

## 7. SOFTWARE DE APLICACIÓN

### 7.1. INTRODUCCIÓN

Tras haber comprobado que las redes neuronales se pueden aplicar al problema de optimización de fármacos, se ha diseñado e implementado una herramienta Software capaz de automatizar el proceso de optimización. Esta herramienta optimizará los datos directamente obtenidos mediante DOE en el laboratorio. En esta optimización, el usuario elegirá la respuesta que desea optimizar y la forma de optimización: minimizar, maximizar o aproximar a un valor dado.

La red neuronal implementada en la herramienta es un perceptrón multicapa de tres capas y el algoritmo de entrenamiento es el *Backpropagation con momento*. El usuario puede elegir ciertos parámetros como el número de neuronas de cada capa y la tasa de aprendizaje y el momento en el aprendizaje. Para validar la red neuronal se puede elegir entre dos métodos: empleando muestras de verificación o el método de remuestreo *Bootstrap*.

Al ejecutar la aplicación aparece la ventana principal que se muestra en la Figura 7.1. Cada parte de la aplicación se detallará en los siguientes subapartados del capítulo.

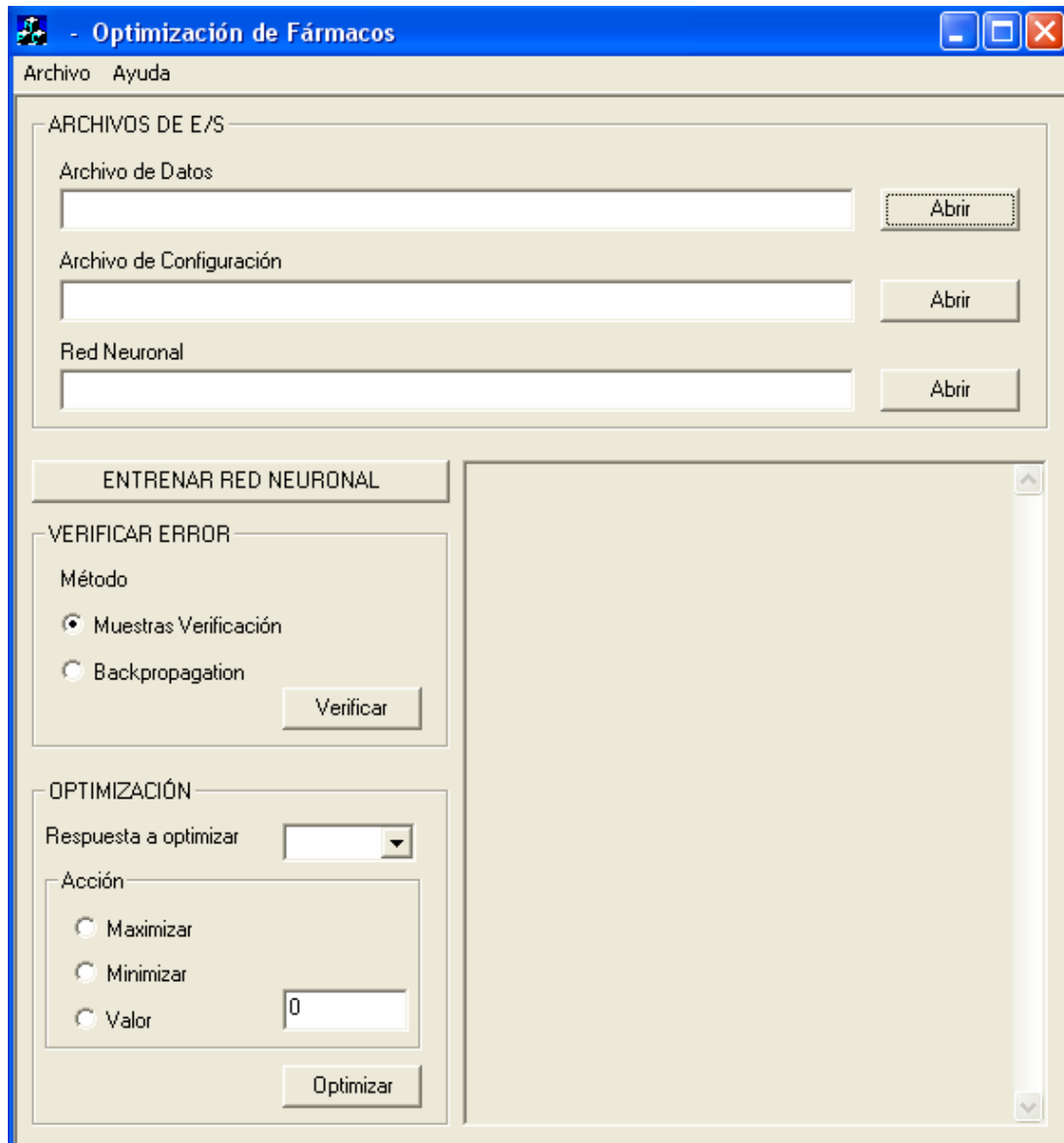


Figura 7.1: Ventana principal de la aplicación.

## 7.2. CREACIÓN DE ARCHIVOS

### 7.2.1. Archivo de configuración

Para crear un archivo de configuración que será utilizado posteriormente para el entrenamiento, validación de la red neuronal y optimización del fármaco se hace click en Archivo/Configuración y se abrirá el siguiente cuadro de diálogo:

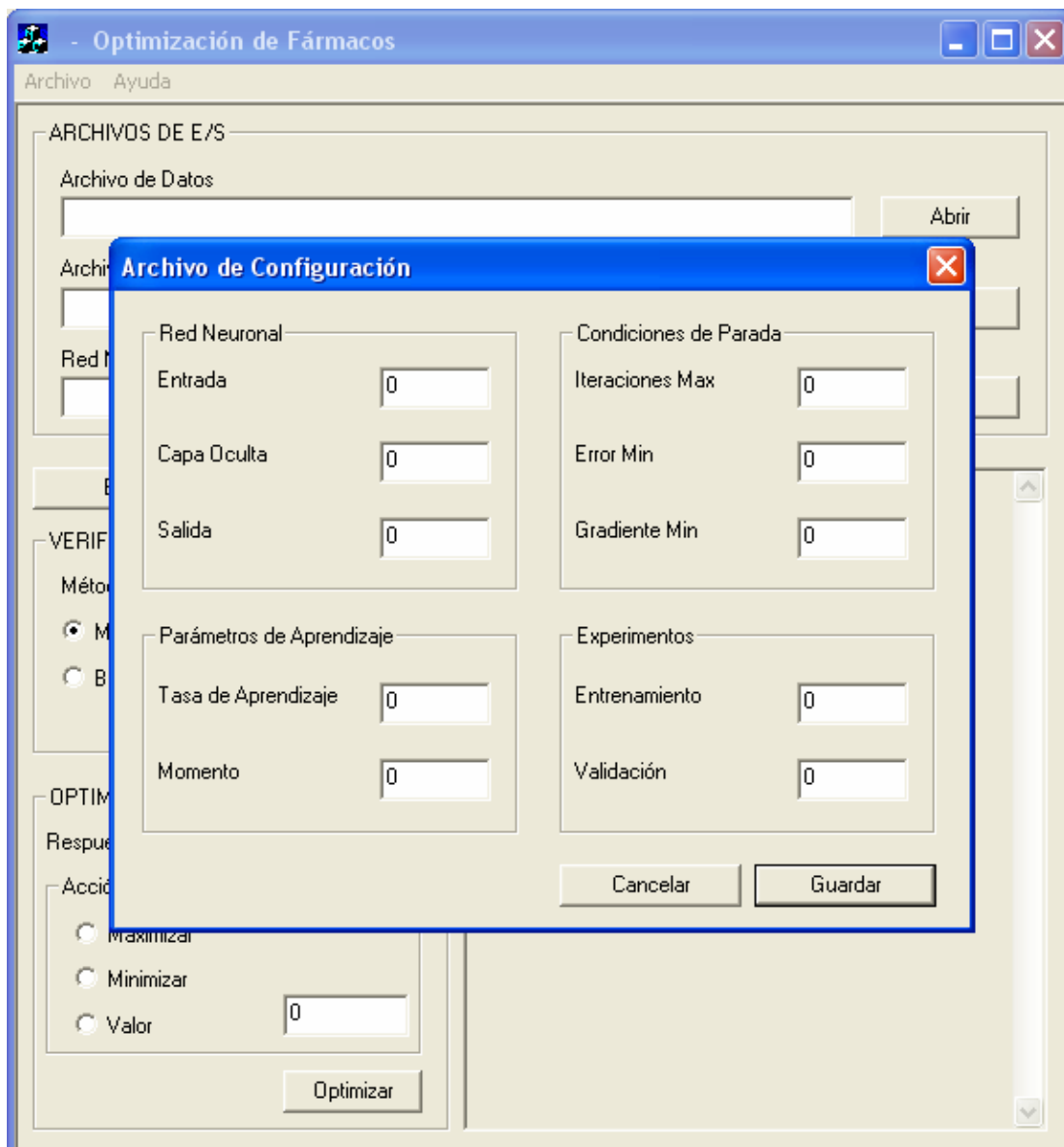


Figura 7.2: Ventana de archivo de configuración.

En este cuadro de diálogo se introducen los parámetros para configurar la arquitectura de la red neuronal, su aprendizaje, las condiciones de parada del entrenamiento y el número de experimentos.

### Red Neuronal:

Aquí se definirá la arquitectura de la red neuronal. La red neuronal implementada en el programa es un perceptrón multicapa. Así, se tendrá que definir el número de neuronas que compone cada capa con las siguientes consideraciones:

- El número de nodos de entrada será el mismo número de variables de entrada de los experimentos.
- El número de nodos de salida que será el mismo número de respuestas de los experimentos.
- N° de nodos de la capa oculta será un número mayor que cero.

### Parámetros de Aprendizaje:

El algoritmo para entrenar la red neuronal es el *Backpropagation* con momento. Los parámetros que se deben de definir son la tasa de aprendizaje y el momento:

- Tasa de aprendizaje: su valor estará comprendido entre 0 y 1. Cuanto menor sea su valor mayor será la precisión obtenida por la red, sin embargo, el entrenamiento será más largo. Si su valor es elevado, el aprendizaje será más rápido pero se perderá precisión.
- Momento: su valor estará comprendido entre 0 y 1. Cuanto mayor sea su valor mayor será la velocidad de convergencia de la red neuronal.

### Condiciones de Parada:

El entrenamiento de la red terminará cuando se alcance alguno de estos tres parámetros, esto es, iteraciones máximas, error mínimo y gradiente mínimo. Dependerá del investigador la preferencia por el número de iteraciones o por el error mínimo. Si se desea que el entrenamiento se detenga cuando se haya completado el número de iteraciones indicado, el error mínimo será un número cercano a cero, imposible de alcanzar por la red. Si, al contrario, se desea que el entrenamiento se detenga cuando el error cometido por la red neuronal en los patrones de entrenamiento sea el indicado, el número de iteraciones será un número elevado.

Si el algoritmo de aprendizaje se detiene debido a que ha alcanzado el gradiente mínimo establecido, la función de optimización del algoritmo ha alcanzado una “*planicie*” donde la pendiente de la función es muy pequeña. Si se llega a esta situación, será muy difícil que la red neuronal pueda terminar el aprendizaje satisfactoriamente por lo que se detiene el entrenamiento.

Estos tres parámetros tienen los siguientes rangos de valores:

- Número de iteraciones máxima será un número mayor que 0.
- Error mínimo tendrá un valor comprendido entre 0 y 1.
- Gradiente mínimo que se permitirá será un número mayor que 0.

### Experimentos:

Aquí se indicará cuántos experimentos se dedicará al entrenamiento de la red neuronal y cuántos se dedicará a la validación:

- Entrenamiento: n° de experimentos que se utilizarán para el entrenamiento.
- Validación: n° de experimentos que se utilizaran para la validación.

Con  $N^{\circ} \text{ Experimentos} = \text{Entrenamiento} + \text{Validación}$

Si se desea validar la red con muestras de validación se dejará el 70%, 80% de las muestras para el entrenamiento y el resto para la validación. Si se validará la red neuronal mediante el método de *bootstrap*, entonces el parámetro Validación será cero y se entrenará la red con todos los datos.

Cuando se haya finalizado la configuración de la red neuronal, se guardará los datos en un fichero de texto.

### **7.2.2. Archivo de datos**

Los datos experimentales que se utilizarán para el entrenamiento y validación de la red neuronal estarán en un archivo de texto. Para crear dicho archivo se seleccionará Archivo/Datos del menú de la ventana principal del programa y se abrirá el siguiente cuadro de diálogo:

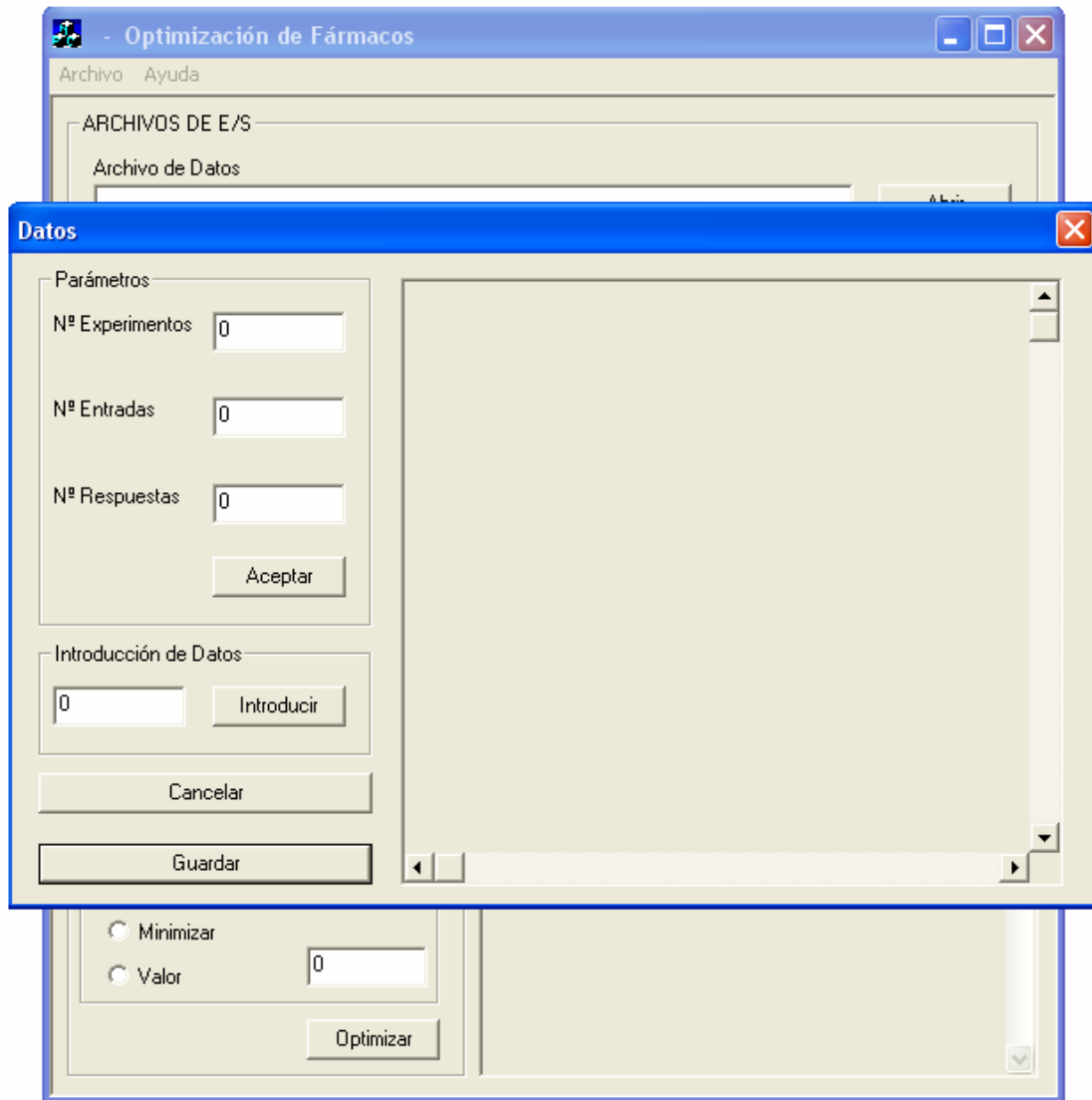


Figura 7.3: Ventana de archivo de datos.

Antes de introducir datos habrá que indicar el número de experimentos en total del que se dispongan, el número de variables de entrada y el número de respuestas que están formadas los experimentos. Al terminar se hará click en aceptar y aparecerá una rejilla donde se introducirán los datos de los experimentos.

La rejilla de datos que aparecerá tras aceptar los parámetros será de tamaño  $N^{\circ} \text{ Experimentos} \times (N^{\circ} \text{ Entradas} + N^{\circ} \text{ Salidas})$ . En cada fila de la rejilla se introducirá

los datos de un experimento. Las primeras columnas están dedicadas a las variables de entrada y las respuestas irán a continuación en las siguientes columnas.

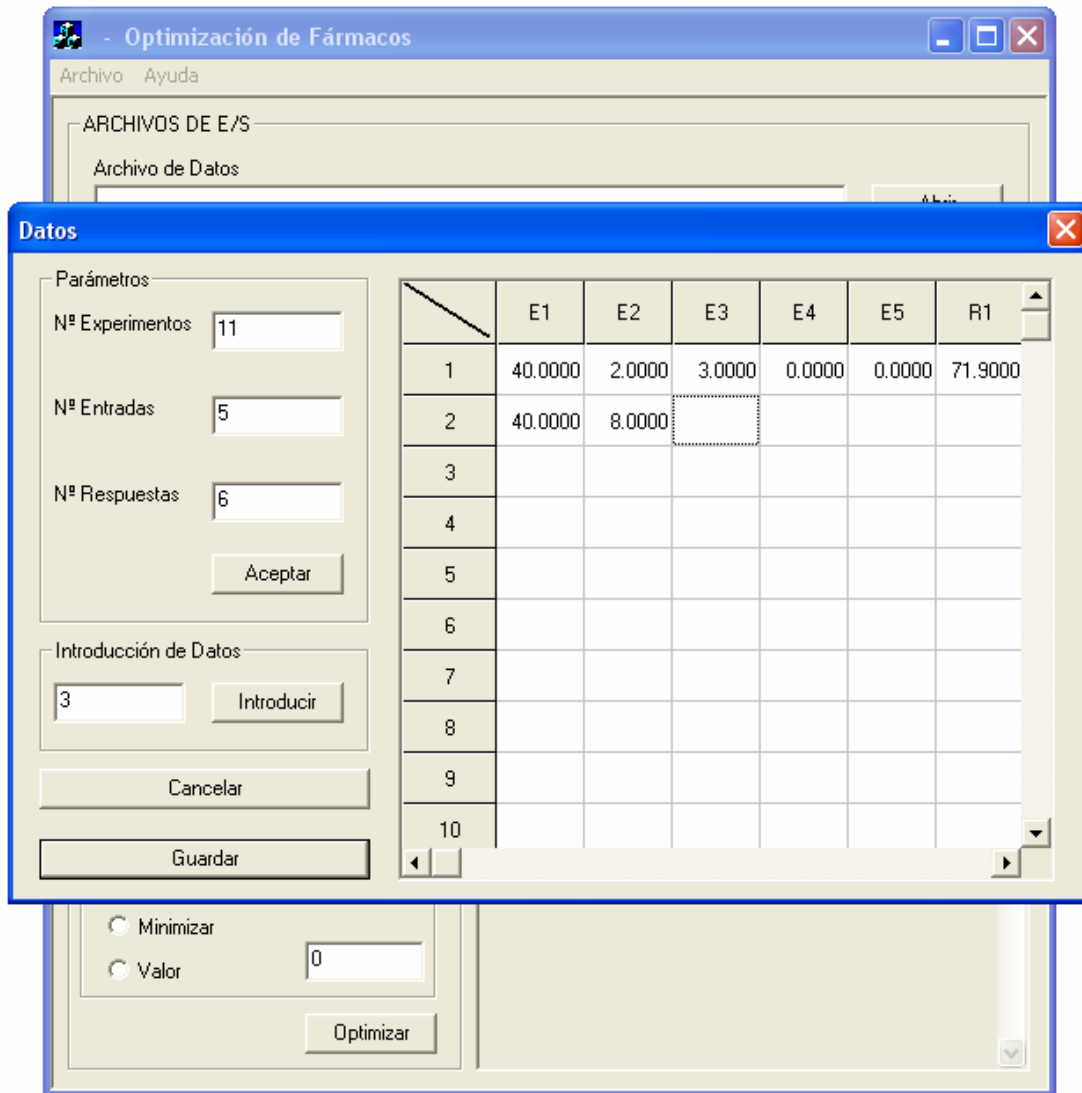


Figura 7.4: Insertando datos en la ventana de archivos de datos.

Para introducir los datos experimentales se pulsará el botón *Introducir* y el dato escrito en la caja de texto aparecerá en la celda seleccionada de la rejilla y automáticamente se seleccionará la siguiente celda para introducir el siguiente dato. Se puede seleccionar otra celda que no sea la de defecto haciendo click en dicha celda.



Una vez terminado de introducir todos los datos, se guardará en un archivo de texto.

### **7.3. ENTRENAR LA RED NEURONAL**

La arquitectura de la red neuronal y los parámetros para el algoritmo de entrenamiento se tomará del archivo de configuración indicado. Los datos que se tomarán para el entrenamiento son los *Nº Entrenamiento* primeros experimentos del archivo de datos. El resto de datos se utilizarán para la validación en caso que se valide la red mediante patrones de validación.

El final del entrenamiento se indicará en la caja de texto junto con el número de iteraciones que ha efectuado el algoritmo de aprendizaje y el resultado, esto es, los pesos de la red neuronal, se guardarán en el archivo de texto indicado en el campo *Red Neuronal*.

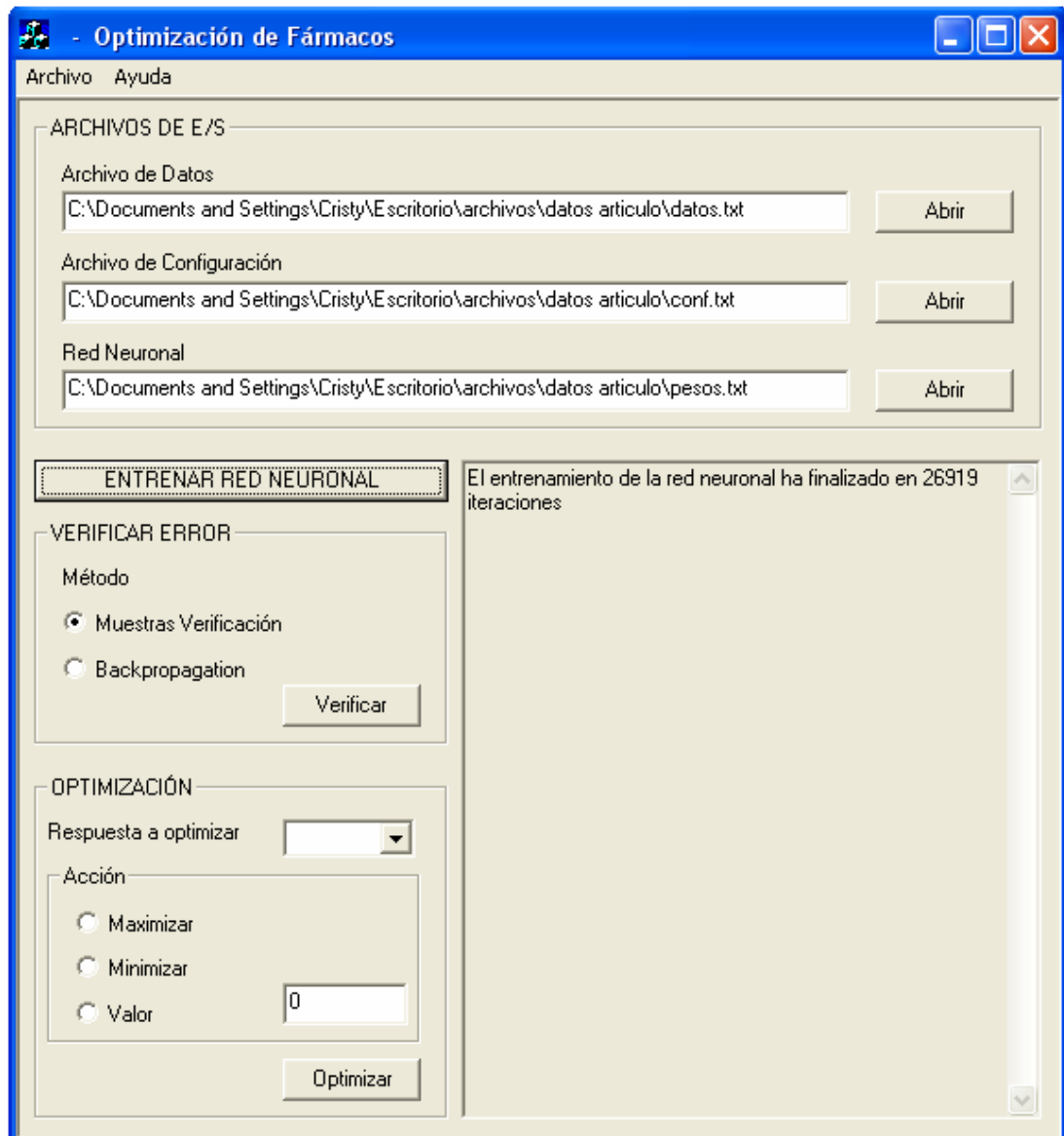


Figura 7.5: Entrenando la red neuronal.

## 7.4. VALIDACIÓN DE LA RED NEURONAL

La arquitectura de la red neuronal y los parámetros para la validación de la red neuronal se tomará del archivo de configuración indicado. Los pesos de la red se tomarán del archivo de red neuronal. Este archivo debería de haber sido creado en algún entrenamiento anterior.

Los datos para la verificación de la red neuronal procederán del archivo de datos. Los datos utilizados dependerán del método de verificación que se elija.

Se elegirá primero el método para la verificación y a continuación se pulsará el botón *Verificar*:

- *Muestras de verificación*: Con este método se verifica el comportamiento de la red neuronal con patrones de verificación. El número de patrones utilizados será el indicado en el archivo de configuración en el parámetro *Verificación*. Estos patrones serán los últimos experimentos del archivo de datos. En la caja de texto aparecerán el resultado de la validación: el error RMS de las muestras de entrenamiento y el error RMS de las muestras de validación.

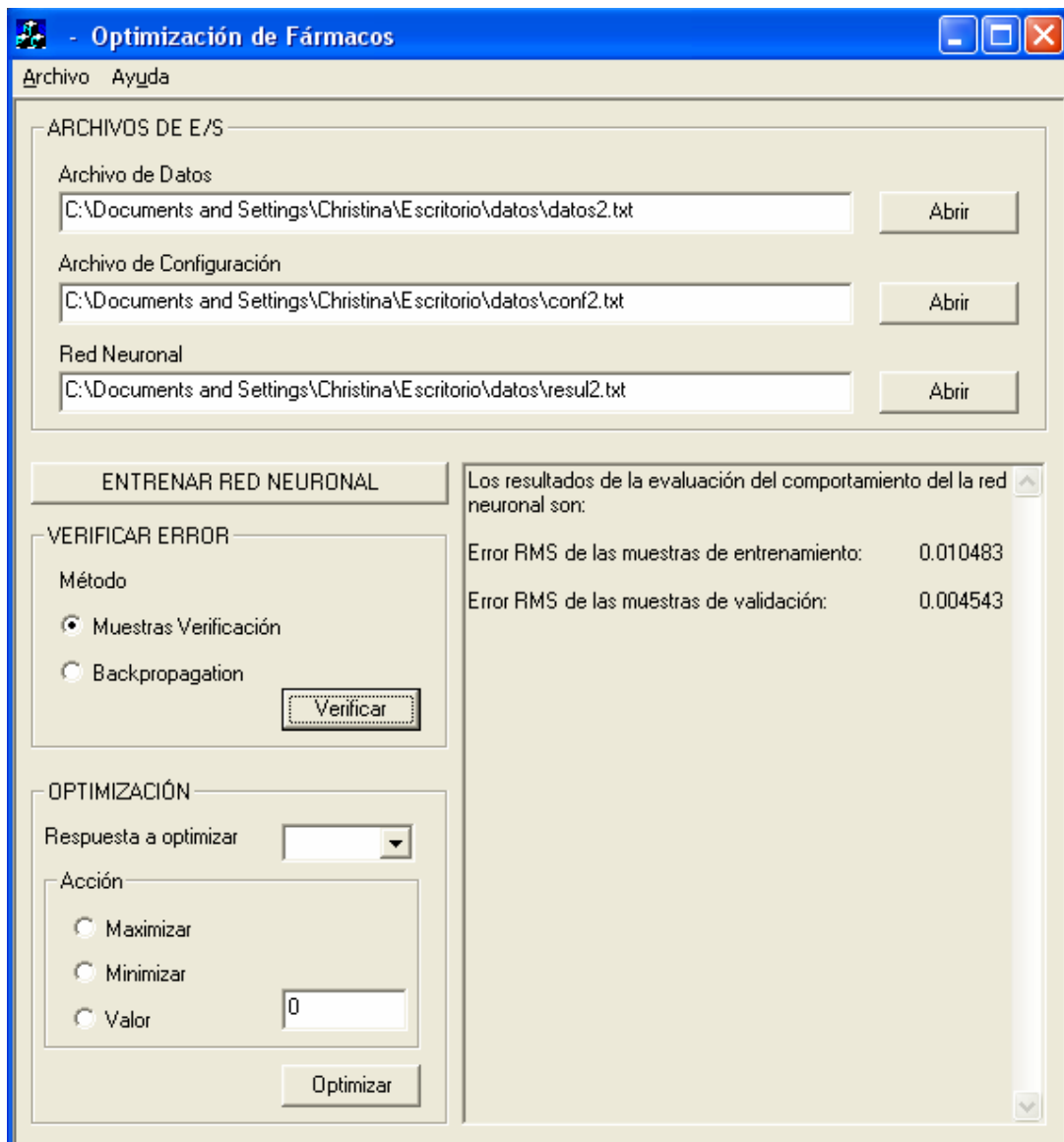


Figura 7.6: Validando la red neuronal con muestras de verificación.

- *Bootstrap*: En este caso, para realizar la validación correctamente, el parámetro *Verificación* debería de ser 0 y *Entrenamiento* debería ser igual al número de experimentos tanto en el entrenamiento como en la validación mediante el método de *Bootstrap*. Se verificará el error de la red neuronal con todos los datos. Al finalizar la verificación se mostrarán los siguientes datos en la caja de texto: el error RMS, el error estándar SE y la desviación o BIAS.

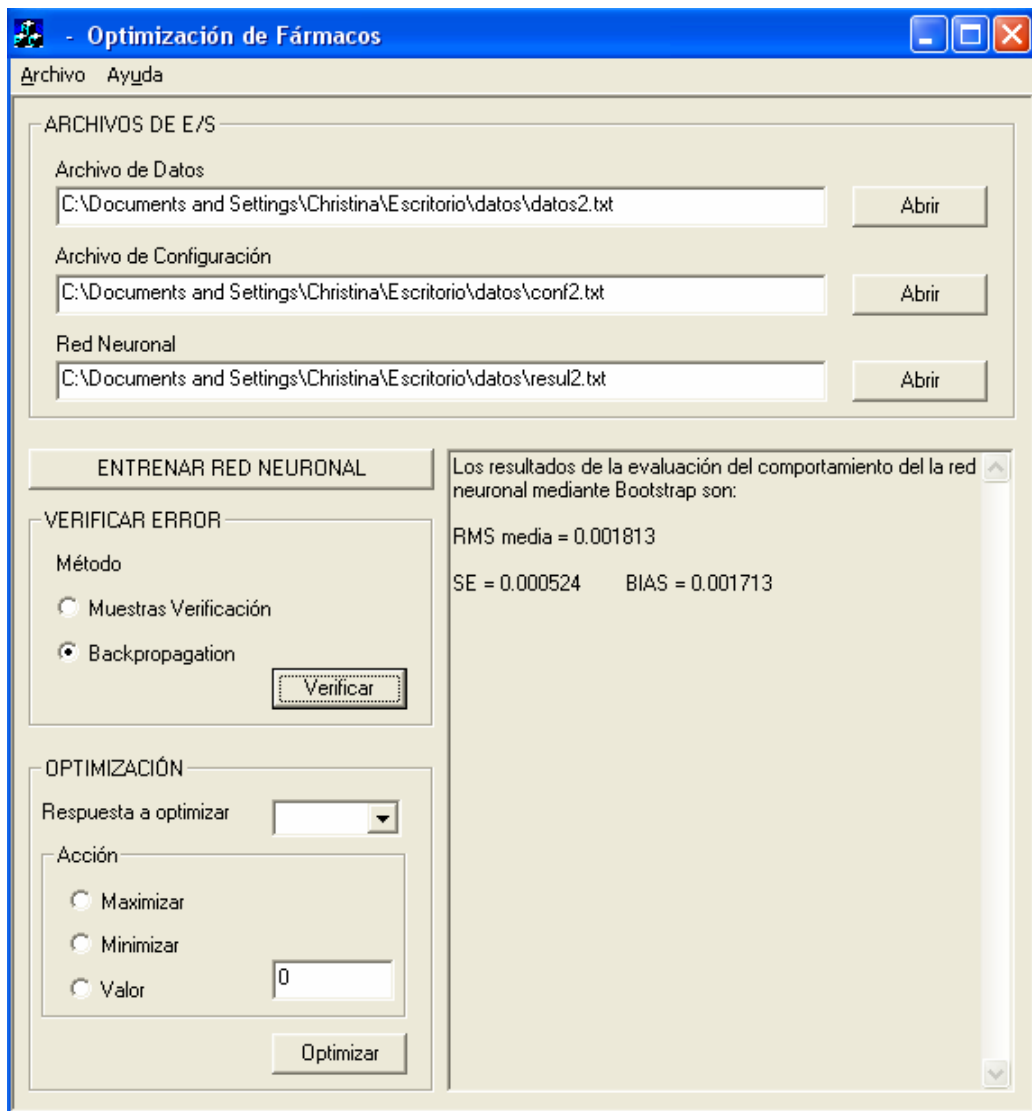


Figura 7.7: Validando la red neuronal mediante Bootstrap.

## 7.5. OPTIMIZACIÓN

Finalmente, tras haber entrenado y validado la red neuronal, se pasa a la parte de optimización del fármaco. Se elige la respuesta que se quiere optimizar y la herramienta hallará las variables de entrada óptimas para esa respuesta. Estas entradas serán las que maximiza la respuesta, la que la minimice o la que consiga que se aproxime más a un valor determinado.

Para ello, primero se elige el número de la respuesta que se desea optimizar en la lista desplegable o introduciendo el dato directamente. A continuación se elige cómo se quiere optimizar dicha respuesta: maximizándola minimizándola o aproximarla a un valor dado seleccionando una de las opciones que se ofrece. En caso que se seleccione un valor para aproximar, se introducirá dicho valor en la caja de texto correspondiente. Una vez finalizado se pulsará el botón de *Optimizar* y los cinco mejores resultados se mostrarán en la caja de texto. Cada uno de los resultados consistirá en los valores de todas las variables de entrada que haya optimizado la respuesta seleccionada.

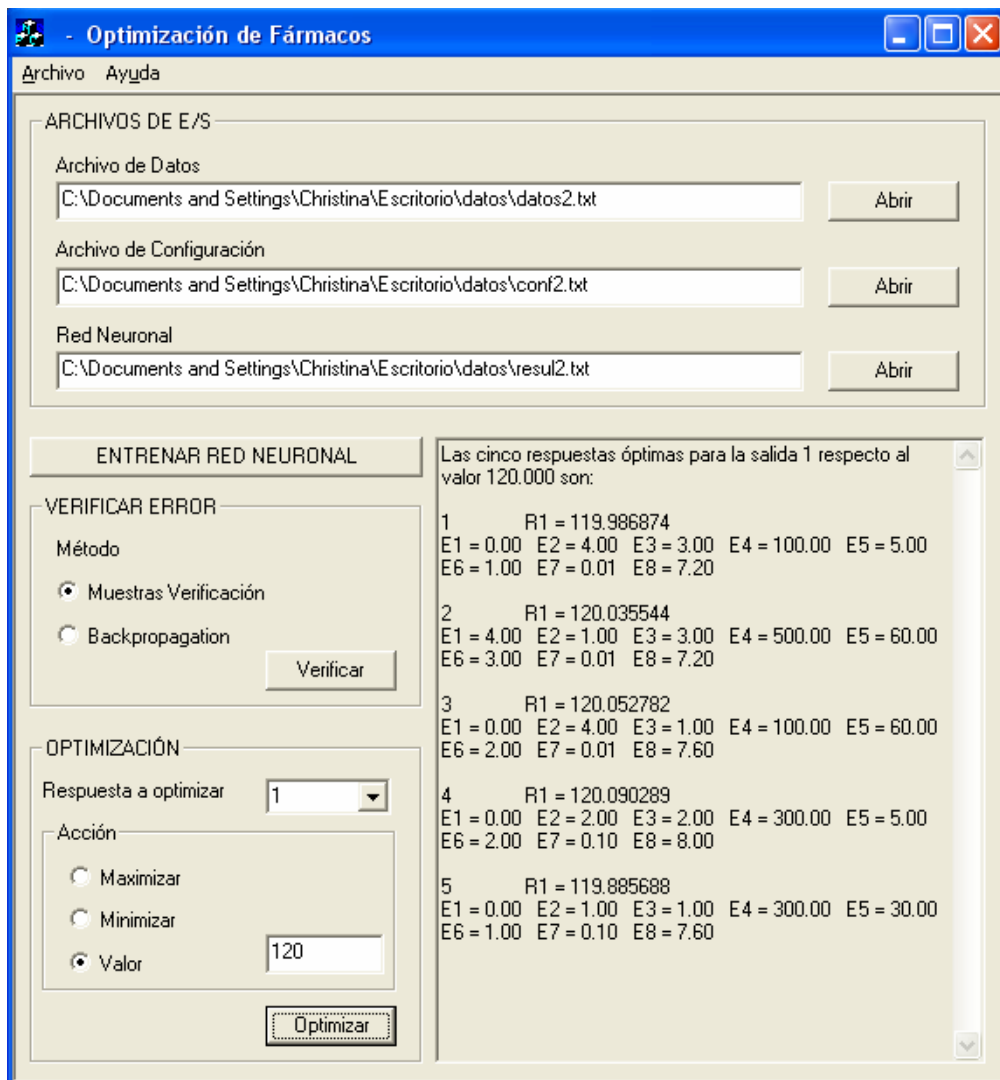


Figura 7.8: Optimización de una salida.