

CAPÍTULO 5

EVALUACIÓN DEL PROGRAMA

5.1 Imágenes de Entrenamiento

Las imágenes usadas para el entrenamiento del perceptrón corresponden a objetos sencillos de dibujar (y representar) y las letras vocales mayúsculas creadas con el editor de imágenes The Gimp. Para las letras se usó y el tipo de fuente 'comic' que es el más parecido a los caracteres manuscritos legibles:

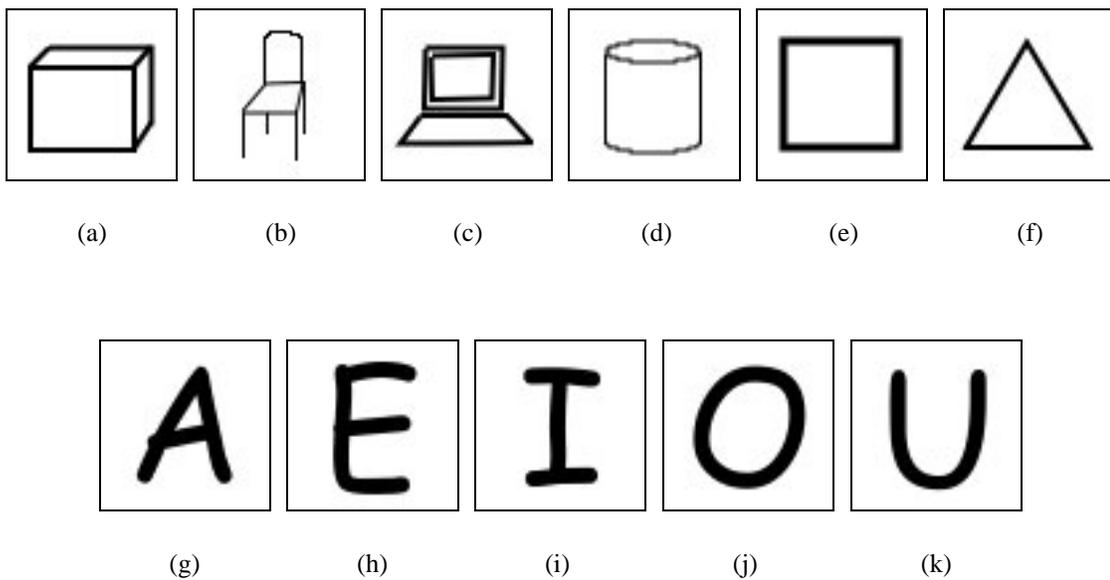


Figura 5.1. Imágenes de entrenamiento: (a) caja; (b) silla; (c) computadora; (d) cilindro; (e) cuadrado; (f) triángulo; (g) letra A; (h) letra E; (i) letra I; (j) letra O; (k) letra U.

Las 11 imágenes de entrenamiento mostradas en la Figura 5.1 son *grayscale* de 64x64 píxeles, y en todas se intentó que el objeto ocupara el mayor espacio posible y que estuviera centrado.

También es posible utilizar otras imágenes como serían los números del 0 al 9 u otros objetos o figuras geométricas básicas.

5.2 Entrenamiento

El objetivo del entrenamiento es, en este caso, entrenar 11 perceptrones multicapa iguales, cada uno con la transformada de distancia de un objeto de la Figura 5.1. Esto se hace 11 veces siguiendo los pasos:

 Leer una imagen de la Figura 5.1:

```
image=imread('C:\Path\image.jpg');
```

 Calcular la transformada de distancia de la imagen leída:

```
DistanceTransform(image);
```

 Entrenar un perceptrón multicapa de 2x9x25x1 con la transformada calculada:

```
ObjectX=NNDT(dt);
```

donde *ObjectX* es el nombre con el que se quiere guardar la red entrenada: *Box*, *Chair*, *LetterA*, *LetterE*, etc.

Para hacer más fácil los pasos anteriores se puede desarrollar un programa como el que agrupe los pasos anteriores como el programa desarrollado que se explica en el Apéndice A.

La evaluación del trabajo desarrollado se dividió en dos partes. Primero se trabajó con los objetos de la parte de arriba de la Figura 5.1 y luego con las letras de la misma figura.

En las siguientes páginas se muestran y se describen las gráficas del MSE que Matlab despliega automáticamente al entrenar una red neuronal, en nuestro caso, al ejecutar el programa *LearnImage.m* descrito en el Apéndice A.

La Figura 5.2 muestra la gráfica del comportamiento del MSE para la red que se entrenó con la transformada de distancia de Figura 5.1 (a) que representa una caja. Esta gráfica, al igual que las que se muestran en las figuras consecutivas, muestra el número de *epochs* que se requirieron para alcanzar el MSE deseado así como el valor del MSE en que se detuvo el entrenamiento. Más adelante, en la Tabla 5.1 se resumen los resultados obtenidos, así como los tiempos que se tardó cada entrenamiento.

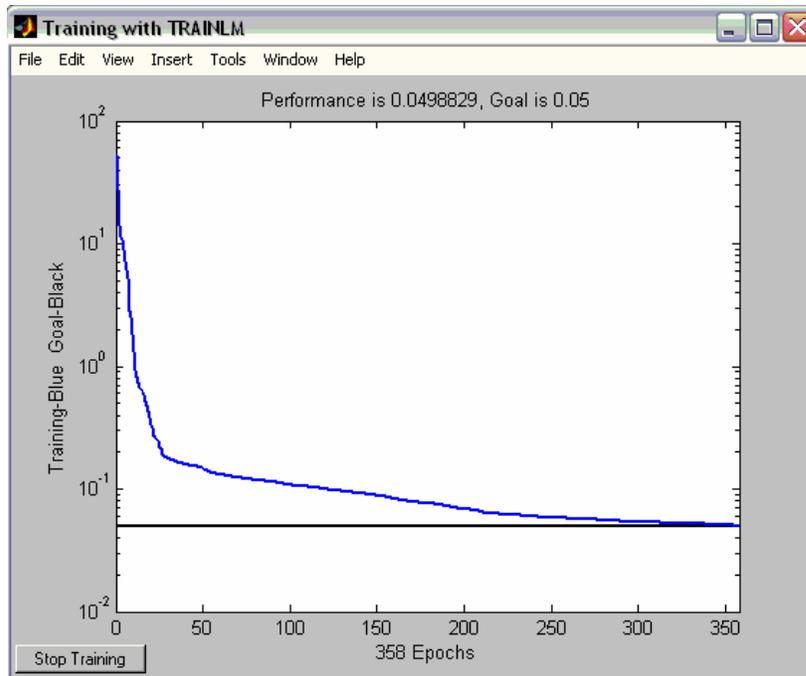


Figura 5.2. Gráfica del MSE del MLP entrenado para la Figura 5.1(a) (caja).

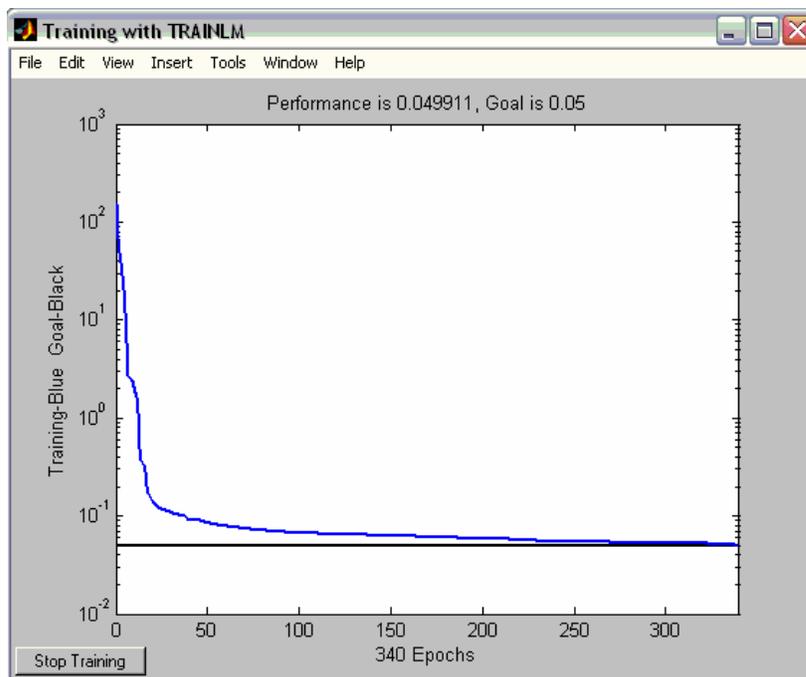


Figura 5.3. Gráfica del MSE del MLP entrenado para la Figura 5.1 (b) (silla).

La Figura 5.3 muestra la gráfica del comportamiento del MSE para el MLP que se entrenó con la transformada de distancia de la Figura 5.1 (b) que representa una silla. La siguiente figura muestra la gráfica del comportamiento del MSE para la red que se entrenó con la transformada de distancia de Figura 5.1 (c) que representa una computadora:

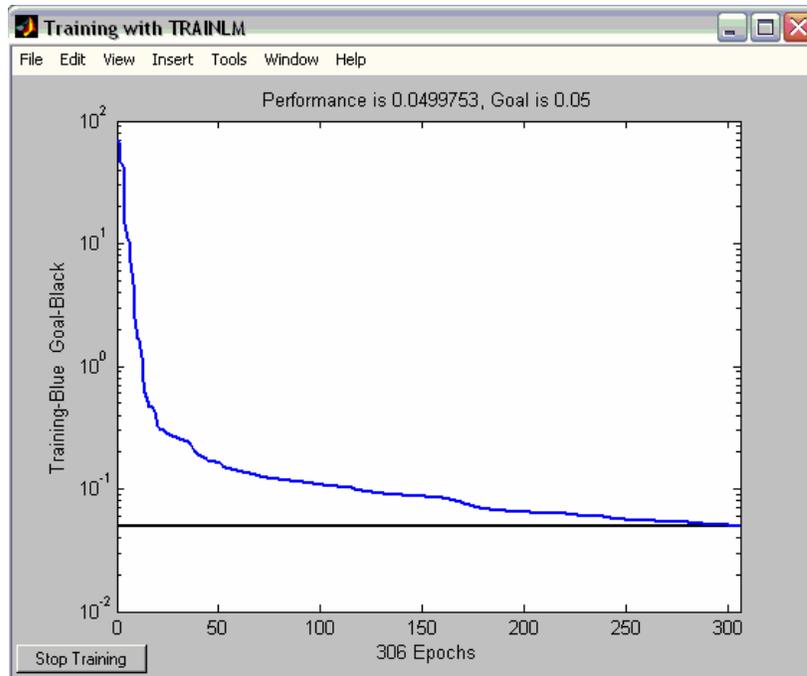


Figura 5.4. Gráfica del MSE del MLP entrenado para la Figura 5.1 (c) (computadora).

La Figura 5.5 muestra la gráfica del comportamiento del MSE de la red que se entrenó con la transformada de distancia de la Figura 5.1 (d) que representa un cilindro, y la Figura 5.6 muestra la gráfica del comportamiento del MSE de la red que se entrenó con la transformada de distancia de la Figura 5.1 (e) que representa un cuadrado. Más adelante, en la Figura 5.7 se muestra la gráfica del comportamiento del MSE del MLP que se entrenó con la transformada de distancia de la Figura 5.1 (f) que representa un triángulo.

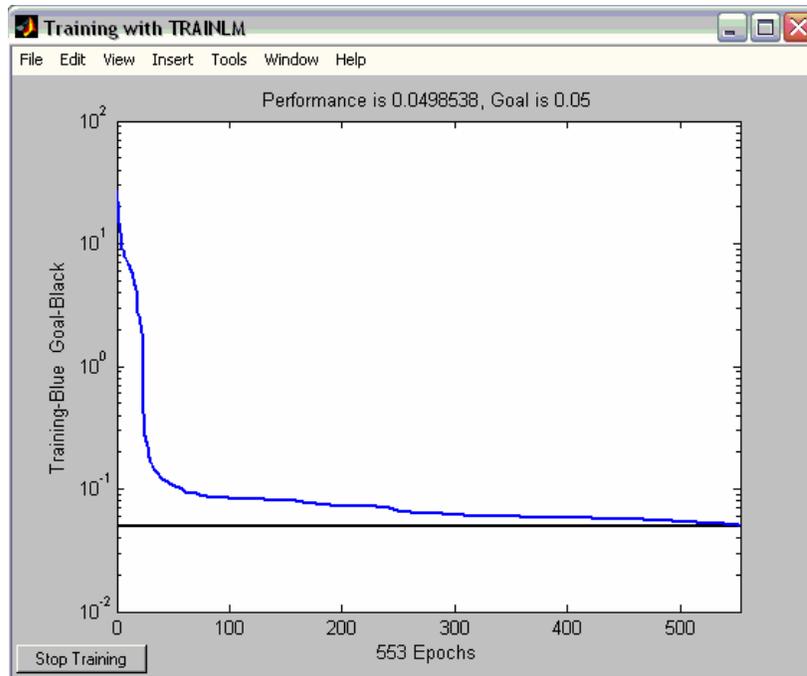


Figura 5.5. Gráfica del MSE del MLP entrenado para la Figura 5.1 (d) (cilindro).

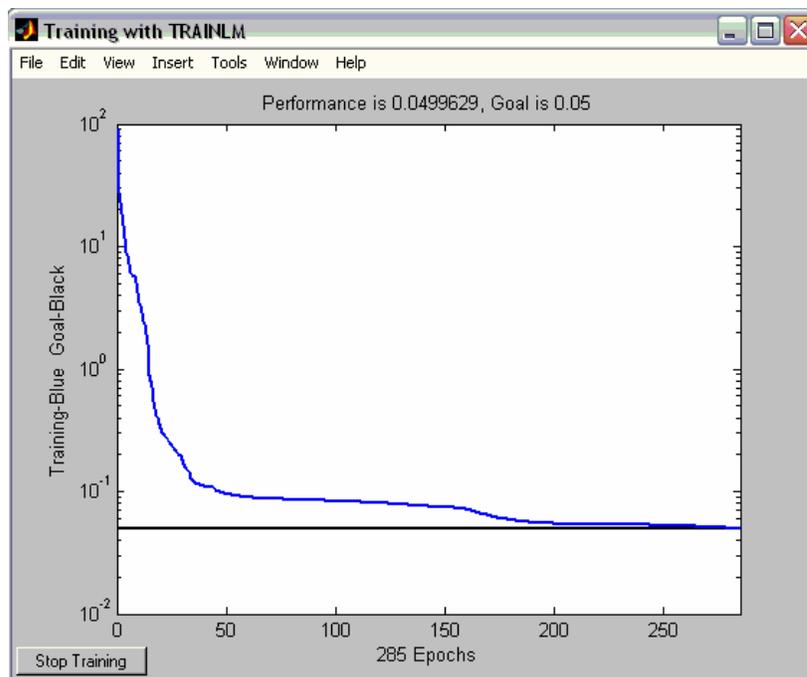


Figura 5.6. Gráfica del MSE del MLP entrenado para la Figura 5.1 (e) (cuadrado).

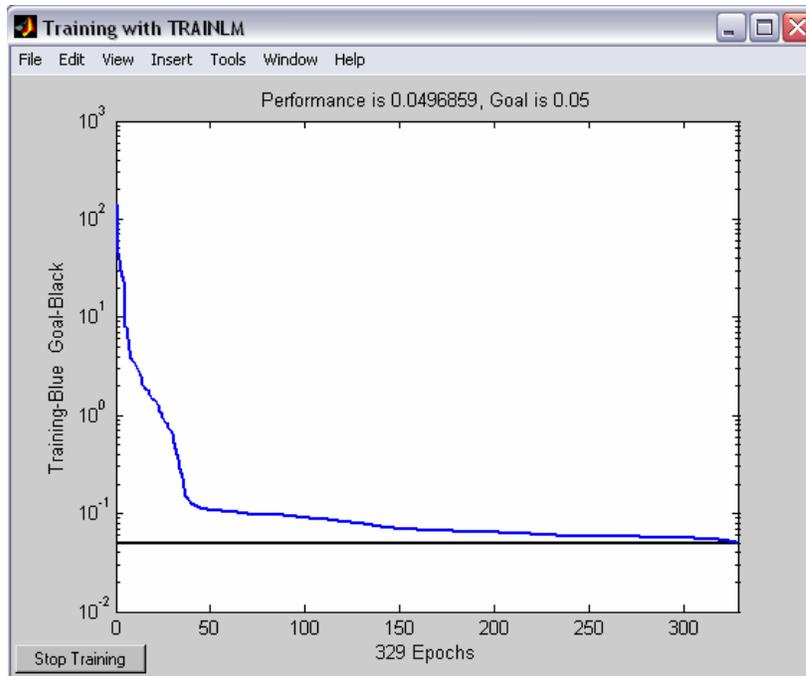


Figura 5.7. Gráfica del MSE del MLP entrenado para la Figura 5.1 (f) (triángulo).

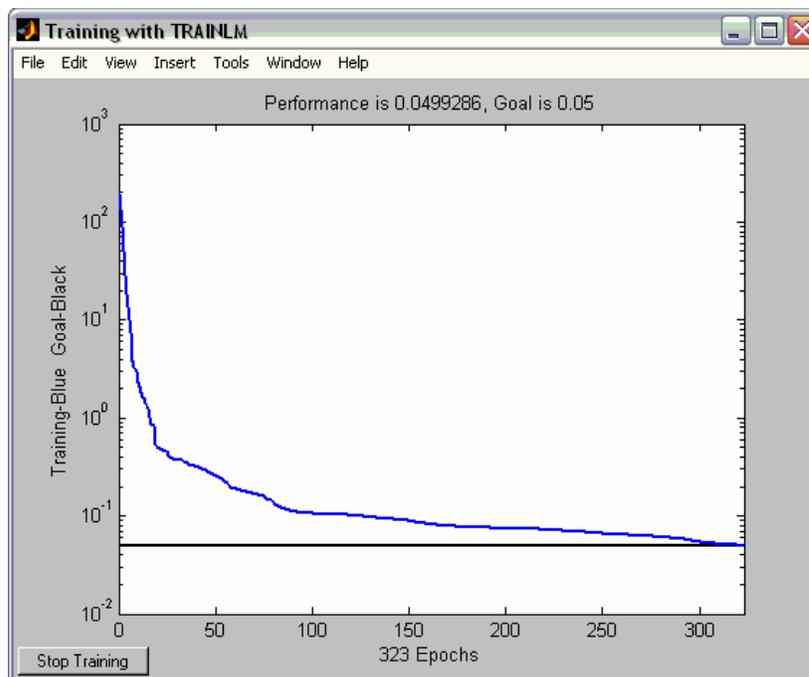


Figura 5.8. Gráfica del MSE del MLP entrenado para la Figura 5.1 (g) (letra A).

En la figura anterior (Figura 5.8) se muestra la gráfica del comportamiento del MSE de la red entrenada con la transformada de distancia de la Figura 5.1 (g) que representa la letra A. En la siguiente figura se muestra la gráfica del comportamiento del MSE de la red entrenada con la transformada de distancia de la Figura 5.1 (h) que representa la letra E:

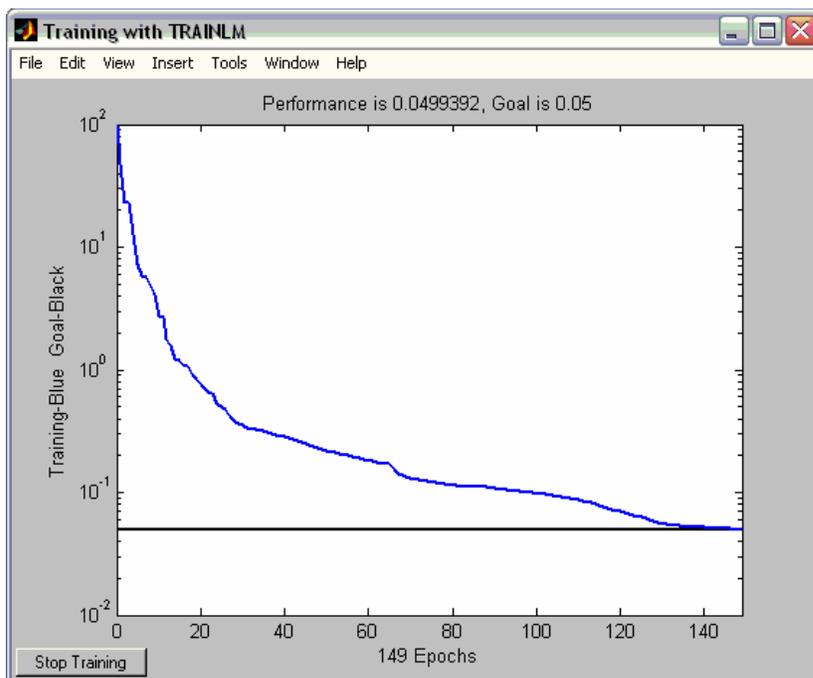


Figura 5.9. Gráfica del MSE del MLP entrenado para la Figura 5.1 (h) (letra E).

En las Figuras 5.10, 5.11 y 5.12 se muestran las gráficas del comportamiento de los MSE de las redes entrenadas con las transformadas de distancia de las Figuras 5.1 (i), (j) y (k) que representan las letras I, O y U respectivamente.

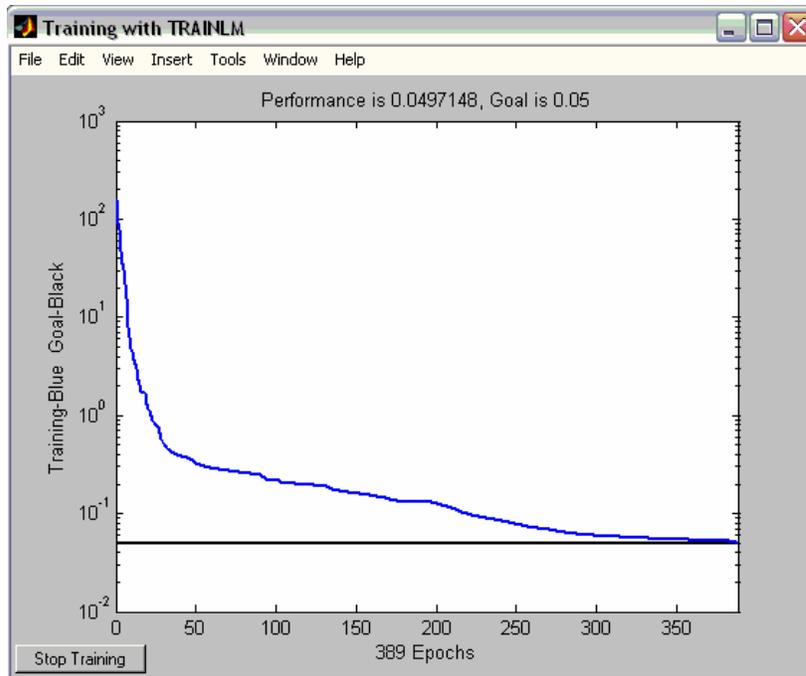


Figura 5.10. Gráfica del MSE del MLP entrenado para la Figura 5.1 (i) (letra I).

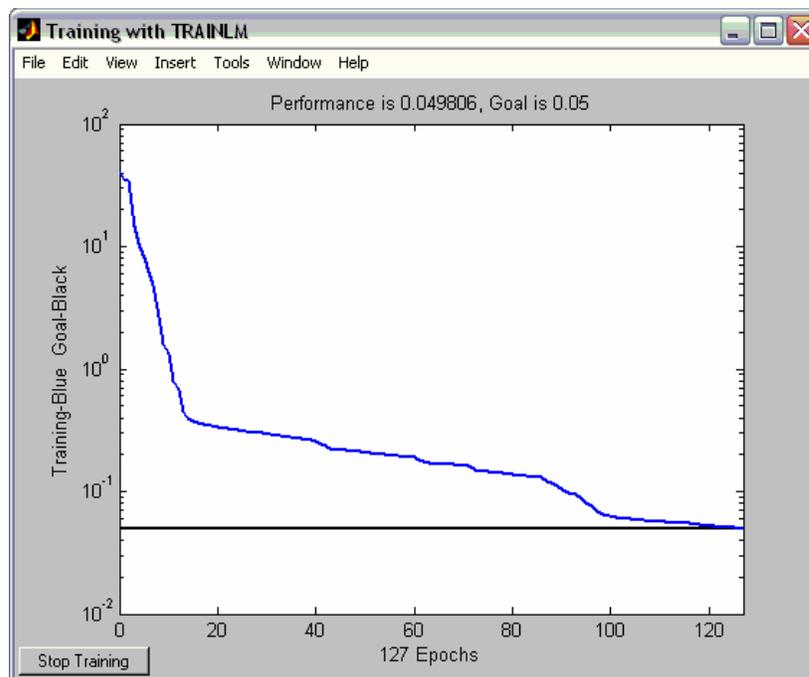


Figura 5.11. Gráfica del MSE del MLP entrenado para la Figura 5.1 (j) (letra O).

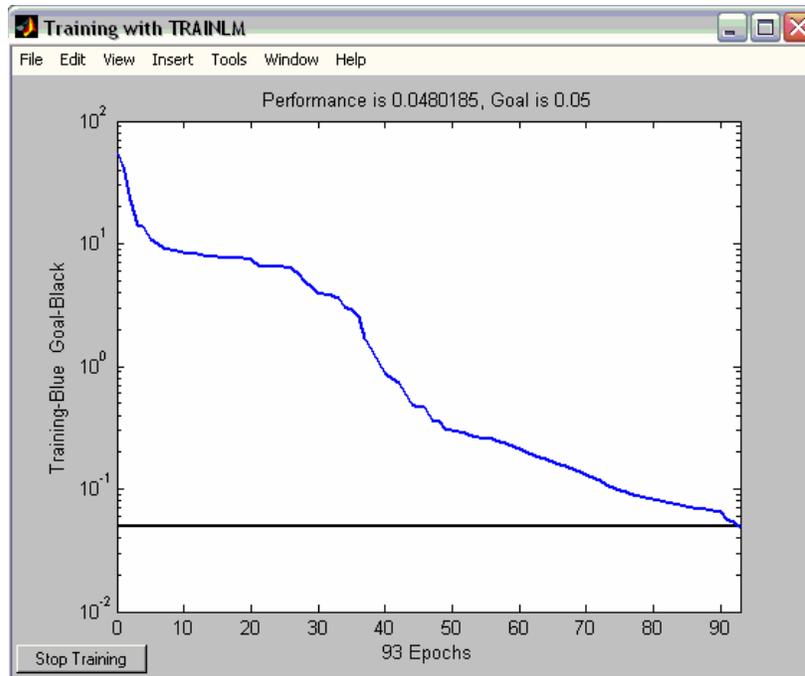


Figura 5.12. Gráfica del MSE del MLP entrenado para la Figura 5.1 (k) (letra U).

Matlab muestra el progreso del MSE gráficamente y por escrito cada tantos *epochs* como se le indique con el objeto *net.trainParam.show*. En las gráficas se va mostrando con una línea azul la gráfica del MSE y con una línea negra el valor al que se quiere llegar. Las gráficas tienen la excelente opción de parar el entrenamiento, *Stop Training*, que es de gran utilidad cuando no se ven resultados satisfactorios como por ejemplo cuando el MSE llega a un mínimo local y se queda estancado ahí por un periodo largo de tiempo, en cuyo caso es mejor volver a iniciar el entrenamiento.

Como se puede apreciar, los tiempos de entrenamiento varían ampliamente debido principalmente a los pesos iniciales de la red neuronal. Cuando se percibe un cambio mínimo en el MSE y aún está lejos de su objetivo, es mejor parar el entrenamiento y repetir el proceso, aunque hay casos en los que el MSE parece detenerse en un valor pero al poco

tiempo vuelve a avanzar eficientemente.

La tabla siguiente resume los resultados obtenidos de los entrenamientos anteriores:

Tabla 5.1. Resultados del entrenamiento de los MLP con las transformadas de distancia de imágenes de la

Figura 5.1.

Objeto	MSE deseado	MSE obtenido	Epochs	Tiempo
Caja	0.05	0.0498829	358	158.969s
Silla	0.05	0.049911	340	156.125s
Computadora	0.05	0.0499753	306	135s
Cilindro	0.05	0.0498538	553	245.891s
Cuadrado	0.05	0.0499629	285	128.078s
Triángulo	0.05	0.0496859	329	148.172s
Letra A	0.05	0.0499286	323	143.984s
Letra E	0.05	0.0499392	149	66.672s
Letra I	0.05	0.0497148	389	172.859s
Letra O	0.05	0.049806	127	57.141s
Letra U	0.05	0.0480185	93	42.391s

Los tiempos de entrenamiento varían, pero pueden considerarse eficientes si no pasan de los 5 minutos para este trabajo en específico porque cuando se tarda más es porque el entrenamiento está estancado en un mínimo local del MSE y muchas veces es mejor reiniciar el proceso si no se observa una convergencia rápida. En todos los casos, el MSE deseado fue de 0.05 y se desecharon algunos entrenamientos debido a que no se observaba una convergencia rápida hacia este MSE. Los tiempos dependen enteramente del algoritmo de entrenamiento, el cual a su vez, depende de la velocidad con la que se realizan los cálculos. Para este trabajo se usó una computadora con procesador de 2.4GHz.

5.3 Imágenes de Prueba

Para probar la fiabilidad del programa en conjunto, se dibujaron 91 objetos de los cuales 70 corresponden a las vocales mayúsculas y 21 corresponden a los demás objetos, tratando de hacerlos lo más parecido posible a las imágenes con que se entrenaron los MLPs. Las imágenes son *grayscale* de 64x64 píxeles y se dibujaron en el editor de imágenes The Gimp con el puntero del *mouse*. La Figura 5.13 muestra las imágenes de los objetos elaborados para realizar la primera prueba de reconocimiento que corresponde a los objetos de la parte superior de la Figura 5.1.

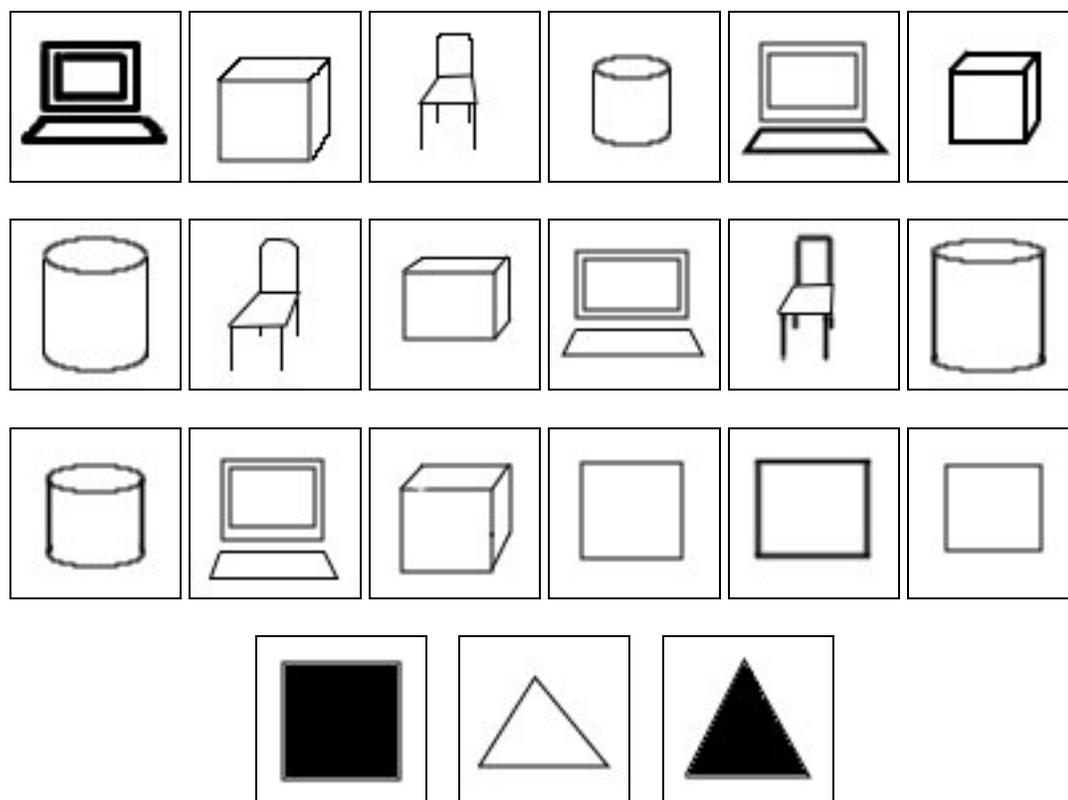


Figura 5.13. Imágenes para la primera prueba.

Las imágenes de la Figura 5.13 se guardaron en el destino que se toma para la elaboración del programa de reconocimiento (Apéndice A) con el nombre imagex.jpg, donde x indica el número de imagen (1-21) y van en orden de derecha a izquierda y de arriba abajo. En la siguiente figura se muestran las imágenes empleadas para otra prueba de reconocimiento, esta vez correspondiente a los objetos de la parte inferior de la Figura 5.1:

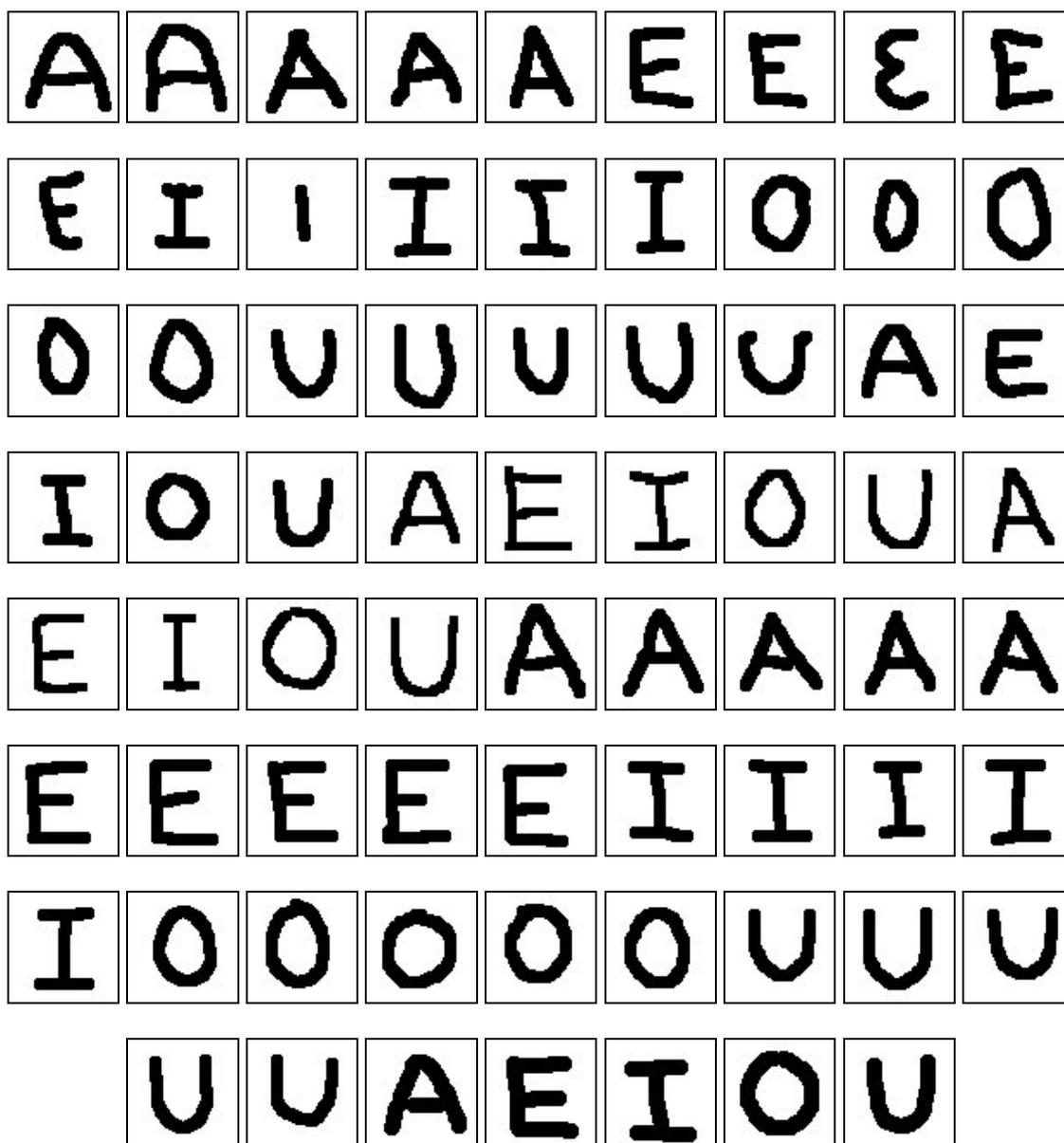


Figura 5.14. Imágenes para la segunda prueba.

5.4 Reconocimiento

En esta etapa se puede reconocer una imagen o todas a la vez, que es más práctico para mostrar resultados con fines de estudio. Se crearon dos programas, uno para mostrar las lecturas de las imágenes de corrido, y otro para mostrar la lectura de una imagen en específico. El primero se describe a continuación:

-  Se lee una imagen.
-  Se saca su contorno.
-  Se invierten los valores de los píxeles.
-  Se detectan los píxeles que conforman el contorno.
-  Se simulan los perceptrones multicapa de la base de datos con los píxeles anteriores.
-  Se saca el promedio de distancias obtenidas en casa red.
-  Se calcula el menor promedio de distancias y se reconoce la imagen.
-  Se repite el proceso para todas las imágenes de prueba desplegando en cada una la letra que se reconoce.

cuyo código se describe en el Apéndice A.

A la hora de ejecutar el programa de reconocimiento se obtuvieron los siguientes resultados:

 De la Figura 5.13 no se reconocieron satisfactoriamente los objetos 4, 6, 12, 13 y 18, es decir, no se reconocieron 5 objetos de 21 que equivale a 76.2% de confiabilidad.

 De la Figura 5.14 no se reconocieron satisfactoriamente los objetos 2, 17, 19 y 68, es decir, no se reconocieron 4 objetos de 70 que equivale a 94.3% de confiabilidad.

La ejecución completa del programa de reconocimiento para las dos pruebas se muestra en el Apéndice C. Obviamente estos resultados varían dependiendo 100% de que las imágenes de prueba se parezcan a las imágenes de entrenamiento, así como también de la cantidad de imágenes con la que se quiere calcular el porcentaje de confiabilidad. Si se tienen sólo dos objetos y uno no es reconocido correctamente, entonces el porcentaje de confiabilidad se calcularía como de 50%, o peor aún, si ningún objeto es reconocido correctamente entonces la confiabilidad sería 0%. Esto no dice nada sobre el buen o mal desempeño del sistema, lo óptimo es contar con una gran cantidad de objetos de prueba para obtener conclusiones eficientes en cuanto al desempeño del sistema.

5.5 Resultados

A continuación se muestran las imágenes que no fueron identificadas correctamente:

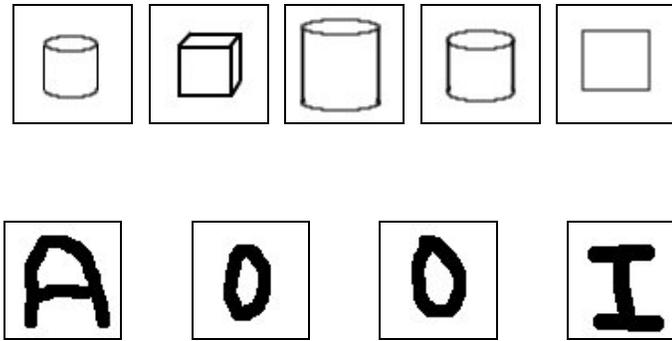


Figura 5.15. Objetos reconocidos incorrectamente.

De la figura 5.15 se puede ver que es necesario que los objetos sean lo más parecido posible a los objetos de entrenamiento. En estos casos la imagen se asocia con la que tenga menor distancia. El grosor de línea también puede “engañar” al reconocimiento así como el tamaño y posición.

El porcentaje de confiabilidad del programa desarrollado no puede encasillarse entre 76% y 94%, ya que depende de qué tan similares sean los objetos de prueba a los objetos de entrenamiento. También depende del entrenamiento porque se ejecutó varias veces esta etapa y el porcentaje de confiabilidad variaba algunas veces al fallar en más ocasiones de las mencionadas arriba.

Los tiempos de reconocimiento que se obtuvieron son relativamente buenos, ya que se encuentran alrededor de los 200 milisegundos. Los tiempos de reconocimiento dependen enteramente de cuantos píxeles conforman el contorno de la imagen, porque es el número total de *inputs* que se presentan a las redes neuronales. Para imágenes con dimensiones mayores a 64x64 tanto el entrenamiento como el reconocimiento se llevarán más tiempo.

Es necesario recordar que este trabajo se hizo en una computadora con procesador a 2.4GHz, lo cual también afecta al tiempo de reconocimiento, pero este factor no es para preocuparse ya que los procesadores continuamente tienden a aumentar considerablemente su velocidad.