



UNIVERSIDAD DE COSTA RICA

Facultad de Ingeniería

Escuela de Ciencias de la Computación e Informática

Documentación

Tarea Programada

CI-1441 Paradigmas Computacionales

Elaborado por:

José Andrés Mena, carné: A93767

Profesora: Yadira Solano Sabatier

8 de Junio del 2012

Contenido

| | | |
|------|---|----|
| 1. | Herramienta implementada | 3 |
| 1.1. | Descripción general..... | 3 |
| 1.2. | Aspectos técnicos..... | 3 |
| 2. | Manual de usuario | 4 |
| 2.1. | Requerimientos de hardware y software | 4 |
| 2.2. | Instalación y ejecución..... | 4 |
| 2.3. | Navegación general..... | 4 |
| 2.4. | Entrenar la red neuronal..... | 5 |
| 2.5. | Probar la red neuronal | 9 |
| 2.6. | Ubicación de los archivos de entrada | 13 |
| 2.7. | Codificación y formato de los archivos de entrada | 14 |
| 2.8. | Interpretación y manejo de resultados..... | 15 |
| 3. | Análisis de los resultados obtenidos..... | 17 |
| 3.1. | Resultados obtenidos al entrenar la red..... | 17 |
| 3.2. | Resultados obtenidos al probar la red..... | 19 |
| 4. | Otras consideraciones..... | 28 |

1. Herramienta implementada

1.1. Descripción general

La aplicación diseñada para esta tarea programada se encarga de administrar una red neuronal de propagación hacia atrás. Provee los comandos necesarios para entrenar y probar dicha red neuronal, de manera que reconozca las vocales tanto mayúsculas como minúsculas.

1.2. Aspectos técnicos

El sistema completo, incluida la red neuronal, están implementados bajo la plataforma Java. La estructura del programa es la siguiente (ver figura 1):

1. Existe una red neuronal de propagación hacia atrás que contiene los métodos de introducción de entradas, propagación hacia adelante, cómputo del error, ajuste de pesos y obtención de salidas. Esta red contiene tres capas: una capa de entrada con veinte unidades, una capa oculta y una capa de salida con diez unidades.
2. La red neuronal es manejada por una clase administradora que se encarga de funciones de alto nivel como el entrenamiento y la configuración de los pesos.
3. Un administrador de archivos se encarga de manejar todo lo referente a entradas y salidas con archivos de texto.
4. El sistema cuenta con interfaz gráfica, la cual utiliza clases auxiliares para implementar una cuadrícula de dibujo y para graficar resultados.
5. El flujo principal del programa es manejado por una clase controladora, en la cual se crean todos los objetos y en ella se pueden modificar todos los valores constantes por defecto.

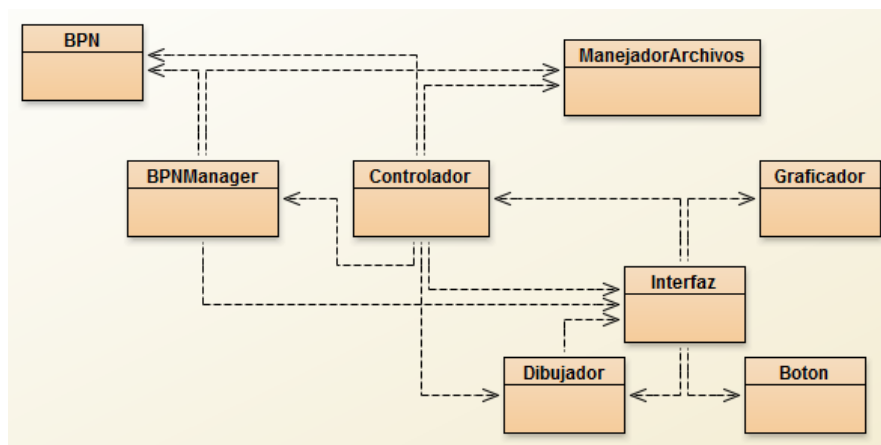


Figura 1. Estructura del sistema

2. Manual de usuario

2.1. *Requerimientos de hardware y software*

El sistema se diseñó haciendo uso del lenguaje de programación Java versión jdk1.7.0_04 con el IDE BlueJ. La implementación y las pruebas se realizaron en el sistema operativo Windows 7 y en máquinas con procesador Intel.

Por lo tanto, para correr la aplicación es necesario que la máquina cuente con la última versión de la plataforma de Java SE 7u4 disponible en <http://www.oracle.com/technetwork/java/javase/downloads/index.html>. Además, se recomienda ejecutarlo en ambiente Windows y en máquinas con procesador Intel (Se detectaron problemas en el manejo de hilos en procesadores AMD, por lo que el graficado de resultados podría presentar problemas).

2.2. *Instalación y ejecución*

Una vez que se tenga el archivo TP_JoseAndresMena.zip, se deben extraer los archivos en su máquina. En la carpeta “Ejecutable” se encuentra el archivo TP_JoseMena.jar, para el cual se requiere una versión de la máquina virtual de java como se mencionó en la sección 2.1.

Ejecute el archivo TP_JoseMena.jar para empezar a utilizar el programa.

En caso de que se quiera ejecutar el programa desde el ambiente en el que fue programado, se debe abrir el proyecto ubicado en la carpeta “Código Fuente” con el IDE BlueJ. Se presiona clic derecho sobre la clase Controlador y se elige el método main, se presiona aceptar y se iniciará la ejecución.

2.3. *Navegación general*

Para utilizar las distintas opciones del programa, se debe hacer uso de la barra de menú superior (Figura 2). En el menú “Acciones” se encuentran las opciones principales del programa que le permiten administrar la red neuronal, los cuales son:

Entrenar la red: Permite introducir los parámetros y el archivo de casos necesarios para iniciar el entrenamiento de la red neuronal y visualizar el comportamiento.

Probar la red: Presenta una pantalla con diferentes comandos para probar la red neuronal: cargar pesos, dibujar y cargar casos de entrada y comprobar los resultados de la red.

En el menú “Ayuda” se muestra información adicional del programa: “Acerca de” y “Créditos”.

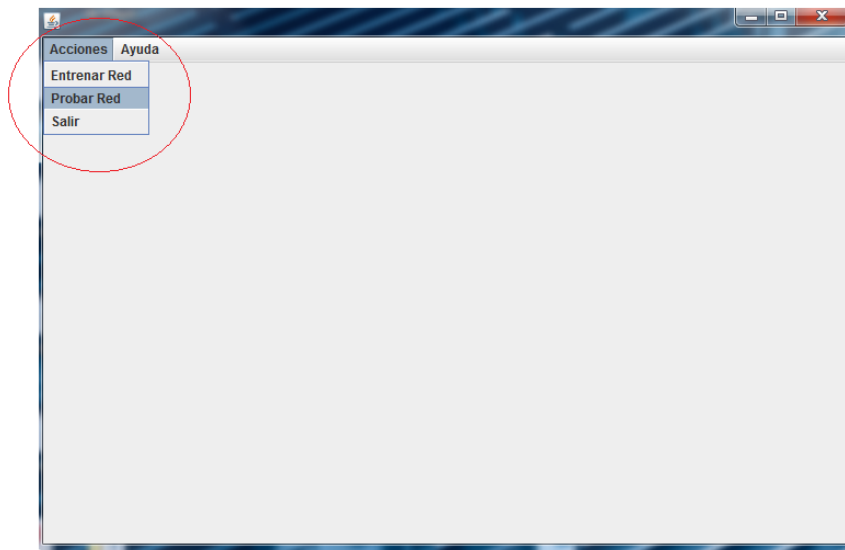


Figura 2. Menú principal en la barra superior.

2.4. Entrenar la red neuronal

Al ingresar a la ventana de entrenamiento (Figura 3), se muestran los campos requeridos para ingresar los valores con los cuales se quiere entrenar la red:

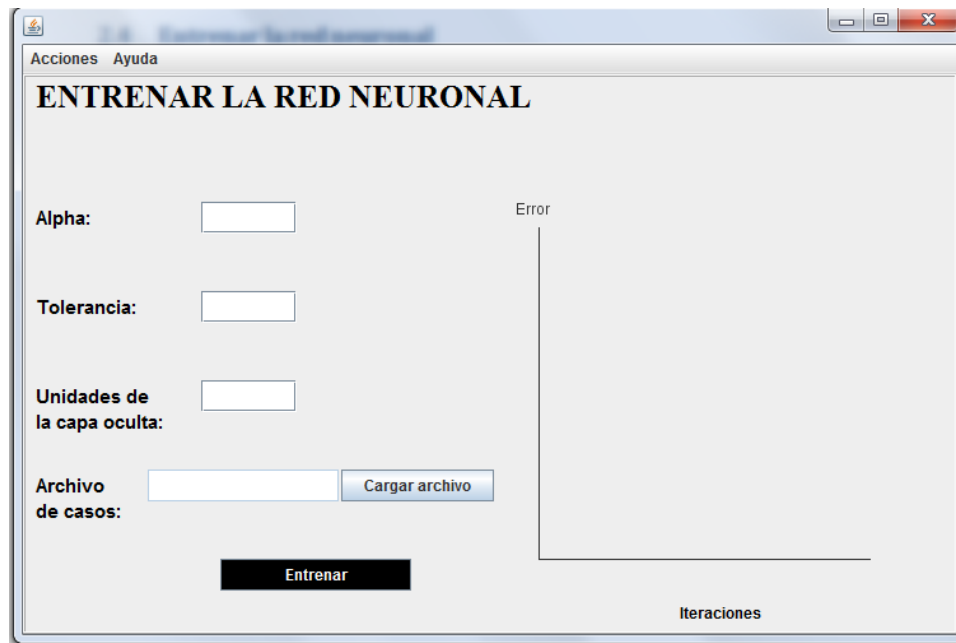


Figura 3. Ventana de entrenamiento de la red neuronal

Alpha: Es un valor numérico que puede contener decimales (utilizando la notación de punto) y que representa el coeficiente de aprendizaje que utilizará la red para ajustar sus pesos durante el entrenamiento. Un ejemplo de valor válido para este campo es “0.8”.

Tolerancia: Valor numérico con decimales, entre cero y uno, que determina el rango aceptado para el cálculo del error en la red durante el entrenamiento. Por ejemplo, si se quiere una tolerancia del 10% se introduce el valor: “0.1”.

Unidades de la capa oculta: Se debe ingresar un valor numérico entero, que representa el número de unidades que contendrá la capa oculta de la red neuronal. Por ejemplo, un valor válido para este campo es: “200”.

Archivo de casos: Este campo no es editable. Para seleccionar un archivo, se debe presionar el botón “Cargar archivo”. Al hacerlo, aparecerá una ventana de navegación, en la cual se debe ubicar la carpeta que contiene un archivo válido que contenga casos de entrenamiento (Sobre el formato del archivo, ver la sección 2.6) como se muestra en la figura 4. Al seleccionar el archivo se debe presionar “open” para elegirlo.

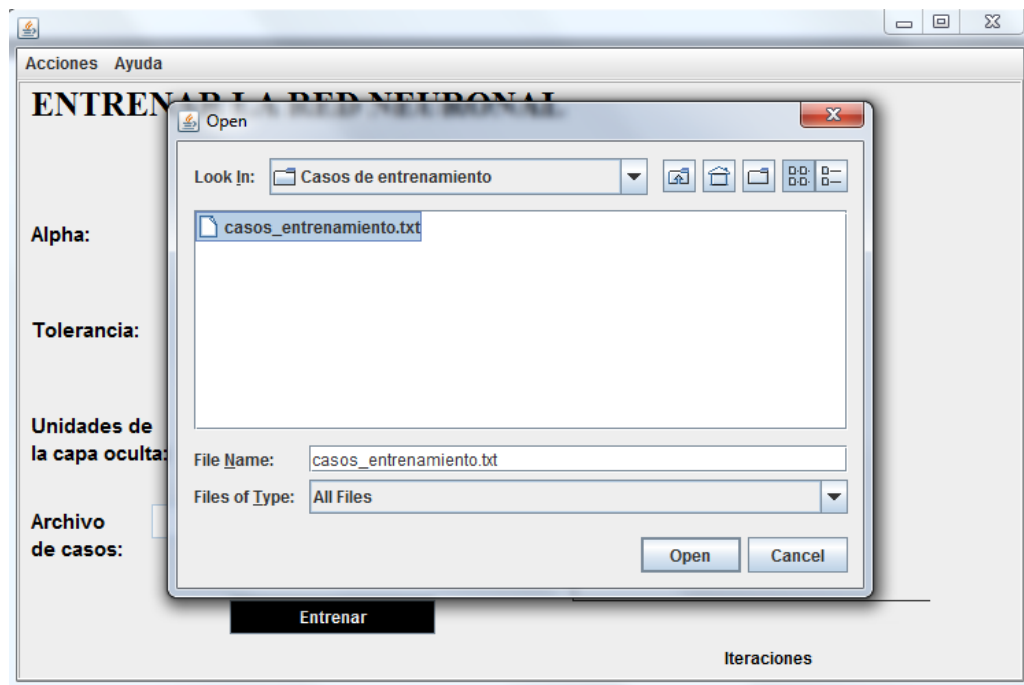


Figura 4. Ventana de navegación para elegir archivo de casos

Una vez elegido el archivo de casos, su nombre aparecerá en el campo correspondiente.

Por lo tanto, para iniciar el entrenamiento, se deben llenar todos los campos mencionados con valores válidos, cargar el archivo de casos y presionar el botón “Entrenar”. El sistema mostrará un mensaje indicando que el entrenamiento está siendo realizado y, al mismo tiempo, se comenzará a desplegar el comportamiento del error en el gráfico de la derecha (figura 5).

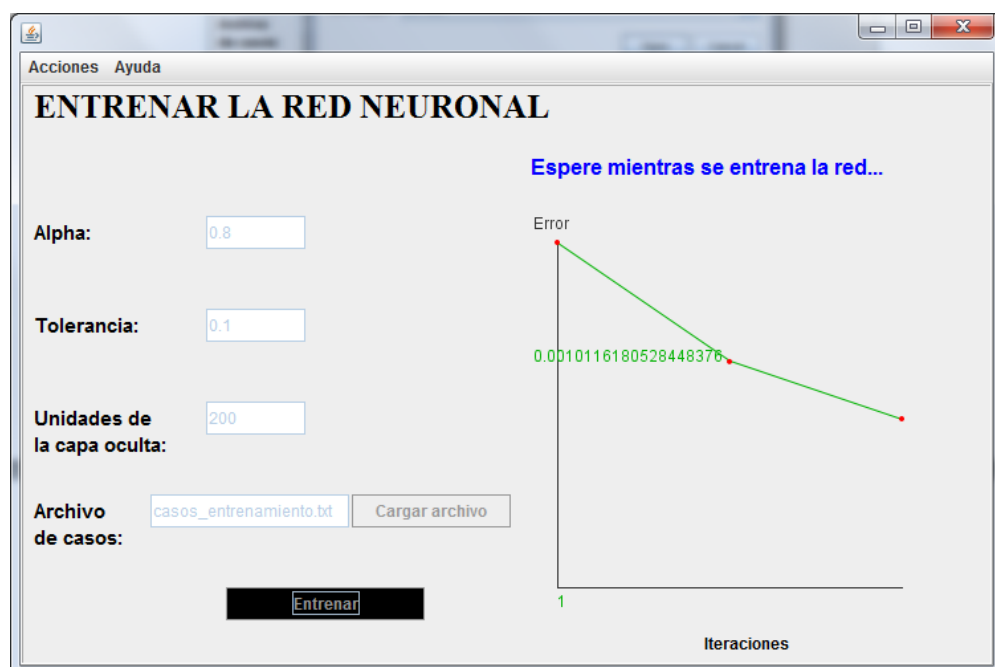


Figura 5. Ventana durante el entrenamiento de la red neuronal

Cuando la red neuronal haya terminado el entrenamiento, se mostrará un mensaje en pantalla indicándolo (figura 6). Al presionar “OK”, se abrirá una ventana de navegación en la cual se debe elegir alguna carpeta, escribir un nombre para el archivo con los pesos y presionar “Save”, como se muestra en la figura 7.

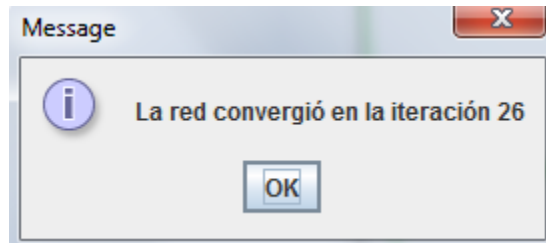


Figura 6. Mensaje que se muestra al finalizar el entrenamiento

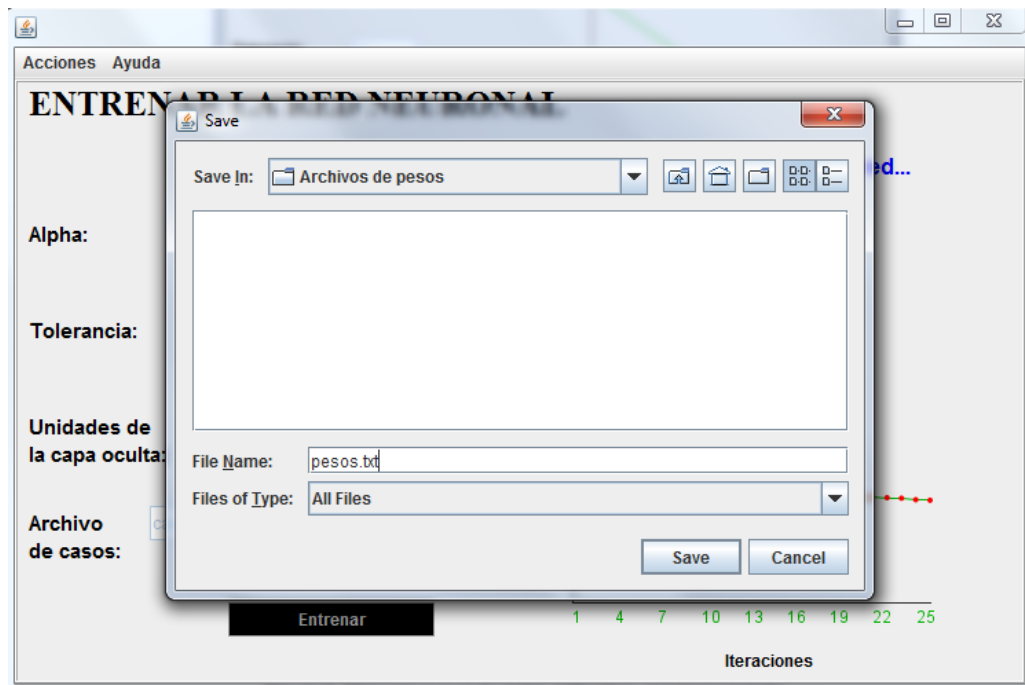


Figura 7. Ventana de navegación para guardar archivo de pesos

En caso de que se deje en blanco algún campo requerido o que se inserte algún valor en formato no válido, el sistema mostrará un mensaje de error y no permitirá iniciar el entrenamiento. De igual manera, se muestra un mensaje de error durante el entrenamiento si se detecta que el archivo de casos no tiene el formato requerido.

2.5. Probar la red neuronal

Al elegir la opción “Probar Red” en el menú, aparece la ventana que se muestra en la figura 8 y en la cual se pueden realizar varias acciones. El principal objetivo de esta ventana es probar la capacidad de la red para reconocer vocales, haciendo uso del conocimiento obtenido durante un entrenamiento previo.

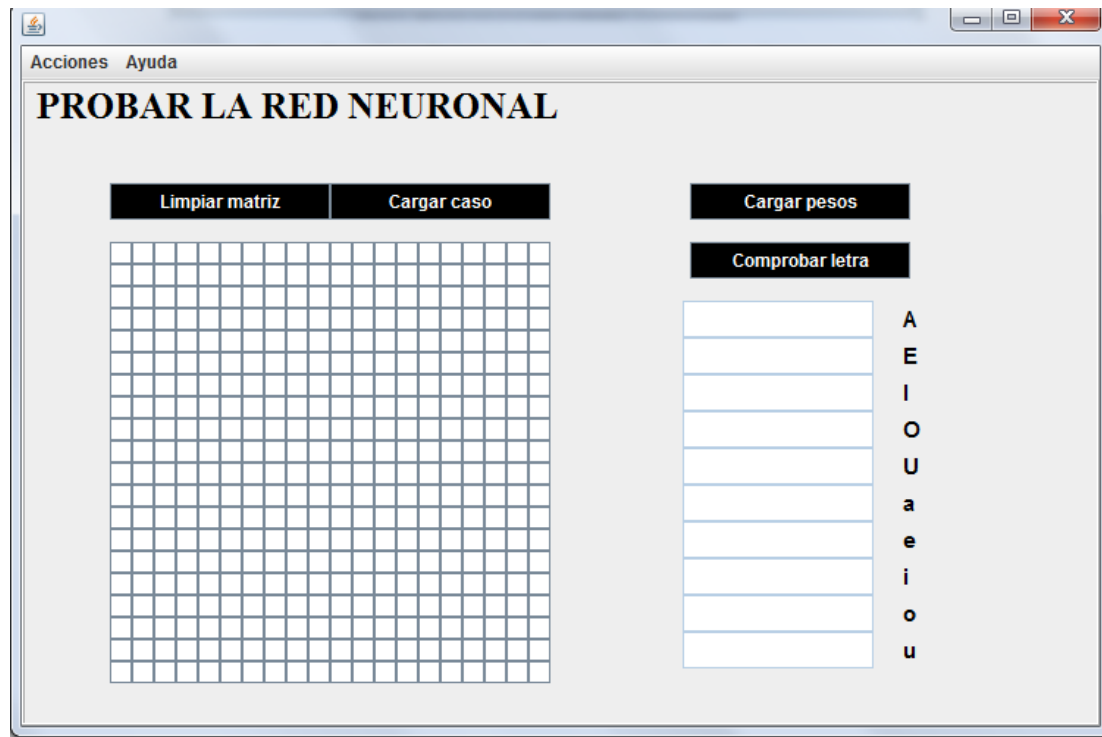


Figura 8. Ventana que permite probar la red neuronal

Si la red aún no ha sido entrenada, quiere decir que sus pesos contienen valores aleatorios, por lo que no tiene sentido probarla en esa condición (Es muy probable que no reconozca ninguna de las vocales). Sobre cómo entrenar la red neuronal, refiérase a la sección 2.4.

Si se cuenta con algún archivo de pesos generados durante un entrenamiento previo, se puede hacer uso del comando “Cargar pesos”. Al presionar dicho botón, se abre una ventana de navegación donde se debe ubicar la carpeta que contiene el archivo de pesos válido, seleccionar el archivo y presionar “Open” (Figura 9). El sistema mostrará un mensaje de confirmación indicando que los pesos de la red neuronal se han cargado satisfactoriamente.

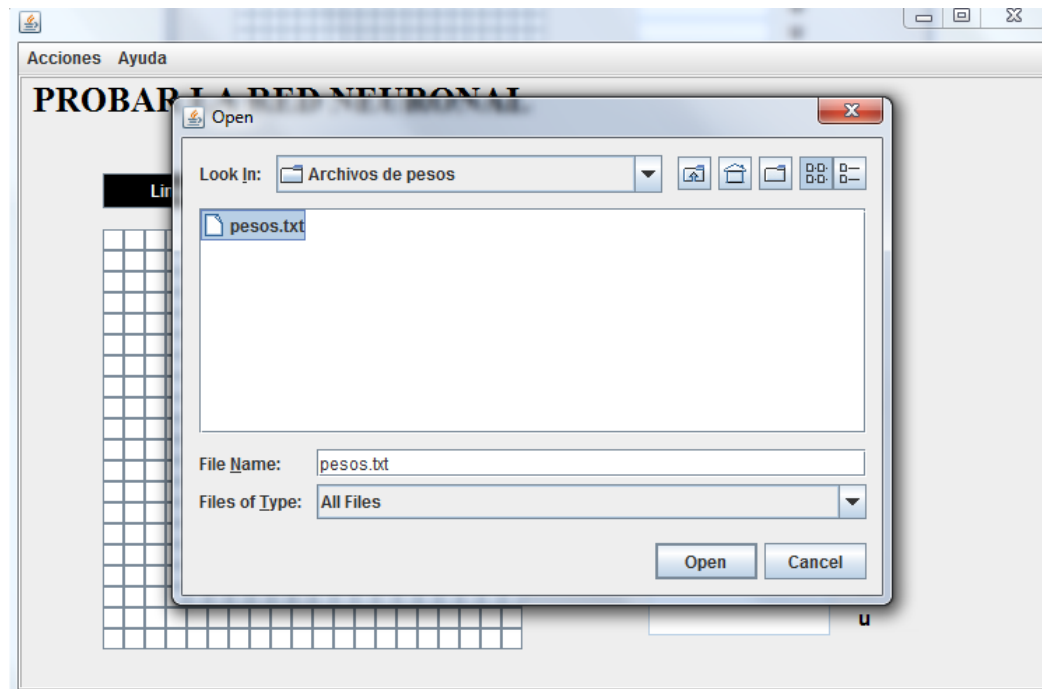


Figura 8. Ventana de navegación para cargar pesos desde un archivo

Para empezar a probar la red neuronal, se deben establecer sus valores de entrada haciendo uso de la matriz que aparece a la izquierda de la ventana. Esto se puede hacer de dos maneras:

Cargando un caso de prueba desde un archivo: Para ello se debe presionar el botón “Cargar caso”, donde aparecerá una ventana de navegación desde la cual se ubica la carpeta que contiene el archivo con el caso de prueba válido, se selecciona y se presiona “Open” como se muestra en la figura 9. En la matriz se dibujará la figura que contiene el archivo, como en el ejemplo el de la figura 10.

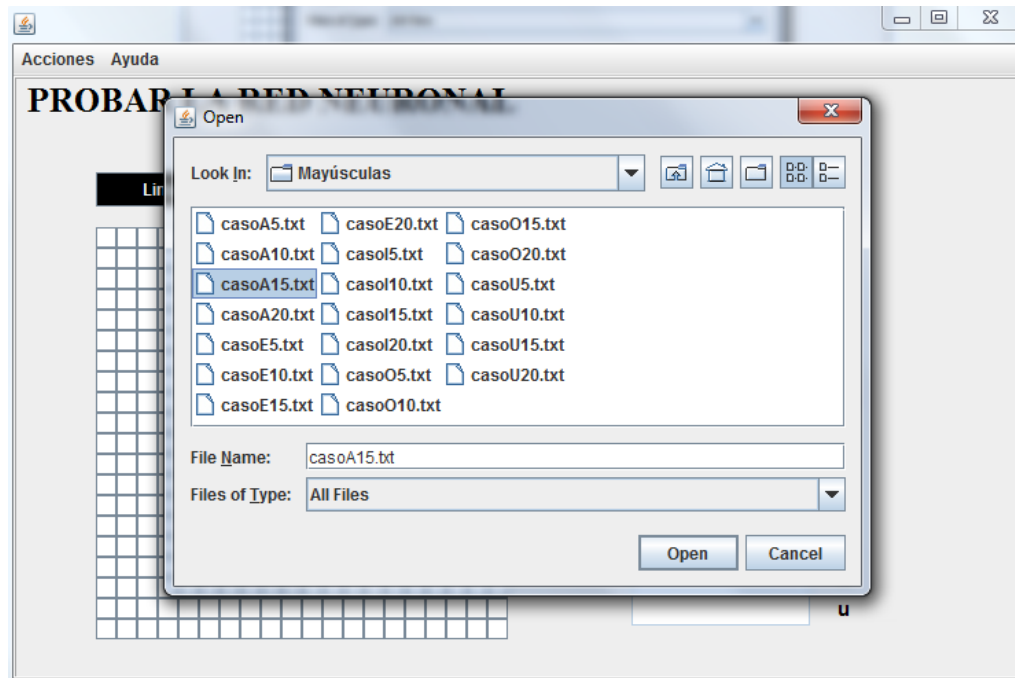


Figura 9. Ventana de navegación para cargar un caso de prueba

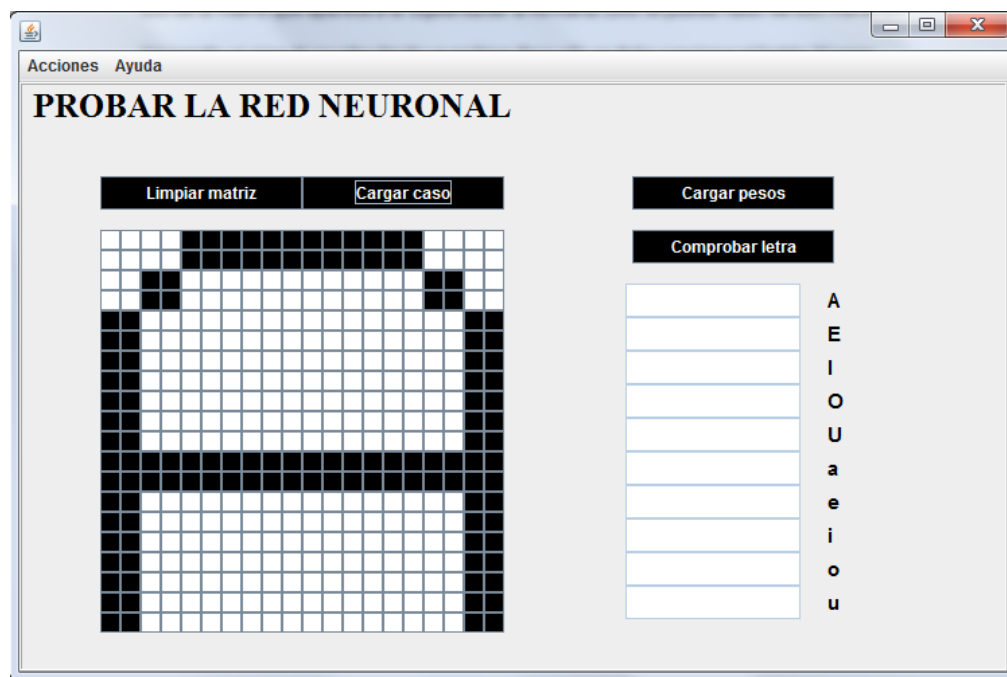


Figura 10. Ejemplo de imagen cargada desde un archivo

Dibujando el caso de prueba en la matriz: La otra manera de establecer la entrada de la red neuronal, es dibujar directamente en la matriz haciendo uso del cursor. Al presionar el botón

izquierdo del mouse sobre alguna de las celdas de la matriz, ésta cambia a color negro. Si se deja presionado el botón izquierdo del mouse y se mueve el cursor a lo largo de la matriz, se podrá ir dibujando el caso de entrada que se desee (Ver figura 11). Para borrar una celda que fue coloreada de negro, simplemente se debe dar “clic” de nuevo en esa celda y cambiará a color blanco. También se puede mantener el botón del mouse presionado y desplazarlo sobre las celdas de color negro para borrarlas.

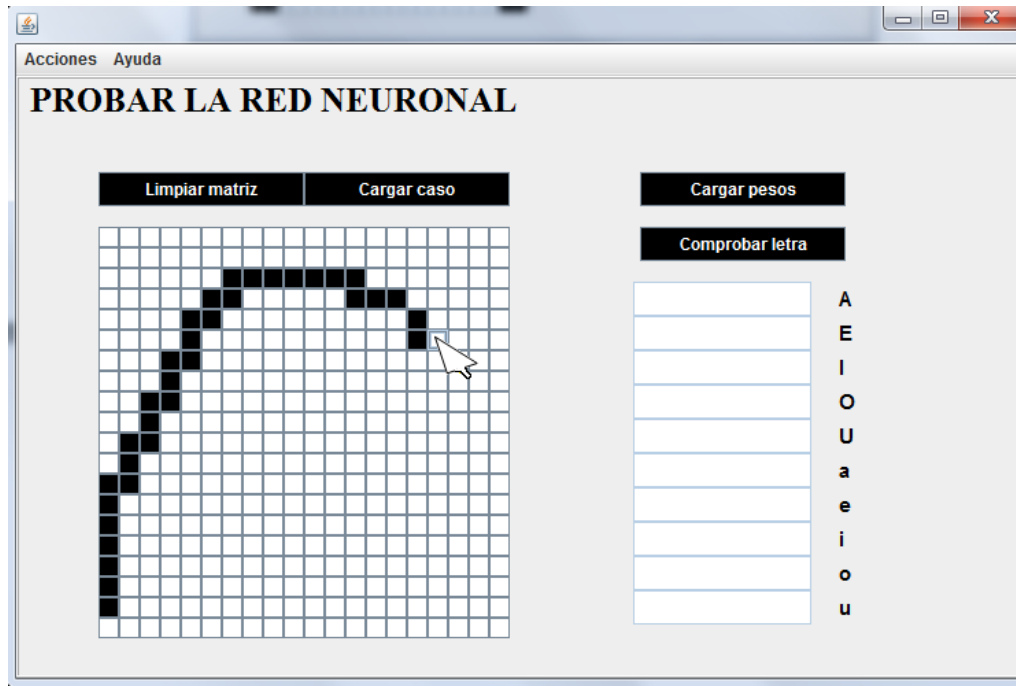


Figura 11. Dibujando con el cursor sobre la matriz

Si se desea borrar todas las celdas coloreadas de la matriz, se puede hacer uso del botón “Limpiar matriz” con el cual se deja la matriz en blanco.

Una vez que el caso de prueba está dibujado en la matriz, se puede comprobar la capacidad de la red para reconocer las vocales, presionando el botón “Comprobar letra”. Esto hace que se cargue en los campos correspondientes el valor que la red obtuvo para cada vocal. Además, aquella vocal que tenga un valor aceptable será recalcada en color rojo (figura 12). Sobre la interpretación de estos valores de salida, ver la sección 2.8.

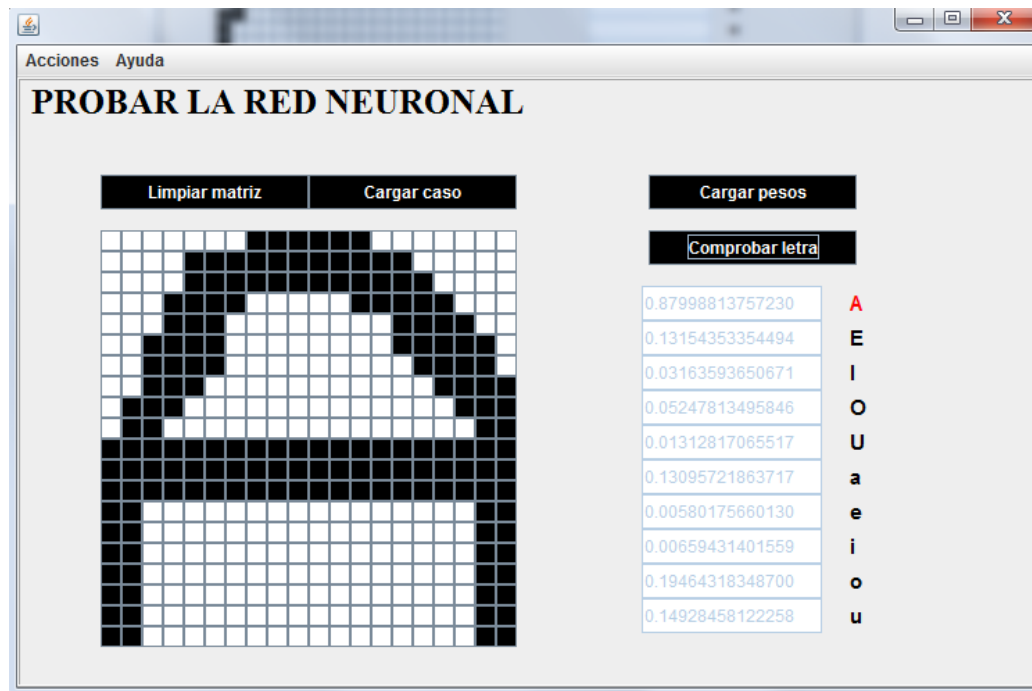


Figura 12. Ejemplo de resultados obtenidos al probar el caso de prueba

2.6. Ubicación de los archivos de entrada

La siguiente tabla muestra la ubicación dentro del archivo TP_JoseAndresMena.zip, de los diferentes archivos que se requieren para el funcionamiento de las algunas del programa:

| Tipo de archivo | Nombre de la carpeta | Observaciones |
|-----------------------------------|------------------------|---|
| Archivo de casos de entrenamiento | Casos de entrenamiento | En la carpeta se encuentra el archivo casos_entrenamiento.txt el cual 160 casos (16 para cada vocal mayúscula y minúscula). |
| Archivo de pesos | Archivos de pesos | La carpeta contiene el archivo pesos.txt, generado luego de entrenar la red y de haber convergido. |

Los diferentes archivos de entrada tienen el siguiente formato:

Archivo de casos de entrenamiento: La primera línea es un número entero que representa la cantidad de casos contiene el archivo. A partir de la segunda línea se pueden encontrar los casos de entrenamiento, cada uno de los cuales se representan con 22 líneas y tienen el siguiente formato:

- La primera línea contiene la palabra Caso (En realidad puede contener cualquier palabra, pues sirve como separador).
- Las siguientes 20 líneas contienen, cada una, 20 números separados por espacios. Cada una de estas líneas representa una fila de la matriz de entrada.
- La última línea contiene 10 valores numéricos separados por espacios, los cuales representan los valores esperados en la capa de la salida para ese caso.

Archivo de pesos: Contiene el valor de pesos para la capa oculta y la capa de salida. Cada valor de pesos tiene el siguiente formato:

- Una línea que contiene el texto "Pesos capa x", donde $x = 1$ para la capa oculta y $x = 2$ para la capa de salida.
- Las siguientes N líneas contienen M valores numéricos con decimales separados por espacio. Donde N es el número de unidades de la capa y M es el número de conexiones que llegan a cada una de las unidades de esa capa.

Archivo de casos de prueba: Un archivo de caso de prueba está formado por:

- Una primera línea de encabezado (Puede contener cualquier texto).
- 20 líneas que contienen 20 valores numéricos, cada una, separados por espacios. Cada línea representa una fila de la matriz de entrada.

2.8. Interpretación y manejo de resultados

En el módulo de entrenamiento de la red neuronal, se pueden obtener dos tipos de resultados:

Gráfico del comportamiento del error: En la parte derecha de la ventana, durante el entrenamiento de la red, se muestra un gráfico en el cual van apareciendo puntos y líneas (Figura 13).

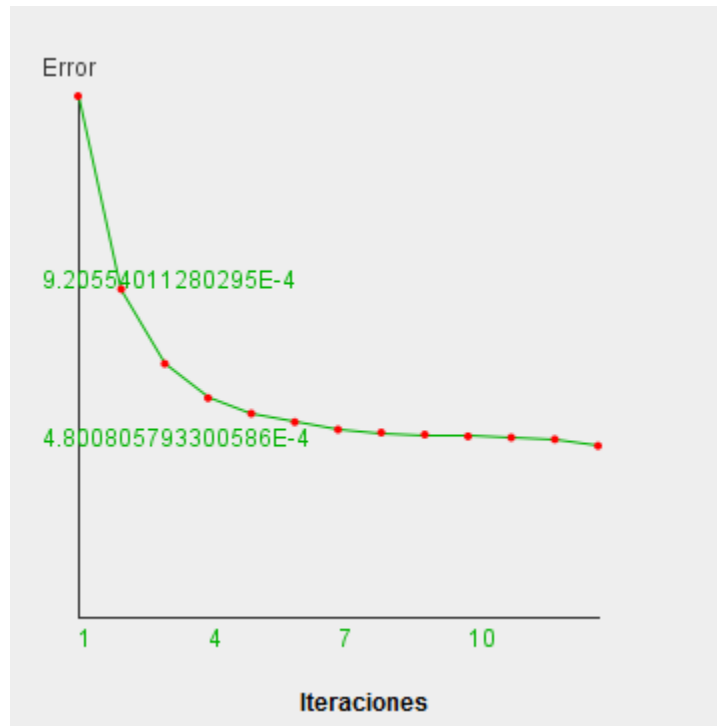


Figura 13. Gráfico del comportamiento del error durante un entrenamiento

En este gráfico, el eje X representa la cantidad de iteraciones que han ocurrido durante el entrenamiento, mientras que el eje Y muestra el valor del promedio de errores obtenidos durante esa iteración. Por ejemplo, en la figura 13 se observa que en la iteración 2, el promedio del error fue de 9.2055×10^{-4} . Es decir, cada punto rojo representa una iteración que ya ha ocurrido durante el entrenamiento, mientras que la línea verde que une un punto con otro, muestra el comportamiento del error entre iteraciones (Si la línea desciende significa que el error disminuyó).

Por efectos de espacio y claridad, no se muestran todas las etiquetas con el número de iteración. Lo mismo sucede con el valor del error (El número de iteración se muestra cada 3 iteraciones, mientras que el valor exacto del error se muestra cada 10 iteraciones).

Pesos ajustados luego del entrenamiento: Otro de los resultados que se obtiene después de entrenar la red, es un archivo con pesos. Este archivo contiene valores de los pesos de la red luego de converger y la manera de interpretar estos resultados es probando la red neuronal como se muestra en la sección 2.5.

Al probar la red neuronal haciendo uso de un determinado caso de prueba (ver sección 2.5), el resultado, como se muestra en la figura 14, se obtiene de la siguiente manera:

- En los 10 campos de la derecha, ubicados al lado de cada una de las vocales en mayúscula y en minúscula, aparecerá el valor que la red neuronal obtuvo para cada

vocal. Sin tomar en cuenta la tolerancia, se espera un valor de 0.9 para la vocal elegida y valores en 0.1 para las demás.

- La etiqueta de la vocal que obtenga un valor de 0.9 (o uno similar dependiendo de la tolerancia) se pondrá de color rojo. En la figura 12 se mostró un ejemplo en el cual la red neuronal reconoce de manera acertada la letra A mayúscula.

3. Análisis de los resultados obtenidos

3.1. Resultados obtenidos al entrenar la red

Para entrenar la red se utilizaron los siguientes parámetros:

Alfa = 0.8

Tolerancia = 0.1

Unidades de la capa oculta = 200

Archivo de casos = casos_entrenamiento.txt

Los casos contenidos en el archivo pueden verse en las imágenes Casos_entrenamiento_mayúsculas.png y Casos_entrenamiento_minúsculas.png que se encuentran en la carpeta Documentación.

Con estos parámetros, la red neuronal converge en pocas iteraciones, normalmente entre 25 y 30 (ver figura 14). El gráfico del comportamiento del error muestra cómo inicialmente el valor del error disminuye rápidamente y cuando se llega a cierto punto empieza a disminuir de manera más lenta. Esto se puede observar más claramente en la figura 14. Además, el error no desaparece por completo al converger e incluso en el gráfico la línea está lejos de llegar al eje de las abscisas. Esto último se debe a la tolerancia del 10% para el manejo del error. En otra prueba en la que se permitió solamente un 2% de tolerancia, el error al converger está más cercano al cero que en el caso anterior (Figura 15).

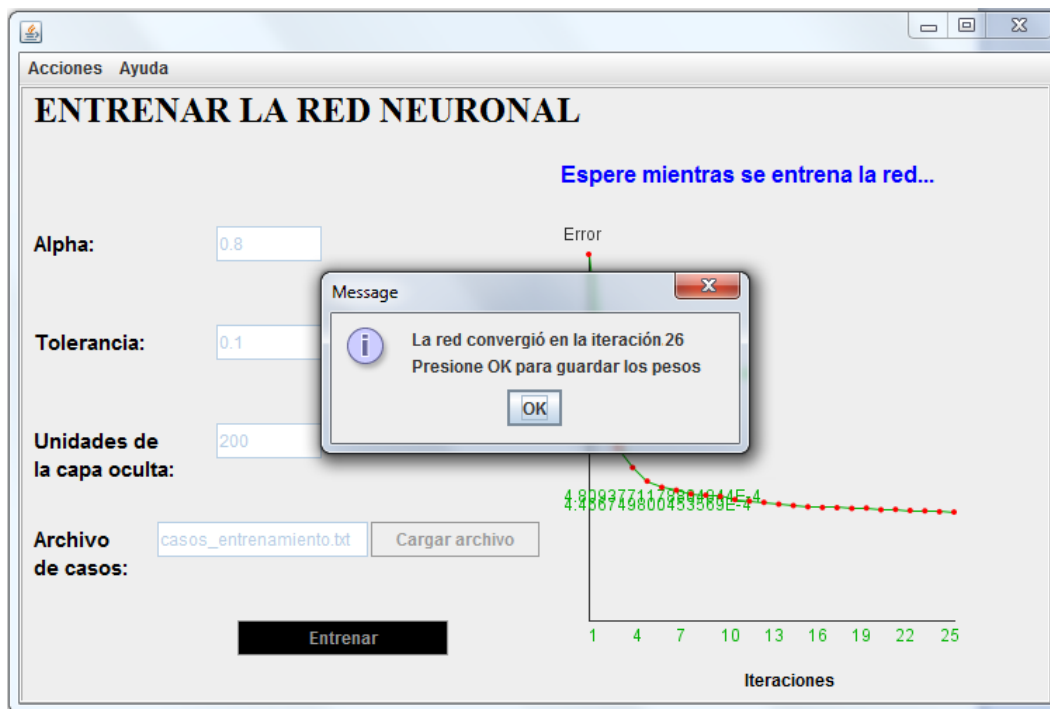


Figura 14. Convergencia de la red en 26 iteraciones

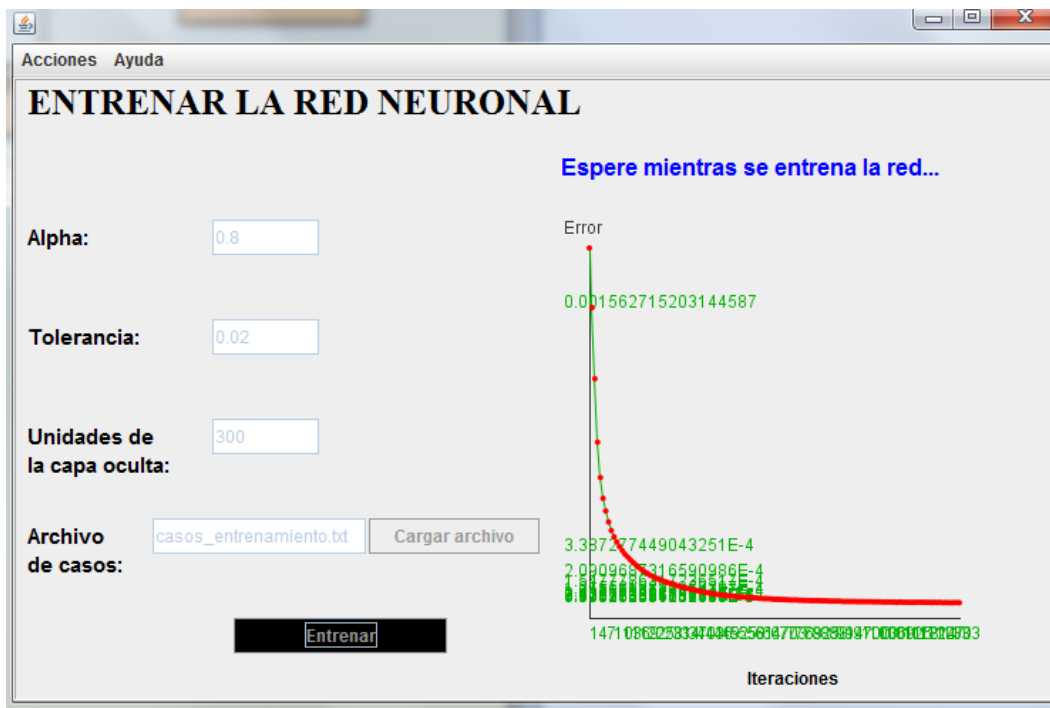
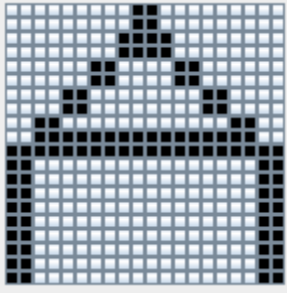
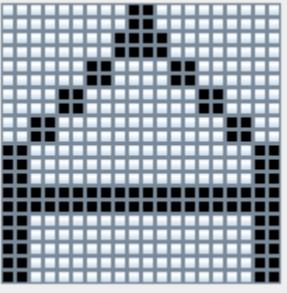
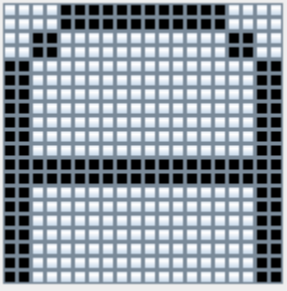


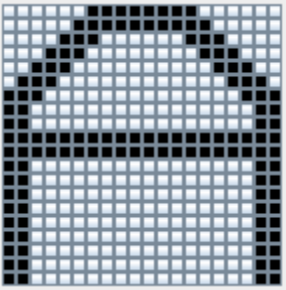
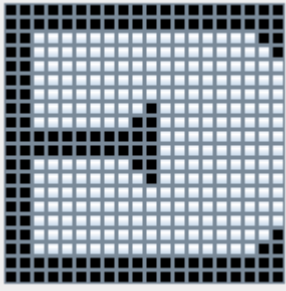
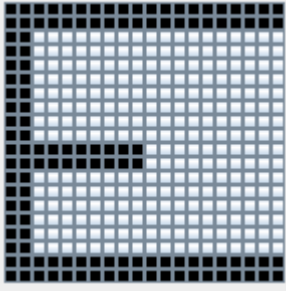
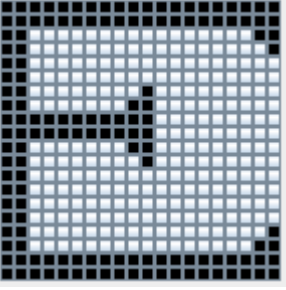
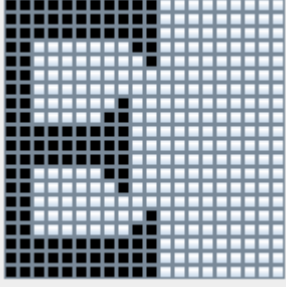
Figura 15. Comportamiento del error con una tolerancia del 2%

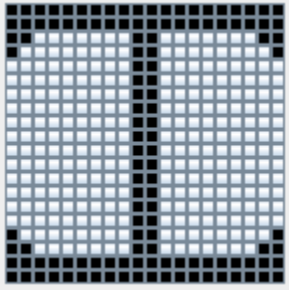
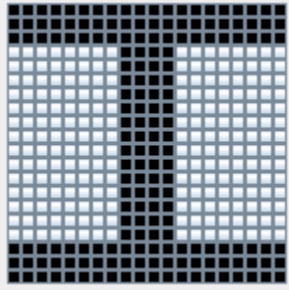
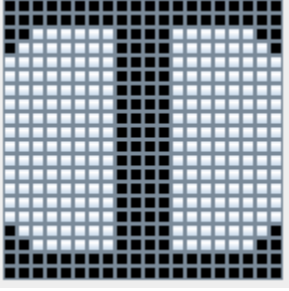
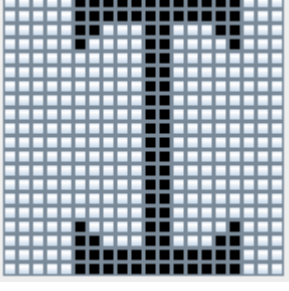
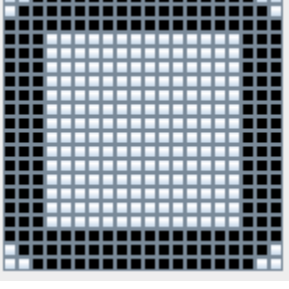
3.2. Resultados obtenidos al probar la red

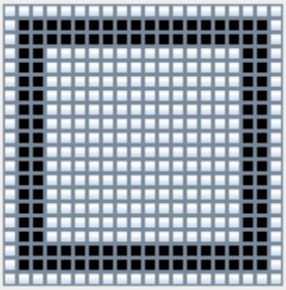
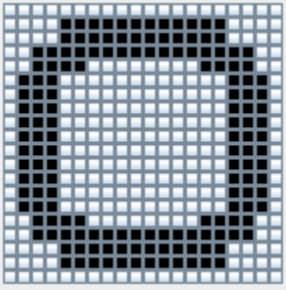
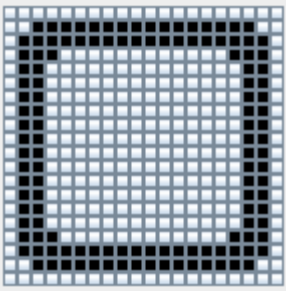
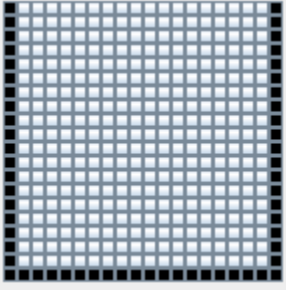
Se realizó la prueba de la red neuronal entrenada, haciendo uso de cada uno de los casos de prueba contenidos en la carpeta Casos de prueba. Para ello se utilizó una tolerancia del 10%, por lo que aquella vocal que obtuviera un valor comprendido entre 0.8 y 1, era considerada por el programa como la vocal elegida por la red.

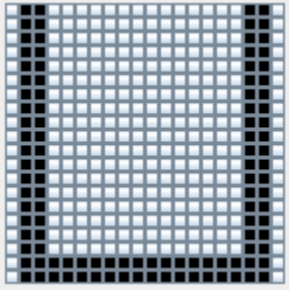
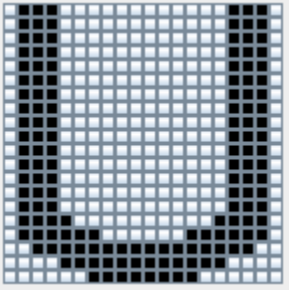
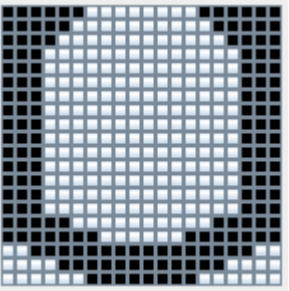
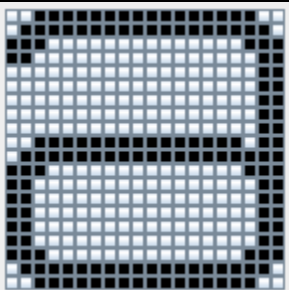
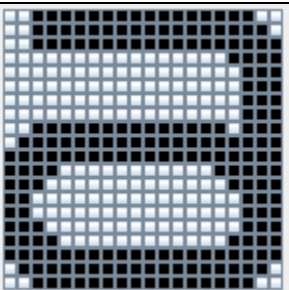
El resultado obtenido con cada uno de los casos y el respectivo análisis para los casos que fallaron se muestra en el cuadro a continuación. Cualquier referencia a los casos de entrenamiento puede apreciarse mejor consultando las imágenes Casos_entrenamiento_mayúsculas.png y Casos_entrenamiento_minúsculas.png ubicadas en la carpeta Documentación.

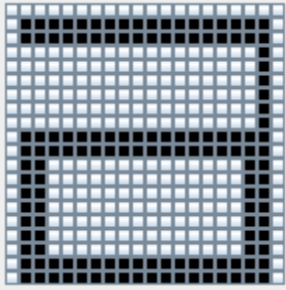
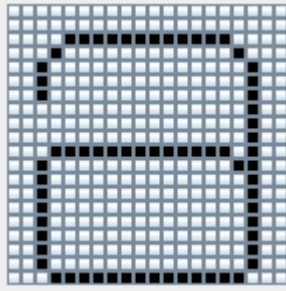
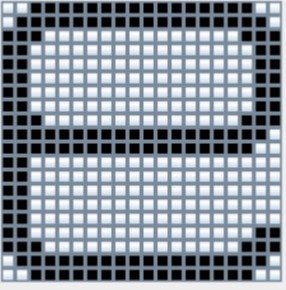
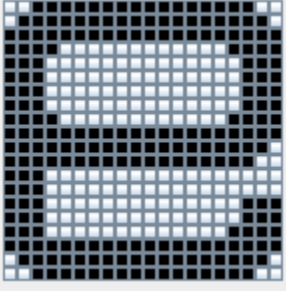
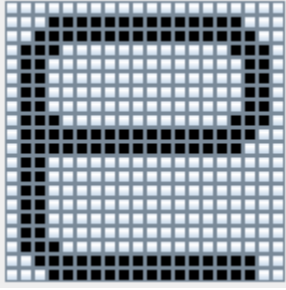
| Caso de prueba | Resultado esperado | Resultado Obtenido | Análisis en caso de fallo |
|---|--------------------|--------------------|---------------------------|
|  | A | A | |
|  | A | A | |
|  | A | A | |

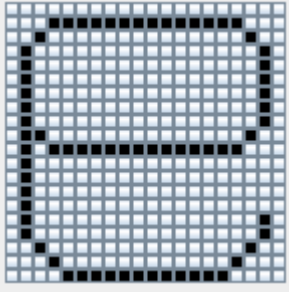
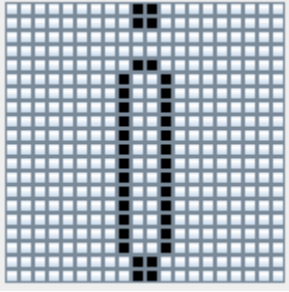
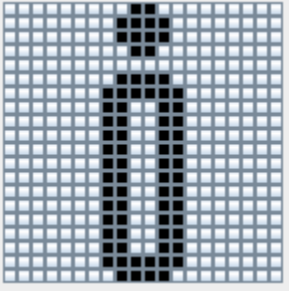
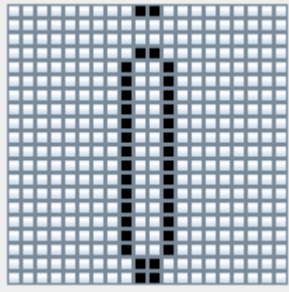
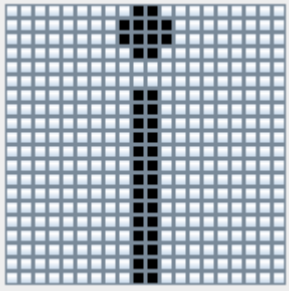
| | | | |
|---|---|---|--|
|  | A | A | |
|  | E | E | |
|  | E | E | |
|  | E | E | |
|  | E | E | |

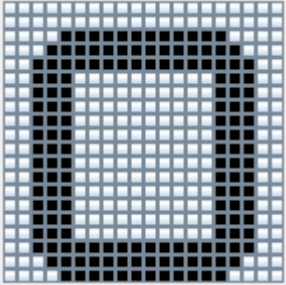
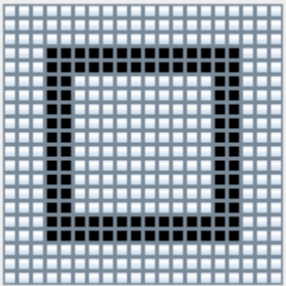
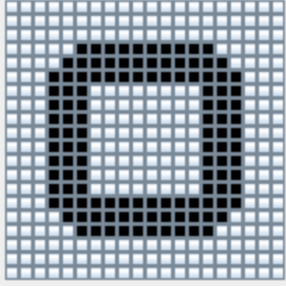
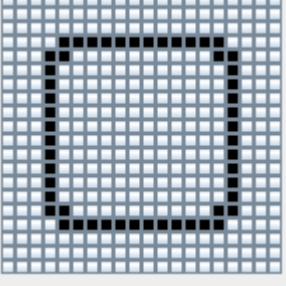
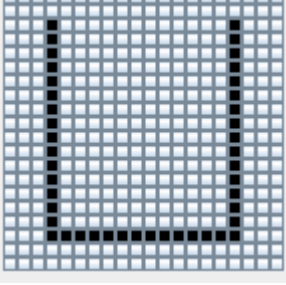
| | | | |
|---|---|---|--|
|  | I | I | |
|  | I | I | |
|  | I | I | |
|  | I | I | |
|  | O | O | |

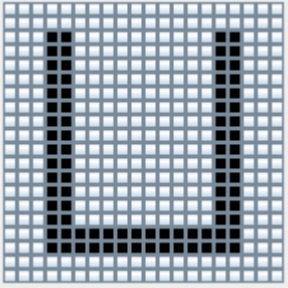
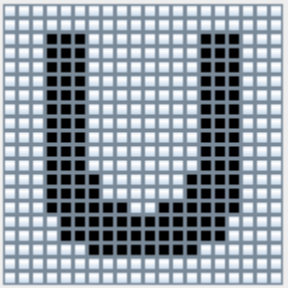
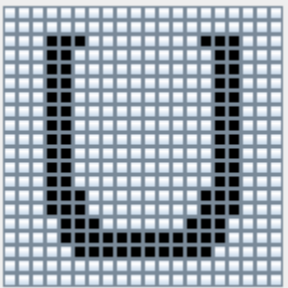
| | | | |
|---|---|--|---|
|  | O | O | |
|  | O | O | |
|  | O | O | |
|  | U | La red neuronal no logró reconocer la vocal. | <p>La red neuronal obtuvo un valor de 0.4299 para la letra U en este caso. Aunque no llegó a 0.9, dicho valor fue mayor que los demás valores obtenidos.</p> <p>Esto ocurrió debido a que en los casos de entrenamiento se utilizaron casos para la U en los cuales la línea era de por lo menos dos líneas de gruesa. Además, hubo un caso de entrenamiento para la letra O muy similar a este que pudo confundir la red (La O también obtuvo un valor alto)</p> |

| | | | |
|---|---|---|--|
|  | U | U | |
|  | U | U | |
|  | U | U | |
|  | a | a | |
|  | a | a | |

| | | | |
|---|---|--|---|
|  | a | a | |
|  | a | La red neuronal no logró reconocer la vocal. | Todos los casos de entrenamiento utilizados para la letra a utilizaban líneas de al menos dos celdas de grosor, además ocupaban toda la matriz. En este caso la letra a tiene una línea delgada y es pequeña en comparación con los otros casos. Aun así, el valor 0.5230 obtenido para letra a en este caso de prueba fue el valor más alto dado por la red. |
|  | e | e | |
|  | e | La red neuronal no logró reconocer la vocal. | Este caso no fue reconocido por la red completamente, pero se obtuvo un valor de 0.7957, el cual está muy cerca del valor permitido. Esto pudo deberse al hecho de que el grosor es bastante en comparación a los otros casos. Sin embargo, aumentando ligeramente la tolerancia la red reconocería fácilmente este caso. |
|  | e | e | |

| | | | |
|---|---|--|---|
|  | e | e | |
|  | i | La red neuronal no logró reconocer la vocal. | Es de esperar que este caso no sea reconocido, dado que ninguno de los casos de entrenamiento tenía este estilo (una letra i sin relleno y ocupando toda la matriz). Sin embargo, de nuevo la red neuronal logra darle a la letra i el valor más alto en este caso: 0.4938 |
|  | i | i | |
|  | i | La red neuronal no logró reconocer la vocal. | Sucede lo mismo que el anterior caso fallido. En el entrenamiento no se le mostró a la red neuronal un patrón similar a este para la letra i. |
|  | i | La red neuronal no logró reconocer la vocal. | En este caso la red se confunde completamente y le da un valor más alto a la letra l mayúscula. Tiene sentido considerando el hecho de que es muy similar a los casos de entrenamiento presentados para la l mayúscula, por lo que en general la letra i minúscula estaba muy propensa a no ser reconocida. |

| | | | |
|---|---|--|---|
|  | o | o | |
|  | o | o | |
|  | o | o | |
|  | o | o | |
|  | u | La red neuronal no logró reconocer la vocal. | Aunque durante el entrenamiento a la red neuronal se le presentaron varios casos de la letra u con línea delgada, el hecho de que la red fallara puede deberse a que en este caso la letra u es más estrecha que los otros. Sin embargo, se obtuvo un valor 0.7049 el cual sí sería válido si se aumenta ligeramente la tolerancia. |

| | | | |
|--|---|---|--|
|  | u | U | |
|  | u | u | |
|  | u | u | |

En resumen, se puede observar que de los 32 casos de prueba presentados, únicamente 7 casos no fueron reconocidos por la red (ver gráfico 1 y gráfico 2), es decir, la red neuronal falló en aproximadamente un 20% de los casos. El porcentaje es bastante bajo, considerando el hecho de que los casos de entrenamiento presentados a la red fueron muy pocos y muy variados. Además, como se observó en el cuadro de resultados, la mayoría de casos fallidos serían exitosos si se aumentaba ligeramente la tolerancia del error.

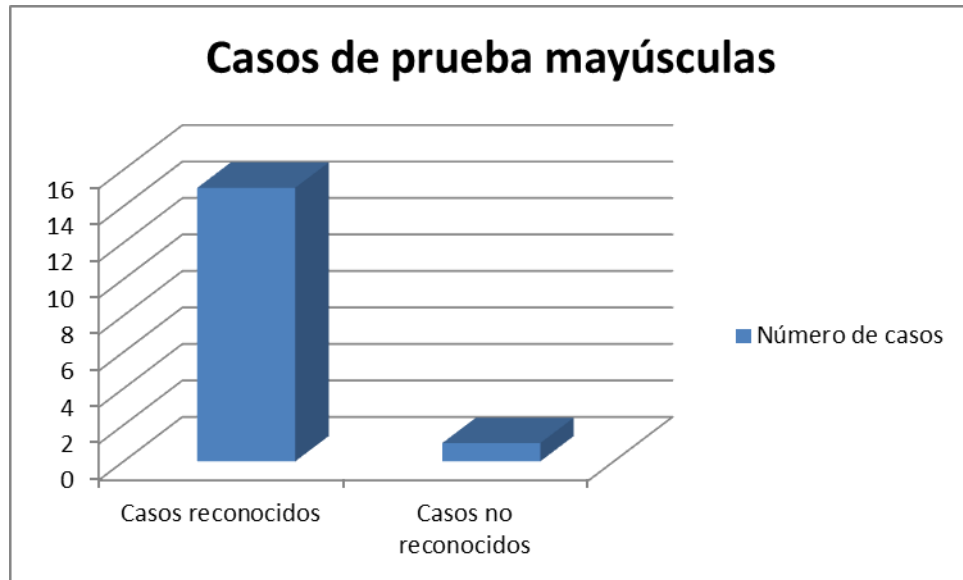


Gráfico 1. Resultados obtenidos con las vocales mayúsculas

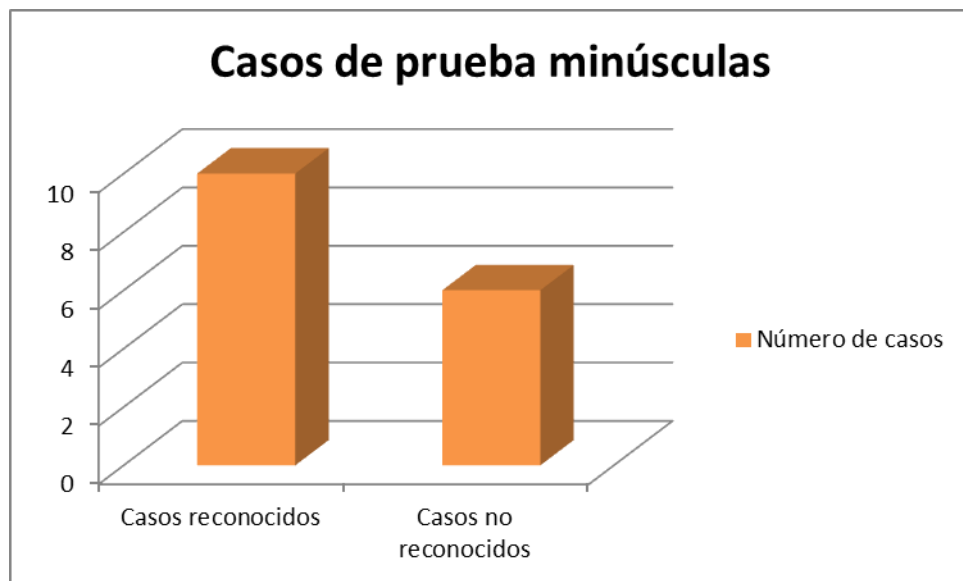


Gráfico 2. Resultados obtenidos con las vocales minúsculas

4. Otras consideraciones

Para la utilización y prueba de esta aplicación es recomendable tomar en cuenta lo siguiente:

- El gráfico que muestra el comportamiento del error durante el entrenamiento hace uso de hilos de ejecución, los cuales suelen presentar problemas al ser ejecutados en procesadores AMD. Por lo tanto, es altamente recomendable realizar las pruebas en procesadores Intel.
- Algunos valores para los parámetros en el entrenamiento de la red pueden llegar a hacer que el proceso de entrenamiento tarde mucho tiempo o que, incluso, la red neuronal sea incapaz de converger. Por lo tanto, es responsabilidad del usuario utilizar los parámetros adecuados y sugeridos en este documento para probar el entrenamiento de la red neuronal.
- Cuando se ejecuta la aplicación, internamente se crea una red neuronal que contiene por defecto los siguientes valores:
 - Alfa = 0.8
 - Tolerancia = 0.1
 - Unidades de la capa oculta = 200

Es importante considerar esto si se quiere ejecutar la aplicación únicamente para probar la red neuronal cargándole pesos generados previamente. Por ejemplo, si se cuenta con un archivo de pesos generados para una red neuronal con 100 unidades de la capa oculta y se ejecuta la aplicación e intentan cargar estos pesos, no será posible debido a que la red neuronal actual contendrá 200 unidades en su capa oculta. Cambiar la cantidad de unidades de la capa oculta únicamente se logra en la pantalla de entrenamiento.