CAPITULO 3

REDES HIBRIDAS-COMPLEJAS

3.1 Descripción de la Red Híbrida Compleja (HCNN)

La predicción de eventos caóticos que se presentan en un mundo que nos rodea es de gran interés. Especialmente en aquellos eventos que pueden poner en peligro su salud, como es en el caso del corazón. Hoy en día los problemas de ataques cardiacos ó taquicardias, son difíciles de prevenir. El poder predecir un evento que ponga en riesgo el funcionamiento del corazón mediante el análisis de electrocardiogramas (ECG) es de gran ayuda para la medicina moderna. El ECG es una señal que muestra un comportamiento caótico, por lo que lo hace una señal difícil de predecir.

En el área de redes neuronales artificiales recurrentes se han desarrollado algunos modelos para la predicción de señales caóticas. Uno de estos modelos es llamado Modelo Complejo. Este modelo consiste en encontrar al sistema neuronal que genere un atractor similar al del sistema dinámico que produce la señal al momento de aprender. Con este sistema se pretende predecir bifurcaciones y eventos catastróficos en un ECG.

Un modelo utilizado para predicción es la Red Híbrida Compleja (HCNN) [8]. Este modelo es una red neuronal parcial ó totalmente recurrente formada por pequeñas redes recurrentes. Estas pequeñas redes son 3 nodos completamente conectados, llamados generadores armónicos. Los generadores armónicos tienen la capacidad de reproducir una señal sinusoidal u otro tipo de señales. La salida dinámica para cada uno de los neuronas está definida por las ecuaciones (2.5) y (2.6) descritas en el capitulo anterior.

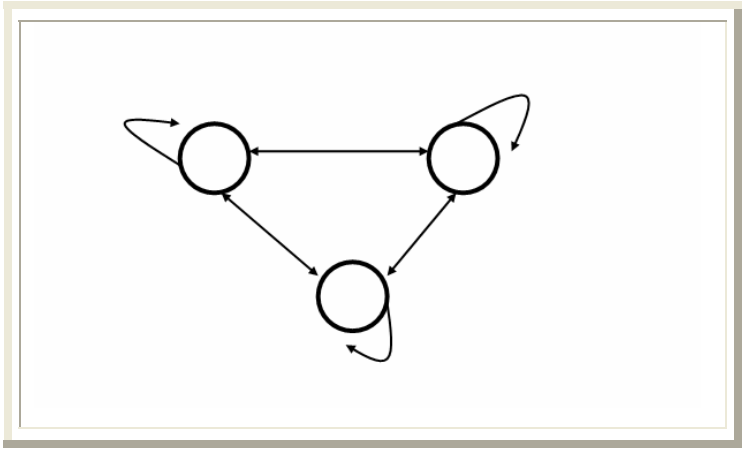

Figura 3.1 Generador Armónico [8].

El modelo HCNN está estructurado de la siguiente forma: generadores armónicos en el primer nivel, y conexiones hacia adelante del primer nivel al nivel escondido, y del nivel escondido al último nivel. La recurrencia de la Red Híbrida se presenta los generadores armónicos y en otras partes de la red. Para realizar el entrenamiento de esta red se utiliza el algoritmo de BPTT.

Para la predicción de señales caóticas (ECG) se necesita primero llevar acabo la etapa de entrenamiento de la red neuronal recurrente, por lo cual el usuario debe de proporcionar al sistema el conjunto de parámetros necesarios para realizar dicho entrenamiento. La aplicación del modelo BPTT es un entrenamiento supervisado, que consiste en aceptar una serie de tiempo que representan la entrada a la red (señal caótica) y una serie de pesos para cada uno de los nodos que forman parte de la red, los pesos representan la intensidad de las conexiones sinápticas de cada nodo. Con esta información y aplicando las formulas (2.5), (2.6) se obtiene la salida de la red. El siguiente paso consiste en calcular el error total utilizando la fórmula (2.8), que consiste en obtener el error para cada período de tiempo, mediante la diferencia entre la salida real y la salida deseada de la red. El error total es de suma importancia

para el proceso de entrenamiento de la red, ya que este debe de ir disminuyendo conforme avanzan las barridas del algoritmo, de lo contrario la red no alcanzará un entrenamiento. Obtener el error, es el medio que el usuario tiene para monitorear el proceso. Posteriormente en cada período de tiempo se lleva a cabo el ajuste de pesos. Los pesos de la red se entrenan hasta alcanzar un mínimo local y estos son utilizados para la etapa de predicción. Lo que se pretende con la etapa de entrenamiento es obtener una nueva señal que gráficamente represente ó sea parecida a la serie de tiempo que se utilizó como entrada a la red neuronal.

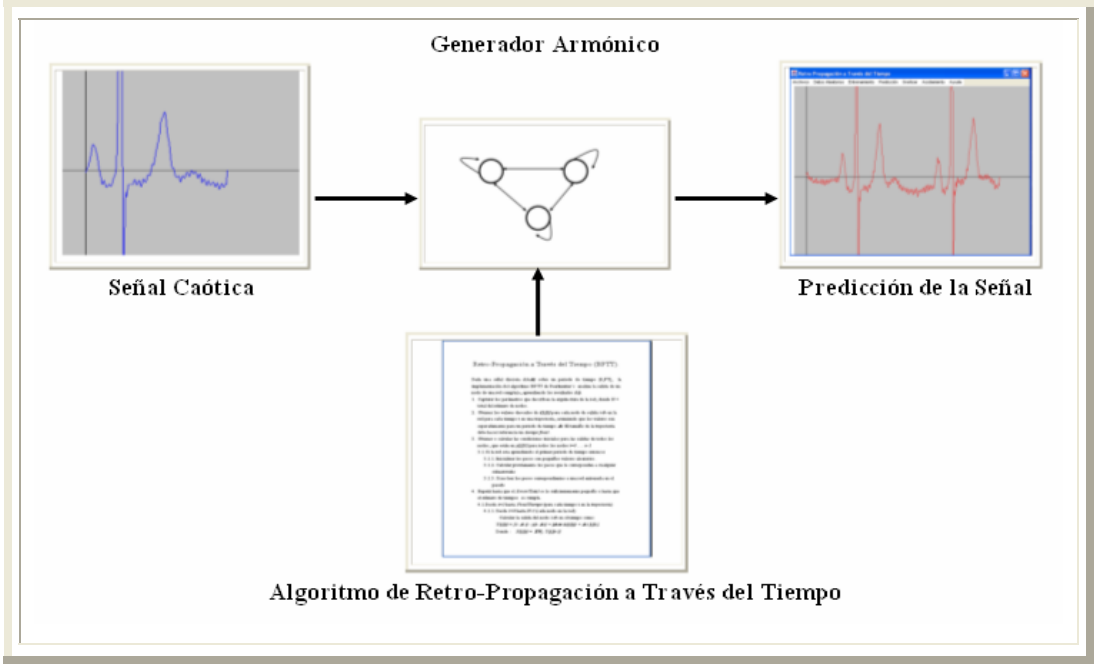

Figura 3.2 Predicción de Señales Caóticas.

Después de obtener un entrenamiento eficiente de la red se puede pasar a la etapa de predicción. Este módulo del sistema consiste en introducir una señal caótica y los pesos entrenados, para calcular la salida de la red. La salida que se obtienen de la red es el resultado de la predicción de la señal.

3.2 Implementación de Redes Recurrentes en ANNSYD

El ***Shell*** Neuronal Artificial Annsyd (Sistema Desarrollador de Redes Neuronales Artificiales) es un sistema que tiene como objetivo principal que aquellos usuarios que tienen conocimientos básicos de redes neuronales puedan diseñar y entrenar redes neuronales artificiales, basadas en diferentes modelos de una manera fácil. La topología de la red puede ser dibujada gráficamente con el ratón ó creada por medio de menús. En este **S*hell*** solo se tiene implementado el algoritmo de Retro-Propagación (Back-Propagation), que es utilizado para realizar el entrenamiento de las redes conectadas hacia adelante. Este método aplica un entrenamiento supervisado. Una de las principales diferencias de este algoritmo con el algoritmo de Retro-Propagación a Través del Tiempo son los datos que emplean, ya que el algoritmo de Retro-Propagación utiliza datos estáticos, mientras el de Retro-Propagación a Través del Tiempo utiliza datos dinámicos.

Este ***Shell*** cuenta con una interfaz de una forma muy amigable, para que al usuario le sea fácil ejecutar las diferentes opciones del sistema, además tiene las herramientas y funciones indispensables para poder definir la topología de las diferentes redes neuronales, así como también realizar el entrenamiento de la red, y posteriormente llevar a cabo el reconocimiento de patrones.

Annsyd está desarrollado en el lenguaje de programación Java, ya que una de las ventajas más significativas de Java es su independencia de la plataforma. En el caso de que se tenga que desarrollar aplicaciones que se ejecuten en sistemas diferentes esta característica es fundamental. Otra característica importante de Java es que es un lenguaje de programación orientado a objetos (POO). Esto nos permite desarrollar aplicaciones con interfaces muy amigables y de alta calidad. El ***Shell*** Neuronal fue diseñado utilizando el paradigma orientado a objetos, en donde se desarrollaron varias clases que comparten objetos entre ellas para funcionar de una manera eficiente.

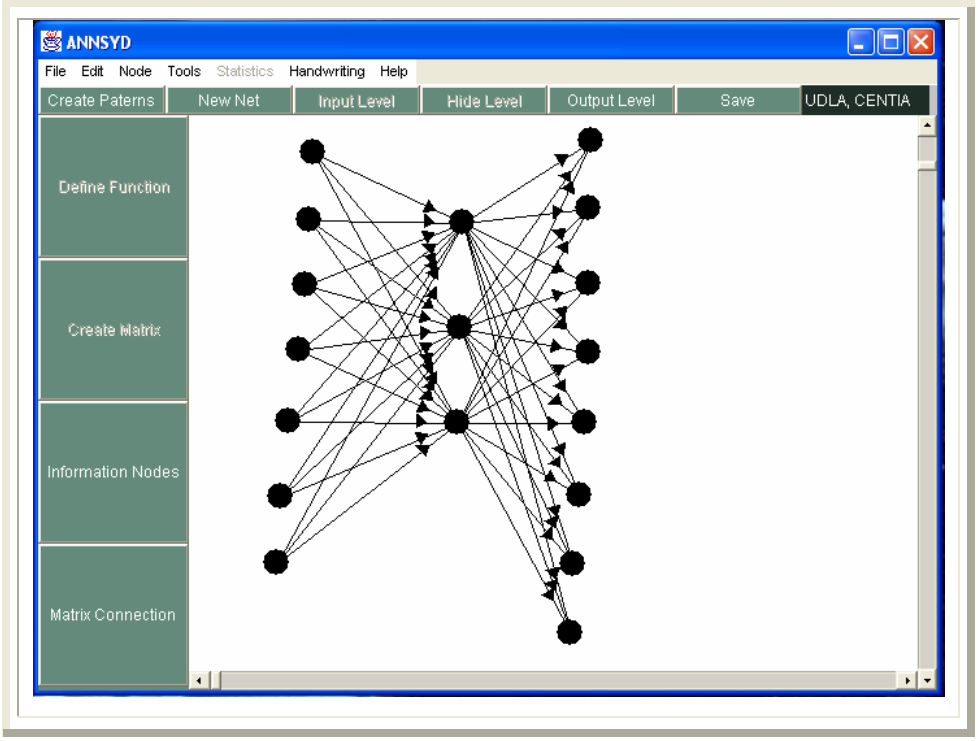

Figura 3.3 Ventana Principal de Annsyd [15].

La figura 3.3 nos muestra la ventana principal de Annsyd, en donde el usuario puede dibujar en un área, los diferentes niveles de nodos que forman a la estructura de la red neuronal y sus conexiones entre ellos, así también puede crear diferentes topologías. La ventana nos presenta también las diferentes opciones que forman parte del ***Shell*** Neuronal.

Después de que el usuario diseña la red neuronal, debe de leer ó crear el archivo de patrones, junto con los parámetros necesarios, para realizar el entrenamiento de la red neuronal. Después de obtener un entrenamiento eficiente se ejecuta el proceso de reconocimiento de letras manuscritas de Porfirio Díaz.

Uno de los objetivo de esta tesis es de desarrollar e implementar el algoritmo de Retro-Propagación a Través del Tiempo en el Shell Neuronal Annsyd, para que de esta forma el sistema ofrezca al usuario la oportunidad de trabajar con otro tipo de redes neuronales como son las Redes Neuronales Recurrente para la predicción de Señales Caóticas.