

Apéndice F

Código del modelo de RNA diseñado

En el apéndice F se presenta el código desarrollado, resultado de las diversas estructuras implementadas a lo largo de la investigación. Este código permite concretar un modelo de RNA cuya configuración interna fue definida por el autor tras ejecutar los procesos de entrenamiento, validación y prueba, utilizando bases de datos experimentales obtenidas de la literatura sobre el flujo bifásico de aceite y agua en tuberías horizontales.

El código es autoexplicativo y fácilmente adaptable a las necesidades del usuario. Contiene secciones descriptivas que explican su funcionamiento y ofrecen sugerencias para que el usuario pueda modificar características topológicas, como las funciones de entrenamiento, funciones de rendimiento, funciones de aprendizaje, número de capas ocultas, número de neuronas en las capas ocultas y la división porcentual de la base de datos para las fases de entrenamiento, validación y prueba. Además, el usuario puede seleccionar las gráficas que desee generar.

El código generado es:

```
%%=====
%%===== Modelo RNA para determinar patrones de flujo de un fluido bifásico de aceite y agua en tubería horizontal=====
%% Los parámetros usados como entradas deben estar organizados en vectores fila y las salidas deben estar organizadas
%% en vectores columna, una para cada variable
%% Las matrices numéricas de entrada y salida son:
% INPUT_RNA - Datos de entrada
% OUTPUT_RNA - Datos de salida
x = INPUT_RNA;
t = OUTPUT_RNA;

% Se procede con la selección de la función de entrenamiento trainFcn:
% Para ver una lista de todas las funciones de entrenamiento escribir: help ntrain
% trainscg : 'Scaled Conjugate Gradient'
% trainbfg : 'BFGS Quasi-Newton'
% trainrp : 'Resilient Backpropagation'
% traincgp : 'Polak Ribiere conjugate gradients'
% traincgb : 'Conjugate gradient with powell/beale restarts'
trainFcn = 'trainrp'

%%=====
%%===== Estructuración de la red neuronal=====
%%=====
%% A continuación se realiza la definición del número de capas ocultas y neuronas que conformarán cada capa oculta.
%% Se presentan las líneas de código para seleccionar hasta 15 capas ocultas y digitar la cantidad de neuronas que se deseen
%% implementar en cada capa, para esto se debe descomentar las líneas de cada capa oculta y digitar el número de neuronas.
%% En el código actual se seleccionan 3 capas ocultas y cada una con 50 neuronas.

hiddenLayer1Size = 50;
hiddenLayer2Size = 50;
hiddenLayer3Size = 50;
% hiddenLayer4Size = XX;
% hiddenLayer5Size = XX;
% hiddenLayer6Size = XX;
% hiddenLayer7Size = XX;
% hiddenLayer8Size = XX;
% hiddenLayer9Size = XX;
% hiddenLayer10Size = XX;
% hiddenLayer11Size = XX;
% hiddenLayer12Size = XX;
% hiddenLayer13Size = XX;
% hiddenLayer14Size = XX;
% hiddenLayer15Size = XX;

% En las líneas de código siguientes se debe descomentar la línea correspondiente a la cantidad de capas ocultas a implementar,
```

```

% si se desean agregar más capas de 15 capas ocultas se debe seguir la misma secuencia de escritura.

% net = patternnet(hiddenLayer1Size,trainFcn);
% net = patternnet([hiddenLayer1Size,hiddenLayer2Size],trainFcn);
% net = patternnet([hiddenLayer1Size,hiddenLayer2Size,hiddenLayer3Size],trainFcn);
% net = patternnet([hiddenLayer1Size,hiddenLayer2Size,hiddenLayer3Size,hiddenLayer4Size],trainFcn);
% net = patternnet([hiddenLayer1Size,hiddenLayer2Size,hiddenLayer3Size,hiddenLayer4Size,hiddenLayer5Size],trainFcn);
% net = patternnet([hiddenLayer1Size,hiddenLayer2Size,hiddenLayer3Size,hiddenLayer4Size,hiddenLayer5Size,
hiddenLayer6Size],trainFcn);
% net = patternnet([hiddenLayer1Size,hiddenLayer2Size,hiddenLayer3Size,hiddenLayer4Size,hiddenLayer5Size,
hiddenLayer6Size,hiddenLayer7Size],trainFcn);
% net = patternnet([hiddenLayer1Size,hiddenLayer2Size,hiddenLayer3Size,hiddenLayer4Size,hiddenLayer5Size,
hiddenLayer6Size,hiddenLayer7Size,hiddenLayer8Size],trainFcn);
% net = patternnet([hiddenLayer1Size,hiddenLayer2Size,hiddenLayer3Size,hiddenLayer4Size,hiddenLayer5Size,
hiddenLayer6Size,hiddenLayer7Size,hiddenLayer8Size,hiddenLayer9Size],trainFcn);
% net = patternnet([hiddenLayer1Size,hiddenLayer2Size,hiddenLayer3Size,hiddenLayer4Size,hiddenLayer5Size,
hiddenLayer6Size,hiddenLayer7Size,hiddenLayer8Size,hiddenLayer9Size,hiddenLayer10Size],trainFcn);
% net = patternnet([hiddenLayer1Size,hiddenLayer2Size,hiddenLayer3Size,hiddenLayer4Size,hiddenLayer5Size,
hiddenLayer6Size,hiddenLayer7Size,hiddenLayer8Size,hiddenLayer9Size,hiddenLayer10Size,hiddenLayer11Size,
hiddenLayer12Size,hiddenLayer13Size,hiddenLayer14Size,hiddenLayer15Size],trainFcn);

% El siguiente paso es configurar la división de datos para entrenamiento, validación y pruebas
% Se le asigna a cada etapa un porcentaje de los datos totales, que en total sumen 100%
net.divideParam.trainRatio = 70/100; % 70% de los datos totales para el entrenamiento
net.divideParam.valRatio = 15/100; % 15% de los datos totales no utilizados para validar que la red está
generalizando y detener el entrenamiento antes de sobreajustar.
net.divideParam.testRatio = 15/100; % 15% de los datos totales no utilizados para probar de forma independiente
la generalización de la red
%% El sistema por defecto realiza de forma automática la división de los datos con la función Dividerand, la hace de forma
%% aleatoria para los tres subconjuntos.
%% =====
%% ===== Entrenamiento de la Red Neuronal Artificial =====
%% =====
[net,tr] = train(net,x,t);
%% =====
%% ===== Prueba de la Red Neuronal Artificial =====
%% =====
%% Una vez se ha entrenado la red, puede utilizarla para calcular las salidas de la red. El siguiente código
%% calcula las salidas de la red, los errores y el rendimiento general.
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y) %% La funcion de performance está definida por la entropía cruzada
%% También puede calcular la fracción de observaciones mal clasificadas
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind);
%% =====
%% ===== Visualización de la estructura general de la Red Neuronal Artificial =====
%% =====
%% Se visualiza el diagrama correspondiente a la estructura general de la RNA.
%% En este diagrama se presenta el número de entradas, el número de capas ocultas, la cantidad de neuronas en cada capa
%% capa oculta y el numero de salidas.
view(net)
%% =====
%% ===== Visualización de gráficas adicionales =====
%% =====
% Si se desean analizar los resultados obtenidos de una forma visual se presentan las siguientes gráficas.

```

```
% Descomente estas líneas para habilitar varias gráficas.
```

```
figure, plotperform(tr)
```

```
figure, plottrainstate(tr)
```

```
figure, ploterrhist(e)
```

```
figure, plotconfusion(t,y)
```

```
figure, plotroc(t,y)
```