

Grupo 3

Participantes:

David Arias Calderón 20181020149

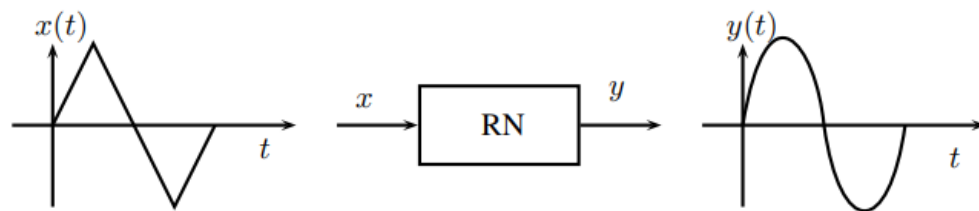
Luis Miguel Polo 20182020158

Taller 2 Ejercicio 1

Enunciado

Considerando la figura 1 diseñar una red neuronal que permita convertir una señal triangular a seno.

Figura 1



Configuraciones:

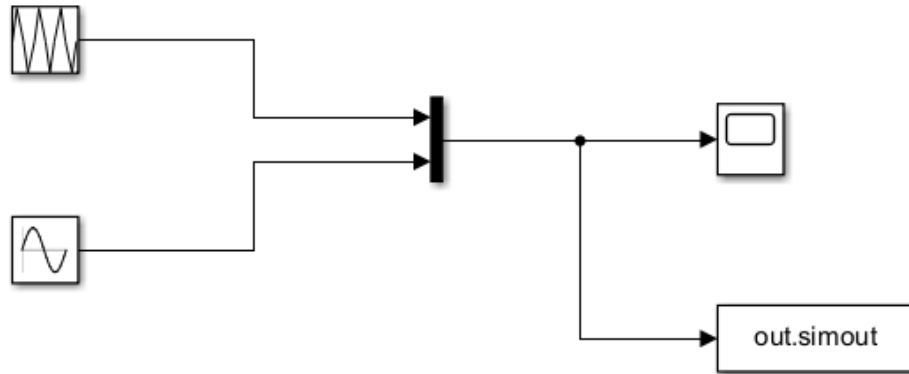
- Capas ocultas: 4
- Neuronas en las capas ocultas: 4
- Tipo de las funciones de activación: libre

Requerimientos de diseño:

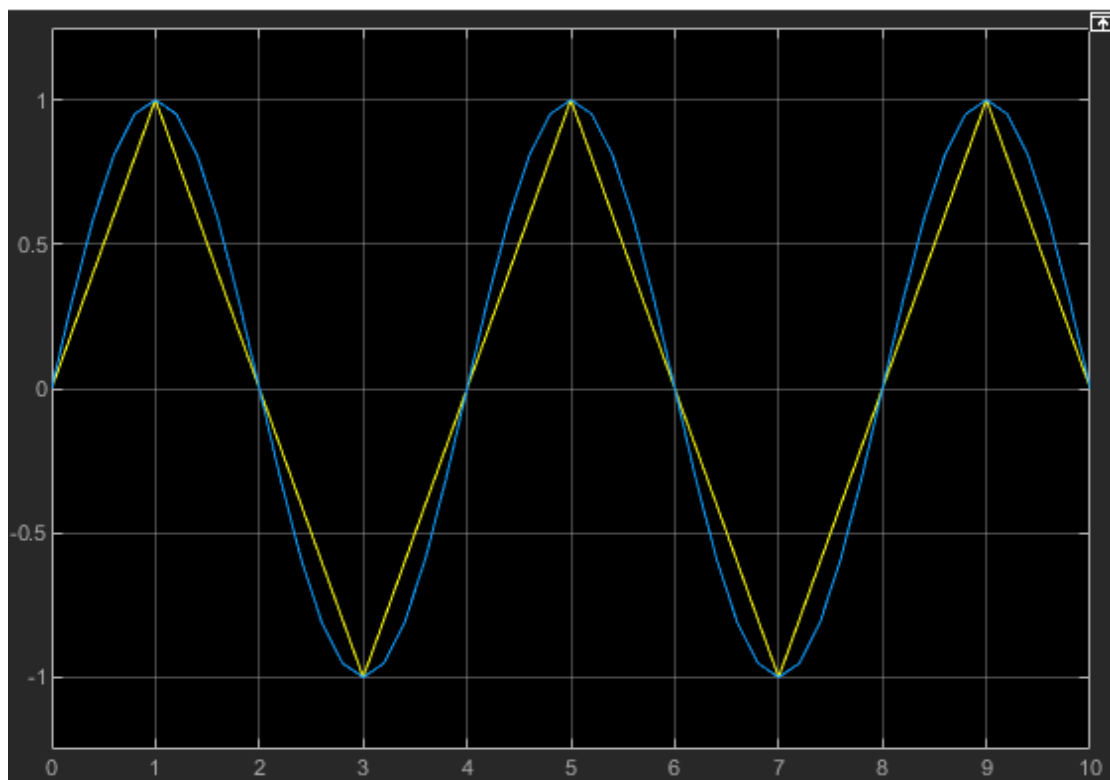
- Considerando el valor máximo de la señal:
- Error máximo del 5%
- Error cuadrático medio inferior al 2%

Solución

Montaje para el registro de datos en simulink



Gráfica del scope en simulink



Script de matlab

```

%Aplicación red neuronal aproximación de funciones

close all
clear all
warning off

%Realizar la simulación
sim('Datos1R16')

%Cargar datos
load Datos

%Rango de la entrada
R = [-1 1];

%Configuración capas y neuronas
S = [4 4 4 4 1];

%Red neuronal FF
net = newff(R,S,{'tansig','purelin','tansig','purelin','tansig','purelin'})

%Tiempo
t=ScopeData(:,1)';

%Datos entrada
P = ScopeData(:,2)';

%Datos salida
T = ScopeData(:,3)';

%Simulación sin entrenar
Y = sim(net,P);
plot(t,T,t,Y)

%Entrenamiento
net.trainParam.min_grad=0.00001;
net = train(net,P,T)

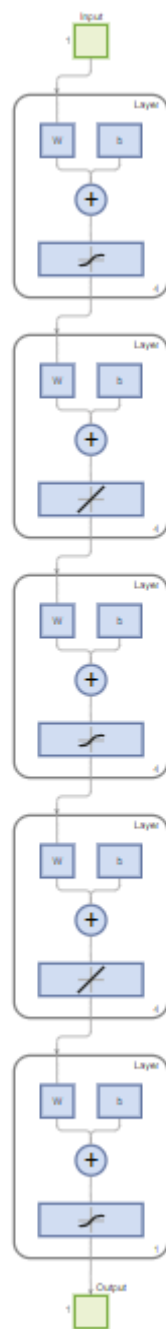
%Simulación con entrenamiento
Y = sim(net,P);
plot(t,T,t,Y)

%Figura del error
e=T-Y;
figure
plot(t,e)

%Valor del MSE
mse = (1/length(e))*sum(e.^2)

```

Esquema de la red neuronal



Resultados del entrenamiento neuronal

Training Results

Training finished: Reached minimum gradient ✓

Training Progress

Unit	Initial Value	Stopped Value	Target Value	
Epoch	0	20	1000	▲
Elapsed Time	-	00:00:06	-	
Performance	0.704	1.71e-08	0	
Gradient	1.69	8.52e-06	1e-05	
Mu	0.001	1e-06	1e+10	
Validation Checks	0	0	6	▼

Training Algorithms

Data Division: Levenberg-Marquardt trainlm

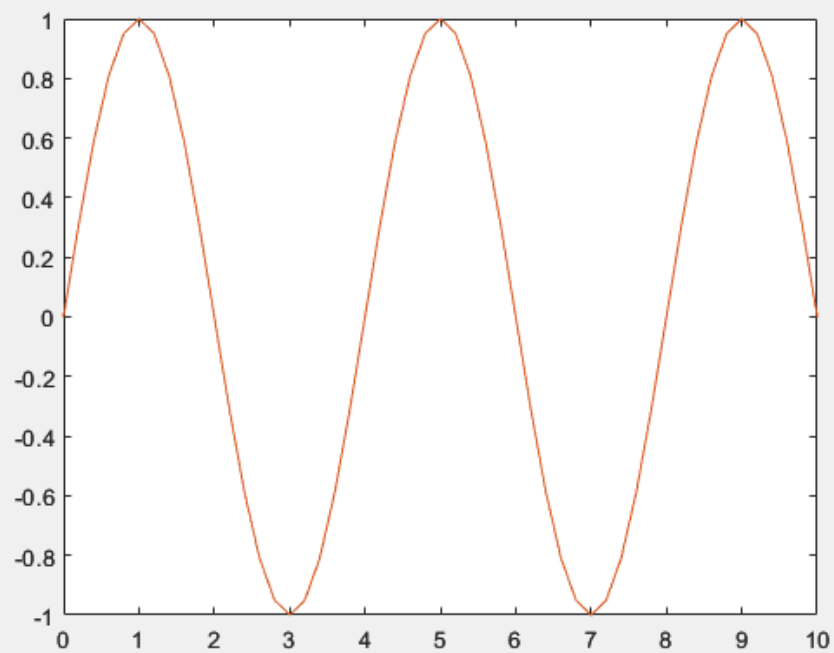
Performance: Mean Squared Error mse

Calculations: MEX

`mse =`

`1.7081e-08`

Gráfica obtenida



Gráfica obtenida aumentada

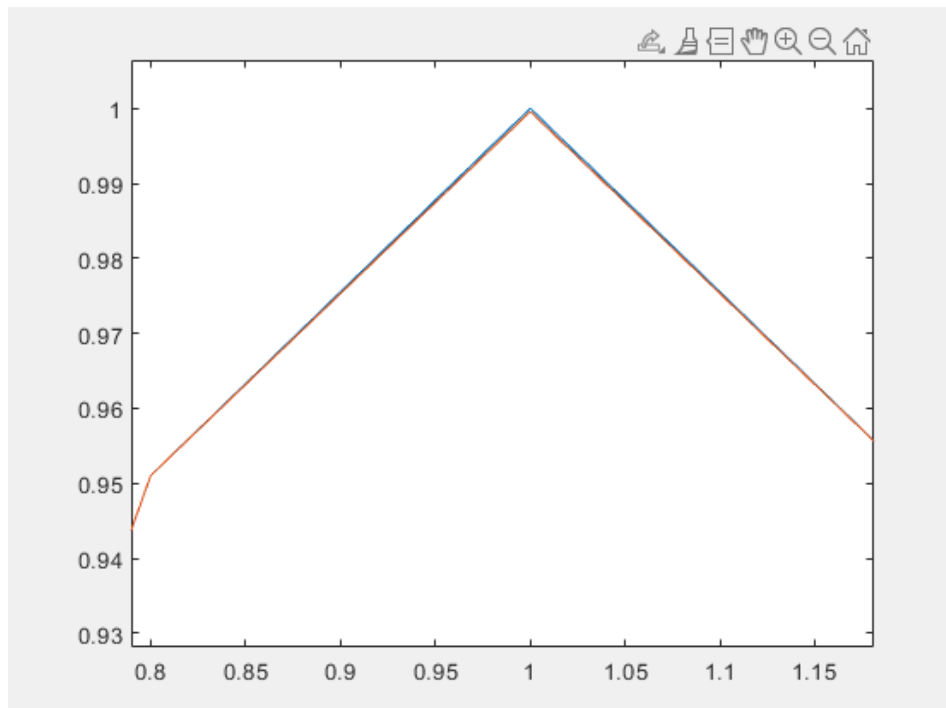


Figura del error

