

Modelado de Funciones Booleanas – Red Madaline

Peralta Rosales Oscar Esaú

Ing. En computación

1002-A

Introducción

La red Madaline es una Red Neuronal Artificial formada por varias neuronas de tipo Adaline, las cuales se encuentran agrupadas en capas. Una de los usos de este tipo de redes es el modelado de Funciones Booleanas, para esta implementación se hará uso de la tabla de verdad generada por una Función Booleana, cuyos valores serán usados para entrenar cada una de las neuronas que conforman la red mediante el algoritmo de Descenso Escalonado.

Desarrollo

Dada la tabla de verdad de una función booleana de N variables, el número de tuplas que conformaran la tabla será de 2^N , sin embargo en este caso solo nos interesan aquellas cuyo evaluación en la función booleana sea igual a 1 o verdadero. Por ejemplo supongamos una función de 3 variables con los siguientes valores verdaderos $\{ [0\ 1\ 0], [1\ 0\ 1], [1\ 1\ 0] \}$, todas las demás combinaciones de valores son ceros, la manera de construir una función que modele este comportamiento es usando los operadores de disyunción y conjunción. Usando el operador de disyunción para los valores $[0\ 1\ 0]$ parte de la función queda como $\bar{X}_1 \wedge X_2 \wedge \bar{X}_3$, para los valores $[1\ 0\ 1]$ queda como $X_1 \wedge \bar{X}_2 \wedge X_3$, y para $[1\ 1\ 0]$ se obtiene $X_1 \wedge X_2 \wedge \bar{X}_3$, ahora uniéndolas con los operadores de conjunción la función obtenida es:

$$f(X_1, X_2, X_3) = (\bar{X}_1 \wedge X_2 \wedge \bar{X}_3) \vee (X_1 \wedge \bar{X}_2 \wedge X_3) \vee (X_1 \wedge X_2 \wedge \bar{X}_3)$$

La función anterior modela perfectamente el comportamiento para nuestras tres combinaciones de valores previas.

La Red Madaline se conformará de solo dos capas, la capa de entrada y la capa de salida, cada neurona de la capa de entrada modelara cada tupla de la tabla de verdad cuyo valor sea verdadero o uno, de donde se observa que la capa de entrada tendrá tantas neuronas como combinaciones haya con valor uno o verdadero cuando son evaluadas en la función booleana, para el ejemplo anterior la capa de entrada contará con 3 neuronas. Cada neurona de la capa de entrada será entrenada para reconocer su combinación correspondiente, es tendrá una salida uno solo cuando sea la combinación de variables correcta para la cual fue entrenada. Cada neurona de la capa de entrada contribuirá para generar una nueva combinación de valores que será la entrada para una única neurona que será parte de la capa de

salida modelando así la conjunción de las funciones. Dichas neurona de salida debe ser entrenada para que reconozca solamente la combinación generada por la salida de las neuronas de la capa de entrada.

La implementación de la Red Madaline se realizó bajo el lenguaje de programación C++ bajo un entorno GNU/Linux.

Resultados

Caso 1.

La primer tabla de verdad usada para probar la red neuronal fue la siguiente:

X1	x2	x3	f(x1, x2, x3)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Tabla 1.1. Valores para la función booleana Caso 1.

Configuración de la red:

- Capa de Entrada: 4 Neuronas, 3 entradas una salida por neurona.
- Capa de Salida: 1 Neurona, 4 entradas una salida.

Parámetros de entrenamiento:

- Número de presentaciones: 100
- Coeficiente de aprendizaje: 0.0032

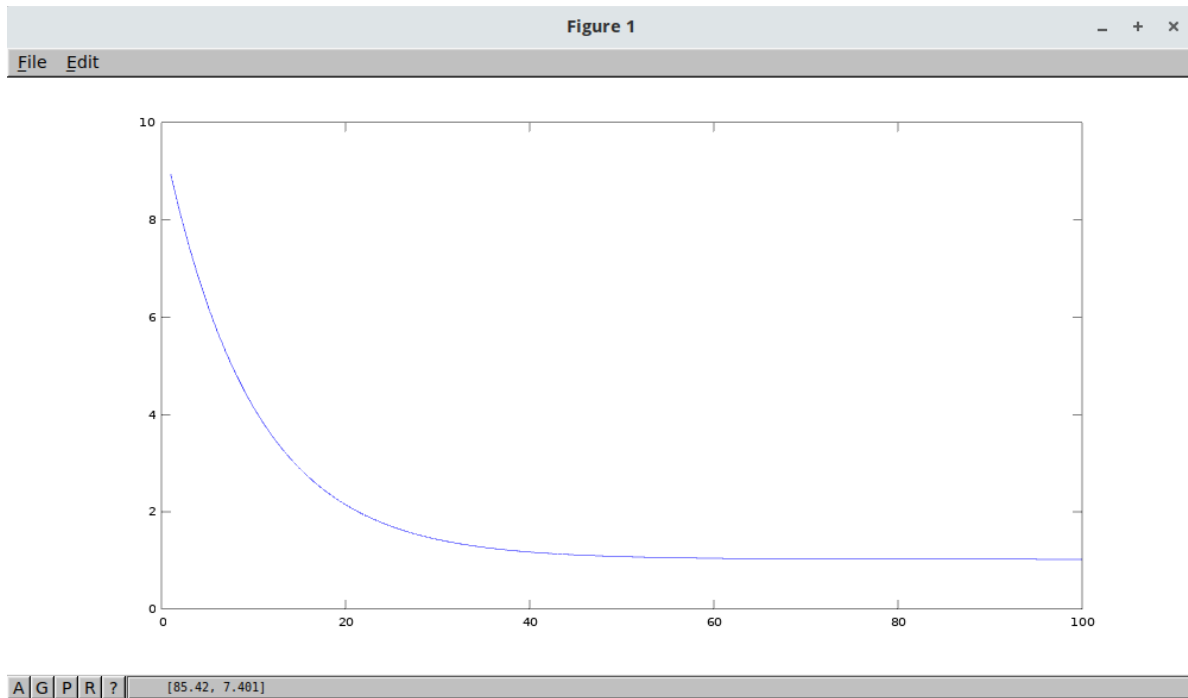
Pesos de las neuronas de la Capa de entrada después del entrenamiento:

- Neurona 1: $d = 0.749274 + -0.243870X1 + -0.242353X2 + 0.250987X3$
- Neurona 2: $d = -0.751620 + -0.252499X1 + 0.242272X2 + -0.251068X3$
- Neurona 3: $d = -0.752211 + 0.238071X1 + -0.245417X2 + -0.243979X3$
- Neurona 4: $d = -0.738981 + 0.251193X1 + 0.257328X2 + 0.255890X3$

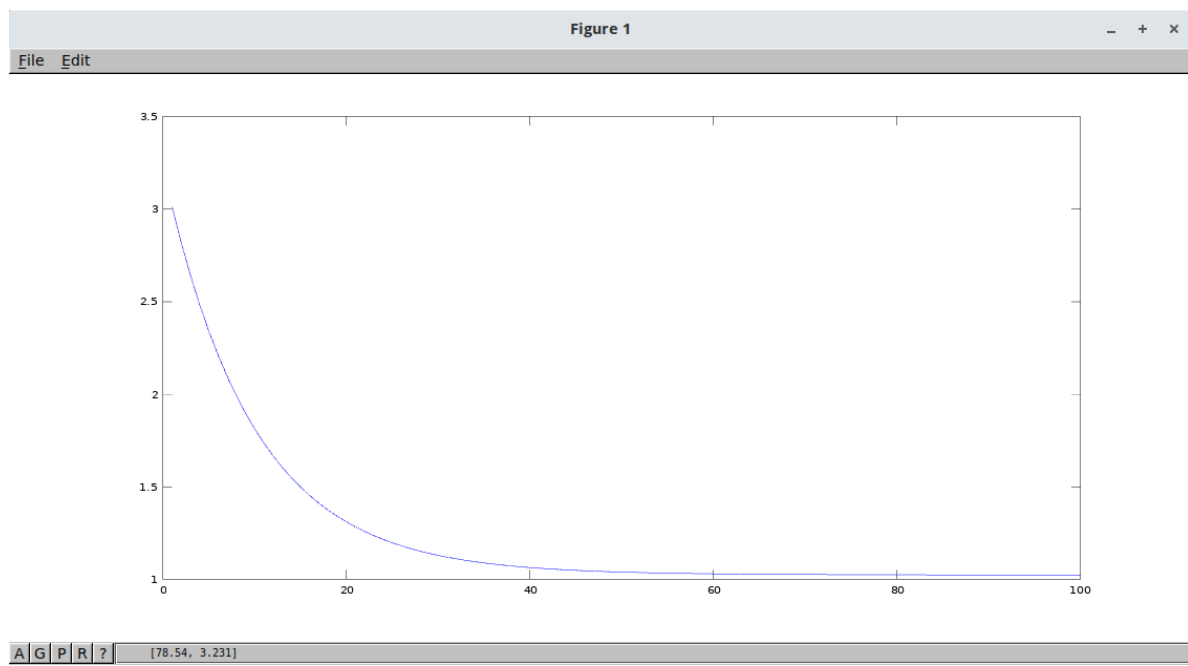
Pesos de la neurona de la Capa de Salida después del entrenamiento:

- Neurona 5: $d = 0.581894 + 0.154942X_1 + 0.171274X_2 + 0.101895X_3 + 0.202074X_4$

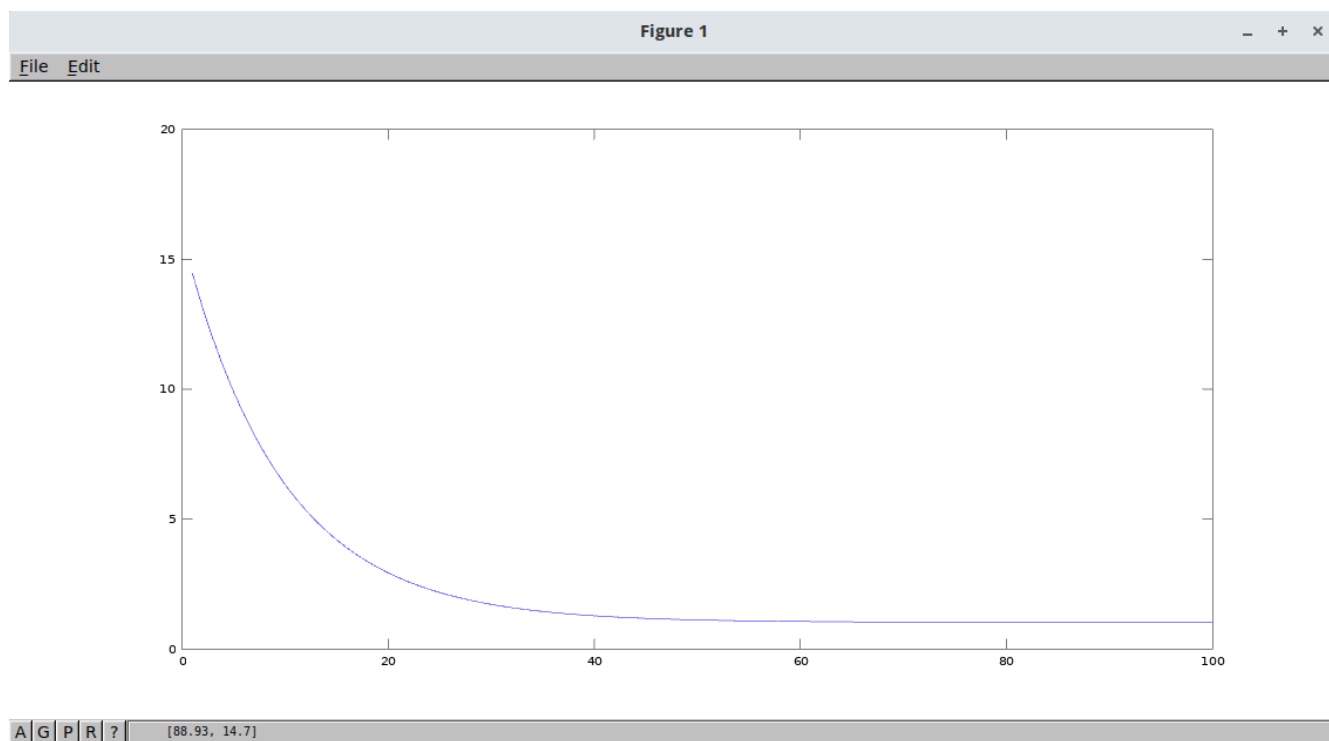
Graficas de error:



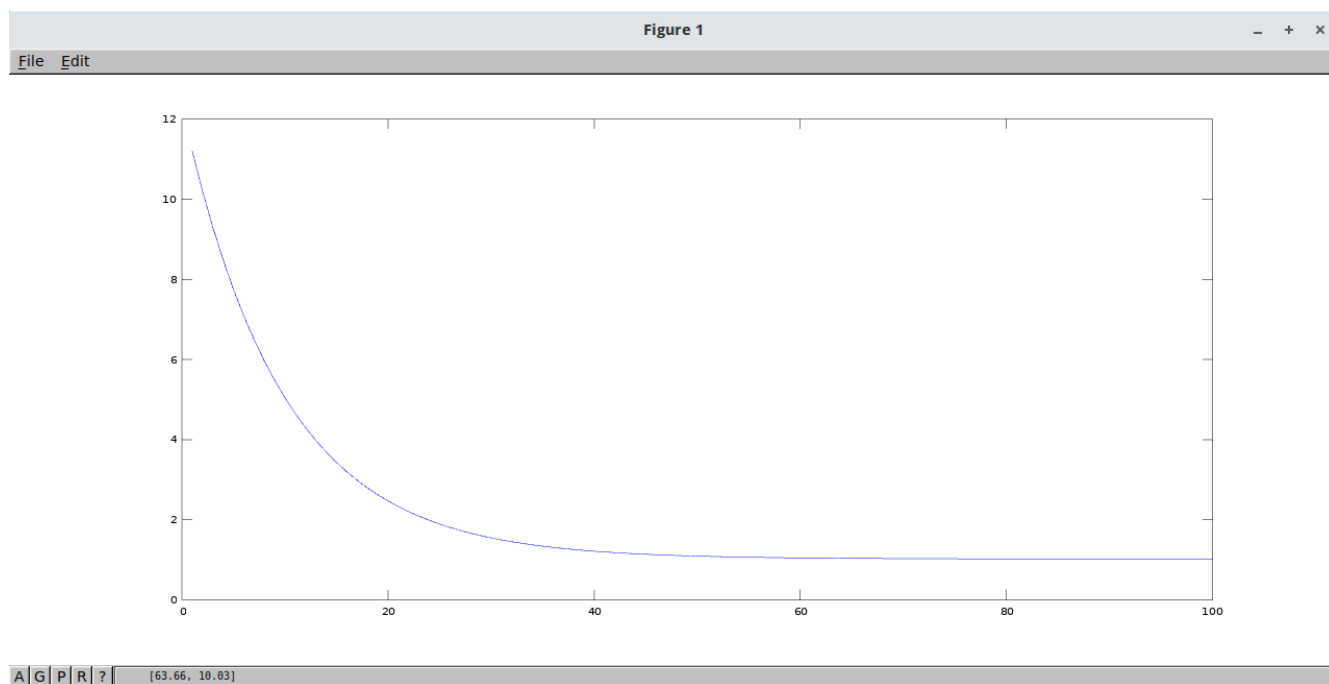
Gráfica 1.1. Error de aprendizaje para la Neurona 1



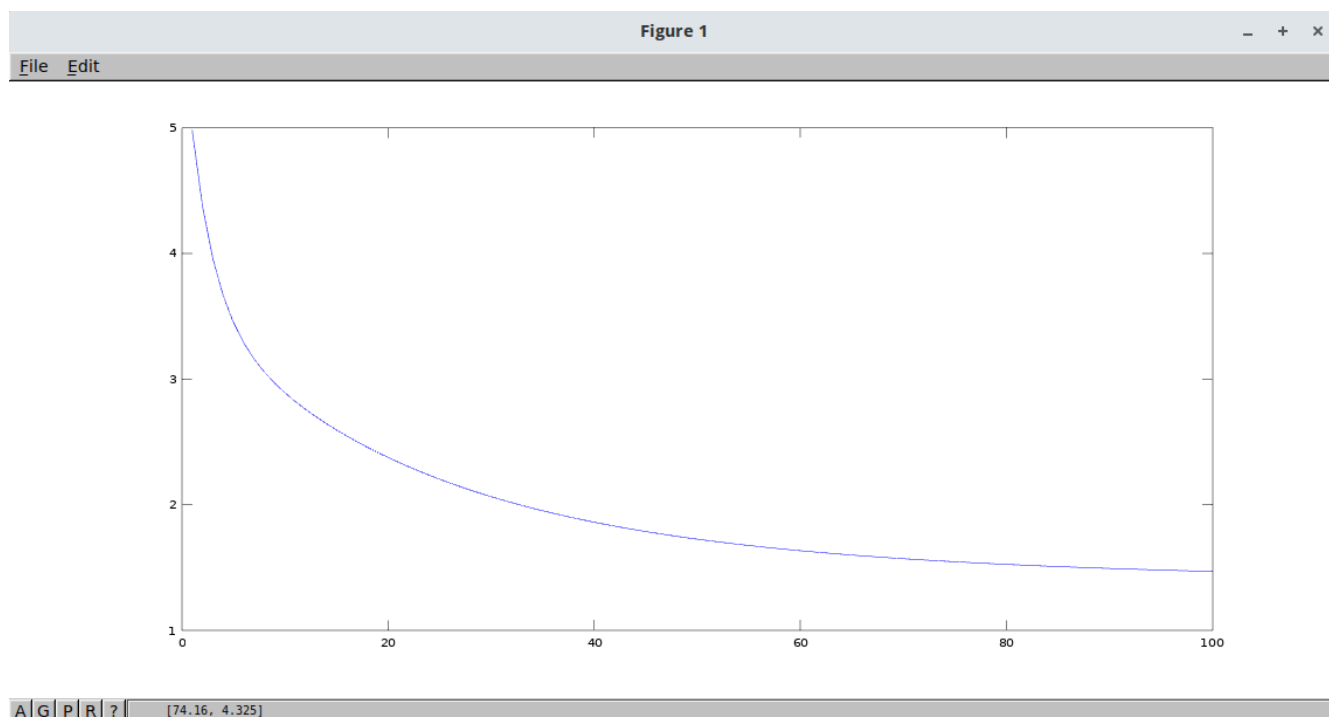
Gráfica 1.2. Error de aprendizaje para la Neurona 2



Gráfica 1.3. Error de aprendizaje para la Neurona 3



Gráfica 1.4. Error de aprendizaje para la Neurona 4



Gráfica 1.5. Error de aprendizaje para la Neurona 5

Caso 1.

La primer tabla de verdad usada para probar la red neuronal fue la siguiente:

X1	x2	x3	x4	x1, x2, x3, x4
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Tabla 1.1. Valores para la función booleana Caso 1.

Configuración de la red:

- Capa de Entrada: 9 Neuronas, 4 entradas una salida por neurona.
- Capa de Salida: 1 Neurona, 9 entradas una salida.

Parámetros de entrenamiento:

- Número de presentaciones: 100
- Coeficiente de aprendizaje: 0.0032

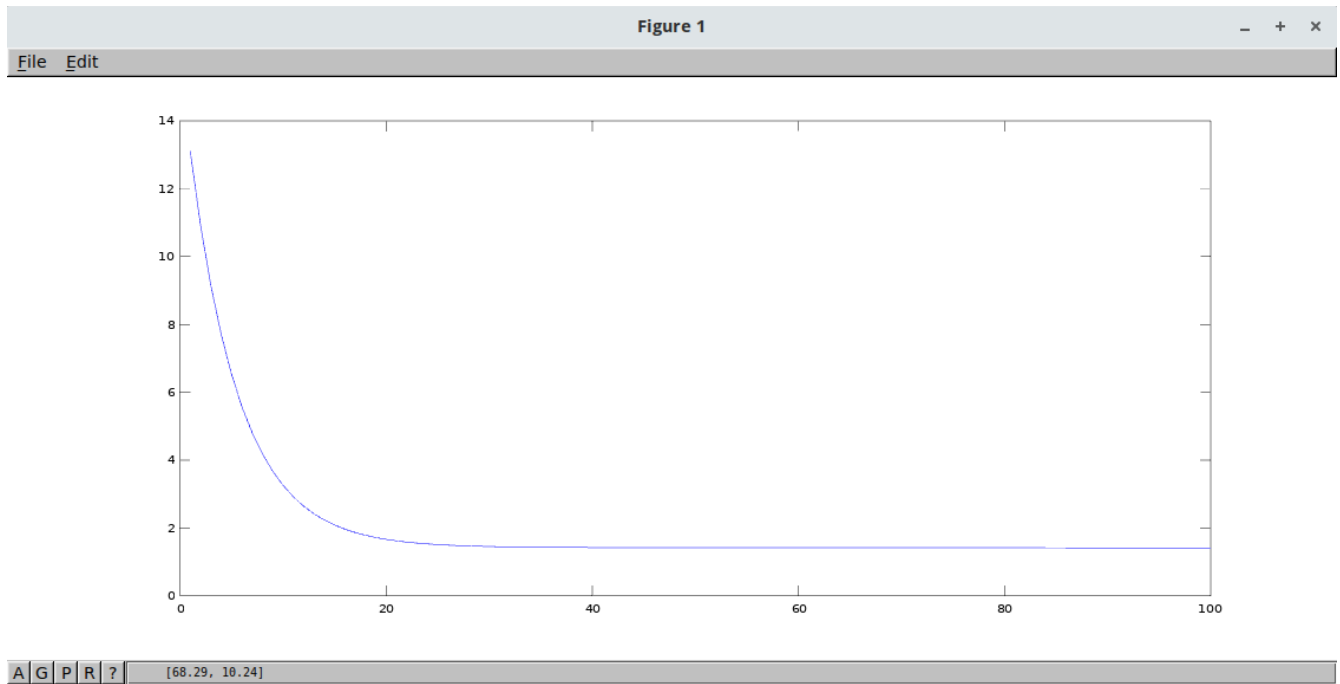
Pesos de las neuronas de la Capa de entrada después del entrenamiento:

- Neurona 1: $d = -0.874992 + -0.123852X_1 + -0.122355X_2 + 0.119489X_3 + -0.121238X_4$
- Neurona 2: $d = -0.874987 + -0.126015X_1 + -0.124568X_2 + 0.121659X_3 + 0.121281X_4$
- Neurona 3: $d = -0.874953 + -0.123818X_1 + 0.119521X_2 + -0.122366X_3 + -0.122694X_4$
- Neurona 4: $d = -0.874929 + -0.129160X_1 + 0.124590X_2 + 0.124624X_3 + 0.124194X_4$
- Neurona 5: $d = -0.875003 + 0.123147X_1 + -0.127523X_2 + -0.127540X_3 + 0.125673X_4$
- Neurona 6: $d = -0.874987 + 0.123842X_1 + -0.128292X_2 + 0.125349X_3 + -0.127163X_4$
- Neurona 7: $d = -0.874979 + 0.126028X_1 + 0.127546X_2 + -0.130688X_3 + 0.128737X_4$
- Neurona 8: $d = -0.874954 + 0.126838X_1 + 0.128360X_2 + 0.128316X_3 + -0.130291X_4$
- Neurona 9: $d = -0.874996 + 0.129200X_1 + 0.130697X_2 + 0.130680X_3 + 0.130245X_4$

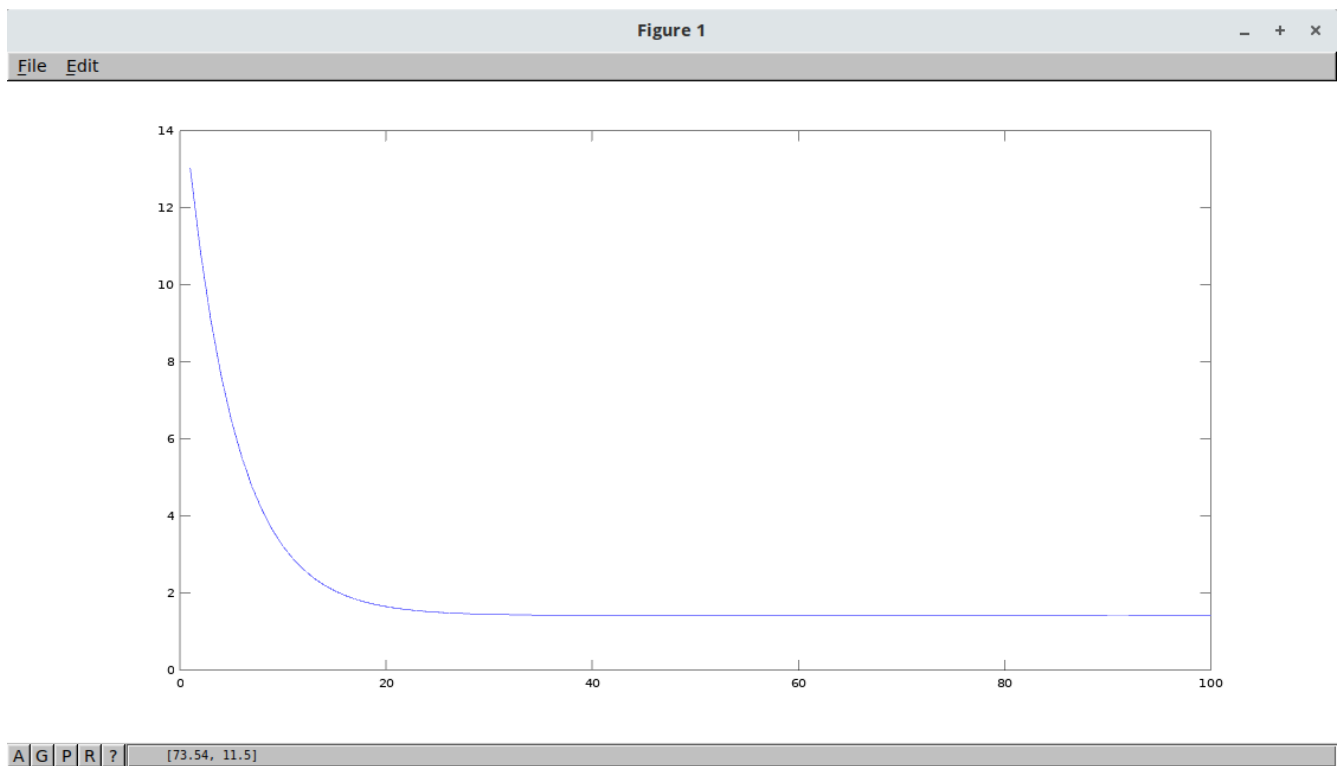
Pesos de la neurona de la Capa de Salida después del entrenamiento:

- Neurona 10: $d = 0.960318 + 0.005038X_1 + 0.005042X_2 + 0.005050X_3 + 0.005064X_4 + 0.005101X_5 + 0.005210X_6 + 0.005518X_7 + 0.006419X_8 + 0.009914X_9$

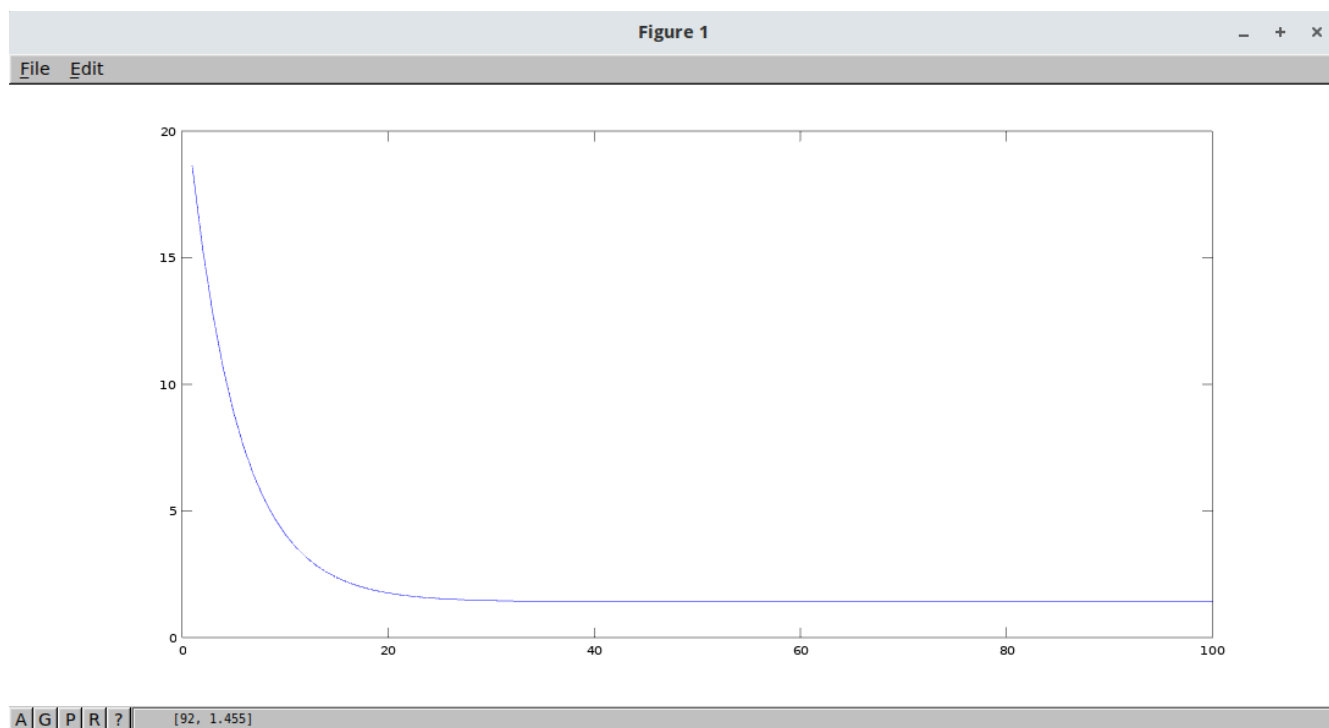
Graficas de error:



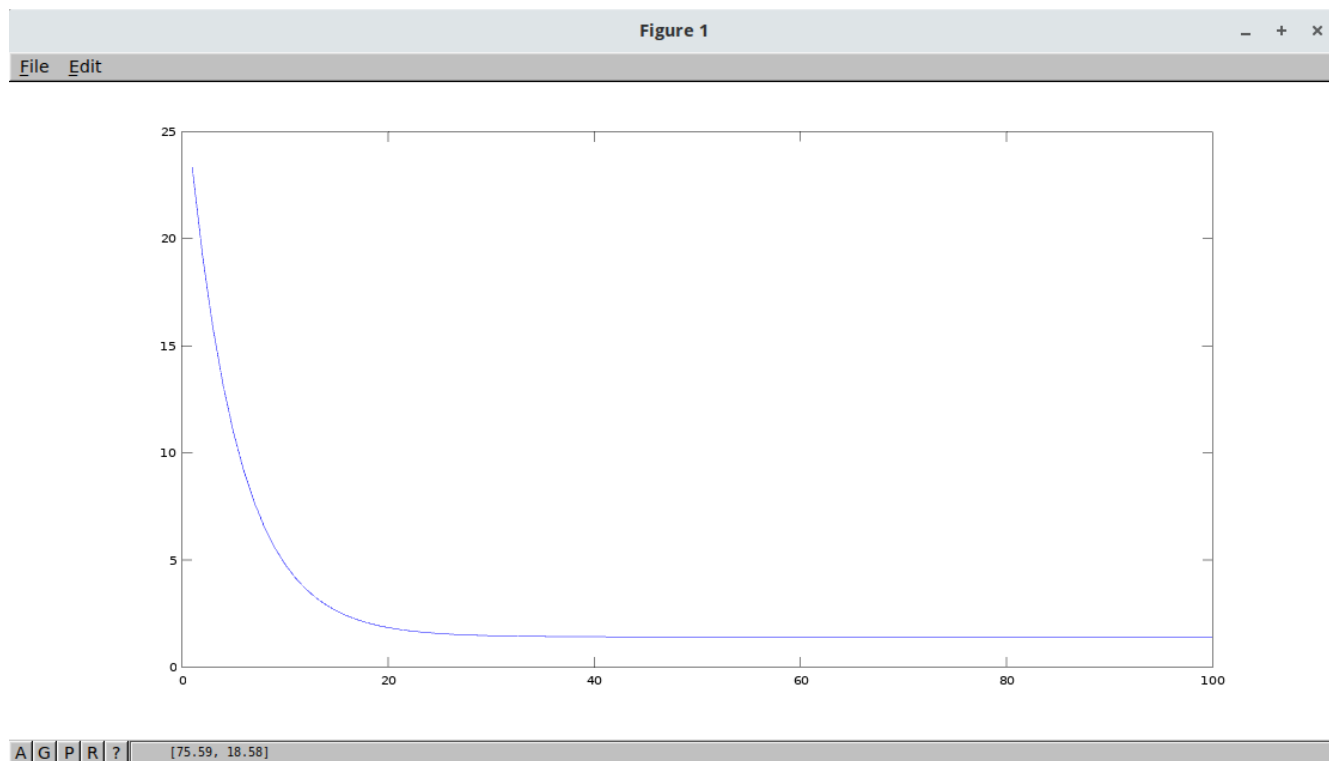
Gráfica 1.6. Error de aprendizaje para la Neurona 1



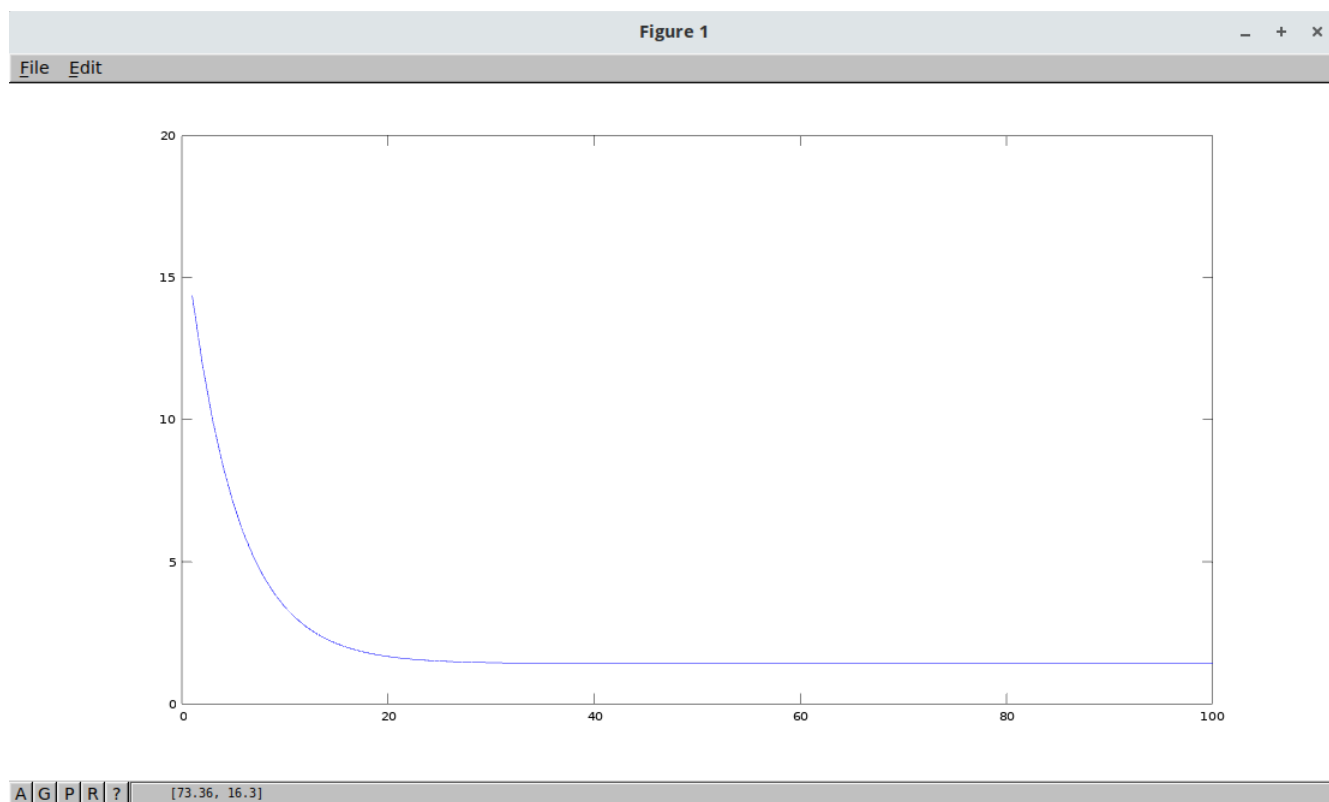
Gráfica 1.7. Error de aprendizaje para la Neurona 2



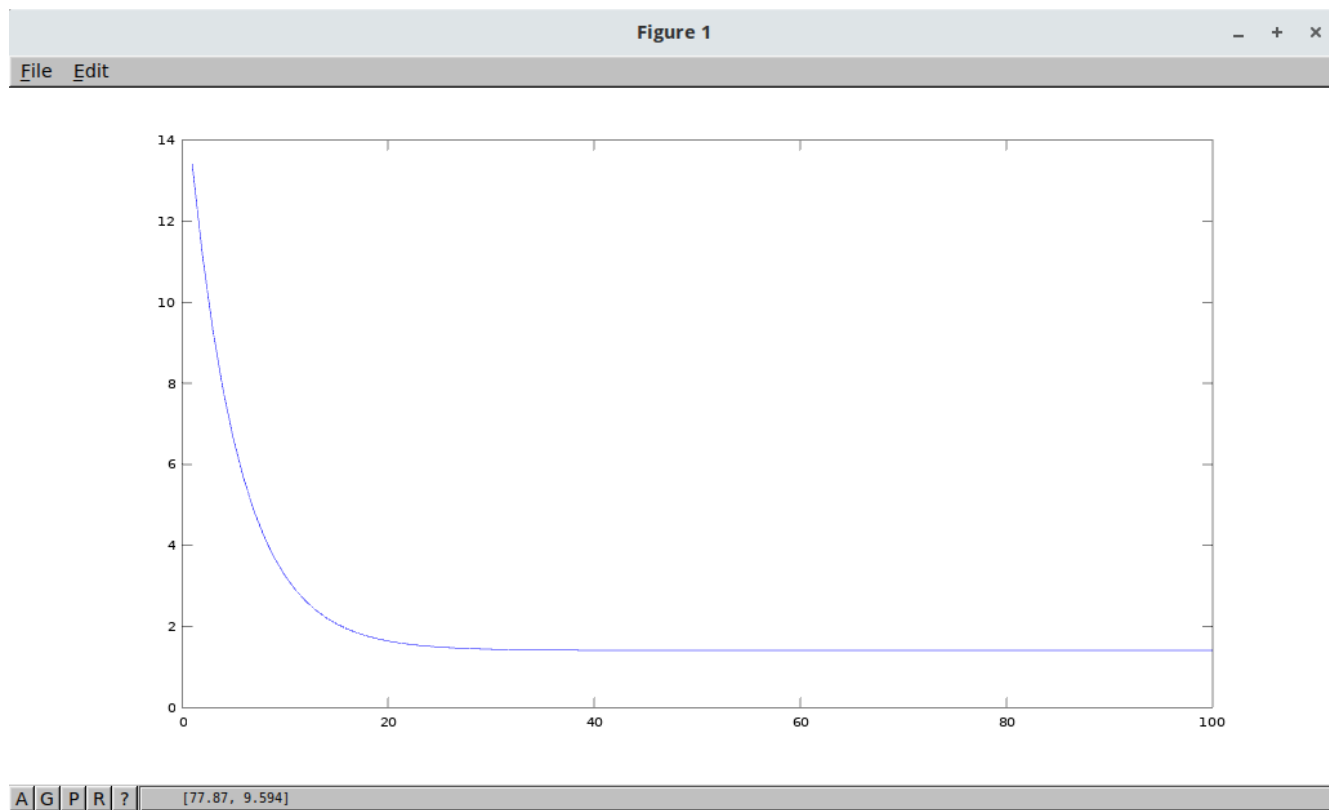
Gráfica 1.8. Error de aprendizaje para la Neurona 3



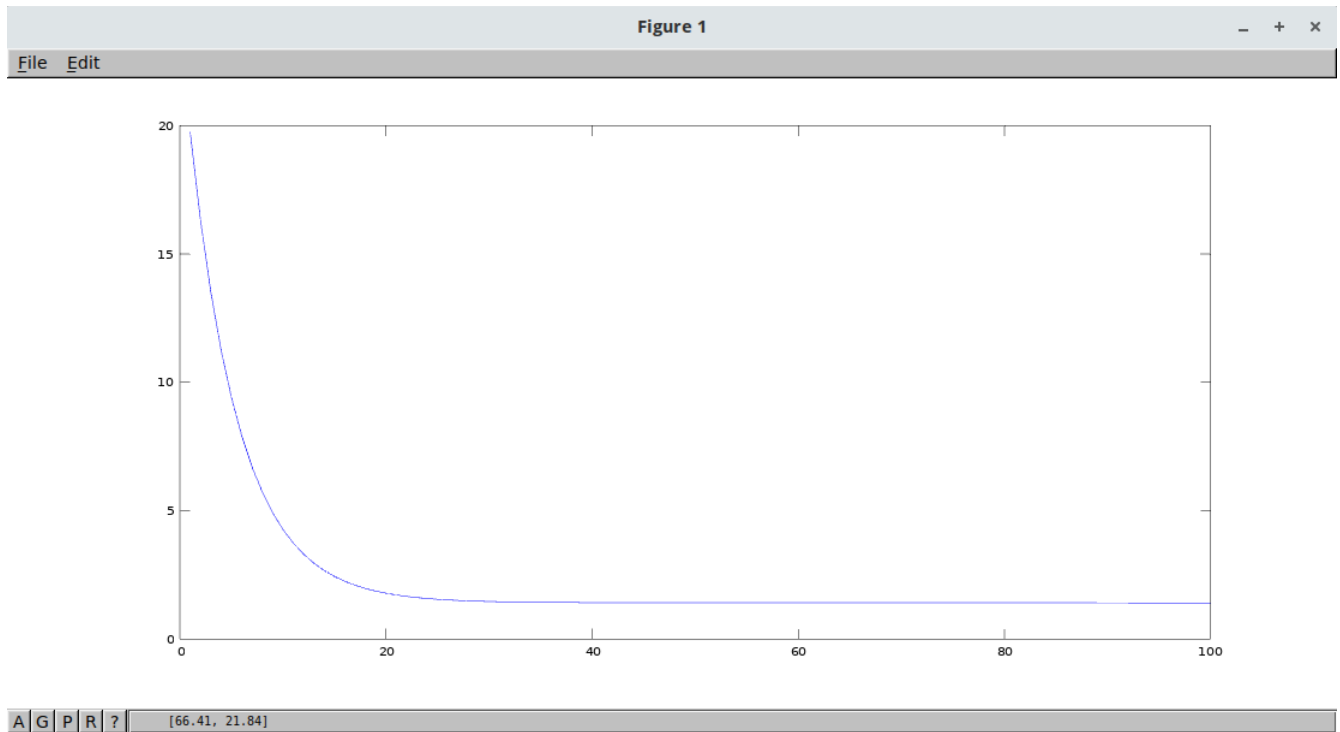
Gráfica 1.9. Error de aprendizaje para la Neurona 4



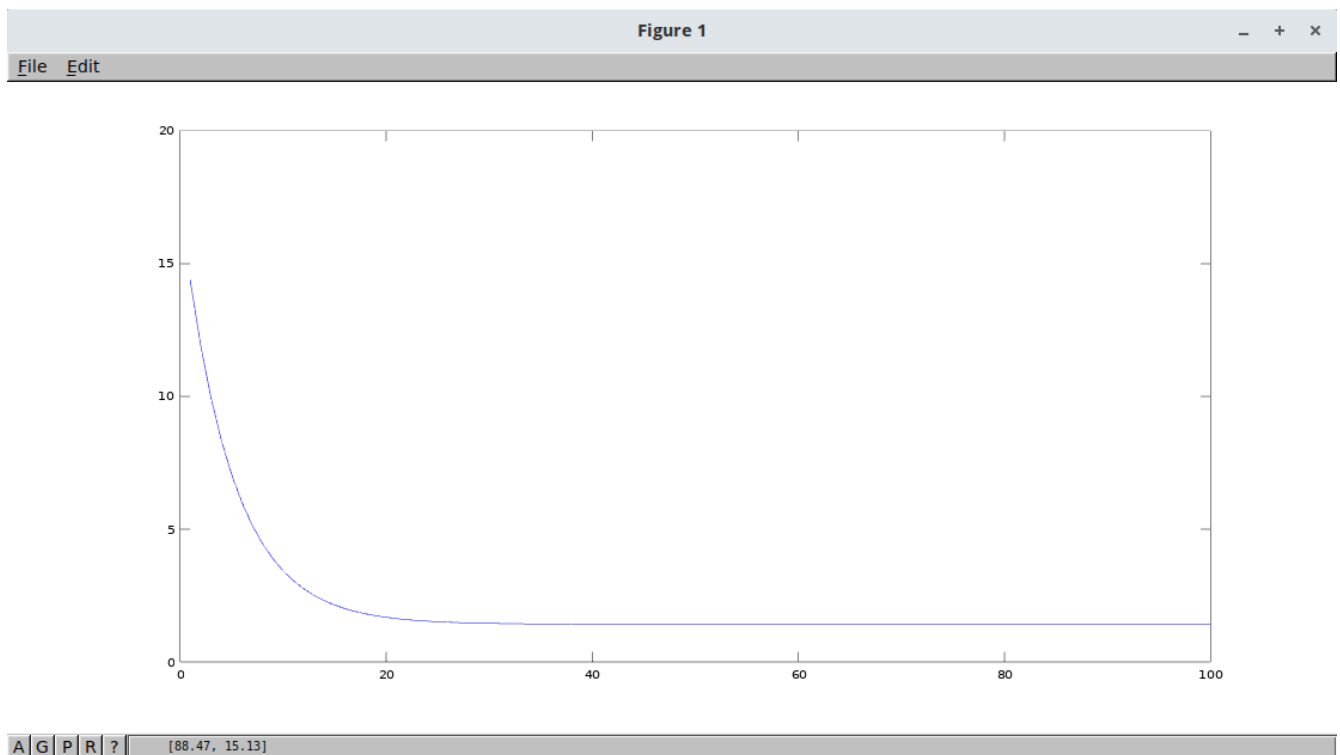
Gráfica 1.10. Error de aprendizaje para la Neurona 5



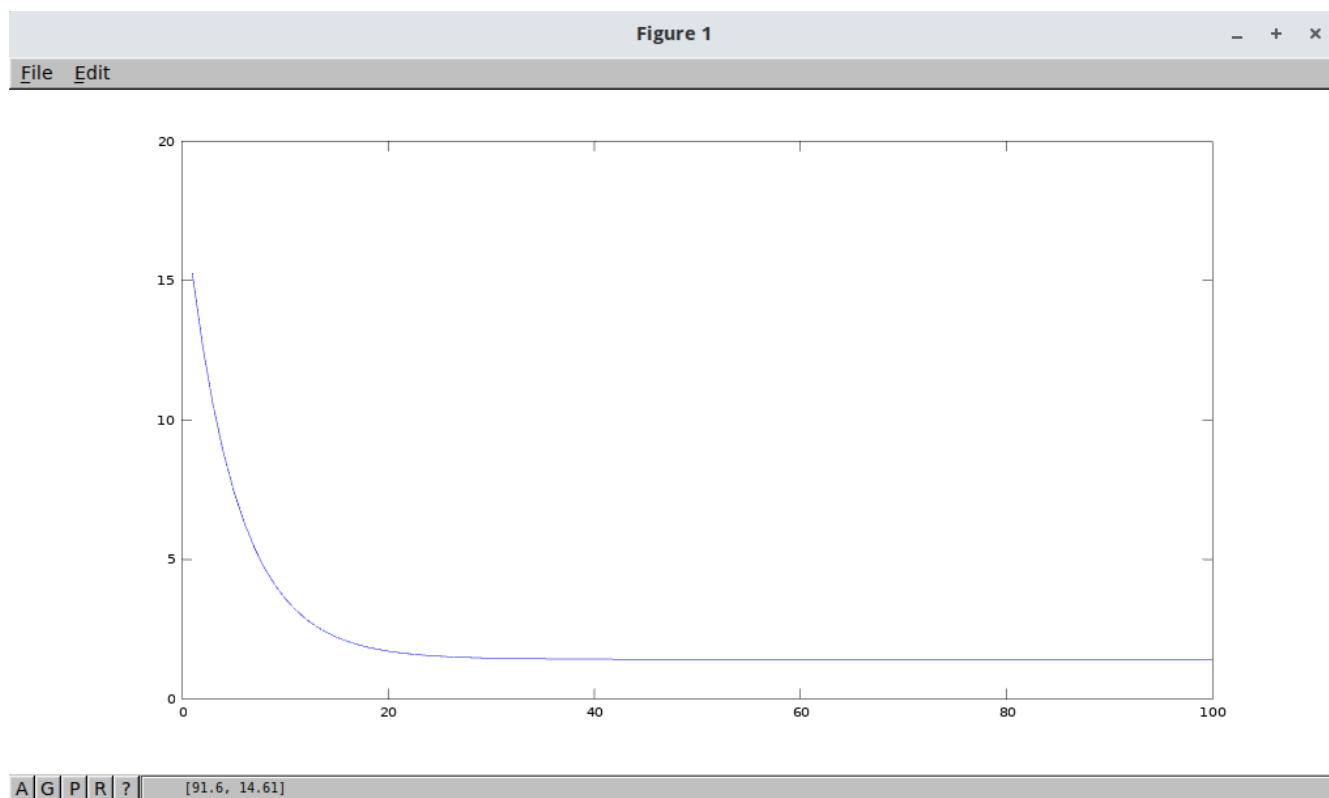
Gráfica 1.11. Error de aprendizaje para la Neurona 6



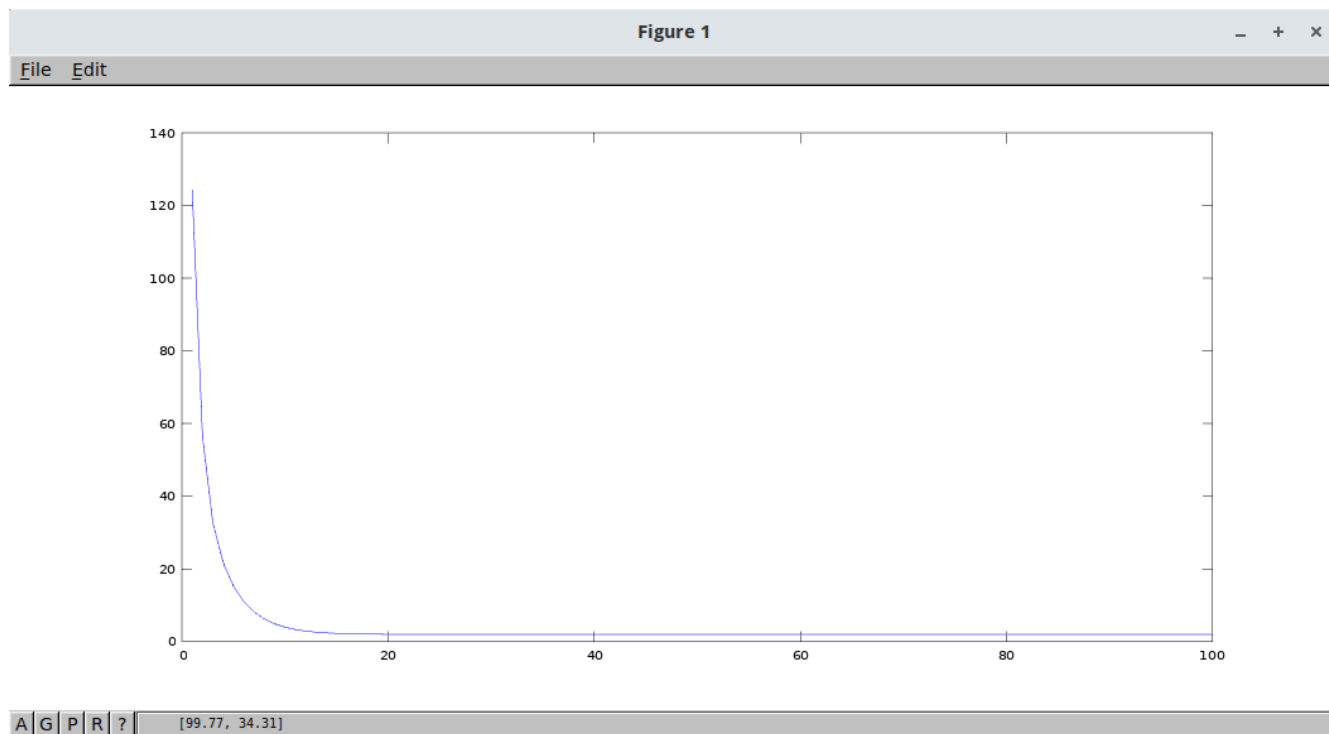
Gráfica 1.12. Error de aprendizaje para la Neurona 7



Gráfica 1.13. Error de aprendizaje para la Neurona 8



Gráfica 1.14. Error de aprendizaje para la Neurona 9



Gráfica 1.15. Error de aprendizaje para la Neurona 10

Caso 3.

La primer tabla de verdad usada para probar la red neuronal fue la siguiente:

X1	x2	x3	x4	f(x1, x2, x3, x4)
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Tabla 1.3. Valores para la función booleana Caso 1.

Configuración de la red:

- Capa de Entrada: 5 Neuronas, 3 entradas una salida por neurona.
- Capa de Salida: 1 Neurona, 5 entradas una salida.

Parámetros de entrenamiento:

- Número de presentaciones: 100
- Coeficiente de aprendizaje: 0.0032

Pesos de las neuronas de la Capa de entrada después del entrenamiento:

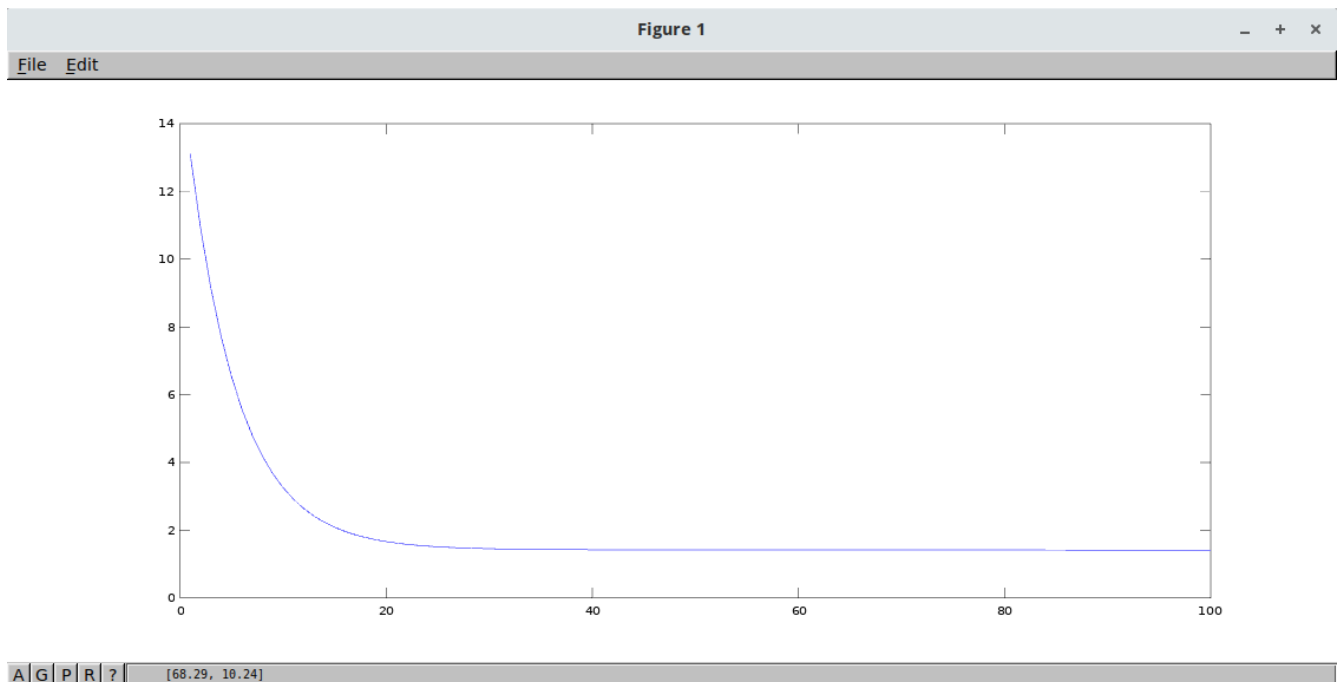
- Neurona 1: $d = -0.874989 + -0.123157X1 + -0.121600X2 + -0.121656X3 + 0.119827X4$
- Neurona 2: $d = -0.874910 + -0.123798X1 + 0.119476X2 + -0.122331X3 + -0.122701X4$
- Neurona 3: $d = -0.874925 + -0.129190X1 + 0.124580X2 + 0.124574X3 + 0.124234X4$

- Neurona 4: $d = -0.874918 + 0.123892X_1 + -0.128293X_2 + 0.125348X_3 + -0.127136X_4$
- Neurona 5: $d = -0.874931 + 0.123848X_1 + 0.125292X_2 + -0.128303X_3 + -0.128750X_4$

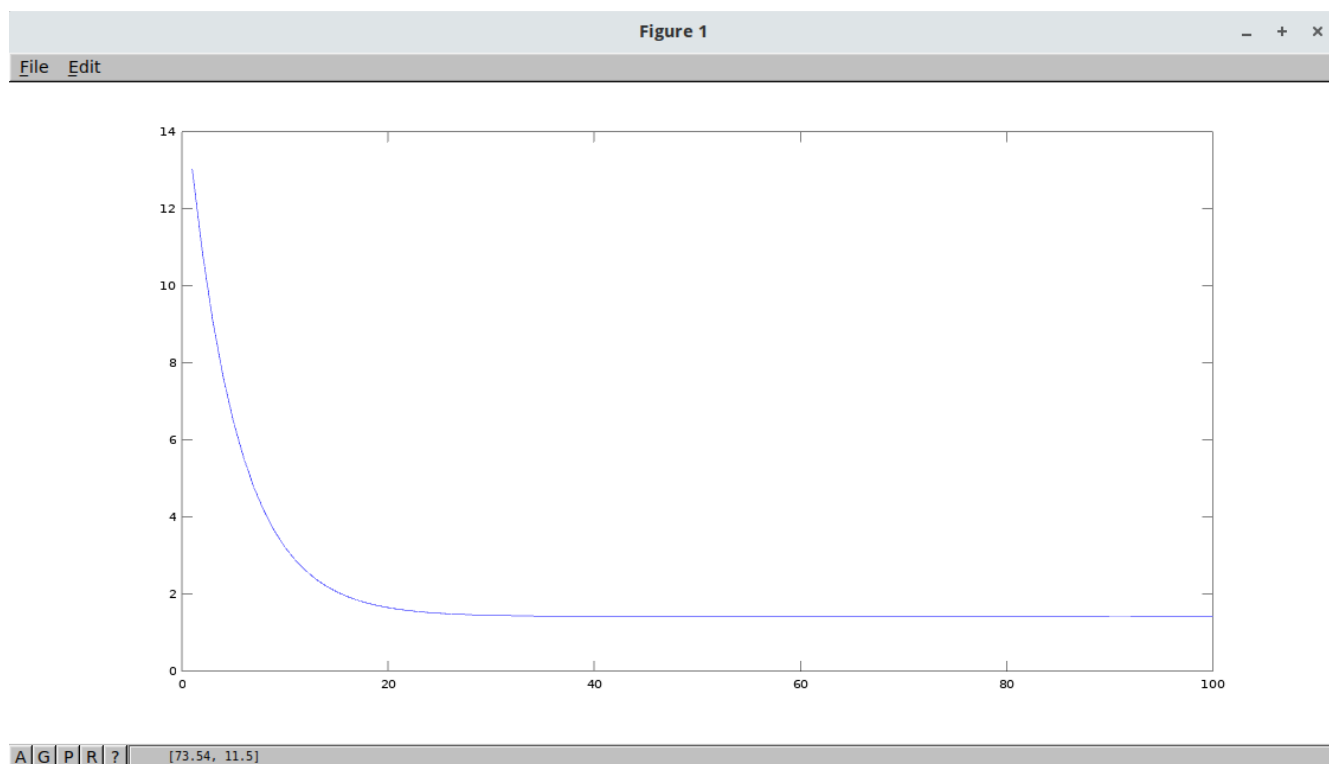
Pesos de la neurona de la Capa de Salida después del entrenamiento:

- Neurona 6: $d = 0.960318 + 0.005038X_1 + 0.005042X_2 + 0.005050X_3 + 0.005064X_4 + 0.005101X_5 + 0.005210X_6 + 0.005518X_7 + 0.006419X_8 + 0.009914X_9$

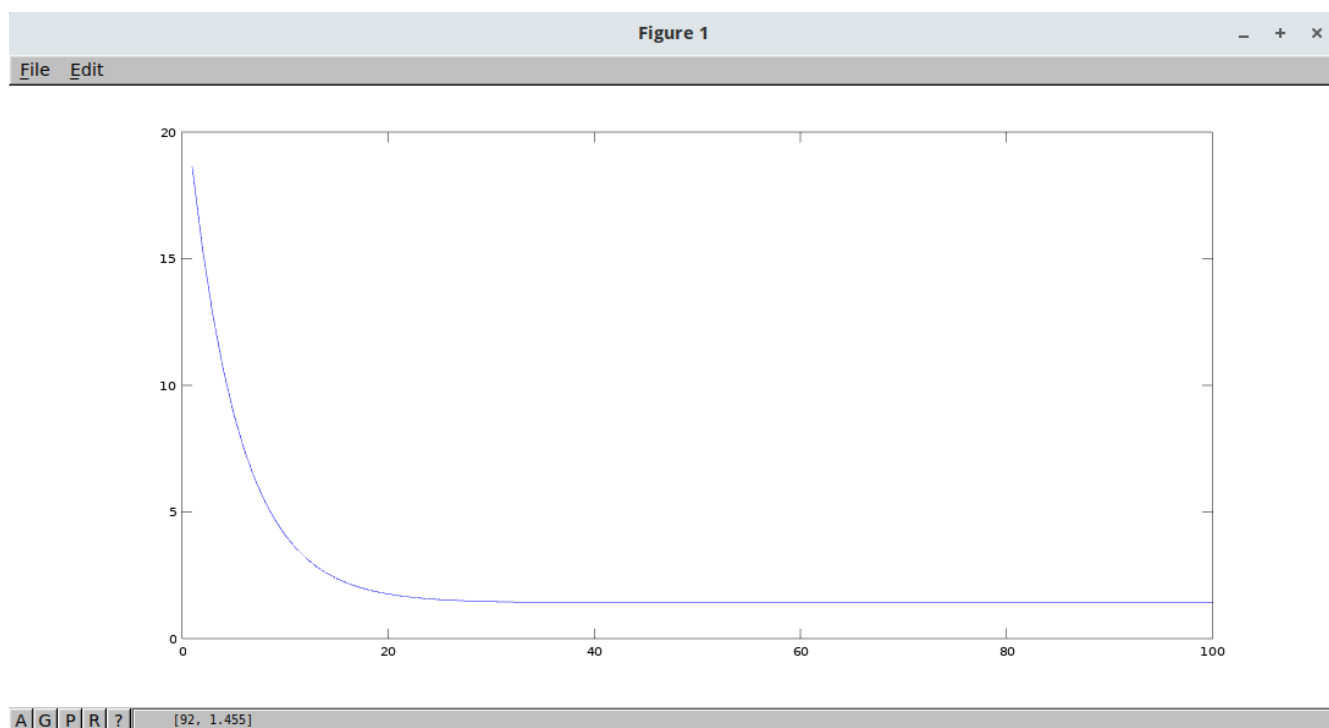
Graficas de error:



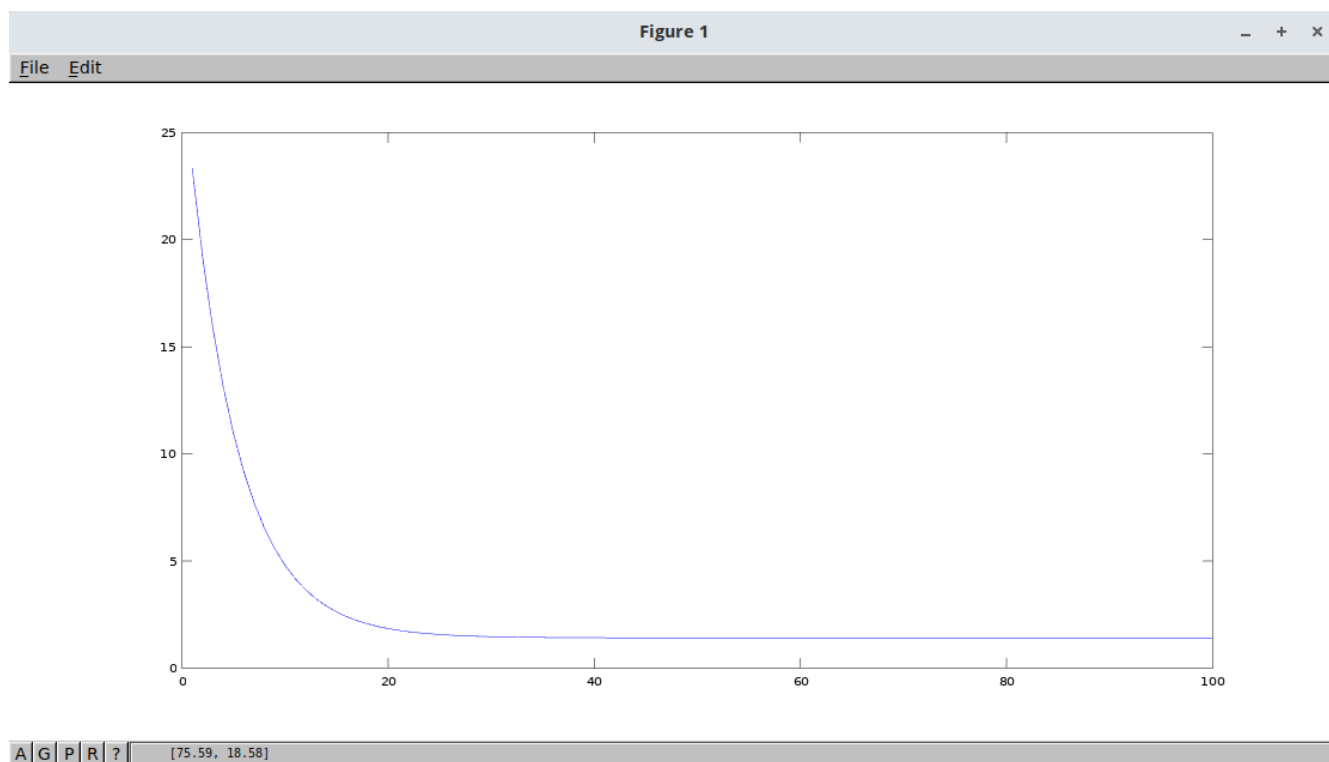
Gráfica 1.16. Error de aprendizaje para la Neurona 1



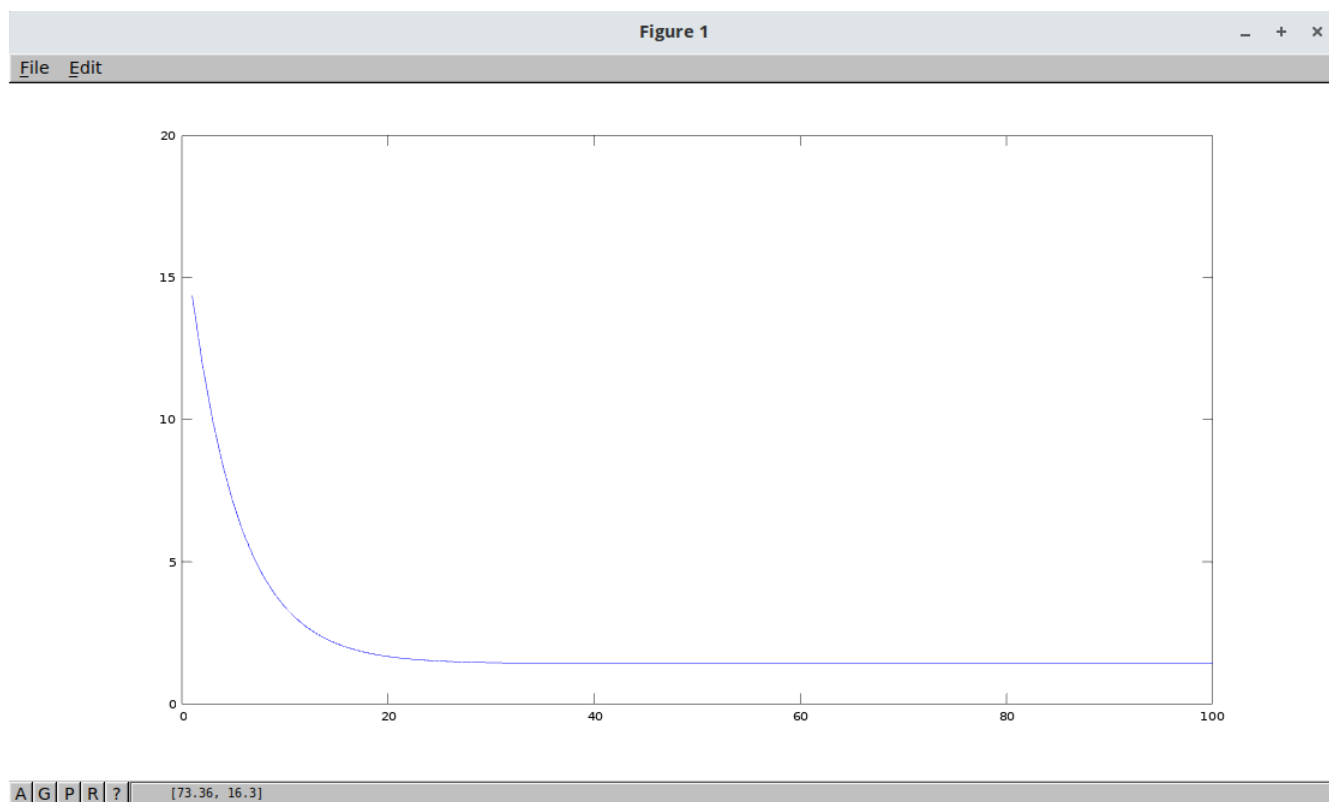
Gráfica 1.17. Error de aprendizaje para la Neurona 2



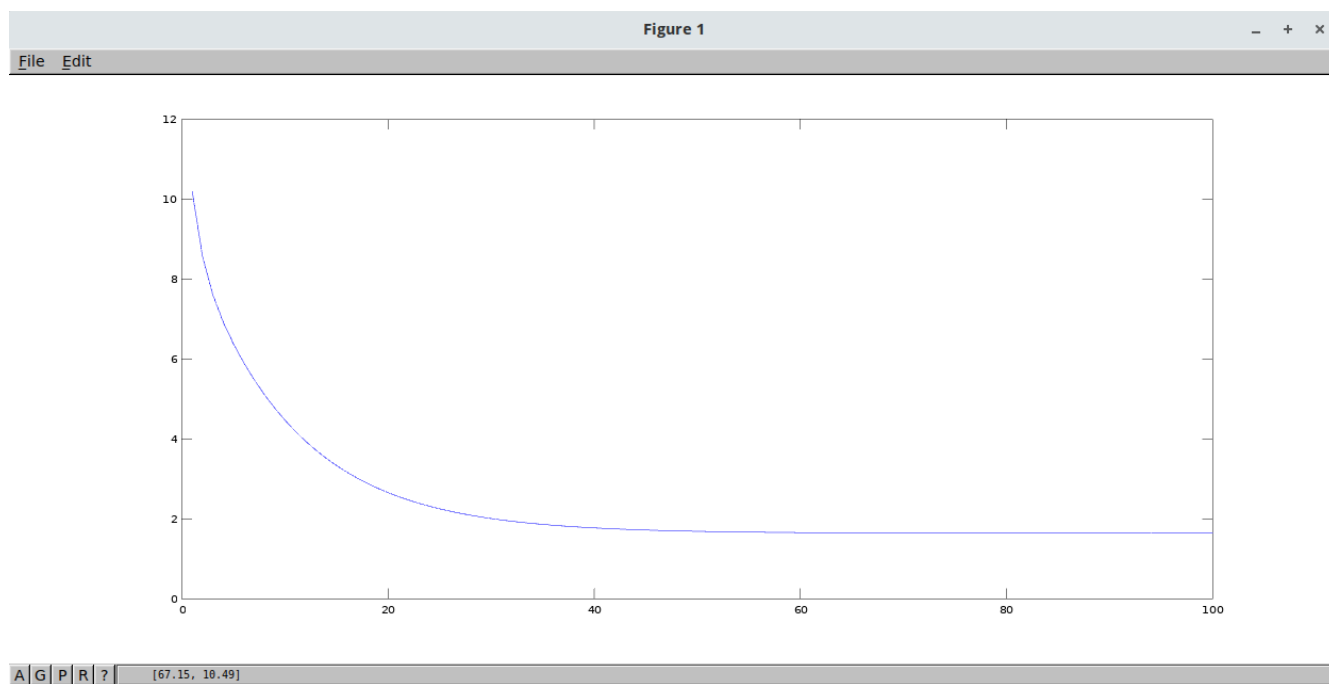
Gráfica 1.18. Error de aprendizaje para la Neurona 3



Gráfica 1.19. Error de aprendizaje para la Neurona 4



Gráfica 1.20. Error de aprendizaje para la Neurona 5



Gráfica 1.21. Error de aprendizaje para la Neurona 6

Referencias:

Una Introducción al Cómputo Neuronal Artificial, Ricardo Pérez Águila, ISBN: 978-1-4135-2424-6