



**UNIVERSIDAD SIMÓN BOLÍVAR**

Departamento de Ciencia de la Información

CI-5438 - Inteligencia Artificial II

Profesora: Ivette Martinez

## **PROYECTO 2**

Elaborado Por:

Edward Fernández 10-11121

Carlos Ferreira 11-10323

Stefani Castellanos 11-11394

Sartenejas, Marzo 2017

## PLANTEAMIENTO DEL PROBLEMA

Se implementó una Red Neuronal Multicapa para ser entrenada con diferentes datos y utilizar los resultados arrojados para predecir la clasificación del *Input* y determinar que tan acertada es dicha red. Se implementó el algoritmo *Backpropagation* en una red multicapa *Feedforward*, dicho algoritmo calcula una aproximación a las derivadas parciales de la función de costo que utiliza la red:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))]$$

donde:

$h_{\theta}$  : valor de la hipótesis (calculada con *feedforward*) en el ejemplo  $x^{(i)}$

$m$  : cantidad de ejemplos

$y^{(i)}$  : clasificación del ejemplo  $x^{(i)}$

En el primer experimento, la red fue entrenada con datos de puntos en un espacio pertenecientes a un cuadrado o a un círculo dentro del cuadrado. En el segundo experimento se realiza una clasificación sobre 3 variaciones de un tipo de flores (Iris).

## ACTIVIDADES

1. Implemente *backpropagation* en una red multicapa *feedforward*.

La implementación del algoritmo se encuentra en el archivo *NeuralNetwork.py*. Se importó la librería *random* para crear las inicializar los valores de las conexiones entre las neuronas ( $\Theta$ ), ya que estas no pueden ser inicializadas con el mismo valor debido a que los resultados de las derivadas serían iguales entre sí en cada iteración. Adicionalmente se utilizó la librería *math* para realizar operaciones matemáticas necesarias. para implementar los algoritmos especificados anteriormente.

2. Pruebe su red con el siguiente experimento de clasificación de patrones: Se pretende lograr la clasificación de puntos en un plano pertenecientes a dos regiones predeterminadas (A y B):
  - El área A es el cuadrado cuya diagonal es el segmento de recta que une los puntos (0, 0) y (20, 20), menos el área que ocupa B.
  - El área B está delimitada por una circunferencia centrada en (10,10) con radio. La ecuación canónica de esta circunferencia es:

$$(x - 10)^2 + (y - 10)^2 = 36$$

- General

El algoritmo *feedforward* es utilizado para calcular la hipótesis de la red, este resultado que contiene una clasificación (está o no en el círculo) para un punto del conjunto de datos de 10000 puntos dentro de la región descrita anteriormente, luego se compara con su clasificación real. Para mostrar la convergencia de los valores calculados por la red se utilizan los errores de la misma ya que, al igual que el costo, este valor no debe aumentar en cada iteración porque la función no lineal (sigmoideal) se ajusta mejor a los datos. Para todas las corridas del algoritmo se usa una tasa de aprendizaje de 0.1 debido a que todas convergen con este valor. Se utilizó 1000 como máximo de iteraciones.

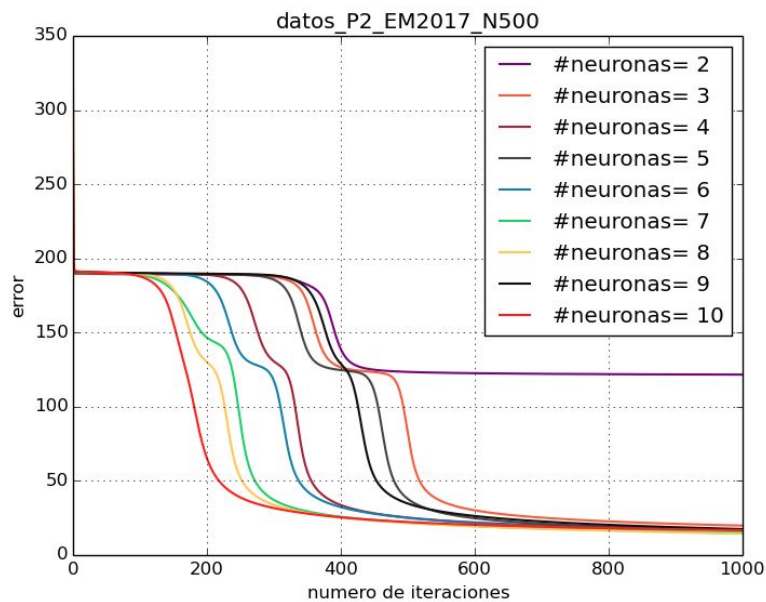


Figura 1. Resultados para los datos con 500 puntos suministrados. Error vs iteraciones

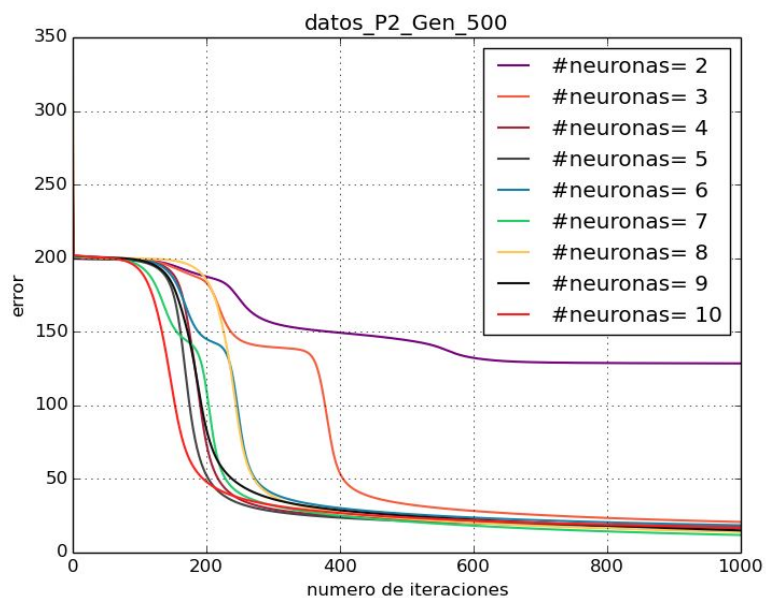


Figura 2. Resultados para los datos con 500 puntos generados aleatoriamente. Error vs iteraciones.

Se observa que para todos los datos con puntos dentro del espacio, cuando se tienen más neuronas en la capa intermedia existe una tendencia a converger más rápido. Sin embargo, hay algunos casos en los que una capa con menos neuronas en su capa intermedia disminuye más rápido el error que se trata de minimizar, esto puede deberse a la inicialización de las conexiones de las neuronas porque son aleatorias.

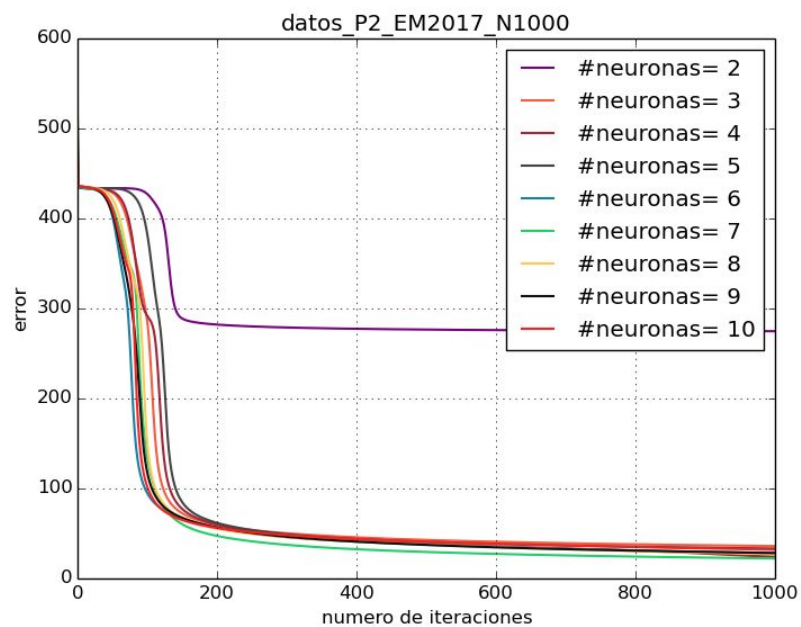


Figura 3. Resultados para los datos con 1000 puntos suministrados. Error vs iteraciones

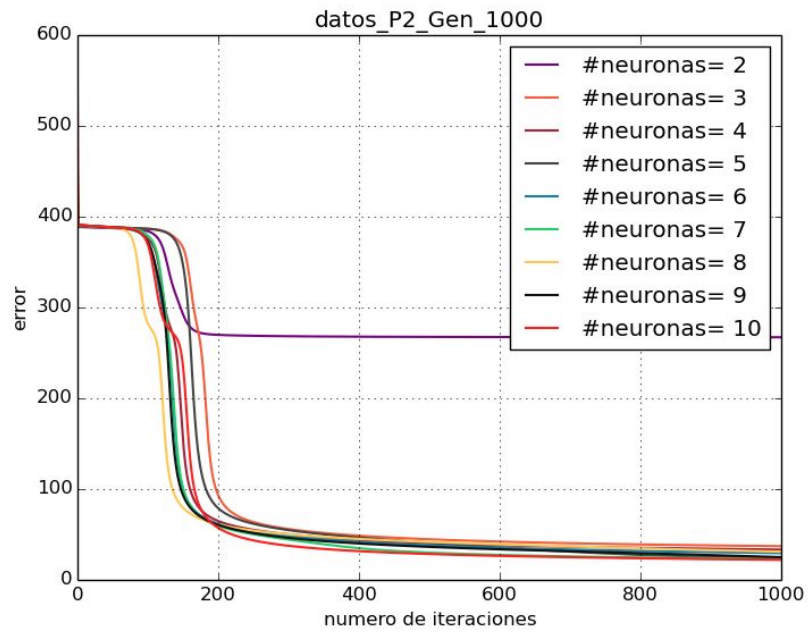


Figura 4. Resultados para los datos con 1000 puntos generados aleatoriamente. Error vs iteraciones.

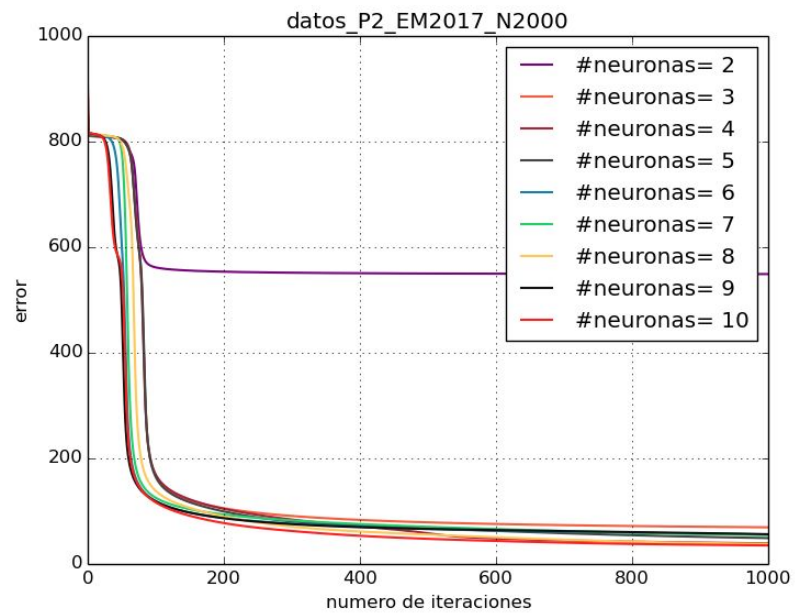


Figura 5. Resultados para los datos con 2000 puntos suministrados. Error vs iteraciones

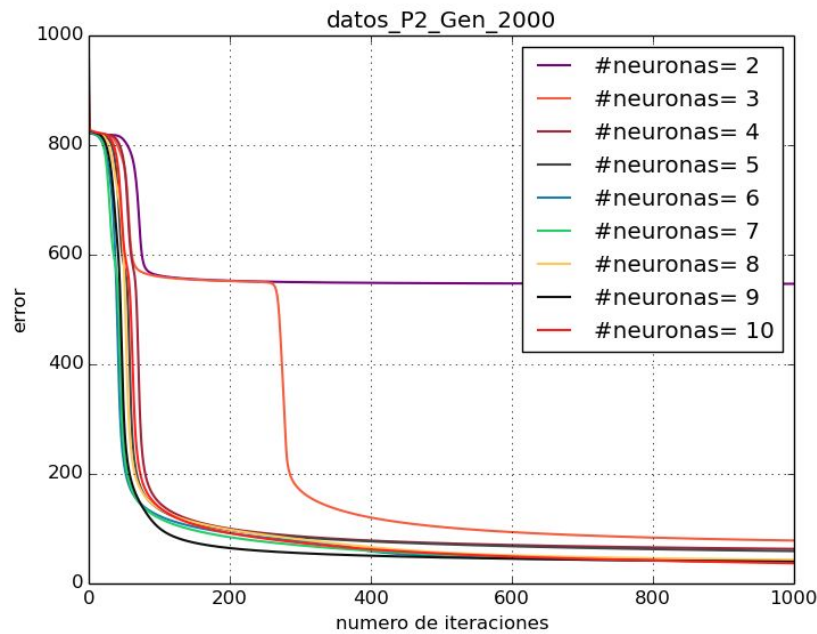


Figura 6. Resultados para los datos con 2000 puntos generados aleatoriamente. Error vs iteraciones

A continuación se observan los mejores y peores resultados graficando 10.000 puntos dentro del área y clasificándolos usando la hipótesis obtenida de cada red neuronal. Es importante destacar que estos resultados pueden variar debido a las inicializaciones de las conexiones. El error en las pruebas fue calculado utilizando la métrica de “Desviación absoluta media” que es calculada promediando la diferencia entre los valores reales y los que predijo la red neuronal.

- Los 500 datos provistos

número de neuronas	error en entrenamiento	error en prueba	falsos positivos	falsos negativos
2	121.663701958	0.2427	1507	920
3	19.8930545314	0.0357	114	243
4	17.1306028912	0.0344	107	237
5	17.0097338838	0.033	123	207
6	14.6274510081	0.0277	92	185
7	15.2521463315	0.0271	82	189
8	14.4492097553	0.031	104	206

9	17.5641948943	0.0279	96	183
10	16.8078238475	0.0298	83	215

Para los 500 datos se obtiene menos error en la predicción utilizando 7 neuronas en la capa intermedia y el mayor error se obtiene utilizando 2 neuronas.

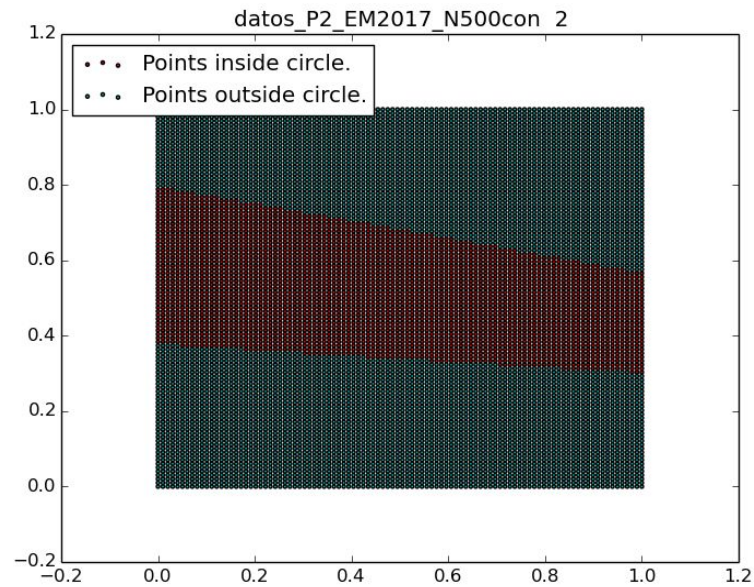


Figura 7. Gráfica para 500 datos con 2 neuronas en la capa intermedia. Peor resultado

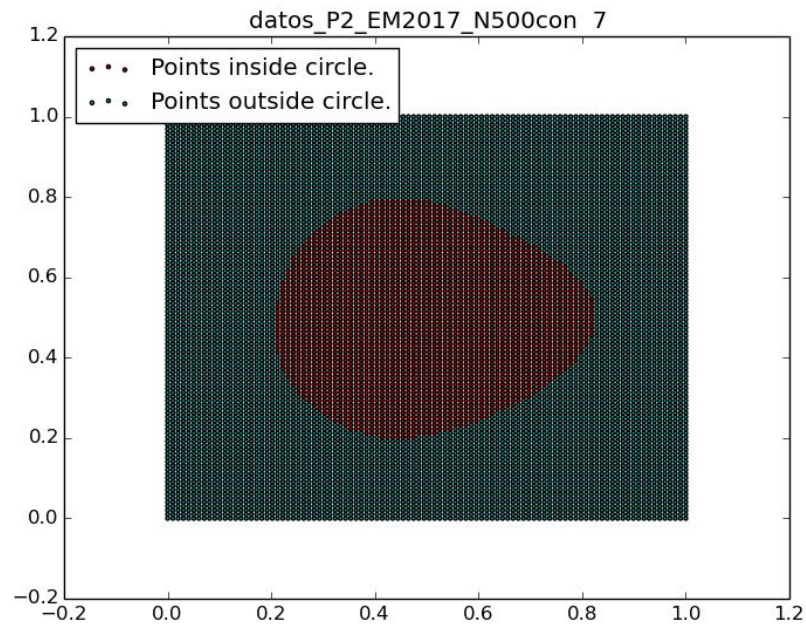


Figura 8. Gráfica para 500 datos con 7 neuronas en la capa intermedia. Mejor resultado



- Los 500 datos generados aleatoriamente

número de neuronas	error en entrenamiento	error en prueba	falsos positivos	falsos negativos
2	128.514473506	0.2423	1544	879
3	20.812151607	0.036	42	318
4	17.6503168778	0.0409	24	385
5	16.7405705314	0.0387	26	361
6	18.555512993	0.0418	3	415
7	11.9643433084	0.0227	35	192
8	14.154325117	0.0251	34	217
9	15.1176808029	0.0413	14	399
10	16.9177728529	0.0463	10	453

Para los 500 datos generados aleatoriamente se obtiene menos error en la predicción utilizando 7 neuronas en la capa intermedia y el mayor error se obtiene utilizando 2 neuronas.

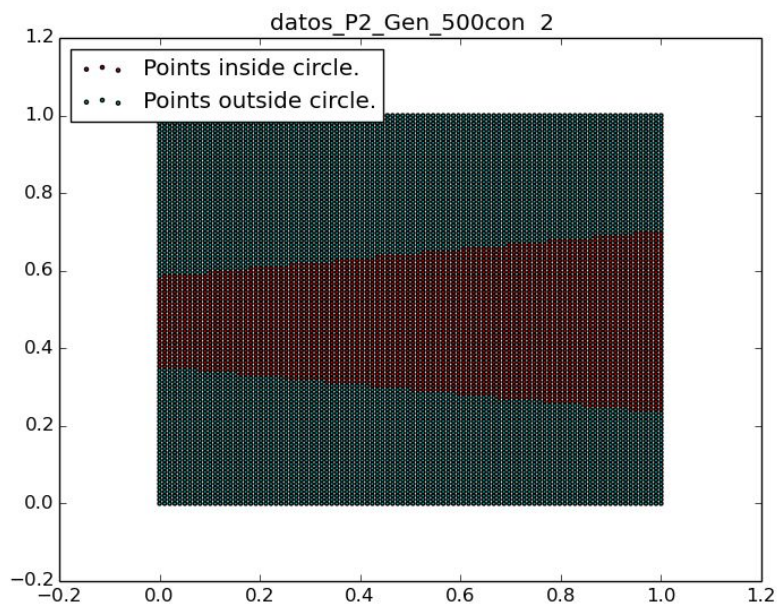


Figura 9. Gráfica para 500 datos generados aleatoriamente con 2 neuronas en la capa intermedia. Peor resultado



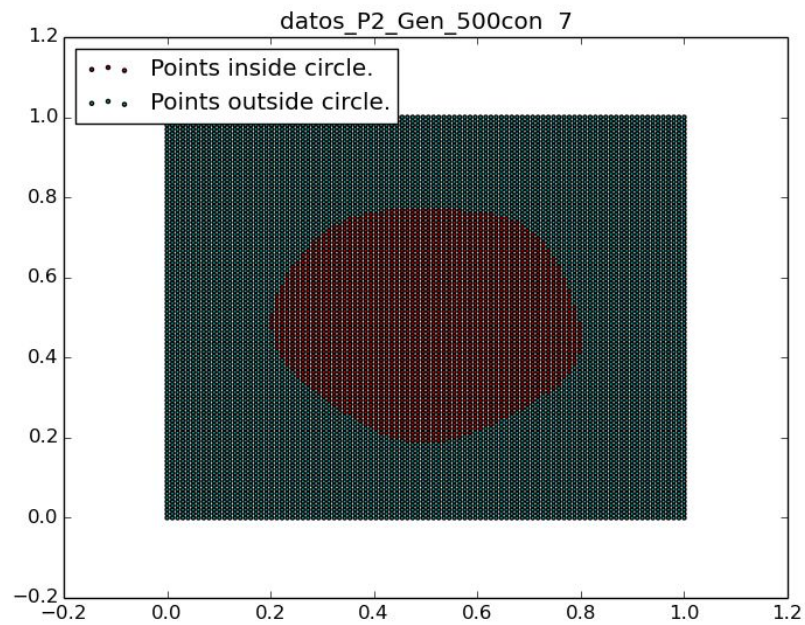


Figura 10. Gráfica para 500 datos generados aleatoriamente con 7 neuronas en la capa intermedia. Mejor resultado.

- Los 1000 datos provistos

número de neuronas	error en entrenamiento	error en prueba	falsos positivos	falsos negativos
2	274.783062848	0.2434	1850	584
3	36.0925235561	0.0263	224	39
4	24.2402672	0.0218	210	8
5	33.2429612491	0.025	214	36
6	28.4007781763	0.0328	305	23
7	22.2695860193	0.0111	98	13
8	27.2909341756	0.0292	275	17
9	28.2880187526	0.0242	231	11
10	32.0698101688	0.0266	249	17

Para los 1000 datos provistos se obtiene menos error en la predicción utilizando 7 neuronas en la capa intermedia y el mayor error se obtiene utilizando 2 neuronas.

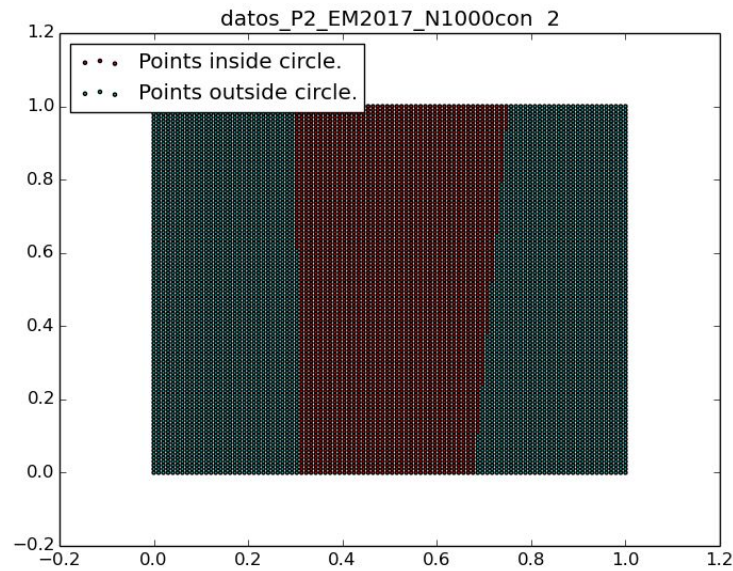


Figura 11. Gráfica para 1000 datos provistos con 2 neuronas en la capa intermedia. Peor resultado

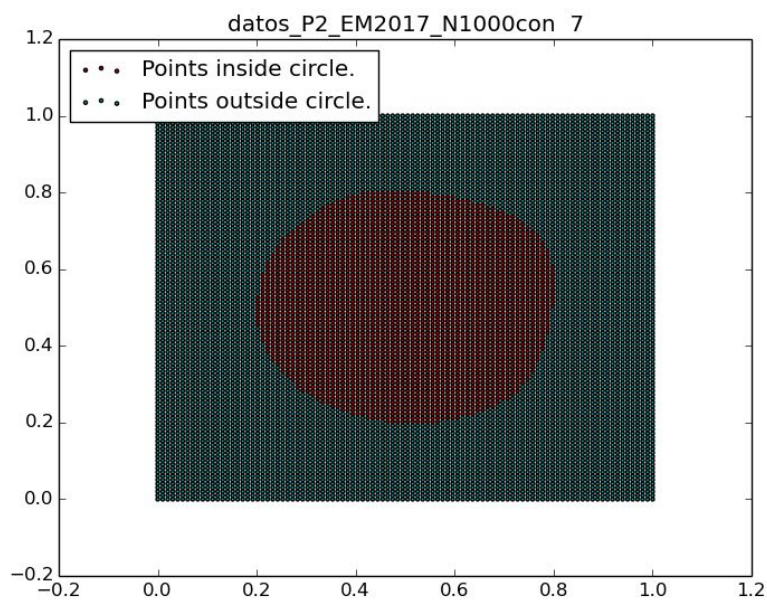


Figura 12. Gráfica para 1000 datos generados aleatoriamente con 7 neuronas en la capa intermedia. Mejor resultado.

- Los 1000 datos generados aleatoriamente

número de neuronas	error en entrenamiento	error en prueba	falsos positivos	falsos negativos
2	267.22460474	0.2413	1405	1008
3	37.1618537572	0.0225	52	173
4	33.3375417299	0.0199	41	158
5	24.1821612109	0.0168	98	70
6	29.4775538029	0.0161	42	119
7	23.0025774145	0.0135	62	73
8	31.1608754823	0.0175	63	112
9	25.1234405983	0.014	70	70
10	22.1193551997	0.0136	72	64

Para los 1000 datos generados aleatoriamente se obtiene menos error en la predicción utilizando 7 neuronas en la capa intermedia y el mayor error se obtiene utilizando 2 neuronas.

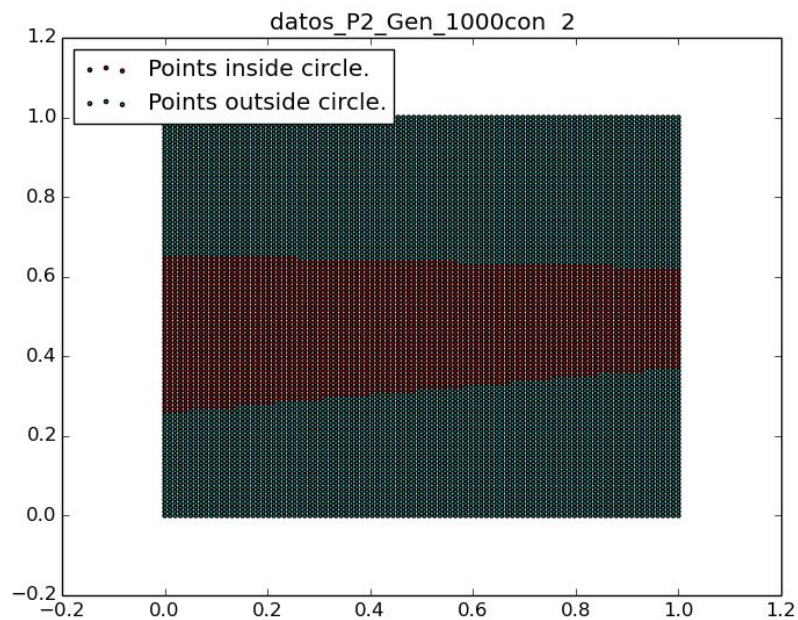


Figura 13. Gráfica para 1000 datos generados aleatoriamente con 2 neuronas en la capa intermedia. Peor resultado.

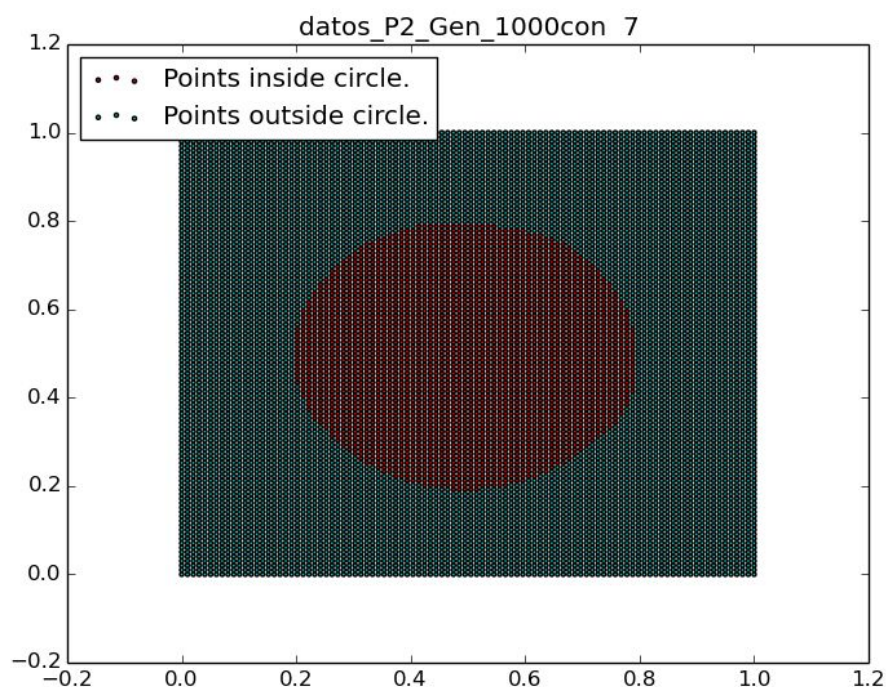


Figura 14. Gráfica para 1000 datos generados aleatoriamente con 7 neuronas en la capa intermedia. Mejor resultado.

- Los 2000 datos provistos

número de neuronas	error en entrenamiento	error en prueba	falsos positivos	falsos negativos
2	549.110980656	0.2375	1432	943
3	69.4752029137	0.0301	0	301
4	39.0747788208	0.0203	20	183
5	49.1656809677	0.0222	11	211
6	55.9444940983	0.0256	0	256
7	54.8860208073	0.026	1	259
8	38.0575846326	0.0188	0	188
9	56.3857945411	0.0209	5	204
10	35.2006028585	0.0202	0	202

Para los 2000 datos provistos se obtiene menos error en la predicción utilizando 8 neuronas en la capa intermedia y el mayor error se obtiene utilizando 2 neuronas.



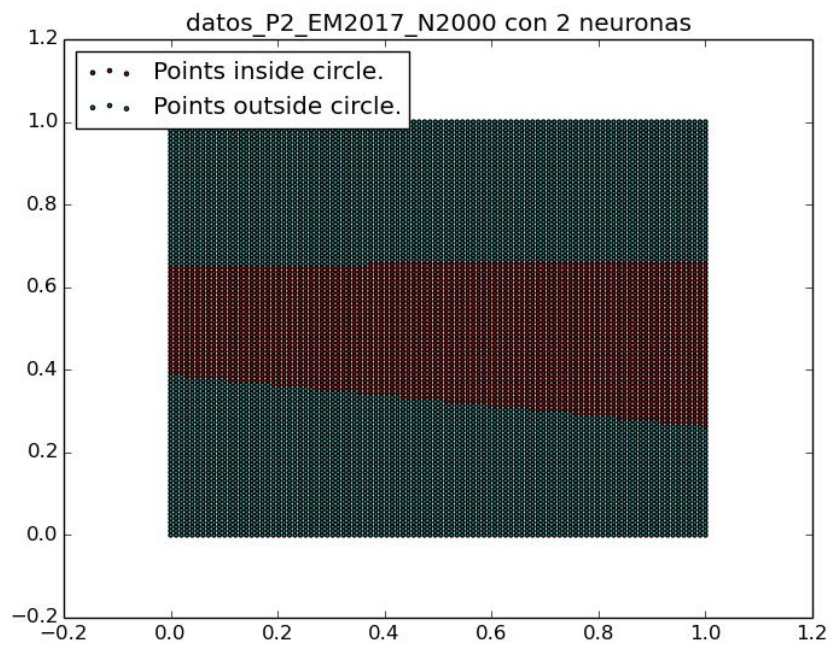


Figura 15. Gráfica para 2000 datos provistos con 2 neuronas en la capa intermedia. Peor resultado

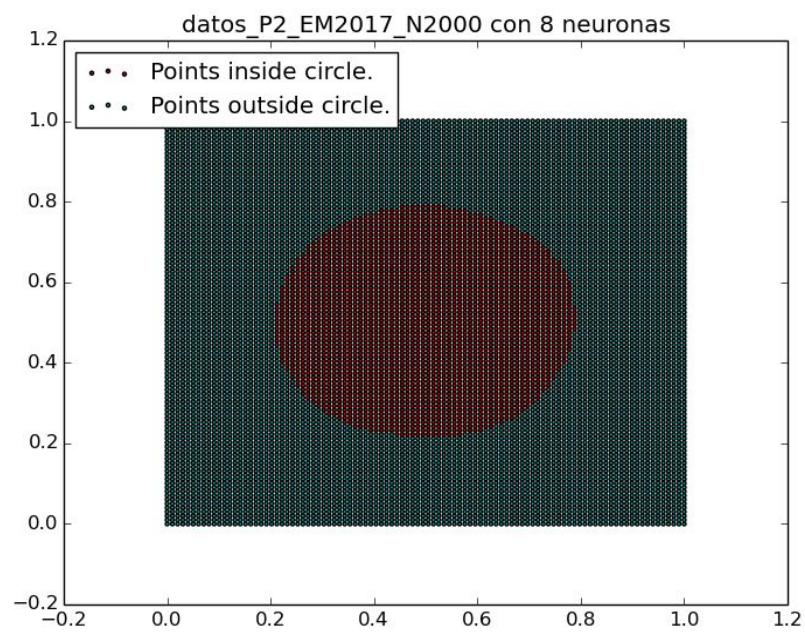


Figura 16. Gráfica para 2000 datos provistos con 8 neuronas en la capa intermedia. Mejor resultado

- Los 2000 datos generados aleatoriamente

número de neuronas	error en entrenamiento	error en prueba	falsos positivos	falsos negativos
2	546.869019681	0.2454	893	1561
3	78.4330880697	0.0135	53	82
4	63.1678862725	0.0132	52	80
5	59.1614407996	0.012	71	49
6	39.5952810826	0.0074	32	42
7	40.3432193575	0.0061	4	57
8	43.269916063	0.0104	44	60
9	39.7750624009	0.0101	81	20
10	36.9859088905	0.0082	41	41

Para los 2000 datos generados aleatoriamente se obtiene menos error en la predicción utilizando 8 neuronas en la capa intermedia y el mayor error se obtiene utilizando 2 neuronas.

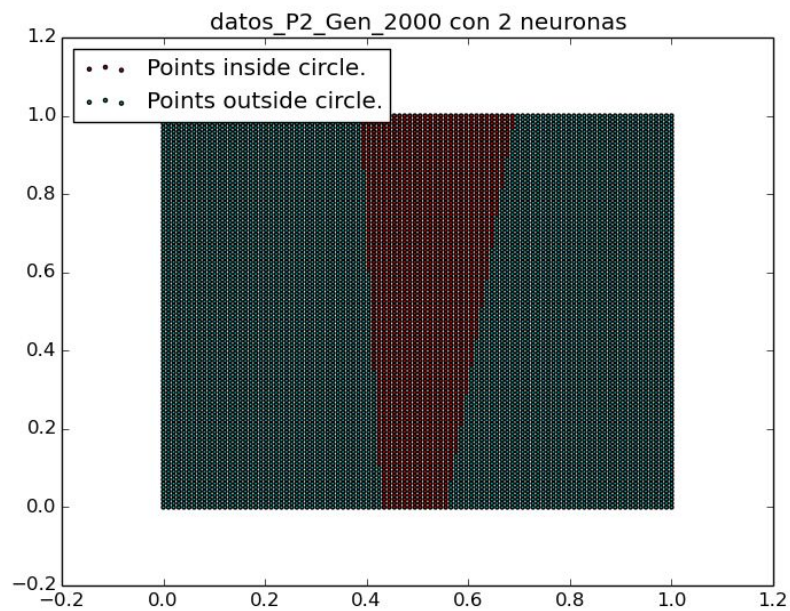


Figura 17. Gráfica para 2000 datos generados aleatoriamente con 2 neuronas en la capa intermedia. Peor resultado.

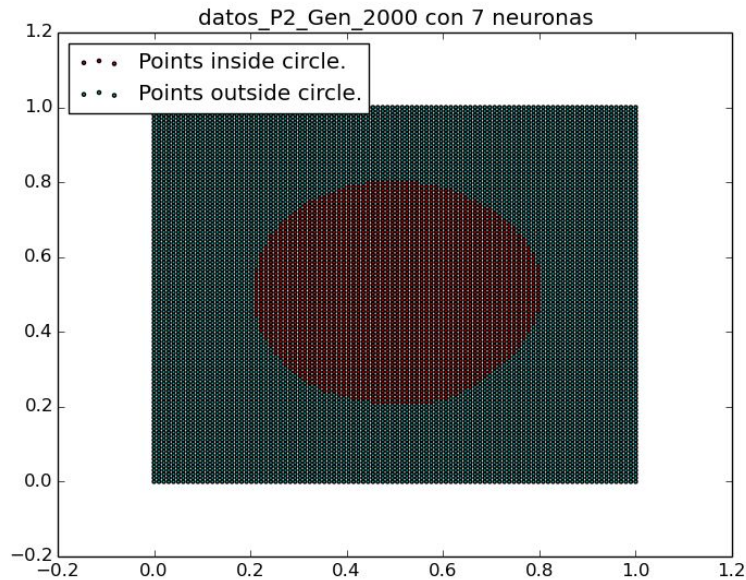


Figura 17. Gráfica para 2000 datos generados aleatoriamente con 7 neuronas en la capa intermedia. Mejor resultado.

Se podría considerar que las redes con 10 neuronas en la capa intermedia serían capaces de predecir mejor los datos que cualquier otra, sin embargo este no es el caso para ninguno de los conjuntos presentados. Esto se debe a que al disminuir tanto el error la función se ajusta muy bien a los datos de entrenamiento causando *overfitting*, impidiéndole generalizar sus hipótesis. En general la mejor cantidad de neuronas para el problema planteado es 7.

3. Entrene su red para construir dos clasificadores sobre los datos del conjunto Iris Data Set: (<http://archive.ics.uci.edu/ml/datasets/Iris> ).

- Uno que separe las “Iris Setosa” del resto (Clasificador binario)
- Uno que separe cada una de las 3 clases.

Pruebe con redes de 4 a 10 neuronas en la capa intermedia, usando como conjunto de entrenamiento los siguientes porcentajes de los datos: 50 %, 60 %, 70 %, 80 %, 90 %.

Para ambos experimentos se crea un diccionario que codifica cada clasificación en un número, en el caso del binario a “Iris Setosa” le corresponde el número 1 y al resto el 0. Para reconocer múltiples clases es necesario aumentar la cantidad de neuronas en la última capa (3) y realizar una codificación del vector de salida, colocando un 1 en donde corresponda: a “Iris-setosa” le corresponde el vector [1, 0, 0], a “Iris-versicolor” [0, 1, 0] y a “Iris-virginica” [0, 0, 1]; en este caso el diccionario mencionado anteriormente contiene los valores que representan la posición de 1 en el vector resultado.



A continuación se muestran los resultados los errores, aciertos y fallos en la predicción para las corridas especificadas en el enunciado utilizando 1000 iteraciones.

- Clasificador Binario. Logra hacer todas las clasificaciones correctamente, pues la cantidad de iteraciones es suficiente para que el error sea despreciable en la red neuronal.

% datos de entrenamiento	número de neuronas	error en prueba	aciertos	fallos
50	4	0.0	75	0
	5	0.0	75	0
	6	0.0	75	0
	7	0.0	75	0
	8	0.0	75	0
	9	0.0	75	0
	10	0.0	75	0
60	4	0.0	60	0
	5	0.0	60	0
	6	0.0	60	0
	7	0.0	60	0
	8	0.0	60	0
	9	0.0	60	0
	10	0.0	60	0
70	4	0.0	45	0
	5	0.0	45	0
	6	0.0	45	0
	7	0.0	45	0
	8	0.0	45	0
	9	0.0	45	0
	10	0.0	45	0

80	4	0.0	30	0
	5	0.0	30	0
	6	0.0	30	0
	7	0.0	30	0
	8	0.0	30	0
	9	0.0	30	0
	10	0.0	30	0
90	4	0.0	15	0
	5	0.0	15	0
	6	0.0	15	0
	7	0.0	15	0
	8	0.0	15	0
	9	0.0	15	0
	10	0.0	15	0

- Diferentes clases. Los mejores resultados se obtienen con una partición 70-30. Para las particiones con más de 70 en el conjunto de entrenamiento probablemente exista *overfitting* y para los menores de 70 el conjunto de entrenamiento es muy pequeño y tampoco logra generalizar, en especial el caso de 50-50. Adicionalmente, esta clasificación es más compleja que la binaria, razón por la cual presenta más errores

% datos de entrenamiento	número de neuronas	error en prueba	aciertos	fallos
50	4	0.05333333333333	71	4
	5	0.04	72	3
	6	0.04	72	3
	7	0.04	72	3
	8	0.04	72	3
	9	0.04	72	3
	10	0.04	72	3

60	4	0.03333	58	2
	5	0.03333	58	2
	6	0.03333	58	2
	7	0.03333	58	2
	8	0.03333	58	2
	9	0.03333	58	2
	10	0.03333	58	2
70	4	0.0	45	0
	5	0.0	45	0
	6	0.0	45	0
	7	0.0	45	0
	8	0.0	45	0
	9	0.0	45	0
	10	0.0	45	0
80	4	0.03333333333333	29	1
	5	0.03333333333333	29	1
	6	0.03333333333333	29	1
	7	0.03333333333333	29	1
	8	0.03333333333333	29	1
	9	0.03333333333333	29	1
	10	0.03333333333333	29	1
90	4	0.13333333333333	13	2
	5	0.13333333333333	13	2
	6	0.13333333333333	13	2
	7	0.13333333333333	13	2
	8	0.13333333333333	13	2
	9	0.13333333333333	13	2
	10	0.13333333333333	13	2

