ITBA
SIA
TP 2: Redes Neuronales

Grupo 1
José Torreguitar - 57519 - jtorreguitar@itba.edu.ar
Tomas Soracco - 56002 - tsoracco@itba.edu.ar
Sofía Picasso - 57700 - spicasso@itba.edu.ar

Introducción

En este informe se detalla la implementación de la red neuronal multicapa creada para enfrentar un problema otorgado por la Cátedra.

El problema consiste en analizar el aprendizaje de la red al recibir datos topográficos, evaluando su capacidad de entrenamiento y generalización.

La implementación en sí se realizó utilizando Octave, el lenguaje orientado a la resolución de problemas matemáticos.

Se realizaron varios testeos con distintas optimizaciones como momentum, eta adaptativo y una variación de backpropagation para evitar la saturación de los datos, y tambień distinta arquitecturas con distintas cantidades de capas y neuronas en cada capa.

Modelaje de la Red

Aunque se utilizaron distintas arquitecturas a la hora de realizar los testeos, se mantuvo la cantidad de neuronas de entrada (dos) y de salida (una).

La cantidad de capas ocultas y neuronas en esas capas puede ser determinada por el usuario.

La Cátedra otorgó los datos topográficos utilizados a la hora de entrenar a la red y evaluar su desempeño. El usuario también puede decidir que cantidad (o porcentaje de los datos) quiere utilizar para que la red aprenda. El resto se utiliza para analizar la capacidad de generalización de la misma.

Los parámetros que pueden ser modificados se encuentra en el archivo config.data. Entre ellos se encuentran:

- el nivel de aprendizaje o eta (η).
- la optimización eta adaptativo: puede mantenerse apagada o prendida. También se puede elegir su factor de crecimiento, de decrecimiento y la cantidad de pasos requeridos para que el error decrezca constantemente antes de que el eta aumente.
- la cantidad de épocas (o ciclos de entrenamiento).
- la optimización momentum y el porcentaje de variación alpha.
- la función que se desea aplicar al realizar el backpropagation, sea tanh o exponencial sigmoidea.
- la optimización de prevención de saturación, la cual agrega un valor constante al realizar el cálculo de delta en caso de que la derivada al calcular devuelva 0.
- dos formas de calcular el cambio de pesos, batch e incremental.

Estos se encuentran detallados en el archivo readme.

Métricas

Se tomaron en cuenta el testing error y el training error. Al comparar ambos uno puede notar el nivel de generalización de la red. Si el error de training es muy bajo y el de testing muy alto, esto significa que la red está memorizando.

Si son similares, esto indica un buen nivel de generalización.

Ambos son evaluados en base de la variable independiente épocas. En este caso, se utilizaron 1000 épocas para todos los testeos y un error épsilon de 0.1.

Resultados

Número de capas ocultas y neuronas

En la figura 1 se observa la arquitectura que se tomó como base. Se trata de una red con dos capas ocultas, la primera de 15 neuronas y la segunda de 10. El aprendizaje es incremental con un eta de 0.01 no adaptativo sin momentum, sin prevención de saturación y utilizando la tangente hiperbólica como función de activación.

Esta red ya de por si da buenos resultados, con un error cercano a 0.0025 tanto, para entrenamiento como para testeo (el de entrenamiento siendo menor, como es esperado). Los errores son similares y la tasa de éxito también. Es un buen primer paso pero tiene lugar para mejora, por esto se lo eligió como caso base.

La segunda red (Figura 2) reduce la cantidad de neuronas a una sola capa de 10 neuronas. El razonamiento para este cambio requiere cambiar como uno considera el problema. En el modelo se usa un valor epsilon para determinar si el resultado de la red es acertado o no. Esto es, si el resultado correcto está contenido en el intervalo (resultado de la red - epsilon, resultado de la red + epsilon) se considera que la red acertó en su predicción.

Puesto que usamos un valor de epsilon de 0.1 y los resultados de z van de -1 a 1 el problema puede considerarse como uno de categorización en 20 intervalos diferentes. Luego, uno podría decir que si se tiene una sola capa con menos de 20 neuronas es muy probable que se le introduzca un cuello de botella a la información, poniendo una cota a la capacidad de la red de aprender el problema.

Es muy posible que este sea el caso ya que es la arquitectura que peores resultados dió. Sorprendentemente, sin embargo, también se ve que la red tiene mejores resultados en el conjunto de testeo que en el de entrenamiento.

La siguiente red (Figura 3) posee dos capas ocultas: la primera de 30 neuronas, la segunda de 20. Se ve que el aumento en el tamaño de las capas aumentó tanto el poder de generalización de la red como su poder de predicción ya que el error para ambos conjuntos baja considerablemente.

Valor de el eta, η

Tanto para la Figura 4 como para la Figura 5 se aumentó el valor del eta comparad con la Figura 1. El aumento de eta produce mayor variación en el error. Sin embargo, permite acercarse a un mínimo de forma más rápido.

En el caso de la Figura 4, el eta utilizado de 0.1 fue demasiado grande y bajo el poder de generalización de la red.

Pero en la Figura 5, aumentar el eta de 0.01 a 0.05 ayudó a la red a incrementar su poder predictivo, permitiéndole explorar más y encontrar un mínimo mejor con el cual quedarse.

Momentum

En la figura 6, donde se activa el momentum con un alfa de 0.9, se puede notar que al comienzo hay picos de error más pronunciados. Sin embargo, parecería que este valor funciona bien con el terreno utilizado, dado que tiene un buen nivel de generalización y valores de error bajos. Este valor de momentum le permite a la red encontrar un mejor mínimo que si se le restringe más la exploración.

En la figura 7 también se utilizó momentum, esta vez con un alfa de 0.99. Se debió aumentar la escala del gráfico debido a que los errores fueron mayores a 0.1. Se puede notar que en este caso, el alfa elegido fue demasiado grande. Los picos del gráfico también son más extremos.

Eta Adaptativo

En la figura 8 se observan resultados muy inferiores a los de las demás arquitecturas. En este caso puede que el eta haya alcanzado valores muy bajos estando cerca de un mínimo local mucho peor que los encontrados para los demás casos.

Es posible que el valor del eta inicial elegido en este caso (0.01) esté cerca del óptimo mientras que los parámetros del eta adaptativo (un factor de crecimiento de 0.01, uno de decrecimiento de 0.95 y 7 pasos realizados donde el error baja constantemente para dejar que el eta crezca) sean demasiado grandes en este caso, alejando al eta del valor que lleva a la mejor solución.

Valores distintos podrían suavizar la curva de errores que se puede notar tiene muchos picos.

Se realizó otro intento con eta adaptativo (Figura 9). Esta vez se usó un factor de crecimiento de 0.001, uno de decrecimiento de 0.05 y 10 pasos realizados donde el error baja constantemente para dejar que el eta crezca. Estos valores afectan al eta de menor manera, y se puede notar debido a como se suavizaron las curvas de errores.

Variación al proceso de Backpropagation

En la figura 10 se intentó de ver los resultados tomando una medida para hacer más lento el proceso de saturación de los valores. Se decidió agregar un valor de 0.1 al calcular el delta, para evitar que este sea nulo si la derivada de la función elegida devolvía un valor muy cercano a 0. De esta forma, se intenta de que los pesos siempre se vean afectados en cada paso. Esto permitió a la

red explorar más a la hora de aprender, lo cual redujo su training error, pero también la ayudó a generalizar mejor, ya que el testing error también se redujo.

El gráfico parece demostrar sin embargo que deben realizarse varias épocas para que se ajusten los niveles de error, ya que al comienzo los cambios pueden ser muy radicales

Tamaño de Entrenamiento

En la figura 11, se ve que la arquitectura a la cual se le restringen los casos de entrenamiento a solo 100 performa mucho peor que las demás, teniendo también una diferencia marcada entre error de entrenamiento y de testeo, indicando que lo poco que aprendió la red en realidad fue memorización.

Batch vs. Incremental

En la Figura 12 se probó utilizar el aprendizaje tipo batch en vez de incremental que se utilizó anteriormente. Los errores fueron mucho mayores y se pueden notar picos más abruptos. Batch tiene la ventaja de ser más rápido que el proceso de aprendizaje incremental, pero también tarda más en converger. Los errores son más grandes ya que se hace una menor cantidad de pasos, y esto explica también los picos de error, ya que la red aprende cada tanto.

Función de Activación

Se utilizó la función tangente hiperbólica por sobre la función de sigmoide exponencial ya que la imagen de la sigmoide se encuentra entre los valores positivos, lo cual no se adecúa a nuestro terreno

Conclusión

Cambiar los parámetros de la red neuronal puede afectar los resultados que esta produce de gran manera.

Un aumento en el eta puede producir variaciones de error mayores. Esto ayuda a prevenir que entre en mínimos locales, sin embargo, si es demasiado grande puede también pasarse del óptimo buscado.

El uso del eta adaptativo puede empeorar ambos el nivel de aprendizaje y el de generalización si los valores que se eligen para sus parámetros son los incorrectos para el terreno explorado. Valores mayores de incremento y decremento o cantidades de pasos mínimos requeridos muy pequeños harán que el eta tenga grandes cambios y cambios constantes. Para suavizar la curva de errores, se deben utilizar valores que cuadren bien con el terreno. Si se eligen los correctos, el nivel de aprendizaje y generalización pueden mejorar de gran manera.

La arquitectura también es de gran importancia. Una arquitectura con pocas neuronas o falta de capas ocultas puede producir cuellos de botella en el entrenamiento. Una mayor cantidad de conexiones puede mejorar mucho el rendimiento de la red.

El momentum, al igual que el eta adaptativo, es solo una optimización si se eligen los factores correctos. Un momentum con un alfa de gran tamaño puede alejar a la red de la solución más óptima. Se debe elegir uno adecuado al terreno.

Una variación para extender la saturación de los valores y permitir que sigan cambiando constantemente ayuda a la red a hacer un proceso de exploración mayor, pero también puede alejarla de la solución si los valores de la variación son muy altos.

La cantidad de muestras que se le permiten a la red para aprender también afecta mucho su desempeño, ya que un nivel muy bajo de éstas no permitirá que la red aprenda y por lo tanto bajará mucho su nivel de generalización, haciendo que la red dependa de la memorización.

El aprendizaje batch tiene una mejor performance y es más rápido que el proceso incremental, sin embargo, suele tardar más en converger al realizar una menor cantidad de pasos. Se debe evaluar que se valora más según el caso en cuestión a la hora de elegir entre estos dos.

Se debe testear la red con distintos parámetros para lograr encontrar los ideales para ese caso específico, ya que no existen parámetros ideales constantes para todas las redes.

Anexo

TABLA DE PARÁMETROS

Fig.	*Arqui.*	*Training size*	*Aprendizaje*	*Eta*	*Eta Adapt*	*Momentum*	*Funcion*	*Prev. de Saturación*
1	2,15,10,1	300	Incremental	0.01	No	No	tanh	No
2	2,10,1	300	Incremental	0.01	No	No	tanh	No
3	2,30,20,1	300	Incremental	0.01	No	No	tanh	No
4	2,15,10,1	300	Incremental	0.1	No	No	tanh	No
5	2,15,10,1	300	Incremental	0.05	No	No	tanh	No
6	2,15,10,1	300	Incremental	0.01	No	Sí, alfa = 0.9	tanh	No
7	2,15,10,1	300	Incremental	0.01	No	Sí, alfa = 0.99	tanh	No
8	2,15,10,1	300	Incremental	0.01	Sí, a = 0.01 b = 0.95 steps = 7	No	tanh	No
9	2,15,10,1	300	Incremental	0.01	Sí, a = 0.001 b = 0.05 steps =10	No	tanh	No
10	2,15,10,1	300	Incremental	0.01	No	No	tanh	Sí, 0.1
11	2,15,10,1	100	Incremental	No	No	No	tanh	No
12	2,15,10,1	300	Batch 5	No	No	No	tanh	No

TABLA DE RESULTADOS

Fig.	*Training error*	*Testing error*	*Training success rate*	*Testing success rate*
1	0.0022366	0.0025531	87.3333%	82.2695%
2	0.0047929	0.0041049	74.3333%	81.5603%
3	0.0010003	0.0019097	95%	92.9078%
4	0.0015415	0.0039599	93.6667%	78.0142%
5	0.0016994	0.0025072	91%	83.6879%
6	0.0014231	0.0016422	93.6667%	92.1986%
7	0.52244	0.50111	5%	4.25532%
8	0.016618	0.016136	47.6667%	43.9716%
9	9.7906e-04	9.6789e-04	95.6667%	95.7447%
10	0.0017686	0.0021263	91%	87.9433%
11	0.0079526	0.013595	76%	45.7478%
12	0.037926	0.018614	22.6667%	39.0071%

FIGURA 1

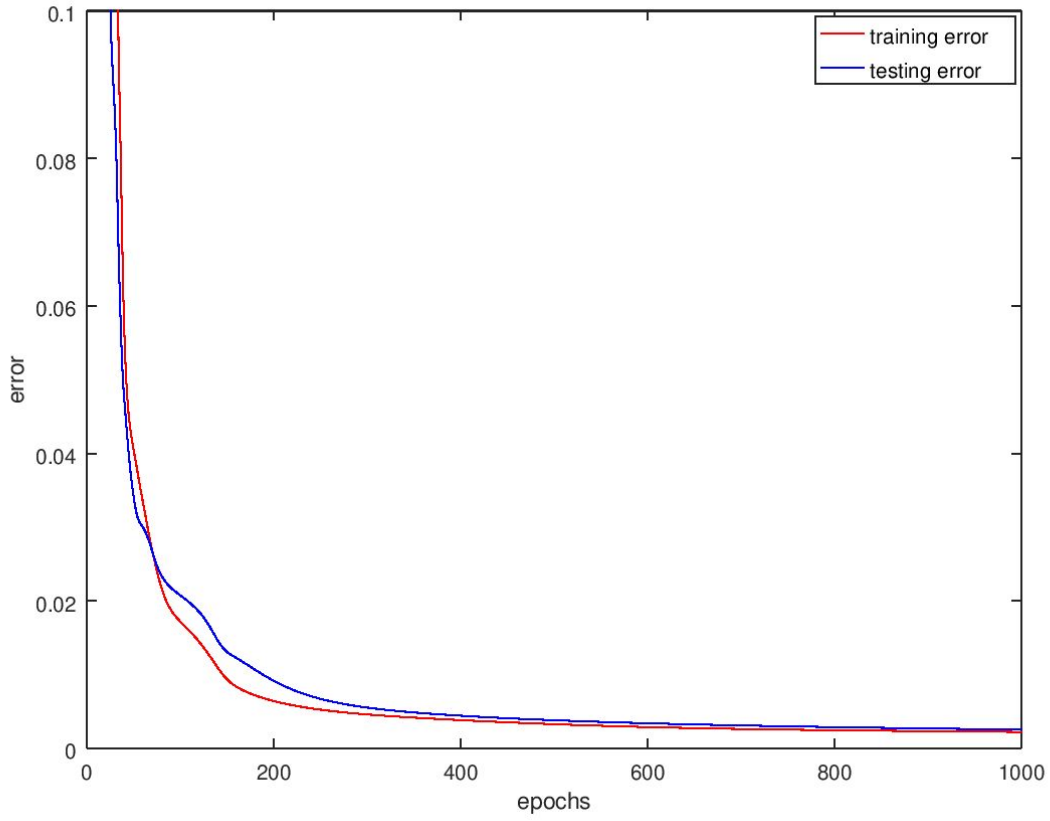

FIGURA 2

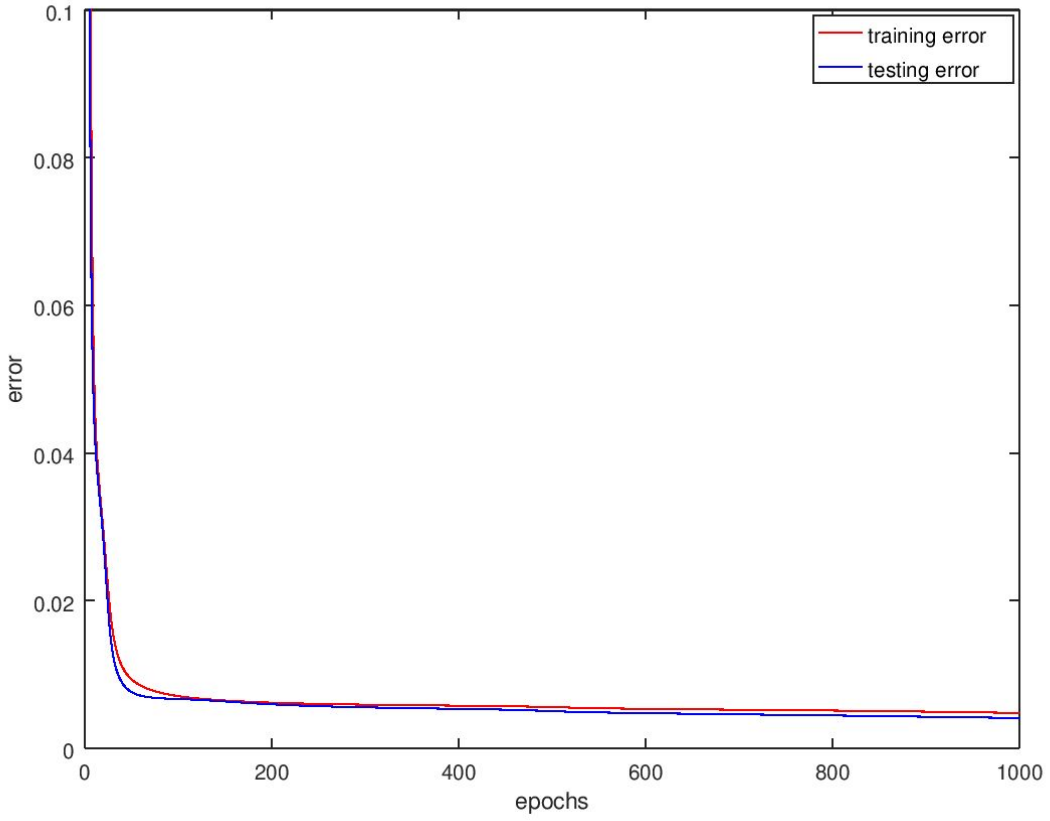

FIGURA 3

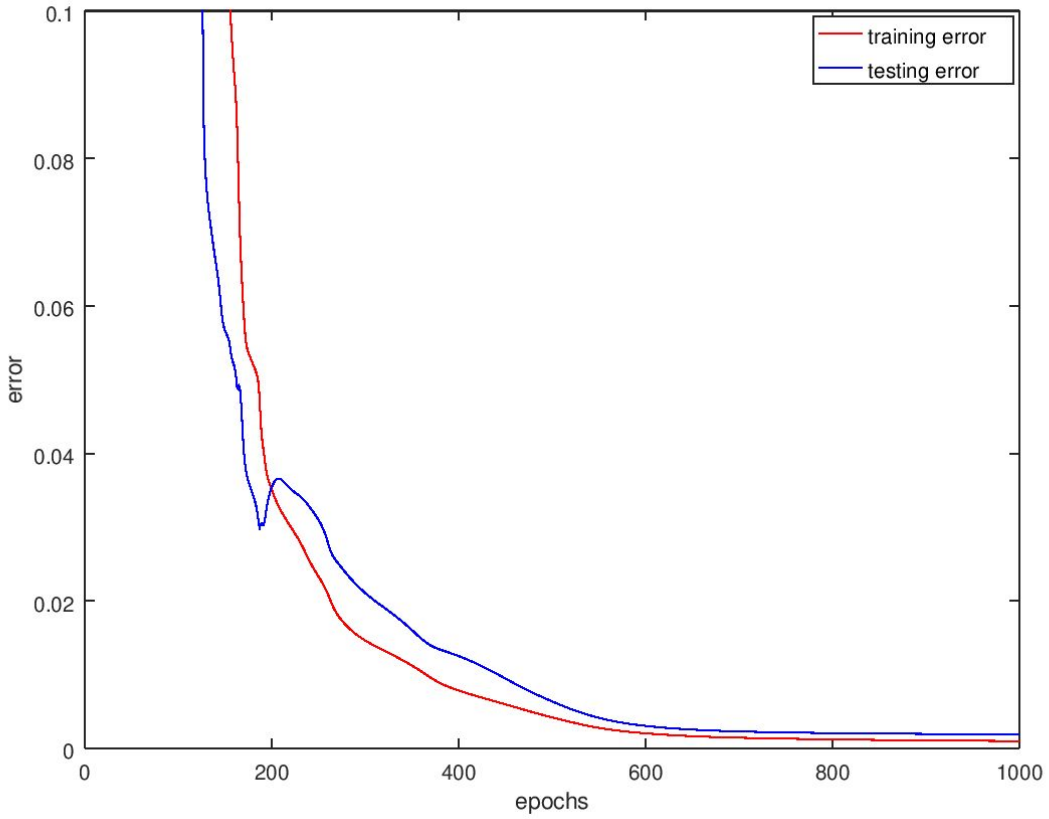

FIGURA 4

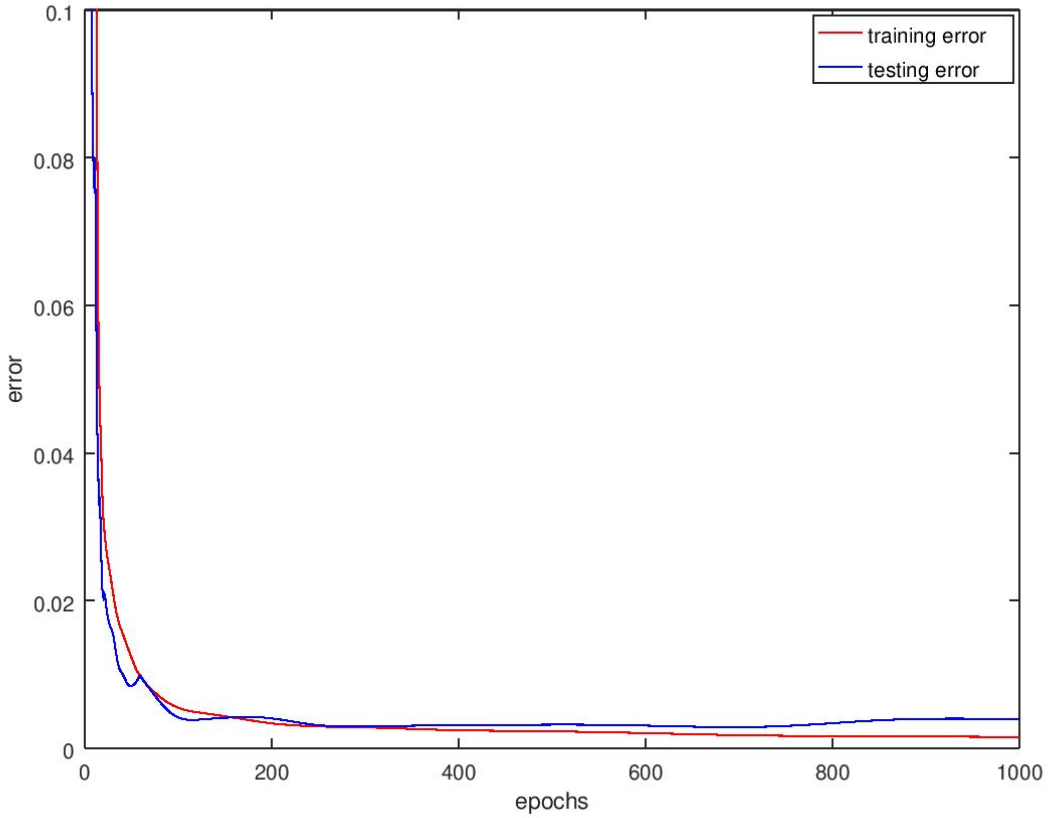

FIGURA 5

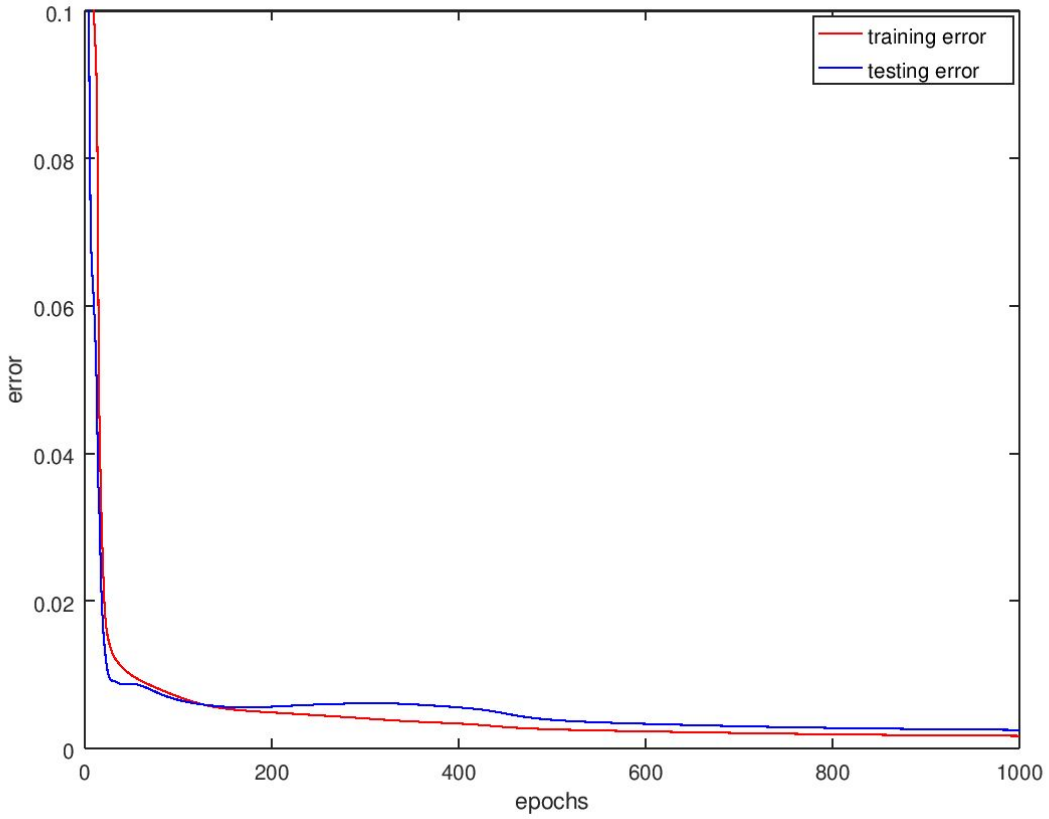

FIGURA 6

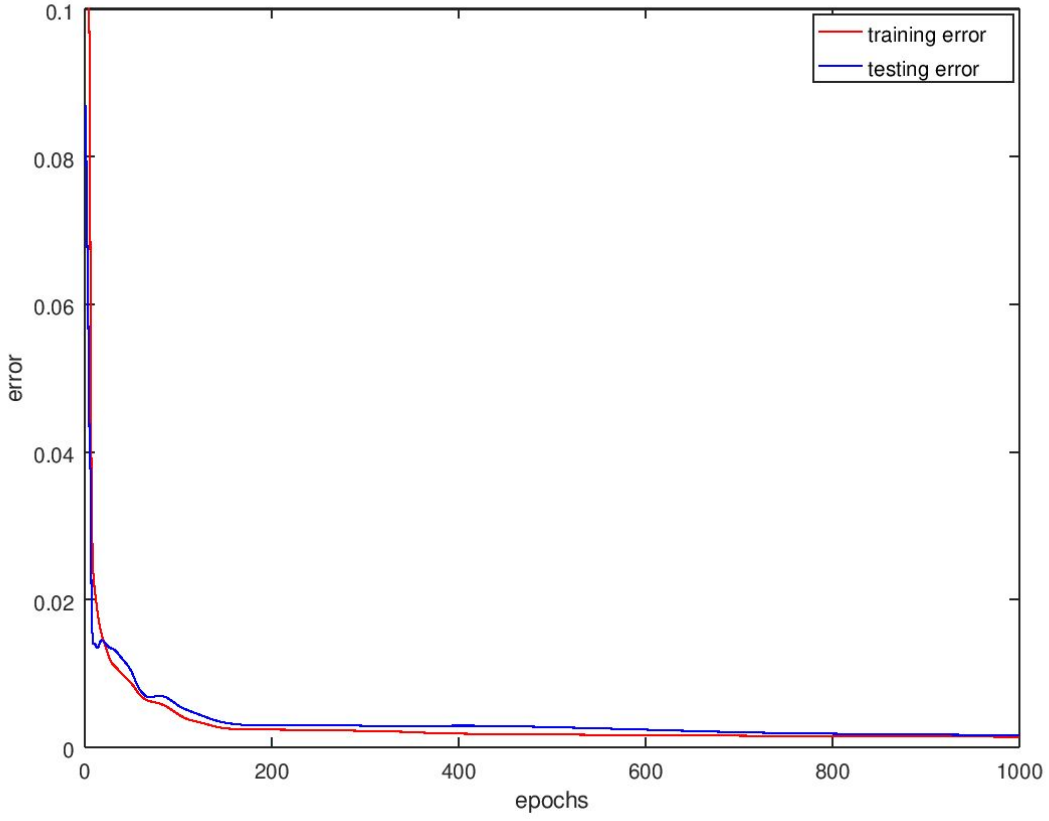

FIGURA 7

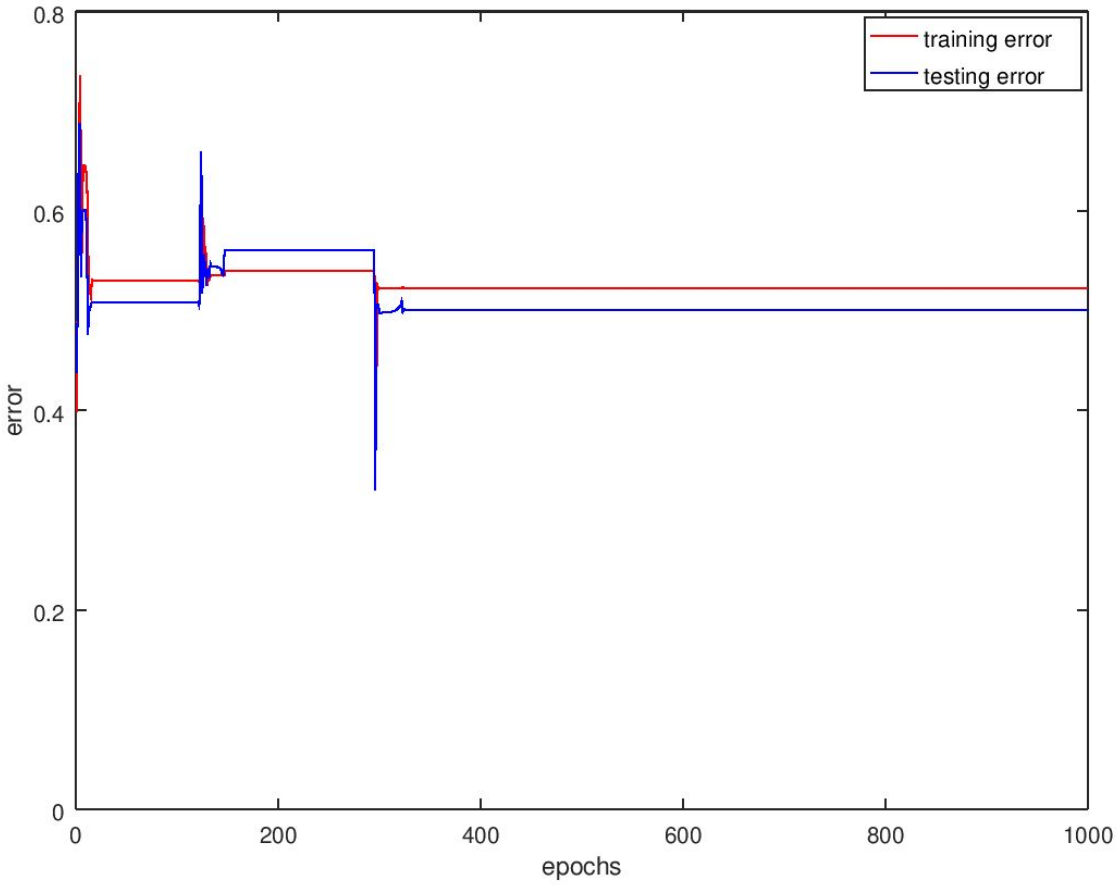

FIGURA 8

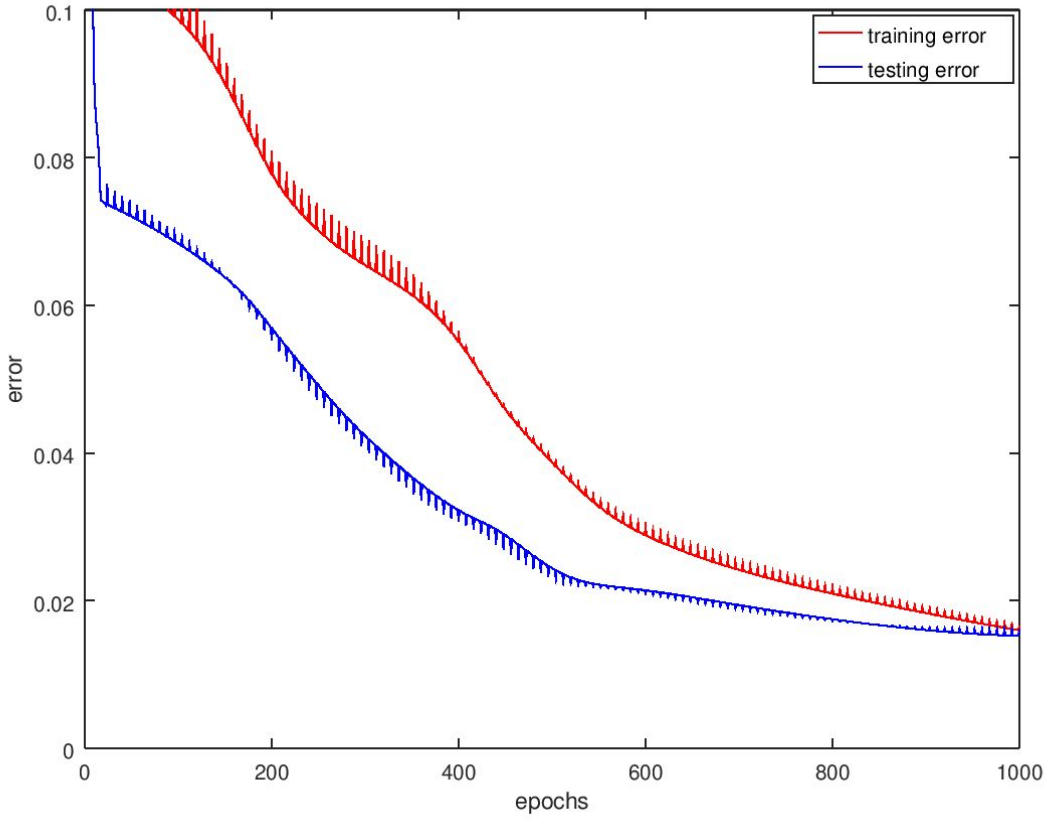

FIGURA 9

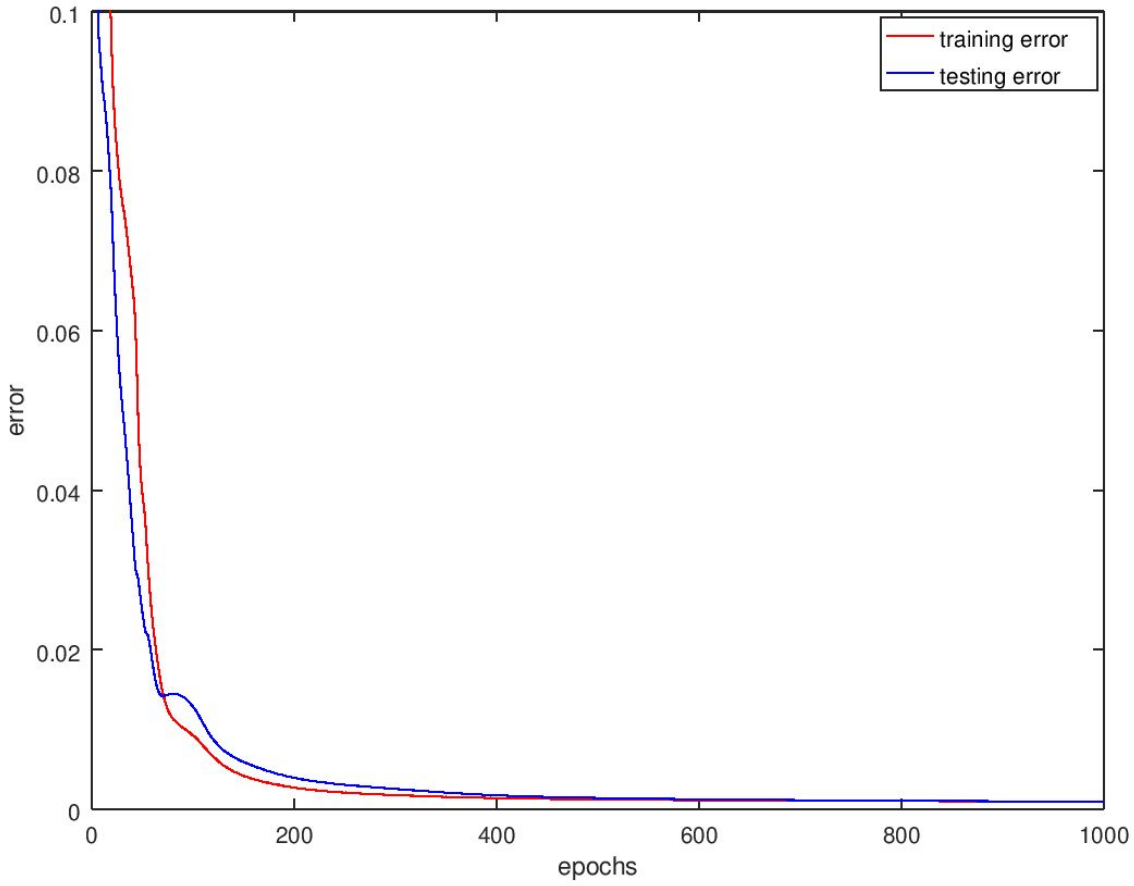

FIGURA 10

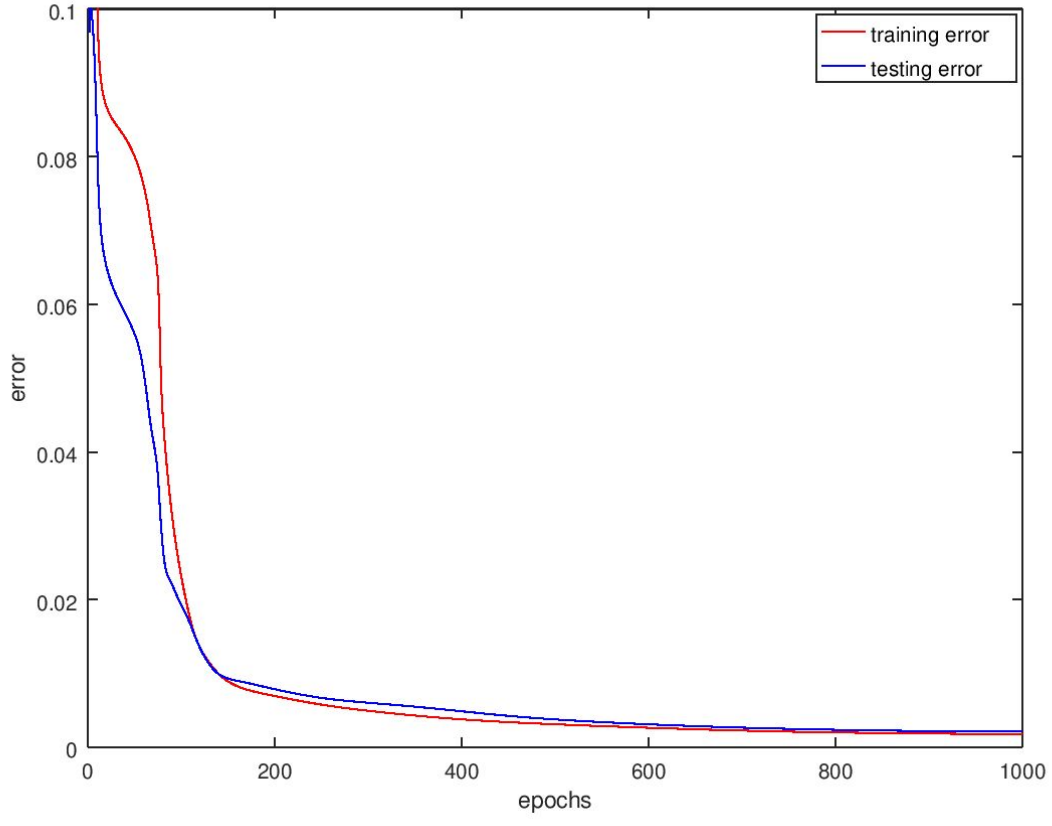

FIGURA 11

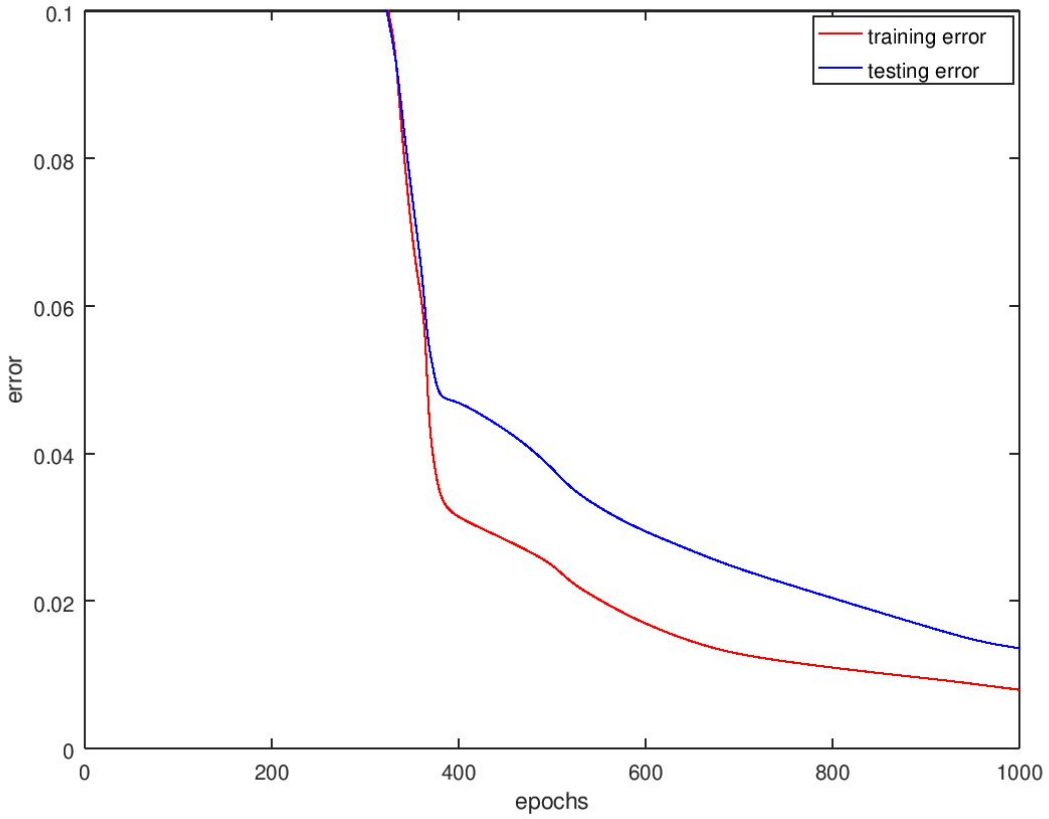

FIGURA 12