



GUÍA DEL USUARIO

Profesor:
ING. LUIS FERNANDO PINILLOS

Autor:
PABLO ALBERTO BORBÓN ARIAS

CONTROL INTELIGENTE
BOGOTÁ D.C.
2009



CONTENIDO

El presente documento pretende ser una guía rápida de uso de la aplicación de redes neuronales hecha como trabajo para la asignatura de Control Inteligente en la Escuela Colombiana de Ingeniería. El documento se ha dividido en los siguientes ítems:

1. INTRODUCCIÓN Y REQUERIMIENTOS
2. ELEMENTOS DE LA INTERFAZ GRÁFICA
3. MODO APRENDIZAJE
4. MODO EJECUCIÓN
5. COMENTARIOS FINALES

BIBLIOGRAFÍA



Figura 2.2 Interfaz gráfica para la aplicación de Java compatible con Matlab

La interfaz gráfica para la aplicación de Java y la aplicación compatible con Matlab se muestra en las figuras 2.1 y 2.2 respectivamente.

Se distinguen las siguientes zonas:

- Zona 0: Contiene los elementos de menú, desde los cuales se puede observar la información del programa (“Acerca de”), salir de la aplicación o reiniciar la aplicación.
- Zona 1: Es una zona en la cual aparecerán los mensajes de estado, proceso y respuesta de la aplicación. En todo momento desplegará información útil sobre los procesos de la red. En la parte superior se encuentra el indicador de modo de la aplicación y el resultado más probable de la ejecución de la red.
- Zona 2: Es el área de dibujo. Al posicionar el Mouse sobre ésta área, aparece una rejilla indicando las divisiones de la matriz 7x5.
- Zona 3: Ésta zona corresponde a un campo de texto para indicar el nombre de un patrón determinado y el botón de agregar patrón que asocia dicho nombre con lo que se encuentre dibujado en la zona 2 al momento de presionar el botón.
- Zona 4: Contiene los botones de entrenamiento y calcular para simular la red neuronal una vez se han creado los patrones de entrada.
- Zona 5: Exclusiva de la aplicación compatible con Matlab. Es utilizada para calcular el resultado de la simulación de una red neuronal pre-entrenada en el Toolbox de Matlab.



3. MODO APRENDIZAJE

Una vez ejecutada la aplicación, ésta comenzará en el modo de aprendizaje. En dicho modo de funcionamiento la aplicación espera los 10 patrones de entrenamiento para la red neuronal. El usuario debe entonces proporcionar uno por uno los patrones asignándoles un nombre en el campo de texto de la zona 3, dibujando el patrón en la zona 2 y presionando el botón de “agregar patrón”. Cada patrón será procesado y su equivalente en código binario aparecerá junto con su nombre en la zona 1.

Una vez se hayan procesado los 10 patrones, se mostrarán las matrices de entrenamiento y objetivos en la zona 1, se desactivarán los elementos de la zona 3 y se invitará al usuario a entrenar la red para posteriormente procesar los dibujos en el modo “ejecución”.

Si se está ejecutando la aplicación compatible con Matlab, al finalizar el ingreso de los 10 patrones la aplicación creará un archivo llamado “archivoVectores.mat” que contiene la información de los patrones y los objetivos para ser utilizados en Matlab.

4. MODO EJECUCIÓN

En el modo ejecución el usuario tiene la posibilidad de entrenar la red presionando el botón correspondiente en la zona 4. Para dicho proceso la aplicación irá pidiendo uno por uno los datos de factor de aprendizaje, error mínimo por patrón y número máximo de iteraciones.

Una vez la red ha sido entrenada, la aplicación indica si se ha alcanzado el valor de error mínimo en el proceso de entrenamiento o si se llegó al límite máximo de iteraciones.

A partir de ese proceso, se activa la opción de calcular los resultados de una entrada determinada, en donde el usuario debe hacer un dibujo en la zona 2, luego presionar el botón “calcularJ” y esperar a que el sistema reconozca el carácter y devuelva tanto el nombre en la zona de “Más probable” como las salidas de la red en la zona de mensajes de la aplicación.

Si el usuario ejecuta la aplicación compatible con Matlab, tan sólo deberá ubicarse en la carpeta de la aplicación de Java y ejecutar el comando “hallarRed(alfa,error,iteraciones)” en la consola de Matlab después de iniciado el proceso de ejecución. El comando hallarRed tomará los parámetros de alfa, error e iteraciones dentro del paréntesis y cargará automáticamente los valores de los patrones y objetivos exportados por la aplicación de Java, luego creará una red neuronal con las características requeridas para éste trabajo y la guardará en un archivo llamado “redEntrenada.mat” en la carpeta de la aplicación. Dicha red será utilizada para procesar los dibujos de la matriz de 7x5 presionando el botón “calcularM”.



5. COMENTARIOS FINALES

La aplicación objeto de éste trabajo cuenta con documentación detallada de las clases, algoritmos y métodos utilizados y creados en toda su programación. Dicha documentación se encuentra en las carpetas “javadoc” dentro de la carpeta de distribución de la aplicación. Así mismo se ha suministrado el código fuente de la aplicación y sus respectivos proyectos en Netbeans dentro del CD de distribución.

BIBLIOGRAFÍA

<http://ohm.utp.edu.co/neuronales/Capitulo2/Backpropagation/ReglaB.htm>

KASABOV, Nikola K., “Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering”

FREEMAN, James A. SKAPURA, David M, “Neural Networks - Algorithms, Applications, and Programming Techniques”