correcta.

**Resultados**

**Perceptrón multicapa predictor**

Teniendo 6 rezagos se obtuvieron 4 salidas neuronales en donde se observa el comportamiento real del precio de cierre de IPC y al mismo tiempo se observa el comportamiento esperado (predicción) que se realizó con la red neuronal arrojando un último tiempo (t+1) donde se predice en base a la información anterior.

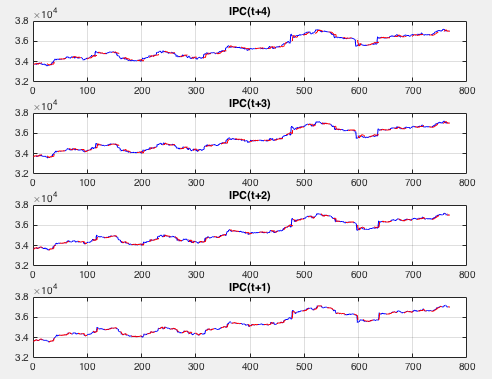
**Perceptrón multicapa clasificador**

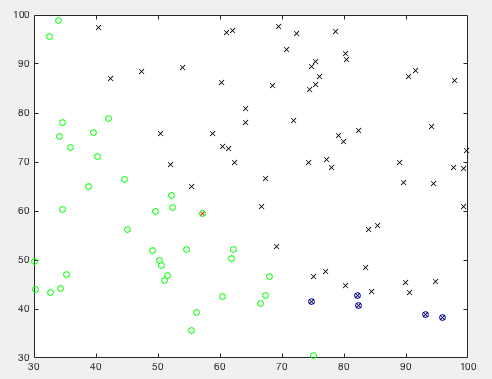
Al correr el código se observa que su error de datos no clasificados de manera correcta es de máximo siete. Esto quiere decir que no hay un error significativo como para cambiar el entrenamiento. El número de neuronas en este caso es de diez.

**Conclusiones**

En conclusión, en ambos casos se considera que existió una buena aproximación ya sea en el predictor como el clasificador. Ya que no era la misma serie de datos no se puede comparar entre sí los dos códigos, en uno se predijo el IPC y en el otro se clasificó una serie de datos; en ambos casos se realizaron con las redes de entrenamiento de Matlab.

**Funcionamiento**

****

****