



Instituto Politécnico Nacional

Escuela Superior de Cómputo

Asignatura:

Neural Networks

Grupo: 3CM2

Práctica EXTRA. **Experimentación MLP**

Alumno:

Garcia Garcia Rafael

Profesor: Moreno Armendáriz Marco Antonio

Experimentos:

Primera parte : Capturar la ecuación y generar 100 puntos en el rango indicado

Para esta parte de la practica se tuvo que crear una función llamada *generar_polinomio_y_target* que generara 2 archivos: "Polinomio_Entrada.txt" y "Polinomio_Target.txt". Cada uno de estos archivos nos servirán para el entrenamiento de nuestro MLP. A continuación, les mostraré la salida de esta función que se anexo a nuestro MLP y los vectores de entrada y target para comprobarlo.

Experimento 1 y 2:

<pre>>> generar_polinomio_y_target Introduzca el rango Ejemplos: [2.7, 7.5] >>[2.7, 7.5] Introduzca la ecuación de G(p): Ejemplos: (sin(p)+sin((10/3).*p)) (sin(p)+sin((10/3).*p)+log(p)-(0.84.*p)+3) >>(sin(p)+sin((10/3).*p)) Numero de puntos generados en p 100 Numero de puntos generados en t 100</pre>	Desplegando p	Desplegando t
	2.7000	0.8395
	2.7485	0.6432
	2.7970	0.4392
	2.8455	0.2318
	2.8939	0.0253
	2.9424	-0.1761
	2.9909	-0.3682
	3.0394	-0.5472
	3.0879	-0.7094
	3.1364	-0.8520
	3.1848	-0.9721
	3.2333	-1.0680
	3.2818	-1.1382
	3.3303	-1.1820
	3.3788	-1.1995
	3.4273	-1.1913
	3.4758	-1.1588
	3.5242	-1.1038
	3.5727	-1.0289
	3.6212	-0.9371
	3.6697	-0.8318
	3.7182	-0.7167
	3.7667	-0.5960
	3.8152	-0.4735
	3.8636	-0.3536
	3.9121	-0.2400
	3.9606	-0.1368
	4.0091	-0.0473
	4.0576	0.0254
	4.1061	0.0786
	4.1545	0.1102
	4.2030	0.1189
	4.2515	0.1037
	4.3000	0.0647
	4.3485	0.0022
	4.3970	-0.0825
	4.4455	-0.1876
	4.4939	-0.3107
	4.5424	-0.4489
	4.5909	-0.5987
	4.6394	-0.7564
	4.6879	-0.9181
	4.7364	-1.0795
	4.7848	-1.2366
	4.8333	-1.3850
	4.8818	-1.5209
	4.9303	-1.6405
	4.9788	-1.7405
	5.0273	-1.8181
	5.0758	-1.8708
	5.1242	-1.8969
	5.1727	-1.8952
	5.2212	-1.8655
	5.2697	-1.8078
	5.3182	-1.7231

	5.3667	-1.6131
	5.4152	-1.4798
	5.4636	-1.3260
	5.5121	-1.1550
	5.5606	-0.9704
	5.6091	-0.7762
	5.6576	-0.5766
	5.7061	-0.3758
	5.7545	-0.1782
	5.8030	0.0121
	5.8515	0.1912
	5.9000	0.3553
	5.9485	0.5013
	5.9970	0.6265
	6.0455	0.7286
	6.0939	0.8061
	6.1424	0.8582
	6.1909	0.8846
	6.2394	0.8858
	6.2879	0.8628
	6.3364	0.8174
	6.3848	0.7520
	6.4333	0.6695
	6.4818	0.5730
	6.5303	0.4662
	6.5788	0.3531
	6.6273	0.2377
	6.6758	0.1242
	6.7242	0.0164
	6.7727	-0.0817
	6.8212	-0.1664
	6.8697	-0.2347
	6.9182	-0.2838
	6.9667	-0.3114
	7.0152	-0.3158
	7.0636	-0.2963
	7.1121	-0.2523
	7.1606	-0.1842
	7.2091	-0.0931
	7.2576	0.0193
	7.3061	0.1509
	7.3545	0.2988
	7.4030	0.4597
	7.4515	0.6300
	7.5000	0.8056

Salida de terminal 1. $G(p) = (\sin(p) + \sin((10/3) \cdot p))$

>> generar_polinomio_y_target	Desplegando p	Desplegando t
Introduzca el rango	2.7000	2.5648
Ejemplos:	2.7485	2.3455
[2.7, 7.5]	2.7970	2.1183
	2.8455	1.8873
>>[2.7, 7.5]	2.8939	1.6570
Introduzca la ecuación de G(p):	2.9424	1.4315
Ejemplos:	2.9909	1.2150
(sin(p)+sin((10/3).*p))	3.0394	1.0114
(sin(p)+sin((10/3).*p)+log(p)-(0.84.*p)+3)	3.0879	0.8242
	3.1364	0.6566
>>(sin(p)+sin((10/3).*p)+log(p)-	3.1848	0.5110
(0.84.*p)+3)	3.2333	0.3895
Numero de puntos generados en p	3.2818	0.2935
100	3.3303	0.2236
	3.3788	0.1798
Numero de puntos generados en t	3.4273	0.1615
100	3.4758	0.1674
	3.5242	0.1955
	3.5727	0.2434
	3.6212	0.3079
	3.6697	0.3858
	3.7182	0.4732
	3.7667	0.5662
	3.8152	0.6607
	3.8636	0.7526
	3.9121	0.8379

3.9606	0.9127
4.0091	0.9737
4.0576	1.0176
4.1061	1.0419
4.1545	1.0446
4.2030	1.0241
4.2515	0.9797
4.3000	0.9113
4.3485	0.8193
4.3970	0.7049
4.4455	0.5701
4.4939	0.4171
4.5424	0.2490
4.5909	0.0690
4.6394	-0.1189
4.6879	-0.3109
4.7364	-0.5028
4.7848	-0.6904
4.8333	-0.8695
4.8818	-1.0361
4.9303	-1.1866
4.9788	-1.3175
5.0273	-1.4261
5.0758	-1.5099
5.1242	-1.5672
5.1727	-1.5969
5.2212	-1.5986
5.2697	-1.5724
5.3182	-1.5193
5.3667	-1.4409
5.4152	-1.3393
5.4636	-1.2173
5.5121	-1.0782
5.5606	-0.9256
5.6091	-0.7634
5.6576	-0.5959
5.7061	-0.4274
5.7545	-0.2620
5.8030	-0.1040
5.8515	0.0426
5.9000	0.1742
5.9485	0.2877
5.9970	0.3803
6.0455	0.4497
6.0939	0.4945
6.1424	0.5138
6.1909	0.5073
6.2394	0.4756
6.2879	0.4196
6.3364	0.3412
6.3848	0.2427
6.4333	0.1270
6.4818	-0.0028
6.5303	-0.1428
6.5788	-0.2892
6.6273	-0.4380
6.6758	-0.5850
6.7242	-0.7262
6.7727	-0.8578
6.8212	-0.9762
6.8697	-1.0781
6.9182	-1.1609
6.9667	-1.2222
7.0152	-1.2605
7.0636	-1.2748
7.1121	-1.2647
7.1606	-1.2305
7.2091	-1.1734
7.2576	-1.0950
7.3061	-0.9975
7.3545	-0.8837
7.4030	-0.7570
7.4515	-0.6209
7.5000	-0.4794

Salida de terminal 2. $G(p) = (\sin(p) + \sin((10/3) \cdot p) + \log(p) - (0.84 \cdot p) + 3)$

Segunda parte: Realizar el adecuado aprendizaje de las ecuaciones, llegando al menos a un error de prueba de 1×10^{-3}

Para la primera función: $G(p) = (\sin(p) + \sin((10/3) \cdot p))$

La siguiente es la tabla de experimentación

Primera parte:

Intento	Early Stopping	Arquitectura	# de épocas	Alpha	Error de entrenamiento	Error de validación	Error de prueba
1	NO	V1 = [1,7,7,1] V2 = [2,2,1]	12,000	0.009	0.5729	0.7058	1.2751
2	NO	V1 = [1,10,7,1] V2 = [3,3,1]	12,000	0.009	0.6756	0.6635	0.6682
3	NO	V1 = [1,8,5,1] V2 = [2, 2, 1]	71,999	0.009	3.283×10^{-5}	5.359×10^{-5}	3.1321×10^{-5}

Tabla 1. Experimentos realizados para $G(p) = (\sin(p) + \sin((10/3) \cdot p))$

Salida de la terminal:

```

Error: 0.000033 de la epoca 71997
Error: 0.000033 de la epoca 71998
Error: 0.000033 de la epoca 71999
Prueba de generalización superada con un error de:
  3.1321e-05

Error final
  3.2833e-05

Valores de pesos finales:
W1
-0.8867
-1.4270
-1.0759
-2.3908
-1.4436
-3.4046
 1.2302
 0.8592

b_1
 6.1107
 5.9155
 0.7134
 6.2323
10.4573
12.6893
-7.1565
-3.2358

W_2
 3.0478  3.0775 -0.9754  1.0321  4.3466  8.8322 -4.7283 -3.2542
 0.1924  0.1843  0.7070  6.3966  0.1628 -3.3124  1.5572 -1.3339
-3.5488 -3.1608 -0.4512 -1.4774 -5.8096 -3.1450  4.5512  1.7704
-3.3602 -2.6654  0.4485  0.5847 -5.0714 -1.8381  2.8159  1.0776
-2.1466 -3.1079  0.0459 -1.3366 -2.3715 -7.8788  4.6433  3.8487

```

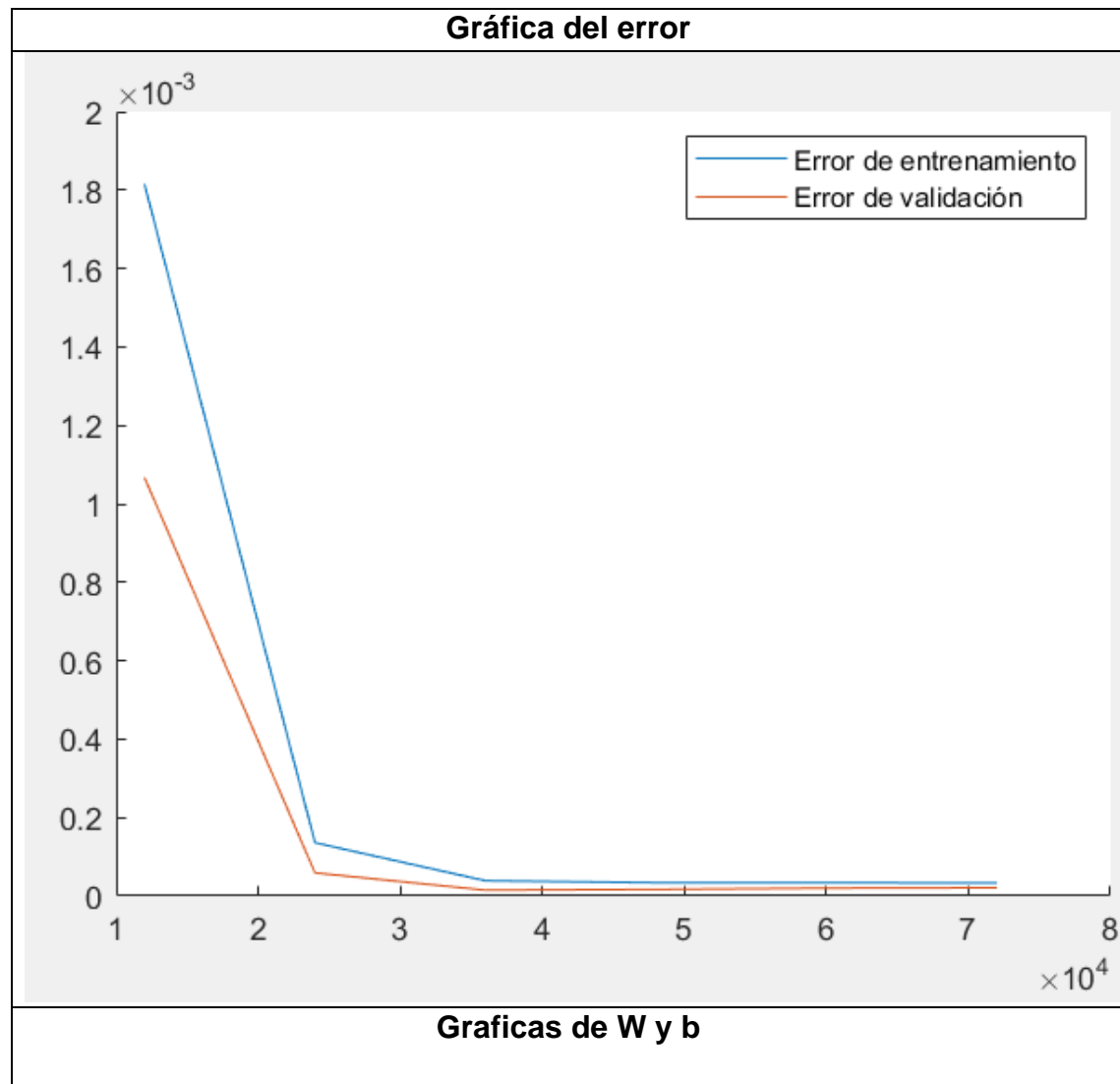
```

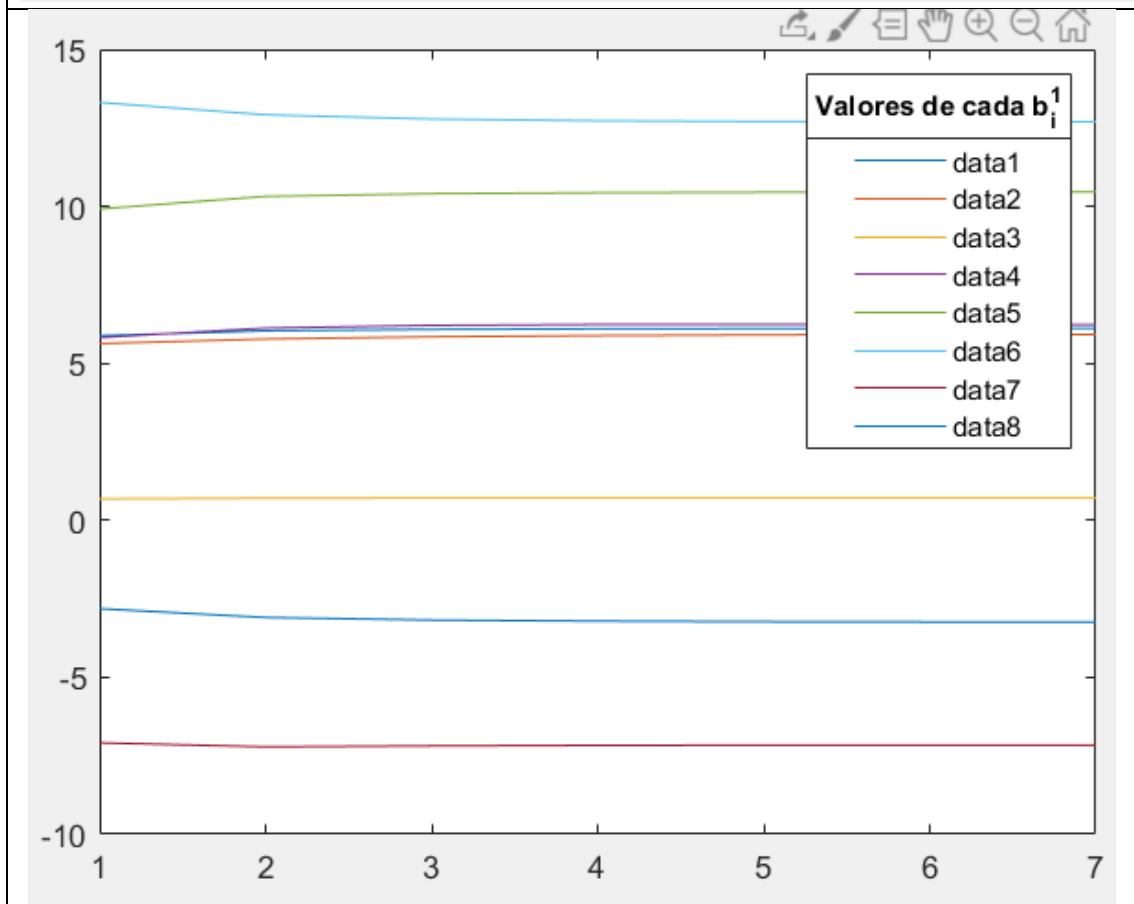
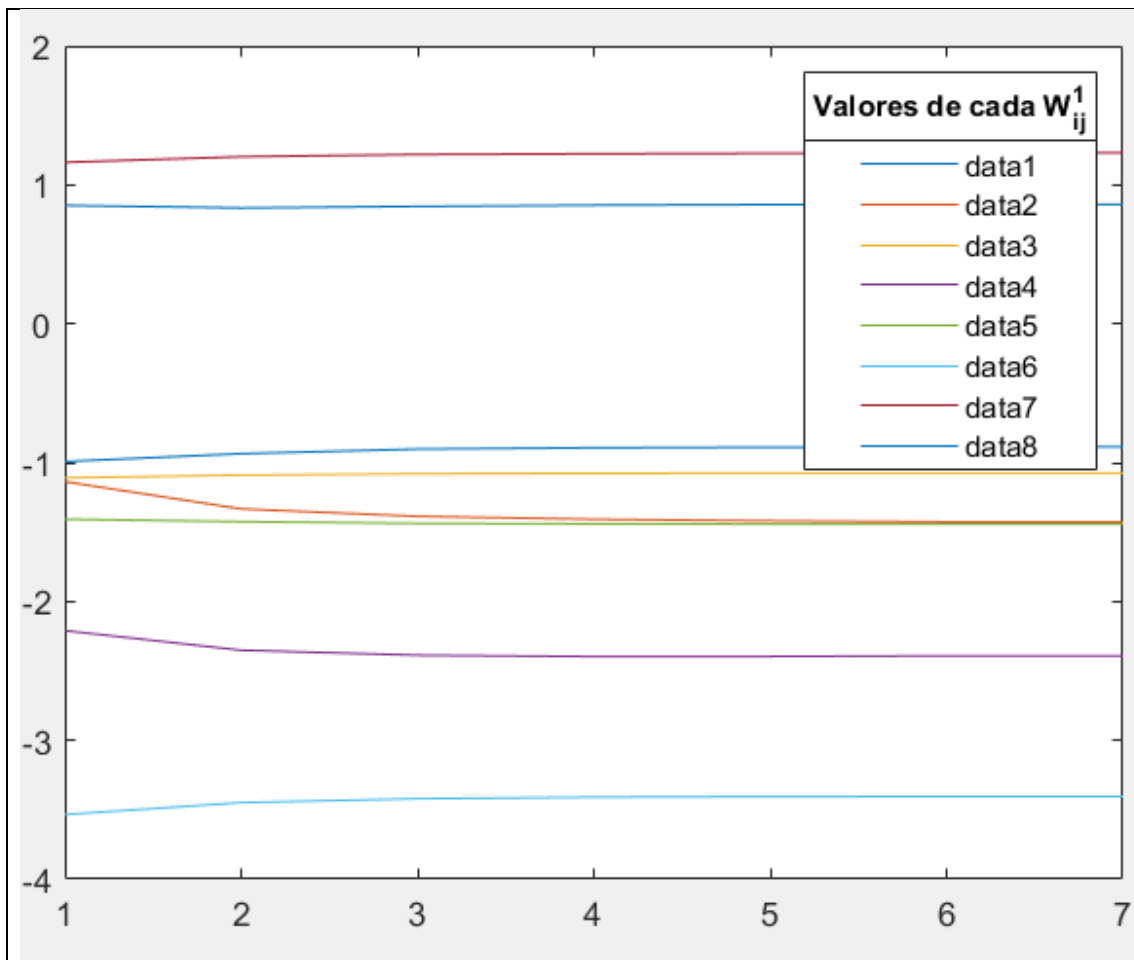
b_2
-1.8909
1.5317
1.4095
-0.8535
1.2602

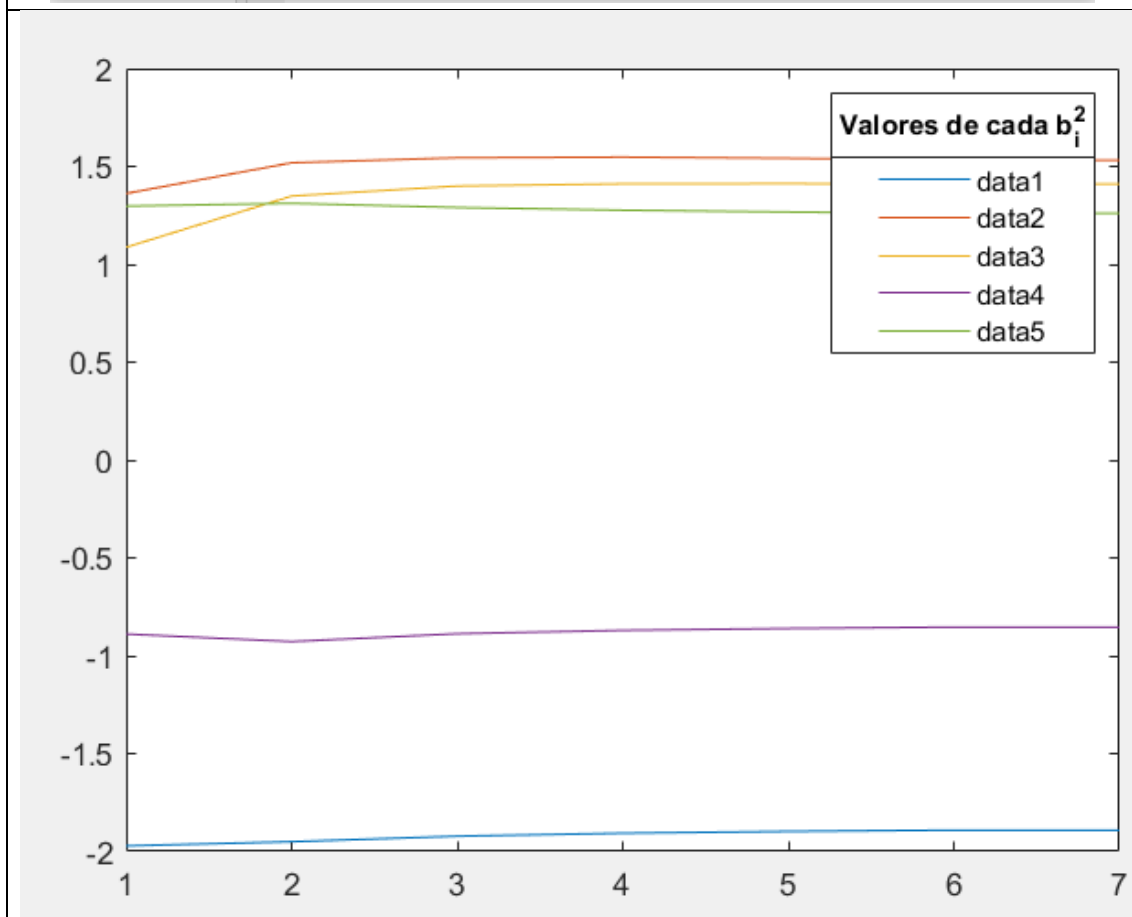
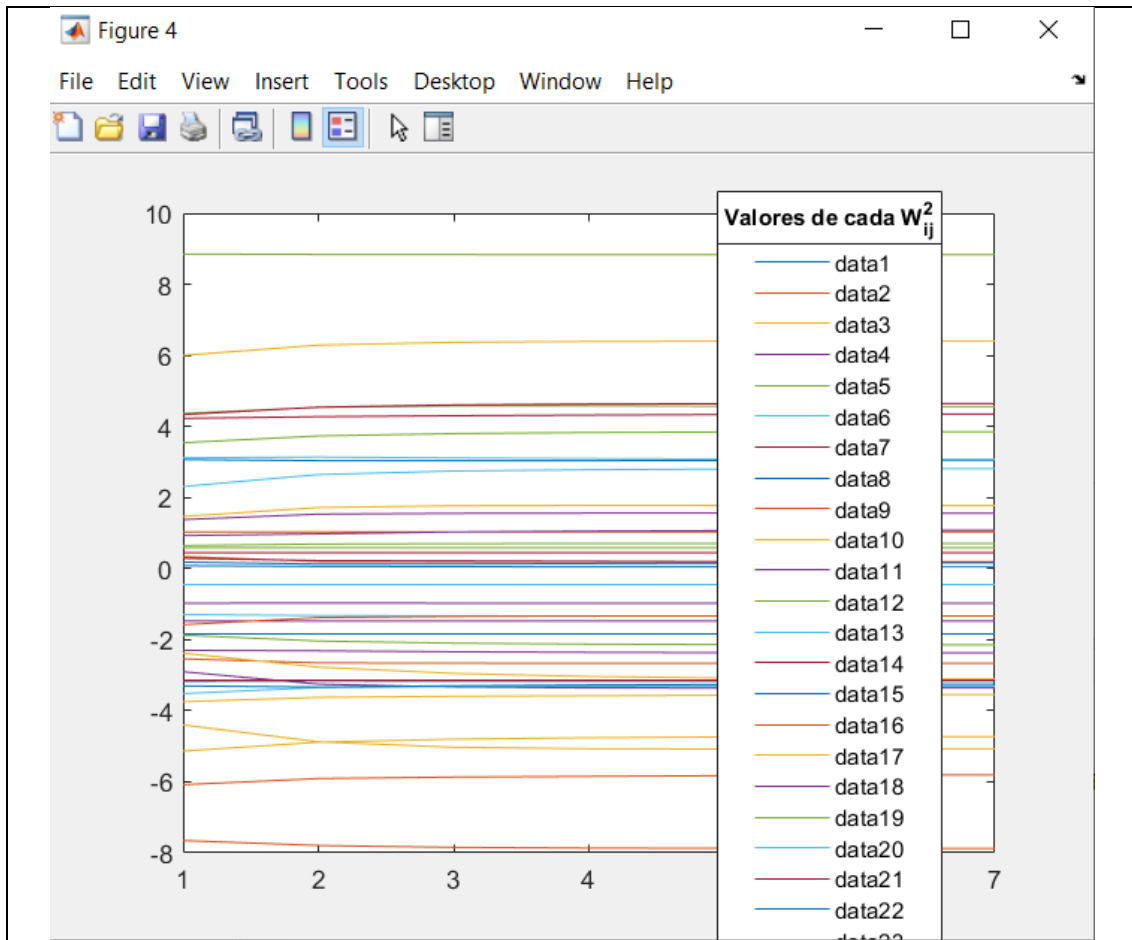
```

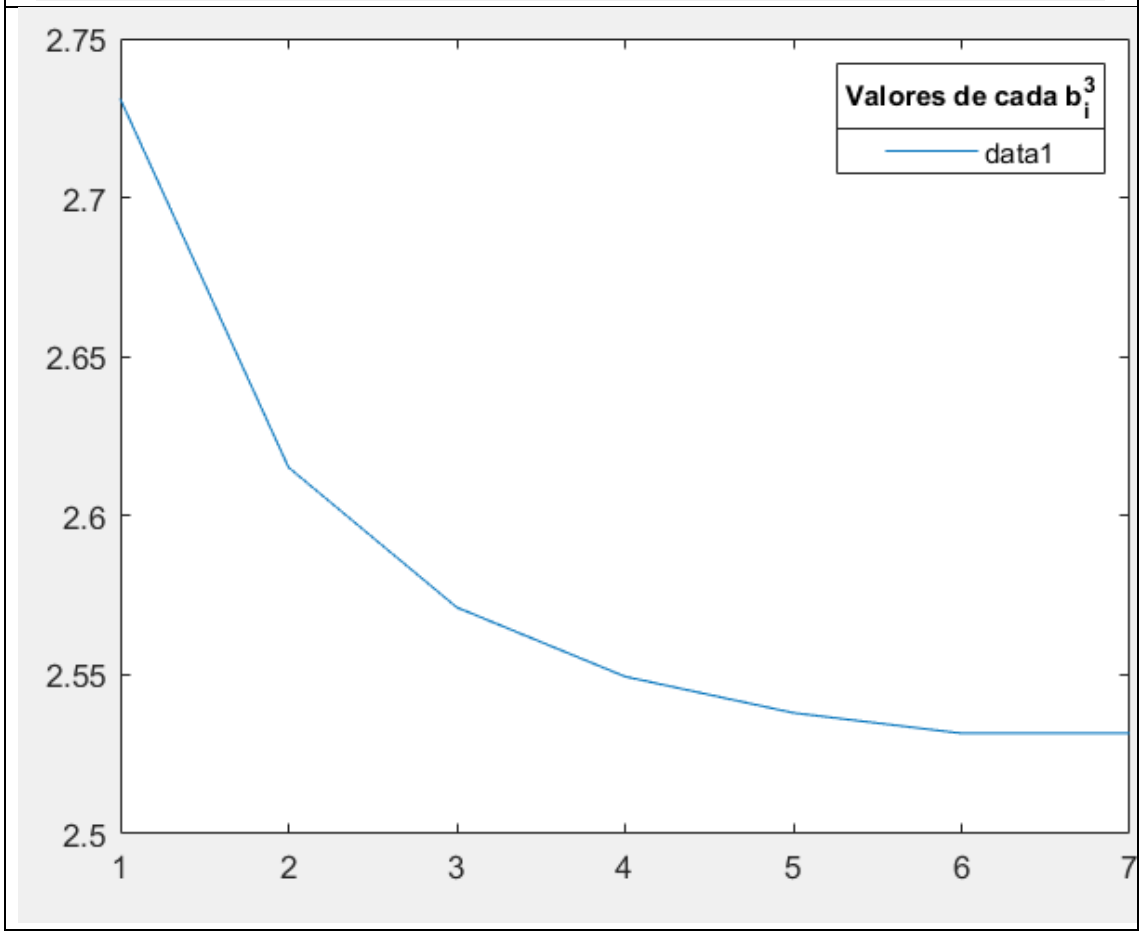
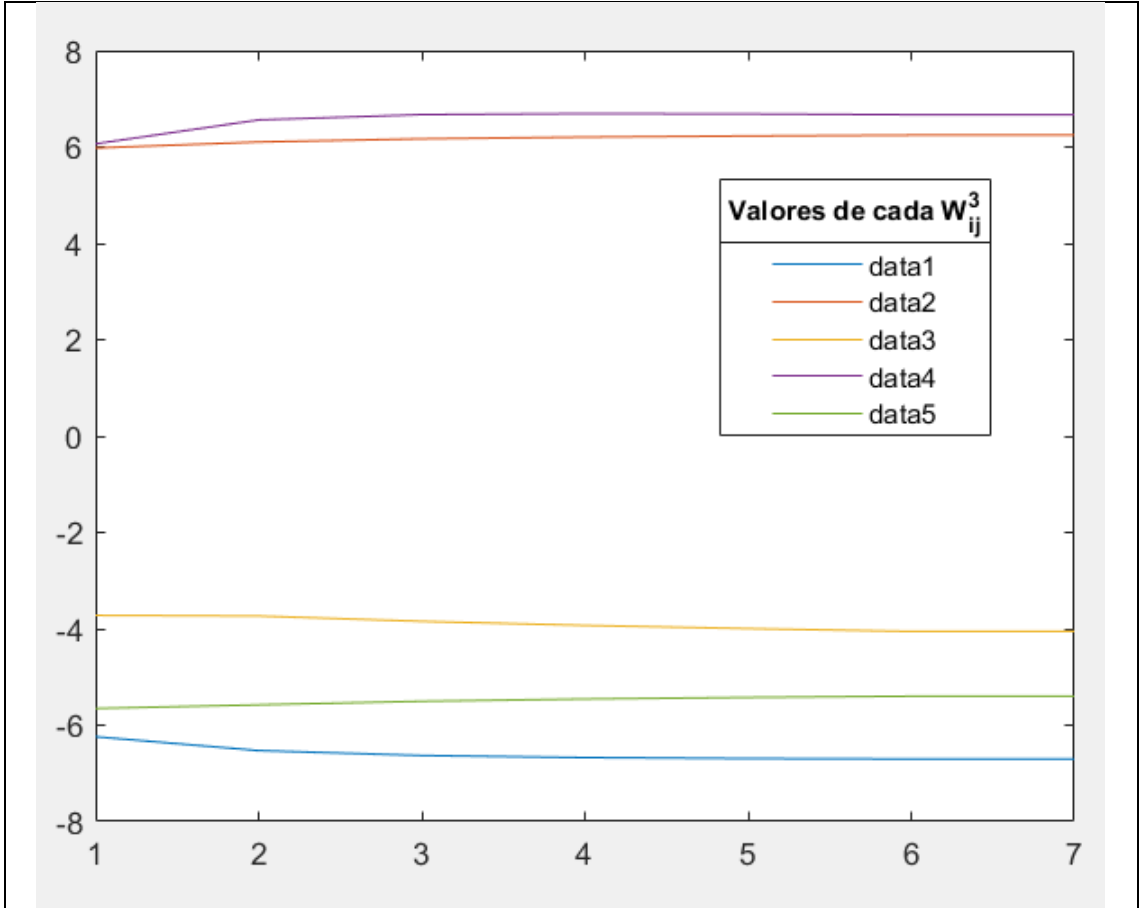
Tabla 2. Salida de la terminal del programa para $G(p) = (\sin(p) + \sin((10/3) \cdot p))$

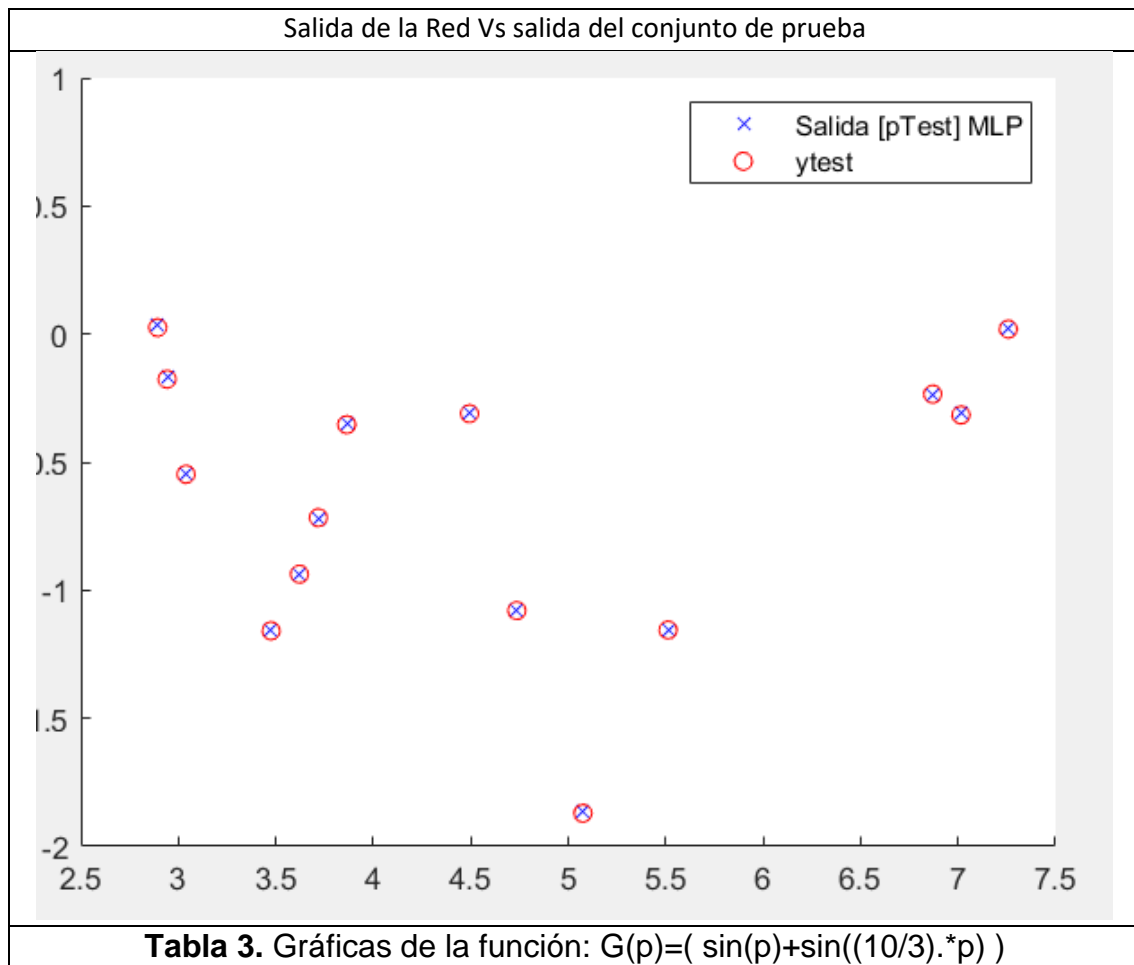
Gráficas:











Segunda parte:

Para la Segunda función: $G(p) = (\sin(p) + \sin((10/3) \cdot p) + \log(p) - (0.84 \cdot p) + 3)$

La siguiente es la tabla de experimentación

Intento	Early Stopping	Arquitectura	# de épocas	Alpha	Error de entrenamiento	Error de validación	Error de prueba
1	NO	V1 = [1,8,5,1] V2 = [2, 2, 1]	239999	0.009	4.8912×10^{-5}	1.1779×10^{-4}	1.1538×10^{-4}

Tabla 4. Experimentos realizados para
 $G(p) = (\sin(p) + \sin((10/3) \cdot p) + \log(p) - (0.84 \cdot p) + 3)$

Salida de la terminal:

```
Error: 0.000049 de la epoca 239997
Error: 0.000049 de la epoca 239998
Error: 0.000049 de la epoca 239999
Prueba de generalización superada con un error de:
  1.1538e-04

Error final
  4.8912e-05

Valores de pesos finales:
```

```

W1
-2.4232
2.2365
-0.6567
-0.6496
-0.9116
-1.0994
-1.6990
-1.7698

b_1
18.9464
-7.9844
3.2447
2.5353
4.9443
-0.0979
-0.1305
10.3432

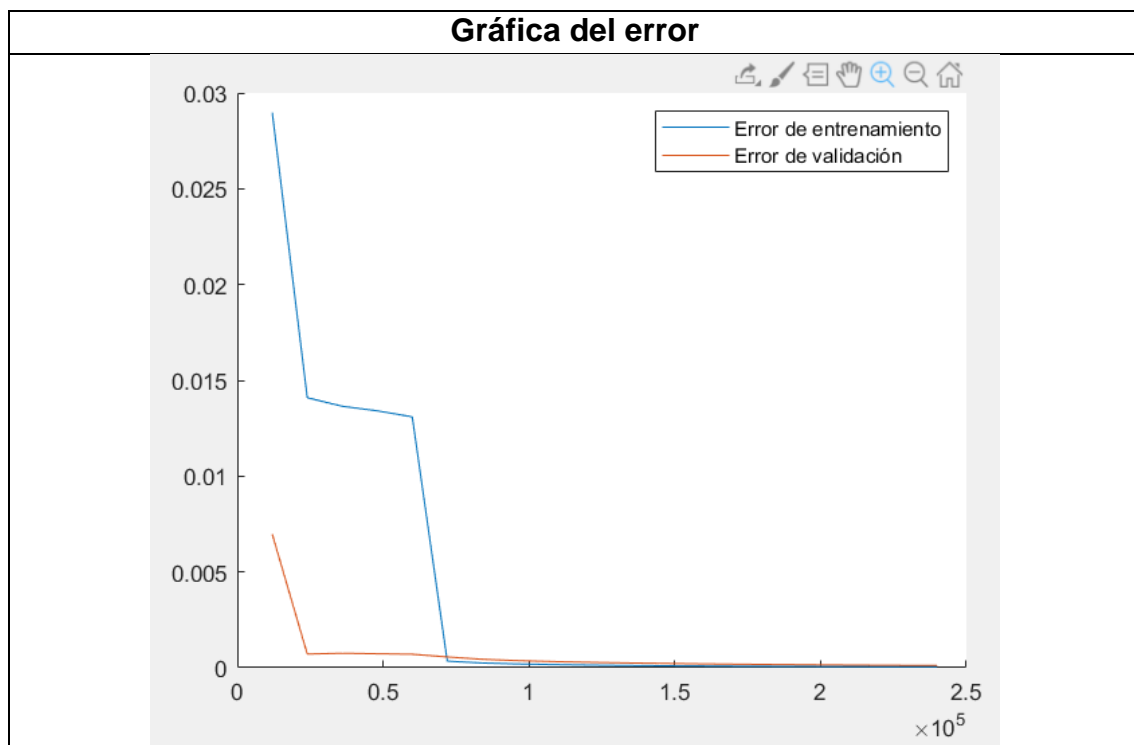
W_2
10.9100 -6.4680 2.3300 1.4537 2.8782 0.1864 0.7554 6.7543
2.2111 -7.9754 1.9428 1.7977 1.4105 0.1663 -0.0606 0.3593
-3.4817 8.5269 -1.3310 -2.2177 -2.8274 0.0640 0.5095 -5.2612
-6.8431 7.2922 -2.0649 -0.6973 -1.5089 -0.2611 0.6095 -5.2589
3.9994 -3.9761 -0.2530 0.4681 0.1796 0.6005 -0.2692 -1.0815

b_2
-6.7862
-5.9162
3.5652
4.5298
0.5552

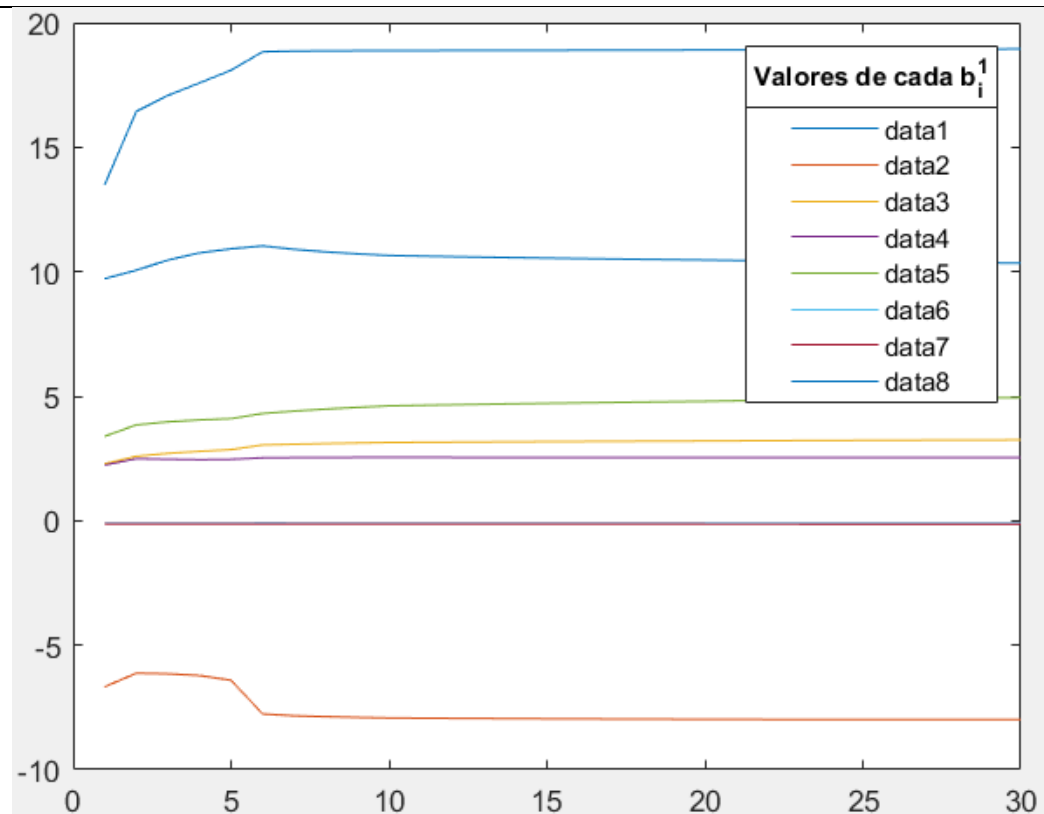
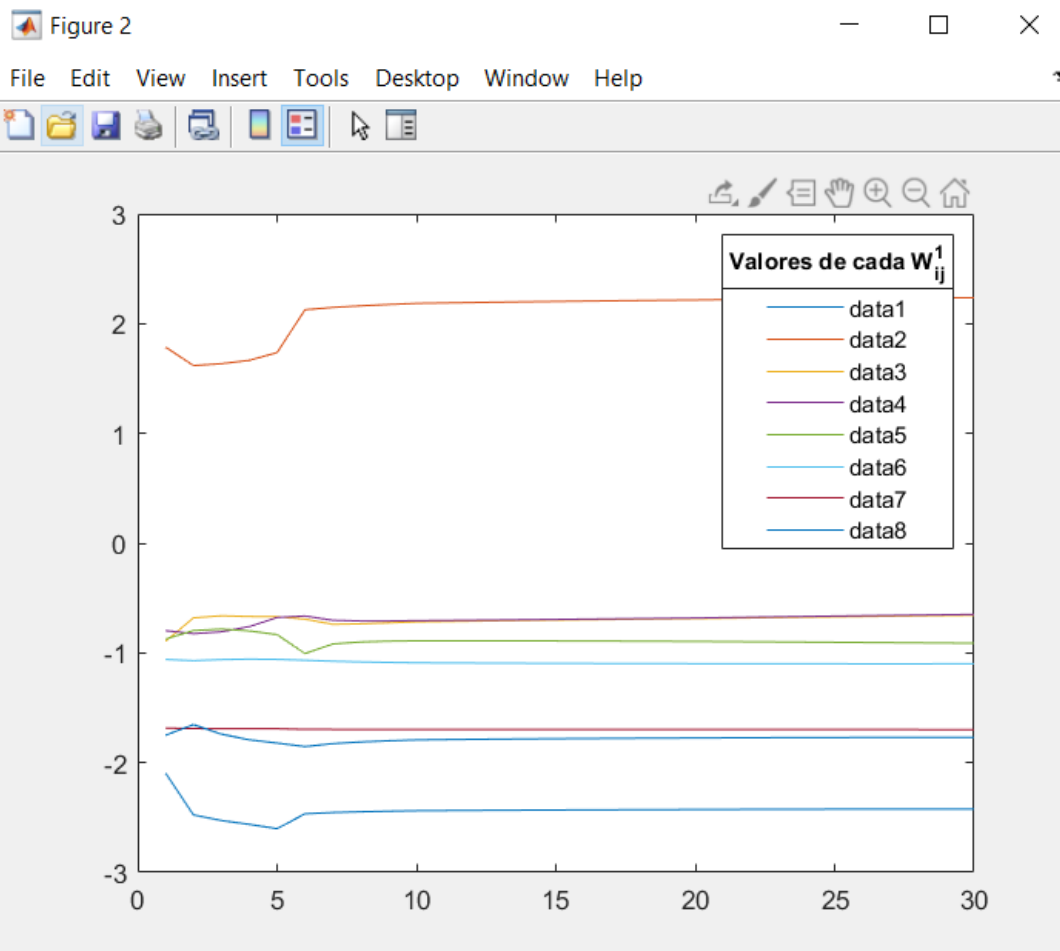
```

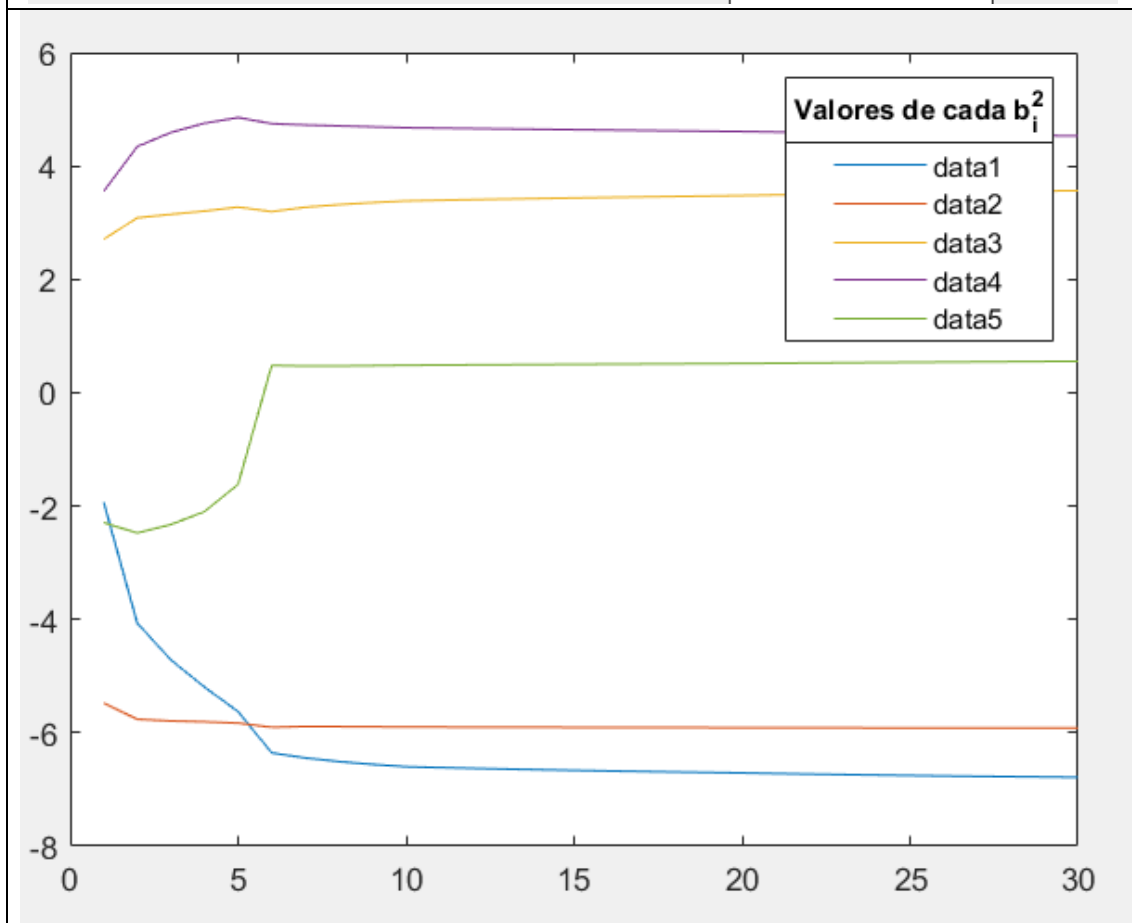
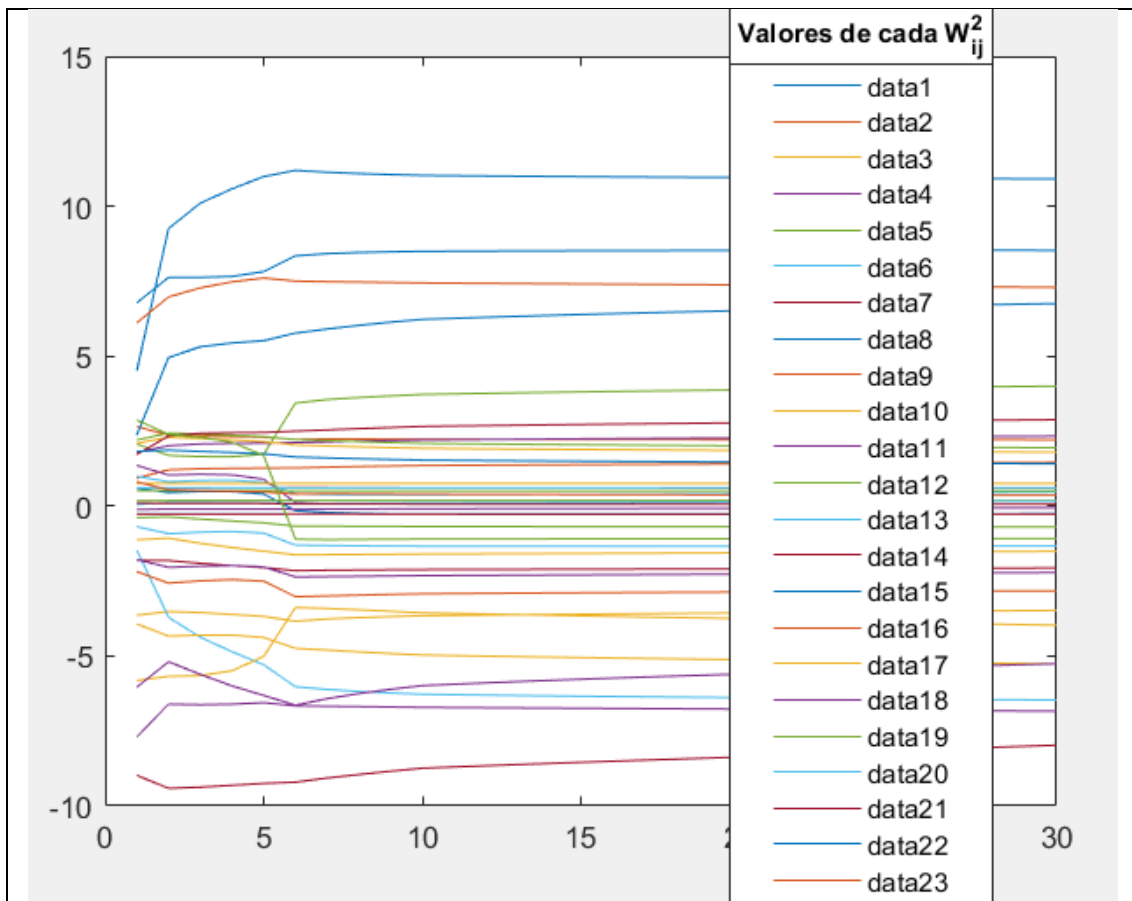
Tabla 5. Salida de la terminal del programa para
 $G(p) = (\sin(p) + \sin((10/3) \cdot p))$

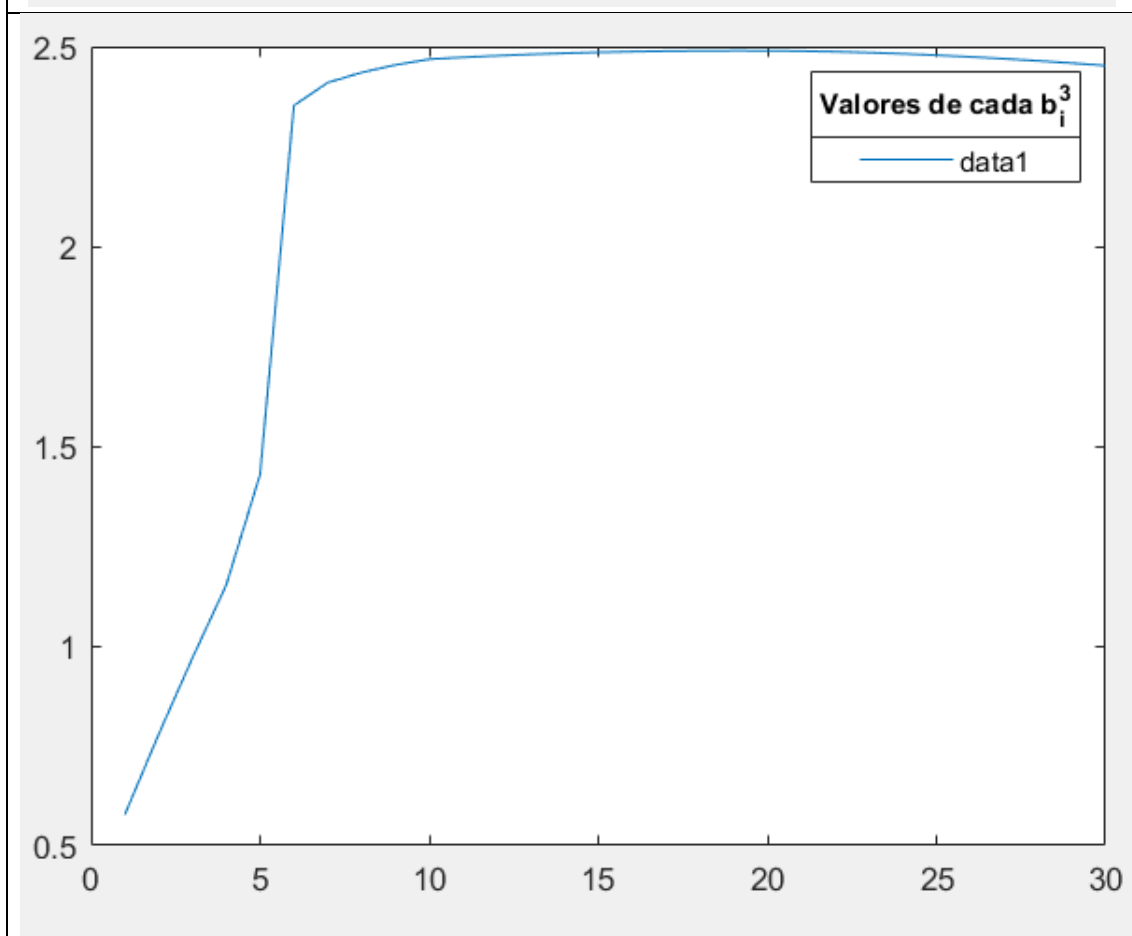
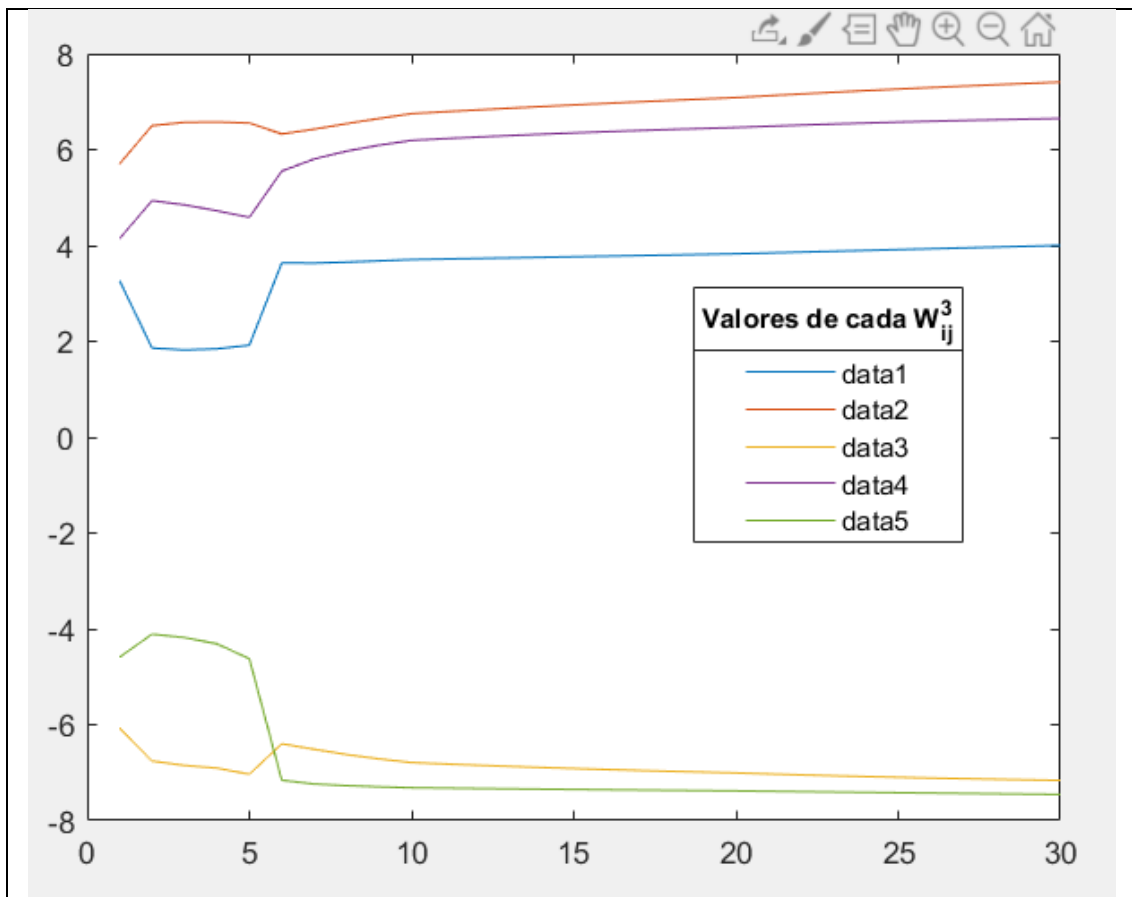
Gráficas:

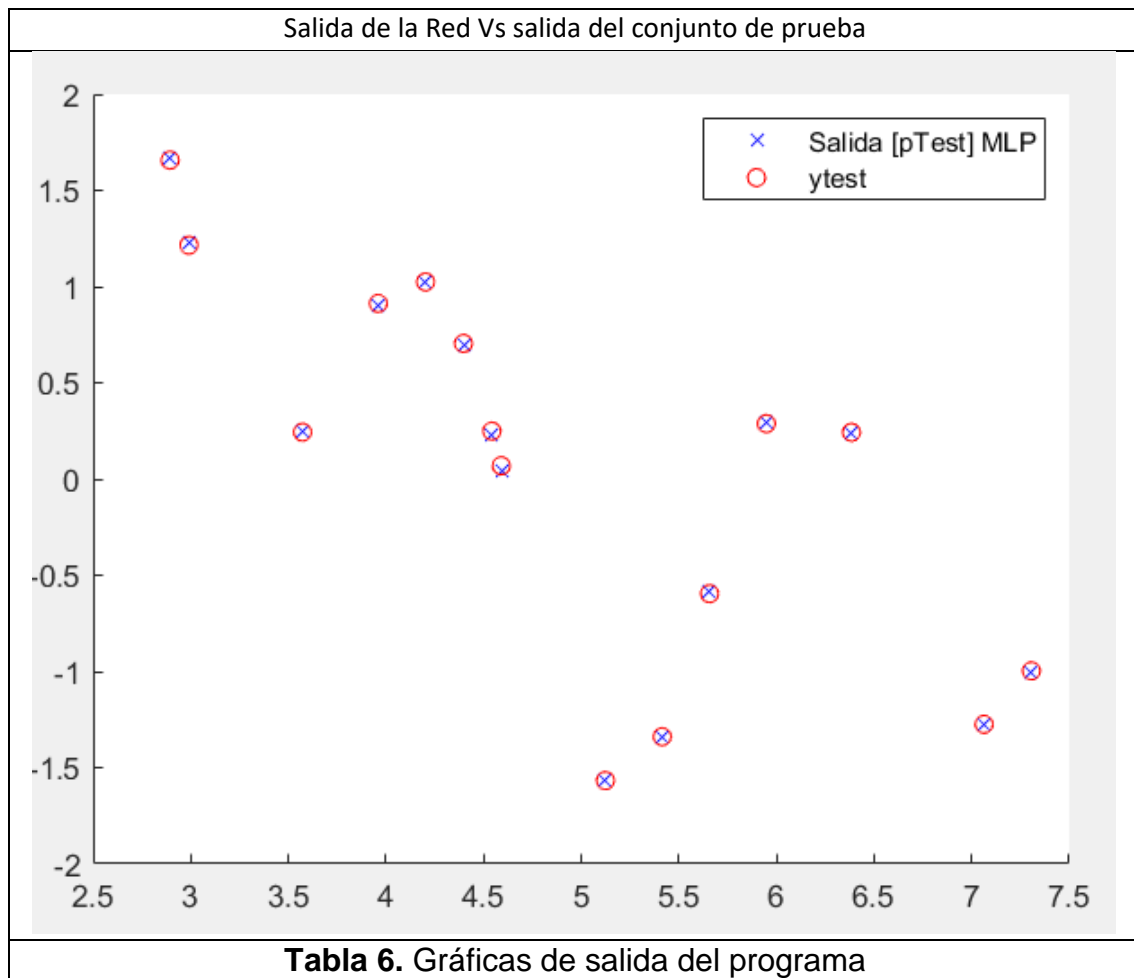


Graficas de W y b









Discusión

Debido a que el MLP se había implementado de una excelente manera en la práctica 5, fue muy fácil agregar la parte de la ecuación para generar nuevas funciones de las cuales aprenda. También no fue muy difícil jugar con la red neuronal para obtener los resultados, debido a que tener muchas neuronas y 2 capas personalizables da muy buenos resultados para funciones seno.

Como era de esperarse tuvimos buenos resultados y nuestro MLP funcionó de manera eficiente. Algo interesante es de que en la segunda función copie la misma configuración que en la primera función pero al ser más compleja esta tardó mucha más épocas para converger, además de que tuvo un error ligeramente más grande pero ambos errores dentro de los objetivos esperados en la clase.

Conclusiones:

Haber desarrollado esta práctica me fue de gran utilidad para agregar funcionalidades extras a mi MLP (Como la introducción de la ecuación y rango) y también para verificar que puede funcionar para diversas funciones el mismo

perceptrón, dependiendo de la complejidad de la función tardará más o menos iteraciones en converger y creo que esta práctica aportó mas conocimientos a mi formación durante la asignatura y reafirmo los que ya tenía.

Fue un excelente curso me lleve muchos conocimientos y su compromiso profesor me ayudo a adquirir los conocimientos esperados de manera eficiente a pesar de la pandemia. :D

Bibliografía

Hagan, M. T. (s.f.). Neural Network Design. En M. T. Hagan, *Neural Network Design* (pág. 1010).