**INSTITUTO POLITÉCNICO NACIONAL**

Alumno: Romero Gamarra Joel Mauricio

Profesor: Moreno Armendáriz Marco Antonio

Asignatura: Neural Networks

Grupo: 3CM2

**Red Adaline**

Como ya vimos en clase, debemos comenzar asignando valores aleatorios a la matriz de pesos (en este caso, no tendremos bias), posteriormente pedimos los datos al usuario para poder comenzar el aprendizaje, uno de ellos es el máximo de iteraciones, que como su nombre lo indica es para que el aprendizaje se detenga en algún momento. El valor al que se desea que llegue la señal del error es para otra condición de paro, y el valor de alfa es un valor multiplicativo para actualizar el valor de la matriz de pesos.

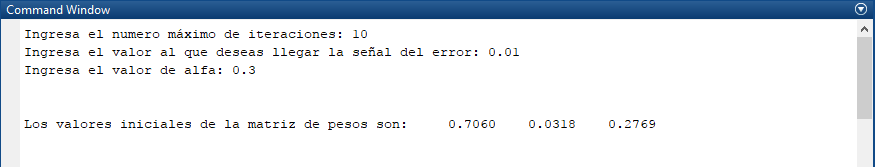


Figura . Valores ingresados por el usuario y valores aleatorios para W

A continuación, se observan las iteraciones necesarias para realizar el aprendizaje de la RNA de Adaline imprimiendo el valor de a y del error por cada dato (en este caso, son 8 datos).

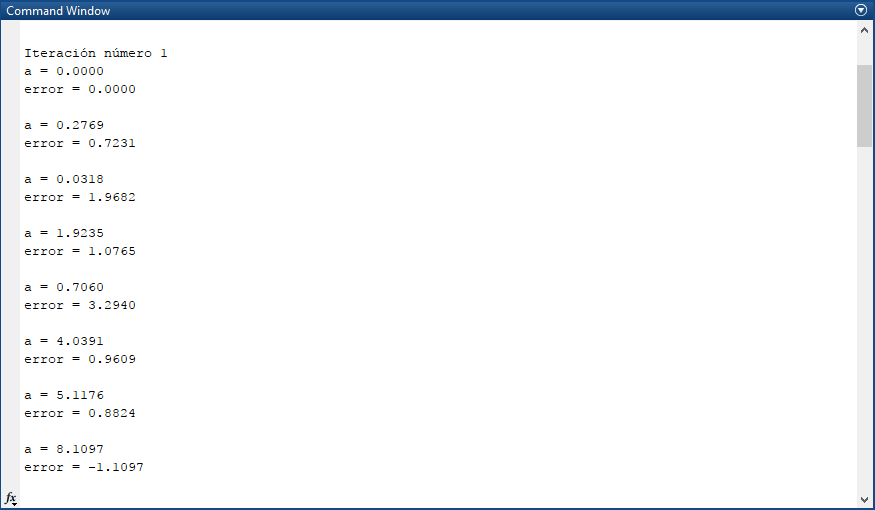


Figura . Iteración número 1 de la red Adaline

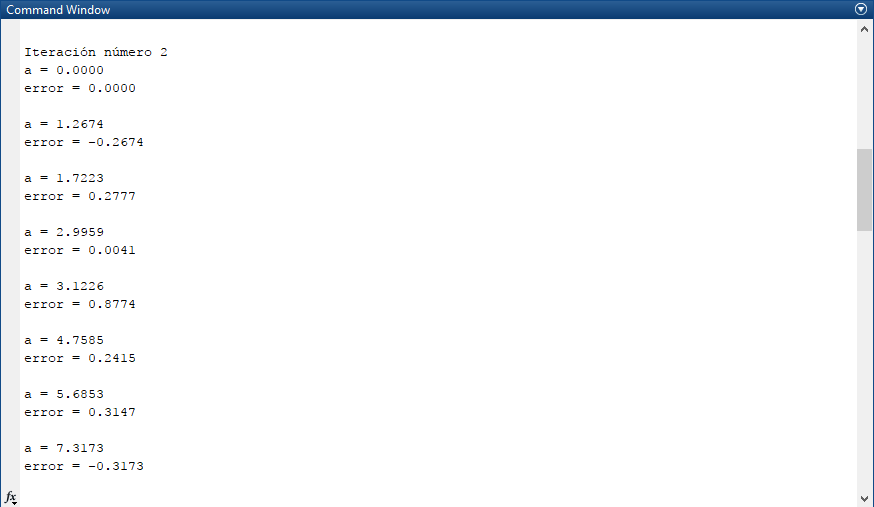


Figura . Iteración número 2 de la red Adaline

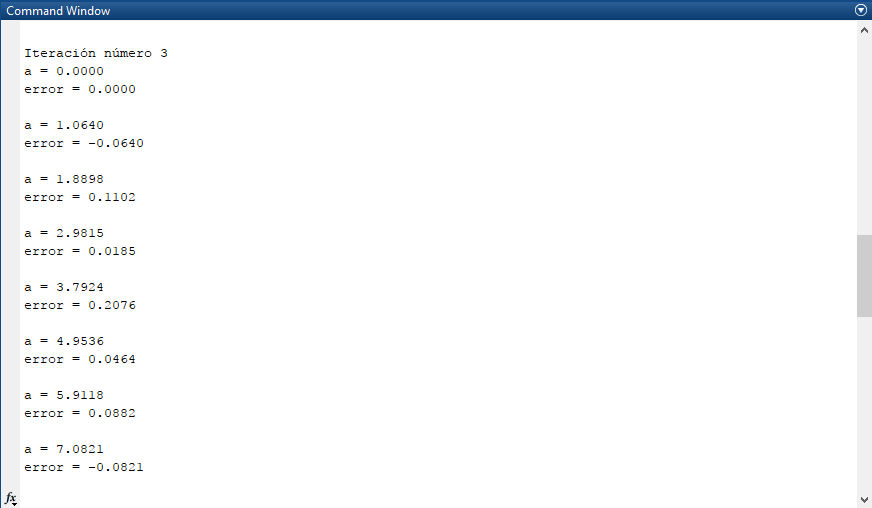


Figura . Iteración número 3 de la red Adaline

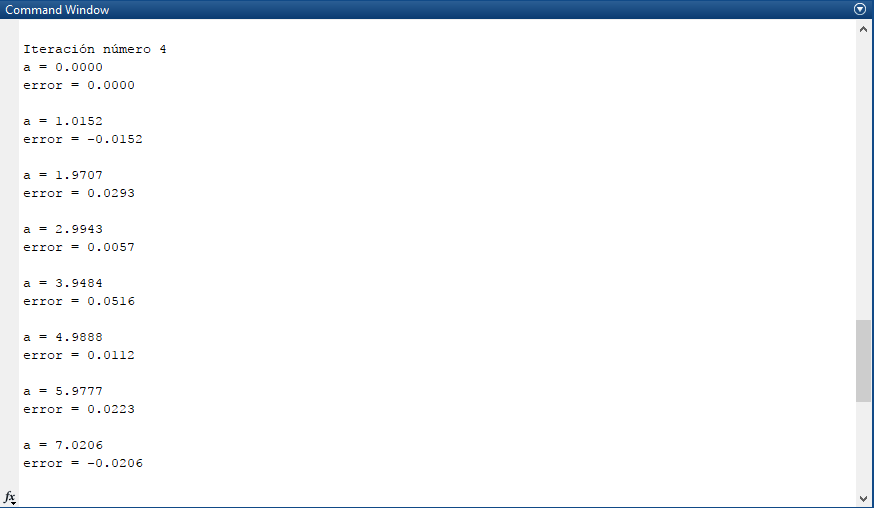


Figura . Iteración número 4 de la red Adaline

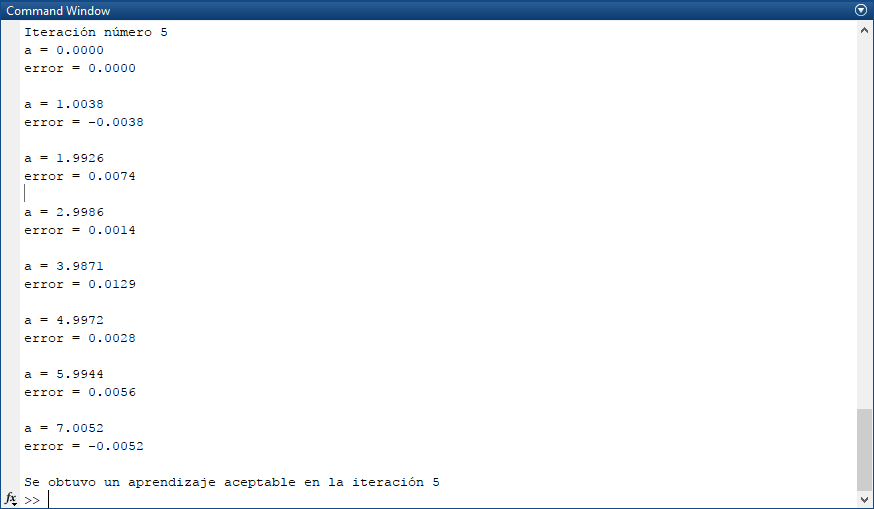


Figura . Iteración número 5 y última de la red Adaline al cumplir la condición de paro (Eit < eit)

A continuación, se muestran las gráficas tanto de la matriz de pesos como de la señal del error, ambas se pusieron en un figure distinto para poder apreciar mejor el comportamiento de la señal del error, que pareciera que no se va acercando a 0, sin embargo al observar la escala, podemos apreciar que si lo está haciendo.

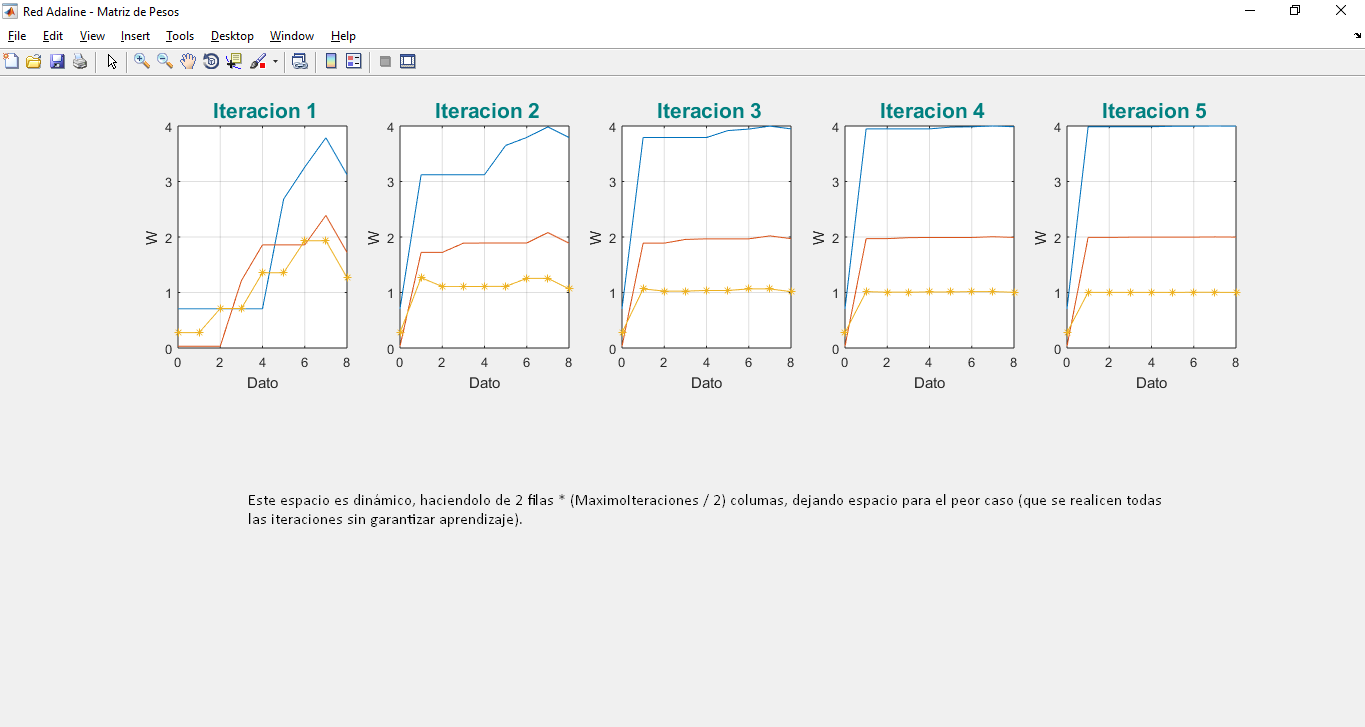


Figura . Gráficas de la matriz de pesos iteración a iteración

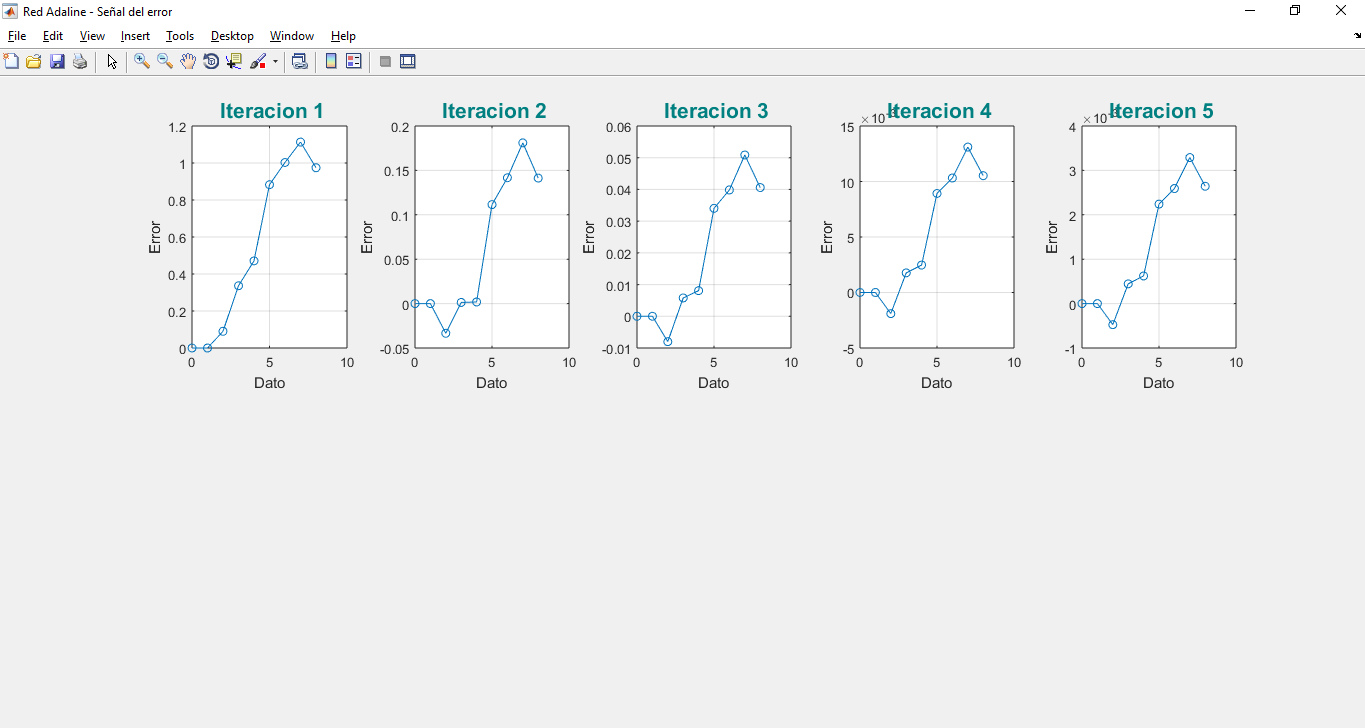


Figura . Gráficas de la señal del error iteración a iteración

**Código**

%Comenzamos limpiando la pantalla y todas las variables

clearvars

clc

%Pedimos los valores dados por el usuario para realizar el aprendizaje de

%una RNA Adaline de un codificador de binario a decimal de 3 bits sin bias

it\_max = input ('Ingresa el numero m�ximo de iteraciones: ');

e\_it = input ('Ingresa el valor al que deseas llegar la se�al del error: ');

alpha = input ('Ingresa el valor de alfa: ');

%Asignamos valores aleatorios a la matriz de pesos W con la funci�n rand

W = rand (1, 3);

W1 = zeros (1, 9);

W2 = zeros (1, 9);

W3 = zeros (1, 9);

%W = [0.84 0.39 0.78];

fprintf ('\n\nLos valores iniciales de la matriz de pesos son: ')

disp (W);

errores = 0;

Pesos = figure('Name','Red Adaline - Matriz de Pesos','NumberTitle','off');

Error = figure('Name','Red Adaline - Se�al del error','NumberTitle','off');

rango = 0:8;

W1 (1) = W (1);

W2 (1) = W (2);

W3 (1) = W (3);

Error\_It = 0;

Error\_Global = zeros (1, 9);

**for** j = 1:it\_max

%k ser� un 0 cada iteraci�n para convertir a binario

k = zeros (1, 1);

fprintf ('\n\n\nIteraci�n n�mero %d\n', j)

%Comenzamos a realizar la propagaci�n hacia adelante de todos los datos

**for** i = 1:8

%Convertimos a k a un n�mero binario de 3 bits

p = dec2bin (k, 3);

%Transponemos el vector, para que sea un vector columna

p = p';

%Conertimos los elementos a numero para realizar la multiplicaci�n

p = str2num (p);

Error\_Global (i) = Error\_It;

%Realizamos la multiplicacion de las matrices

a = purelin (W \* p);

fprintf ('a = %.4f\n', a)

%Obtenemos el error para este dato

errores = (k - a);

fprintf ('error = %.4f\n\n', errores)

%Actualizamos el valor de la matriz de pesos

W = W + ((2 \* alpha \* errores) \* p');

k = k + 1;

%Actutalizamos los valores de los pesos

W1 (i + 1) = W (1);

W2 (i + 1) = W (2);

W3 (i + 1) = W (3);

%Actualizamos el valor del error

Error\_It = Error\_It + ((1/8) \* errores);

**end**

Error\_Global (9) = Error\_It;

%Pintamos en el figure de la se�al del error

figure (Error);

hold on;

subplot (2, ceil(it\_max / 2), j),plot (rango, Error\_Global, 'o-');

grid, ylabel('Error'), xlabel('Dato');

title (['\fontsize{16} \color[rgb]{0 .5 .5}Iteracion ' int2str(j)]);

%Pintamos en el figure de la matriz de pesos

figure (Pesos);

hold on;

subplot (2, ceil(it\_max / 2), j),plot (rango, W1, rango, W2, rango, W3, '\*-');

grid, ylabel('W'), xlabel('Dato');

title (['\fontsize{16} \color[rgb]{0 .5 .5}Iteracion ' int2str(j)]);

%Si el error global es 0, el aprendizaje obtenido es el ideal

**if** Error\_It == 0

fprintf ('Se cumpli� un aprendizaje ideal en la iteraci�n %d\n', j)

**break**;

%Si el error global es menor al error introducido por el usuario, el

%aprendizaje obtenido es 2/3 de exitoso

**elseif** Error\_It < e\_it

fprintf ('Se obtuvo un aprendizaje aceptable en la iteraci�n %d\n', j)

**break**;

**end**

%Reseteamos el valor del error global a 0

Error\_It = 0;

**end**

clearvars