

## **MACHINE LEARNING**

Se trata de una de las tecnologías facilitadoras más relevantes pues gracias a ella se puede desarrollar analítica avanzada de datos, incluso sobre grandes cantidades de datos (big data). De hecho, en general las técnicas de machine learning funcionan mejor si existen grandes cantidades de datos disponibles.

### **Enunciado**

En este caso práctico, se ha tratado de desarrollar unos programas que permitan la carga de diferentes series temporales (se adjuntan algunas como ejemplo), y a partir de sus datos, se pueda realizar una predicción a un cierto horizonte de futuro. Primero se pueden probar algunos métodos clásicos como el de persistencia o el de medias móviles, pero también se puede realizar una predicción basada en regresión múltiple de una serie de valores en el pasado (una ventana del pasado) e igualmente probar con una shallow Artificial Neural Network, cambiando el número de neuronas en la única capa oculta. Los resultados de la predicción se presentan para una semana de test (con datos que no se han usado para el diseño o entrenamiento) de una forma gráfica, así como sus valores de R y MSE.

### **Implementación**

`p_main.m`: es el script principal que va realizando llamadas a otras funciones en función de lo que el usuario del programa vaya seleccionando.

El resto de funciones son:

- `f_cargar_datos.m`
- `f_persistencia.m`
- `f_medias_moviles.m`
- `f_regresion_redes.m`
- `f_menu.m`
- `f_representar.m`

Los datos que pueden usarse sin modificar el programa son:

- `NO2.mat`
- `PM10.mat`
- `Velocidad_viento.mat`

```

%-----
% Programa para predicción usando redes neuronales, regresión lineal,
% medias móviles y persistencia
%-----

% Limpieza de variables.
close all;
clear all;

% Mostramos el menú con las opciones
opcion = f_menu();

% Opción 5 para salir
while opcion ~= 5

    % Se presenta un menú y se cargan los datos de la serie temporal
    [datosTrain, datosTest, tipoSerie] = f_cargar_datos();

    % Se llama a la función correspondiente según la opción elegida por el
    % usuario
    switch opcion

        case 1 % Persistencia

            % Llamada a la función de persistencia para calcular la predicción
            predicción = f_persistencia(datosTrain, datosTest);

        case 2 % Medias móviles

            predicción = f_medias_moviles(datosTrain, datosTest);

        case 3 % Regresión múltiple lineal

            tipoMetodo = 1;
            [predicción, out_test] = f_regresion_redes(datosTrain, datosTest,
            tipoMetodo);

        case 4 % Redes neuronales

            tipoMetodo = 2;
            [predicción, out_test] = f_regresion_redes(datosTrain, datosTest,
            tipoMetodo);
    end

    % Representamos
    if opcion == 1 || opcion == 2
        datosT = datosTest;
    else
        datosT = out_test; % Para redes y regresión el test cambia por el
        % proceso de la geog
    end

    f_representar(datosT, predicción, tipoSerie);

    % Ponemos una pausa
    fprintf('\n')

```

```
disp('Pulse una tecla para volver al menú principal...');
pause;

% Mostramos el menú con las opciones
opcion = f_menu();

end

close all; % Cerramos los gráficos
fprintf('\nEl programa ha finalizado.');
```

```
-----
-- Programa para predicción de series temporales --
-----

Seleccione el método de predicción:

[1] Persistencia
[2] Medias móviles
[3] Regresión lineal
[4] Redes neuronales
[5] Salir

Opcion: |
```

```
-----
-- Programa para predicción de series temporales --
-----

Seleccione serie temporal:

[1] NO2
[2] Velocidad de Viento
[3] Materia particulada PM10

Opcion: 1|
```

*Published with MATLAB® R2018b*

```

%-----
% Función para presentar un menú con los diferentes métodos de predicción
% para que elija el usuario.
%
% Entrada: Ninguna.
% Salidas: Ninguna.
%-----
function valor = f_menu()

clc;

disp('-----');
disp('-- Programa para predicción de series temporales --');
disp('-----');

valor = input('\nSeleccione el método de predicción: \n\n [1] Persistencia\n [2] Medias
móviles\n [3] Regresión lineal\n [4] Redes neuronales\n [5] Salir\n\nOpción: ');

% El bucle se ejecuta hasta que se elija una serie o salga
while valor ~= 1 && valor ~= 2 && valor ~= 3 && valor ~= 4 && valor ~= 5

    valor = input('Opción: ');

end

```

*Published with MATLAB® R2018b*

```

%-----
% Función para presentar un menú con las opciones del programa
%
% Entradas:
%   Ninguna.
% Salidas:
%   datos : Los datos de la serie temporal.
%   tipo  : El tipo de serie temporal (1 no2, 2 viento, 3 PM10)
%-----

function [datosTrain , datosTest, tipo] = f_cargar_datos()

clc;

disp('-----');
disp('-- Programa para predicción de series temporales --');
disp('-----');

% Tipo de serie temporal
tipo = input('\nSelecione serie temporal:\n\n [1] NO2\n [2] Velocidad de Viento\n [3]
Materia particulada PM10\n\nOpcion: ');

while tipo ~= 1 && tipo ~= 2 && tipo ~= 3
    tipo = input('Opcion: ');
end

% Cambiamos al directorio datos
dire = pwd;
dirDatos = strcat(dire, '\datos');
try
    cd (dirDatos);
catch
    % Si no existe, lo creamos
    mkdir(dirDatos);
    cd (dirDatos);
end

% Cargamos los datos
switch tipo
case 1
    load('no2.mat', 'datos_no2');
    datosTrain = datos_no2(1:35040,1); % Datos horarios. 4 años de training
    datosTest = datos_no2(35041:end,1); % Datos horarios. El esto para test
    tipo = 1; % Los datos de las ventas de cosas son mensuales
case 2
    load('velocidad_viento.mat', 'datos_viento');
    datosTrain = datos_viento(1:35040,1); % Datos horarios. 4 años de training
% para training
    datosTest = datos_viento(35041:end,1); % Datos horarios. El esto para test
% para test
    tipo = 2; % Los datos de viento son horarios
case 3
    load('pm10.mat', 'datos_pm10');
    datosTrain = datos_pm10(1:35040,1); % Datos horarios. 4 años de training
% para training
    datosTest = datos_pm10(35041:end,1); % Datos horarios. El esto para test

```

```
        tipo = 3; % Los datos de materia particulada son horarios
end

% Volvemos al directorio principal
cd (dire);
```

*Published with MATLAB® R2018b*

```

%-----
% Función para calcular predicciones usando persistencia
%
% Entradas:
%     datoTrain: Los datos de training de la serie temporal.
%     datosTest: Los datos de test de la serie temporal.
%     salto     : Salto en el pasado para asociar a t+1 (por ejemplo, en
%                 una serie temporal diaria, salto = 7 indica que el valor
%                 de t+1 va a ser el mismo que el valor de 7 días antes.
% % Salidas:
%     predicción: La Predicción calculada.
%-----

funcion [predicción] = f_persistencia(datosTrain, datosTest)

% Pedimos el salto en el pasado al usuario.
% Salto en el pasado para asociar a t+1 (por ejemplo, en una serie
% temporal diaria, salto = 7 indica que el valor de t+1 va a ser
% el mismo que el valor de 7 días antes).
fprintf('\n');
salto = input('Introduzca valor de salto en el pasado para asociar a t+1 (valores
posibles 1 a 35040): ');
while salto < 1 || salto > 35040
    salto = input('Introduzca valor de salto en el pasado para asociar a valores futuros
(valores posibles 1 a 35040): ');
end

% Al valor de t+1 le corresponde t+1-salto. Para t+2 será t+2 - salto,
% y así sucesivamente. Ejemplo: Salto 1, para t+1 le corresponde t.
predicción = zeros(length(datosTest), 1);
ind = length(datosTrain) + 1; % Índice del valor en t + 1
datosTodos = vertcat(datosTrain, datosTest);
indice = ind - salto;

for i = 1: length(datosTest)

    predicción(i, 1) = datosTodos(indice);

    % Para que no llegue a índices no válidos
    if i ~= length(datosTest)
        indice = indice + 1;
    end

end

end

% Calculamos el RMSE y el R:
%-----
[~, ~, r] = postreg(datosTest, predicción);
MSE = f_performindex(datosTest, predicción);
close all; %Cerramos gráfico regresión

fprintf('\n')
disp('*****');
fprintf('Calculos completados para salto: %i. \n', salto);
fprintf('Valor de R: %f \n', r);
fprintf('Valor de MSE: %f \n', MSE);
disp('*****');

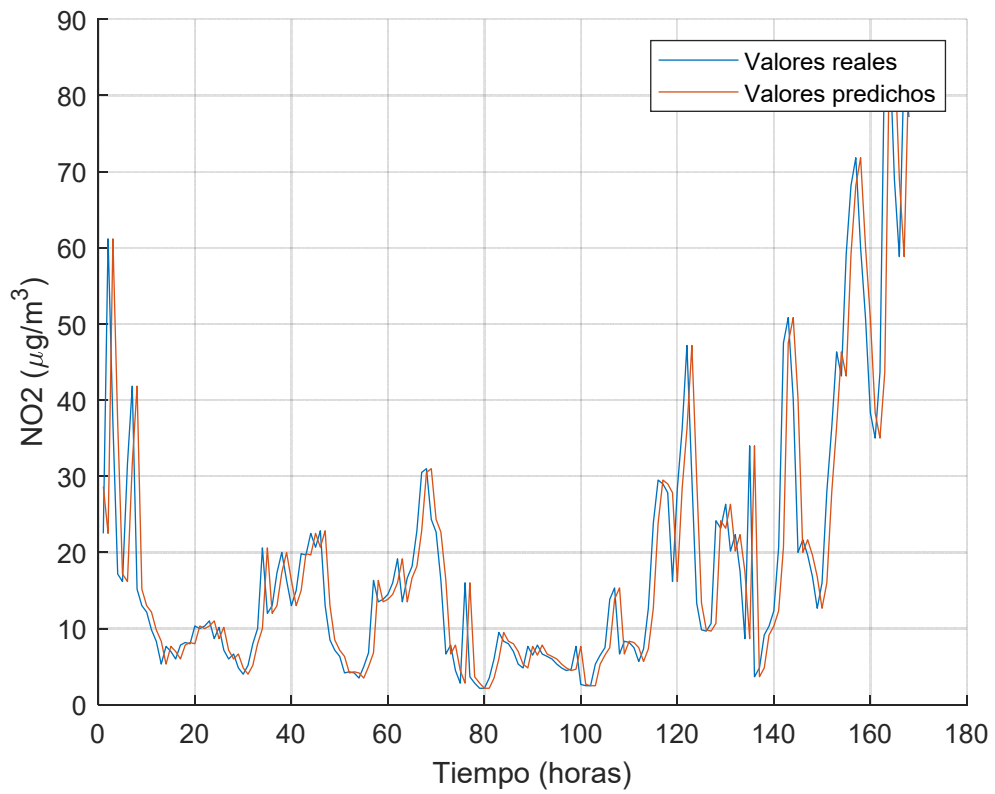
```

end

```
Introduzca valor de salto en el pasado para asociar a t+1 (valores posibles 1 a 35040): 1
*****
Calculos completados para salto: 1.
Valor de R: 0.861847
Valor de MSE: 147.505049
*****

Se procede a representar una semana de los datos de test (valores reales vs. predichos).

Pulse una tecla para volver al menú principal...
```



Published with MATLAB® R2018b



```

%-----
% Función para calcular predicciones usando medias móviles
%
%
% Entradas:
%     datoTrain : Los datos de training de la serie temporal.
%     datosTest : Los datos de test de la serie temporal.
% Salidas:
%     predicción: La Predicción calculada.
%-----

function [predicción] = f_medias_moviles(datosTrain, datosTest)

% Pedimos al usuario el tamaño de la ventana
fprintf('\n');
tamVentana = input('Introduzca el número de valores del pasado que se usarán para la
media móvil (valores posibles 2 a 35040): ');
while tamVentana < 2 || tamVentana > 35040
    tamVentana = input('Introduzca el número de valores del pasado que se usarán para la
media móvil: (valores posibles 2 a 35040): ');
end

predicción = zeros(length(datosTest), 1);

datosTodos = vertcat(datosTrain, datosTest);
ind_sup = length(datosTrain); % La primera vez se corresponde con
% el índice de t, o sea, el último valor de training
ind_inf = ind_sup + 1 - tamVentana; % Primera posición para calcular
% la media móvil. Desde la posición de t+1, vamos al pasado tantas
% posiciones como indique tamVentana

for i = 1:length(datosTest)

    predicción(i, 1) = mean(datosTodos(ind_inf: ind_sup, 1));

    % Para que no llegue a índices no válidos
    if i ~= length(datosTest)
        ind_inf = ind_inf + 1;
        ind_sup = ind_sup + 1;
    end

end

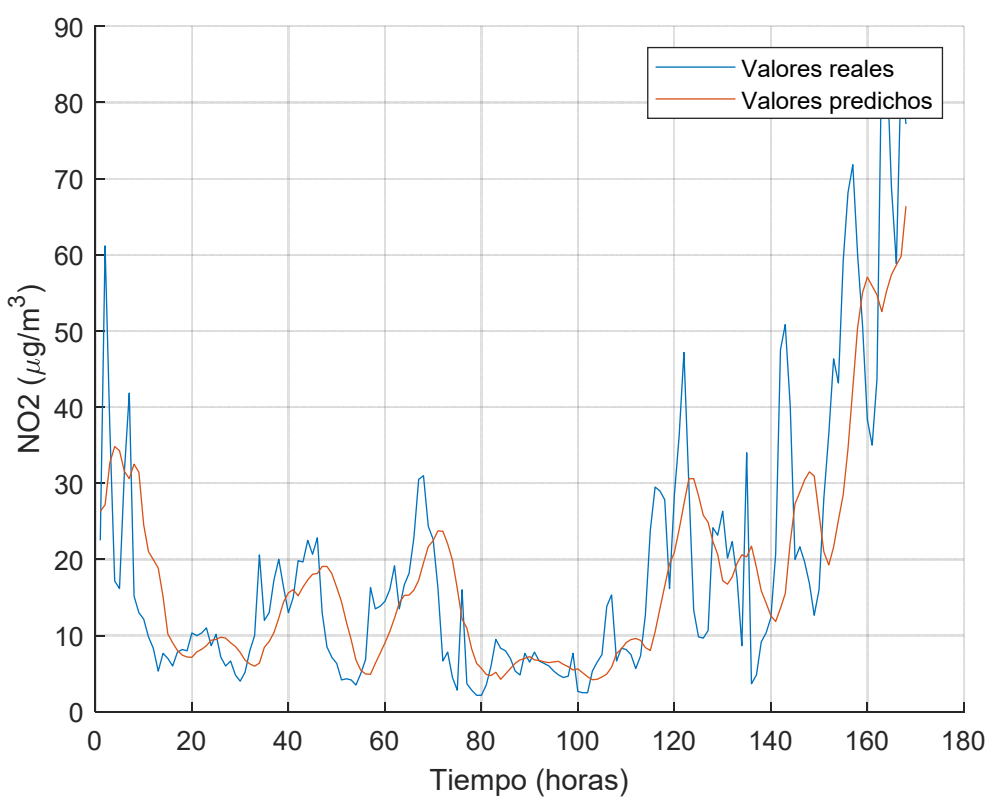
% Calculamos el RMSE y el R:
%-----
[~, ~, r] = postreg(datosTest, predicción);
MSE = f_performindex(datosTest, predicción);
close all; %Cerramos gráfico regresión

fprintf('\n')
disp('*****');
fprintf('Calculos completados media móvil de %i valores del pasado. \n', tamVentana);
fprintf('Valor de R: %f \n', r);
fprintf('Valor de MSE: %f \n', MSE);
disp('*****');

end

```

```
Introduzca el número de valores del pasado que se usarán para la media móvil (valores posibles 2 a 35040): 7
*****
Calculos completados media movil de 7 valores del pasado.
Valor de R: 0.694661
Valor de MSE: 290.029829
*****
Se procede a representar una semana de los datos de test (valores reales vs. predichos).
Pulse una tecla para volver al menú principal...
```



Published with MATLAB® R2018b

```

%-----
% Función para calcular predicciones usando regresión múltiple lineal o
% redes neuronales.
%
% Entradas:
%     datoTrain : Los datos de training de la serie temporal.
%     datosTest : Los datos de test de la serie temporal.
%     tipoMetodo: 1 para regresión, 2 para redes neuronales.
% Salidas:
%     predicción: La Predicción calculada.
%     out_test  : Los datos de test según el proceso de lagueo.
%-----
funcion [predicción, out_test] = f_regresion_redes(datoTrain, datosTest, tipoMetodo)

%-----
% Autocorrelación:
%-----
disp('Se muestra el gráfico de autocorrelación:');
figure
autocorr(datoTrain, 'NumLags', 336, 'NumSTD', 3);

%-----
% Lagueo de variables:
%-----
salto = input('Introduzca valor de salto en el pasado: ');
while salto < 1 || salto > 720
    salto = input('Introduzca valor de salto en el pasado: ');
end

% Pedimos al usuario el tamaño de la ventana
tamVentana = input('Introduzca el tamaño de ventana: ');
while tamVentana < 1 || tamVentana > 720 % Un mes como tope
    tamVentana = input('Introduzca el número de valores del pasado que se usarán: ');
end

% Para poder comparar con los persistencia y medias móviles, se fija el
% hp en 1 hora
hp = 1;

% % Pedimos al usuario el horizonte de predicción
%     hp = input('Introduzca el horizonte de predicción: (valores posibles 1 a 168
horas): ');
%     while hp < 1 || hp > 12
%         hp = input('Introduzca el horizonte de predicción: (valores posibles 1 a 168
horas): ');
%     end

% Lagueamos todas las de ozono (t, t-1, ...)
varLagTrain = f_regrewindow(datoTrain, tamVentana, salto, hp);
varLagTest  = f_regrewindow(datosTest, tamVentana, salto, hp);

% Selección de variables con ramdonForest

inputs  = varLagTrain(:, 1:end-1);
targets = varLagTrain(:, end);

```

```

in_test = varLagTest(:, 1:end-1);
out_test = varLagTest(:, end);

%-----
% Métodos de predicción:
%-----

if tipoMetodo == 1 % Regresión múltiple lineal

    md1 = fitlm(inputs, targets, 'linear');
    predicción = predict(md1, in_test);

else % Redes neuronales

    % Pedimos al usuario el número de neuronas que quiere usar Definimos el número de
    neuronas en capa oculta
    neuronas = input('Introduzca el número de neuronas en la capa oculta a usar (valores
    posibles 1 a 50): ');
    while neuronas < 1 || neuronas > 50
        neuronas = input('Introduzca el número de neuronas en la capa oculta a usar
        (valores posibles 1 a 50): ');
    end

    % Con optimización Bayesiana
    net = fitnet(neuronas, 'trainr');

    % Early Stopping.
    % Set up Division of Data for Training, Validation, Testing.
    net.divideParam.trainRatio = 70/100;
    net.divideParam.valRatio = 15/100;
    net.divideParam.testRatio = 15/100;

    % Ocultar la ventana
    net.trainParam.showWindow = 0;
    % Entrenamiento de la red.
    [net, ~] = train(net, inputs', targets');

    % Test de la red. Valores obtenidos para la entrada de test.
    predicción = net(in_test');
    predicción = predicción';

end

% Calculamos el RMSE y el R:
%-----
[~, ~, r] = postreg(out_test, predicción);
MSE = f_performance(out_test, predicción);
close all; %Cerramos gráfico regresión

if tipoMetodo == 1
    mensaje = 'Cálculos completados para regresión múltiple lineal.';
else
    mensaje = 'Cálculos completados para redes neuronales.';
end

fprintf('\n')
disp('*****');

```

```

di sp(mensaje);
fprintf('Valor de R: %f \n', r);
fprintf('Valor de MSE: %f \n', MSE);
di sp('*****');

end

```

### Regresión Múltiple Lineal

```

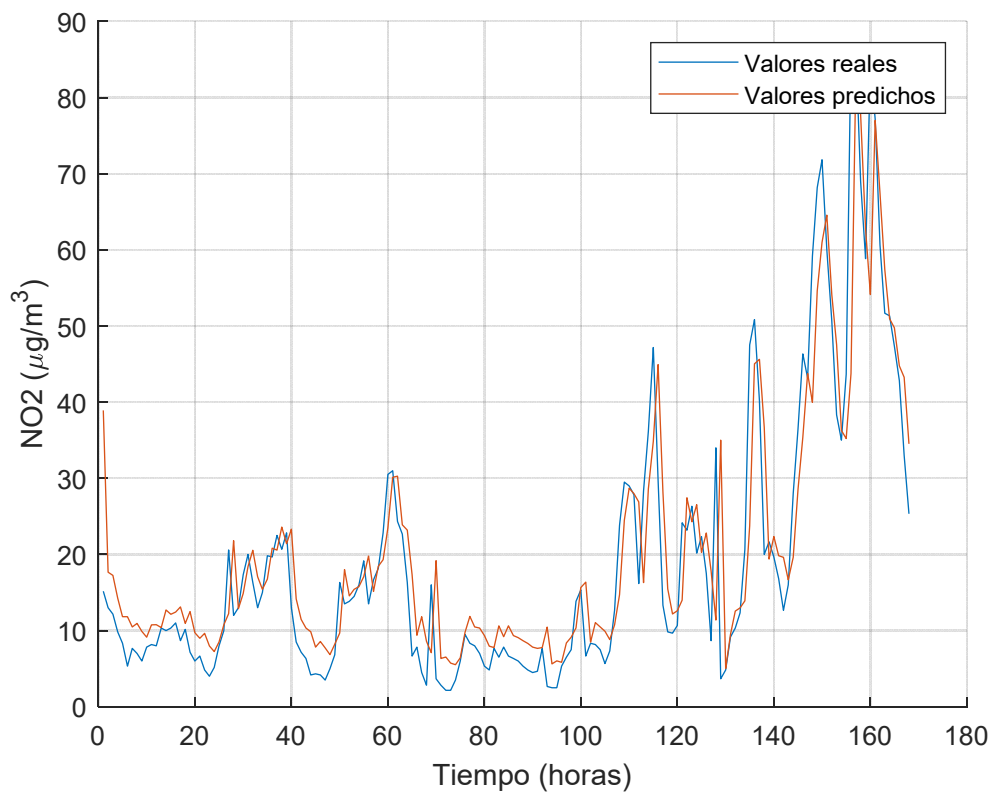
Introduzca valor de salto en el pasado: 1
Introduzca el tamaño de ventana: 7

*****
Calculos completados para regresión múltiple lineal.
Valor de R: 0.864455
Valor de MSE: 135.004670
*****

Se procede a representar una semana de los datos de test (valores reales vs. predichos).

Pulse una tecla para volver al menú principal...

```



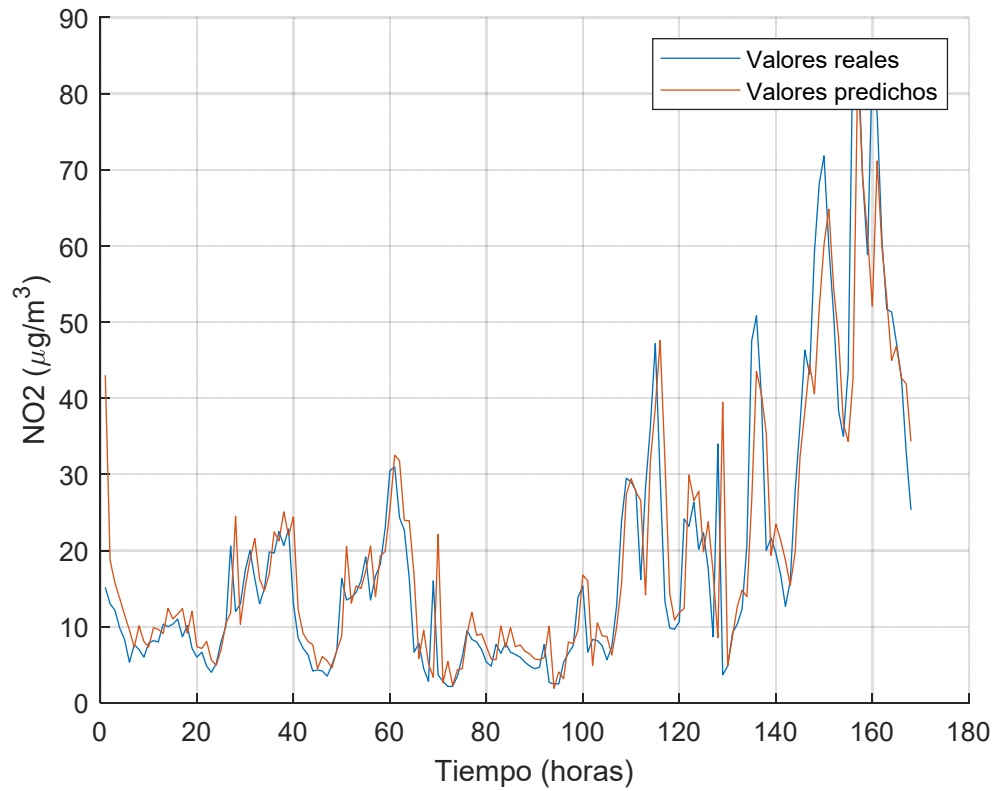
## Redes Neuronales Artificiales

```
Introduzca valor de salto en el pasado: 1
Introduzca el tamaño de ventana: 7
Introduzca el número de neuronas en la capa oculta a usar (valores posibles 1 a 50): 10

*****
Calculos completados para redes neuronales.
Valor de R: 0.867990
Valor de MSE: 131.862639
*****

Se procede a representar una semana de los datos de test (valores reales vs. predichos).

Pulse una tecla para volver al menú principal...
```



```

%-----
% Función para representar gráficamente los datos
%
%
% Entrada:
%     datosTraining.
%     datosTest.
%     tipoSerie : Según la serie, se establece propiedades de los
%                 gráficos.
% Salidas: Ninguna.
%-----
function f_representar(test, predicción, tipo_serie)

% Representamos los valores predichos vs. los valores reales:
%-----

% Gráfica
figure;

inicio = 1;

fprintf('\n');
disp('Se procede a representar una semana de los datos de test (valores reales vs.
predichos).');
test = test(1:168,1); % Representamos una semana
predicción = predicción(1:168,1);
fin = length(test);
hold on, plot(inicio:fin, test(1:end));
xlabel('Tiempo (horas)')

if tipo_serie == 1
    ylabel('NO2 (\u00b5g/m^3)') % Eje y (Microgramos m3)
elseif tipo_serie == 2
    ylabel('Velocidad del viento (m/s)') % Eje y (Microgramos m3)
else
    ylabel('PM10 (\u00b5g/m^3)') % Eje y (Microgramos m3)
end

hold on, plot(inicio:fin, predicción(1:end));
legend('Valores reales', 'Valores predichos')

grid on;

end

```