



UNACAR
Universidad Autónoma del Carmen

Israel Emmanuel Jiménez Lopez.

Alumno

Reconocimiento de patrones.

Curso

Jesús Alejandro Flores Hernández

Profesor

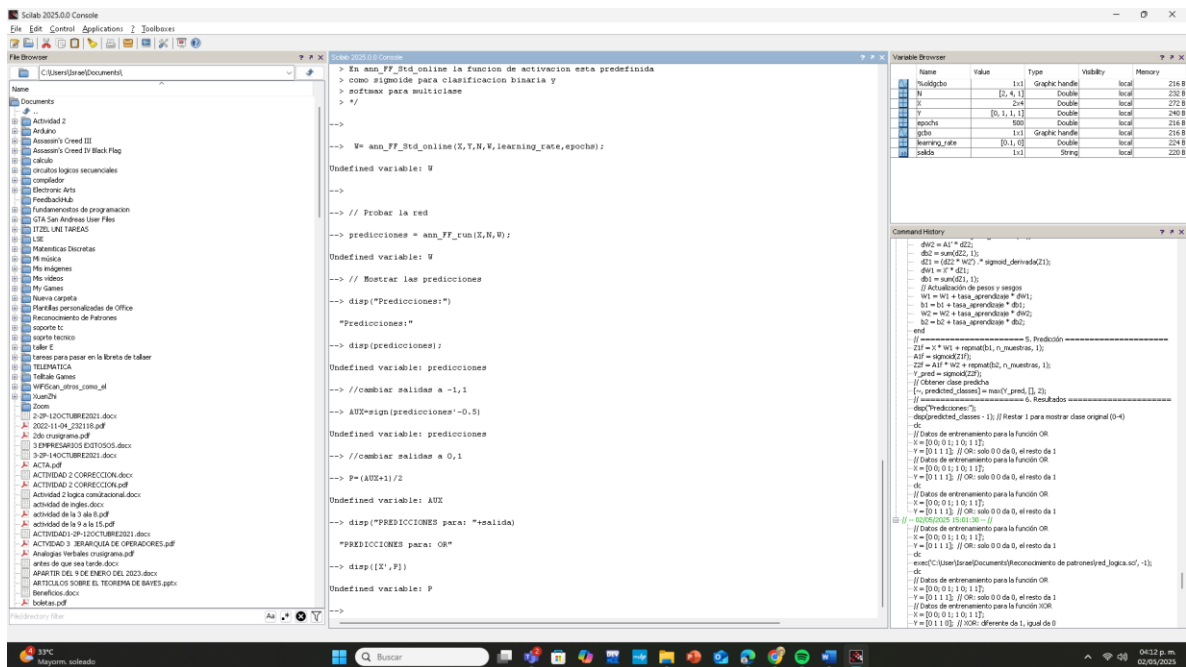


actividad Utilizar biblioteca ANN

1. Pruebe el código "ANNred2x4x1-OR.sci".
2. Modifique para las salidas de AND y XOR.

Entregable

1. haga un video que explique el código y sus resultados.
2. suba el video a un repositorio publico y envíe la liga al profesor.



```
> En ana_ff_std_online la funcion de activacion esta predefinida
> como sigmoid para clasificacion binaria y
> softmax para multiclase
> /*

--> W= ana_ff_std_online(X,T,N,W, learning_rate, epocha);

Undefined variable: W

-->

--> // Probar la red

--> predicciones = ana_ff_cum(X,N,W);

Undefined variable: W

--> // Mostrar las predicciones

--> disp("Predicciones:")

"Predicciones:"

--> disp(predicciones);

Undefined variable: predicciones

--> // cambiar salidas a -1,1

--> ATX=sign(predicciones*-0.5)

Undefined variable: predicciones

--> // cambiar salidas a 0,1

--> P=(ATX+1)/2

Undefined variable: ATX

--> disp("PREDICCIONES para: "+salida)

"PREDICCIONES para: OR"

--> disp([X',P])

Undefined variable: P

-->
```

Con función AND y XOR.

```
Archivo  Editar  Ver

// Red neuronal para XOR/OR/AND con arquitectura 2x4x1

// Definir la arquitectura de la red
N = [2,4,1]; // 2 entradas, 4 neuronas ocultas, 1 salida

// Datos de entrada (columnas con ejemplos)
X = [0 0; 0 1; 1 0; 1 1]';

// Elegir tipo de salida lógica
salida = "AND"; // Cambiar a "XOR" o "OR" según se desee

// Seleccionar los valores de salida esperados según el tipo
select salida
case "XOR"
    Y = [0; 1; 1; 0]';
case "OR"
    Y = [0; 1; 1; 1]';
case "AND"
    Y = [0; 0; 0; 1]';
else
    error("Salida no reconocida. Usa XOR, OR o AND.");
end

// Parámetros de entrenamiento
learning_rate = [0.1, 0]; // tasa de aprendizaje
epochs = 500;

// Inicializar pesos
W = ann_ff_init(N);

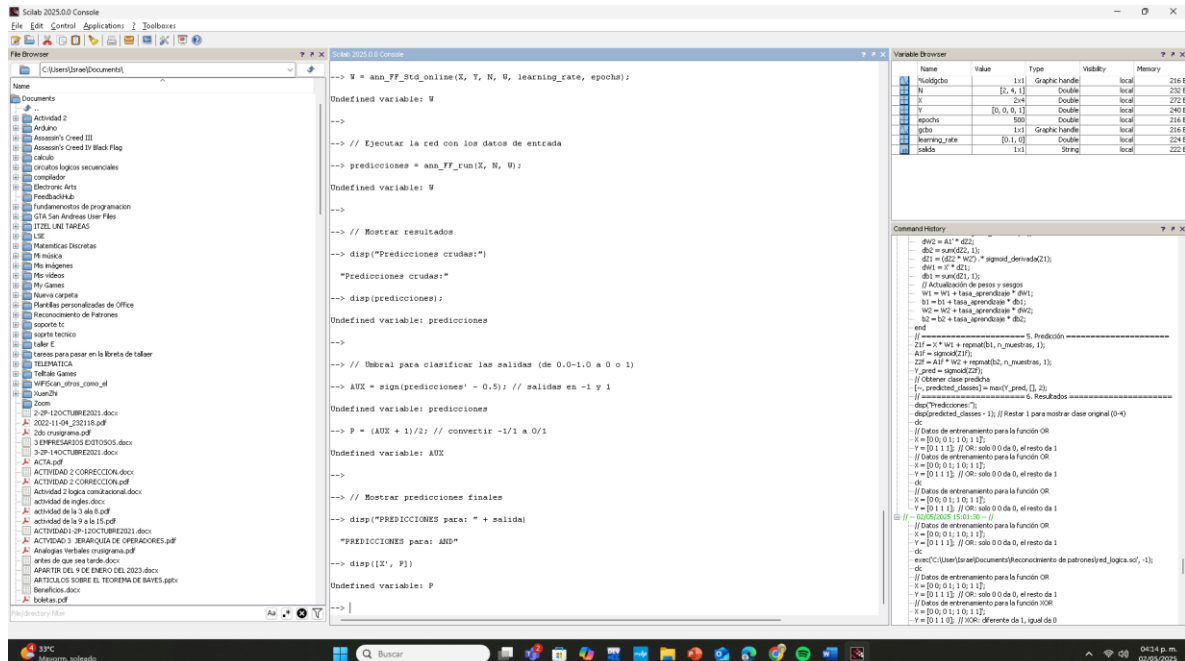
// Entrenar la red (modo en línea, activación sigmoide predeterminada)
W = ann_ff_std_online(X, Y, N, W, learning_rate, epochs);

// Ejecutar la red con los datos de entrada
predicciones = ann_ff_run(X, N, W);

// Mostrar resultados
disp("Predicciones crudas:");
disp(predicciones);

// Umbral para clasificar las salidas (de 0.0-1.0 a 0 o 1)
AUX = sign(predicciones' - 0.5); // salidas en -1 y 1
P = (AUX + 1)/2; // convertir -1/1 a 0/1

// Mostrar predicciones finales
disp("PREDICCIONES para: " + salida)
disp([X', P])
```



Objetivo del código

Entrenar una red neuronal en **Scilab** para aprender una función lógica: **AND**, **OR** o **XOR**, usando una arquitectura simple de 2 capas (2x4x1) y evaluar sus predicciones.

Explicación.	Código.
<ul style="list-style-type: none">2 entradas (porque usamos combinaciones binarias de 2 bits).	$N = [2,4,1];$

<ul style="list-style-type: none"> • 4 neuronas en la capa oculta (puedes cambiar esto). • 1 salida (valor entre 0 y 1 que representa la clase). 	
<p>Son las 4 combinaciones posibles de dos valores binarios.</p> <p>Se transponen (') para que cada columna sea un ejemplo de entrada (formato que requiere la toolbox).</p>	$X = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix};$
<ul style="list-style-type: none"> • Puedes cambiar este valor para probar otras funciones lógicas. 	<code>salida = "AND"; // Cambia por "OR" o "XOR"</code>
<p>Según la lógica elegida, se definen las salidas correctas para cada combinación de entrada.</p> <p>También se transpone (') para que el formato sea 1x4.</p>	<pre>select salida case "XOR" Y = [0; 1; 1; 0]; case "OR" Y = [0; 1; 1; 1]; case "AND" Y = [0; 0; 0; 1]; else error("Salida no reconocida. Usa XOR, OR o AND."); end</pre>
<p>Se inicializan los pesos aleatoriamente.</p>	<code>W = ann_FF_init(N);</code>
<p><code>ann_FF_Std_online</code>: algoritmo de aprendizaje en línea (actualiza los pesos con cada ejemplo).</p> <p>Usa función de activación sigmoide internamente.</p> <p>Se entrena durante epochs iteraciones.</p>	<code>W = ann_FF_Std_online(X, Y, N, W, learning_rate, epochs);</code>
<p>❑ Se pasa la entrada por la red ya entrenada.</p> <p>❑ La salida será un número entre 0 y 1.</p>	<code>predicciones = ann_FF_run(X, N, W);</code>
<p>❑ Se aplica un umbral de 0.5 para convertir la salida continua a binaria.</p> <p>❑ <code>sign(x-0.5)</code> devuelve -1 para $x < 0.5$ y 1 para $x \geq 0.5$.</p> <p>❑ Se ajusta a 0 y 1 con $(AUX + 1)/2$.</p>	<pre>AUX = sign(predicciones' - 0.5); P = (AUX + 1)/2;</pre>
<p>Muestra las entradas y sus salidas predichas.</p>	<pre>disp("PREDICCIONES para: " + salida) disp([X', P])</pre>