CAPÍTULO 3: REDES NEURONALES RECURRENTES

En este capítulo se describen las principales características y elementos de tres tipos de redes neuronales: *feedforward*, Elman y una red recurrente completamente conectada, con énfasis en éstas últimas. Se delinearán las ventajas y desventajas que caracterizan a cada ejemplo y se seleccionará la red más adecuada para la creación de un IDS.

3.1 INTRODUCCIÓN

Parte fundamental de un IDS es, evidentemente, la red neuronal que analiza y clasifica los datos para poder llevar a acciones preventivas o correctivas en términos de seguridad. Existen muchos tipos de algoritmos diferentes para gobernar el comportamiento de dichas redes, y en este capítulo se verán los más sobresalientes incluyendo la justificación de elegir el conocido como *backpropagation* y la regla del *steepest descent*.

Desde un punto de vista biológico, todo sistema nervioso consta de elementos básicos, las neuronas. Para una red neuronal, el elemento básico es conocido en forma general como un combinador lineal adaptativo (*adaptive linear combiner)*, el cual tiene una respuesta s_k para una serie de entradas del vector X_k (ver Figura 3.1). En este elemento, se muestra que la entrada X_k es modificada por ciertos coeficientes, denominados pesos *(weights)* que forman parte del vector Wk y cuyo valor es el resultado de comparar la salida s_k con el valor de salida deseado d_k. La señal de error que es generada se usa a su vez para actualizar los pesos w_{0k}, w_{1k}, etc, de tal manera que mediante un proceso iterativo la salida se aproxime al valor deseado y a un error ε_k de cero.

La estructura general del elemento en su totalidad es la que se muestra en la Figura 3.1 [3]:

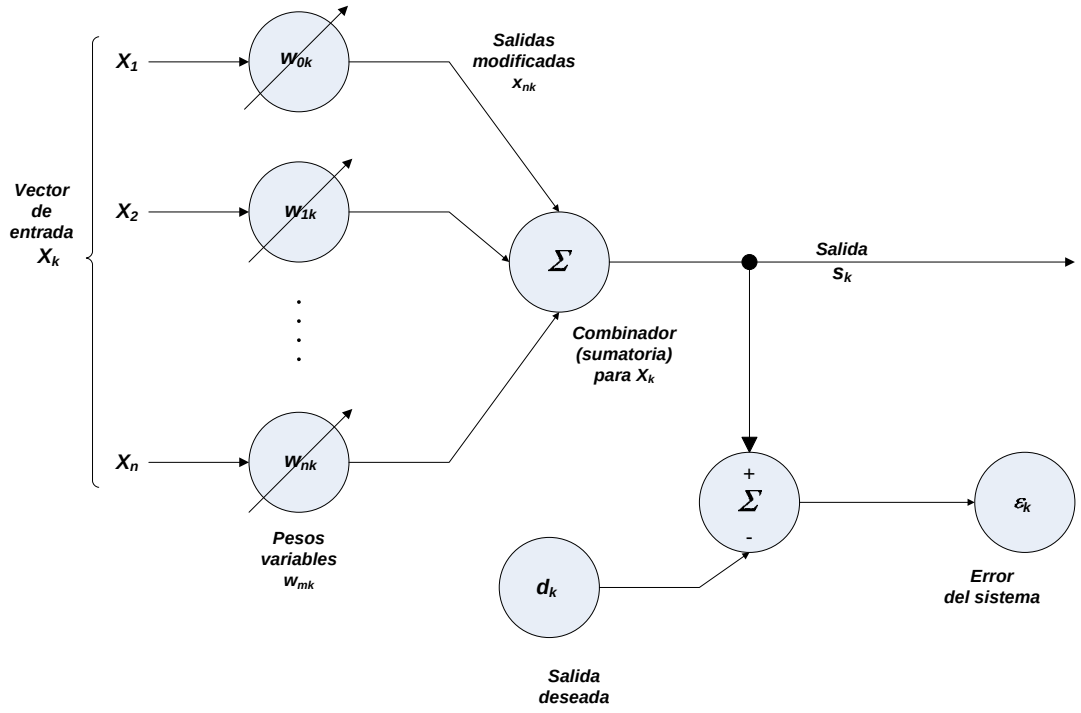

Figura 3.1. Combinador lineal

Este tipo de elementos lineales pueden ser modificados con ciertas funciones que limiten o modifiquen su salida para que ésta esté en un rango determinado de valores. Funciones tales se conocen como funciones de umbral, y las hay de diferentes tipos, con diferentes características. Las funciones de umbral más usuales son las lineales y las no lineales. Dentro de éstas últimas, dos funciones son predominantemente usadas: la función *signum* (limitador binario a ±1) y la función sigmoide (limitador a valores continuos), la cual es usualmente una función tal como la tangente hiperbólica [1,2].

Una vez que la función de umbral se ha incluido en la trayectoria de la señal de salida, el combinador lineal se convierte en un elemento lineal adaptativo (*Adaline*, por sus siglas en inglés). Este es la verdadera unidad fundamental de todas las redes neuronales, la cual realiza la única función de producir una salida acotada para cualquier entrada que se le presente, con fines de clasificación.

3.2 REDES *FEEDFORWARD*

Dependiendo de la arquitectura e interconexión de todas las neuronas de una red, puede clasificársele en distintas categorías. La primera de ellas es la de las redes conocidas como *feedforward.* Como su nombre lo indica, en este tipo de redes se empieza con un vector de entradas el cual es equivalente en magnitud al número de neuronas de la primera capa de la red, las cuales procesan dicho vector elemento por elemento en paralelo. La información, modificada por los factores multiplicativos de los pesos en cada neurona, es transmitida hacia delante por la red pasando por las capas ocultas (si hay) para finalmente ser procesada por la capa de salida. Es por eso que este tipo de redes reciben su nombre.

Es importante mencionar que las redes *feedforward* son las más sencillas en cuanto a implementación y simulación, pero su desempeño es bueno para aplicaciones en los que no se requiera que la red retenga información de eventos pasados como ayuda para evaluar eventos futuros. Cada vector de entrada presentado como entrenamiento para este tipo de redes es una entidad aislada del resto y, al final de dicho periodo de prueba, la red estará lista para comenzar a identificar y clasificar patrones, reconocer imágenes o cualquier otra aplicación que se le quiera dar [5].

Al iniciar las investigaciones sobre redes neuronales, las redes *feedforward* fueron las que recibieron más atención de parte de los investigadores porque sus características en cuanto a tiempos de procesamiento hacían viables simulaciones con los equipos computacionales de la época. Comparadas con las otras redes, las FFNN *(Feedforward Neural Network)* son una opción cuyo balance costo-velocidad y costo-exactitud es tal que da mayor ventaja al costo que a los otros parámetros.

En este tipo de redes no existen interconexiones entre capas más allá de la conexión directa hacia delante para propagar la información. No hay rutas de retroalimentación para desempeñar la función de memoria de la red [3].

En la Figura 3.2 se aprecia una FFNN (*feedforward neural network)* en donde todas las neuronas de una capa están interconectadas con las neuronas de la capa siguiente, iniciando con la capa principal y los elementos del vector X_k, proporcionando su información (las salidas $s_{mn}(t+1)$) propagándola hacia delante dentro de la red. El vector de pesos W_k es actualizado conforme los *epochs* pasan mientras el entrenamiento prosigue, y al final del mismo los pesos individuales $w_{11}\ldots w_{1n}$, $w_{21}\ldots w_{2n}$, etc, asumen sus valores finales para iniciar el trabajo de la FFNN con datos de entrada nuevos una vez que se ha llegado a una o varias salidas globales $s_{on}(t+1)$. La función de umbral *signum* (±1) se encuentra a la salida de la red para convertir $s_{on}(t+1)$ en el valor final y_k.

3.3 REDES ELMAN

Las redes *feedforward* tienen ciertas limitaciones inherentes a su diseño que pueden ser mejoradas con un cambio de arquitectura. Las redes Elman son conocidas como redes recurrentes simples (SRN) y son precisamente una mejora de las redes *feedforward* debido a la inclusión de retroalimentación entre las capas inmediatas contiguas [16].

Los nombres de las variables (ver Figura 3.3) corresponden a las de la Figura 3.2, pero en este caso la retroalimentación simple hace que dichas redes posean una memoria de los eventos inmediatos anteriores, y éstos afectan las actualizaciones sucesivas de los pesos en cada una de las capas de la red. De esta manera, los algoritmos de aprendizaje utilizados pueden mejorar el desempeño del sistema de detección de intrusos para conseguir mejores porcentajes generales al final de una sesión de entrenamiento con un conjunto de datos igual al de la red *feedforward*.

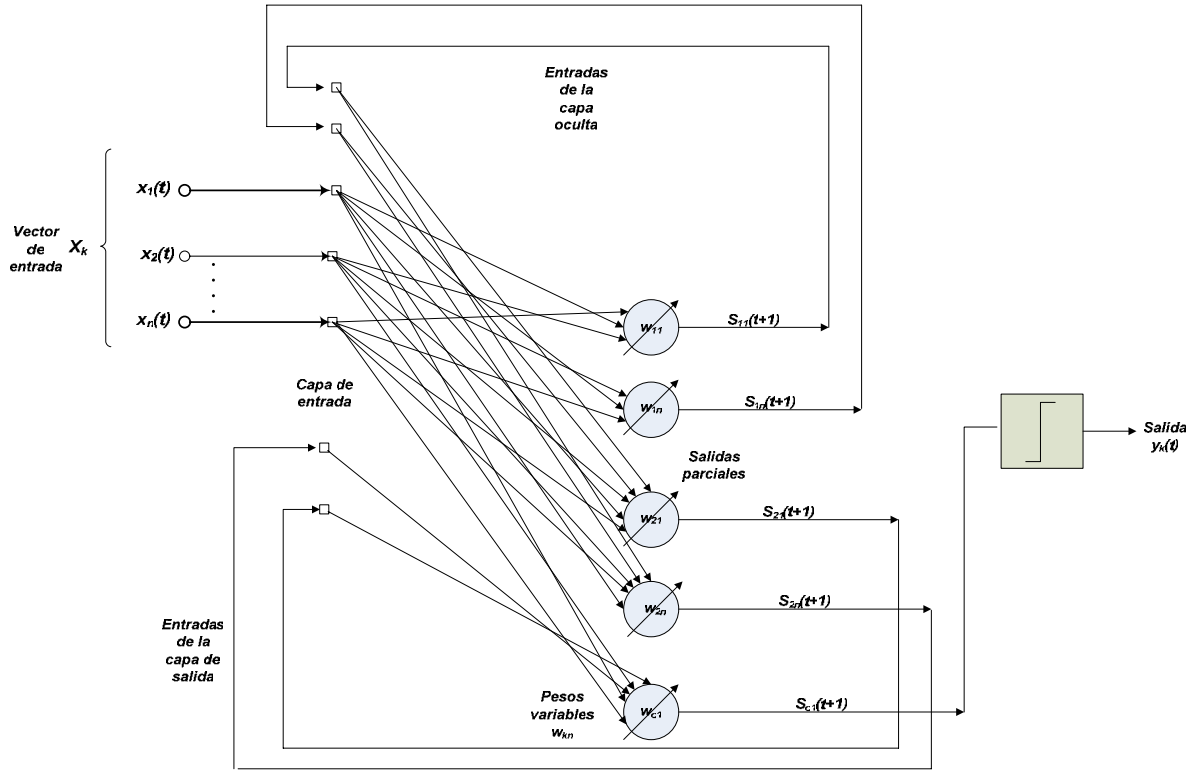

Figura 3.2. Red *Feedforward* función de activación *signum*

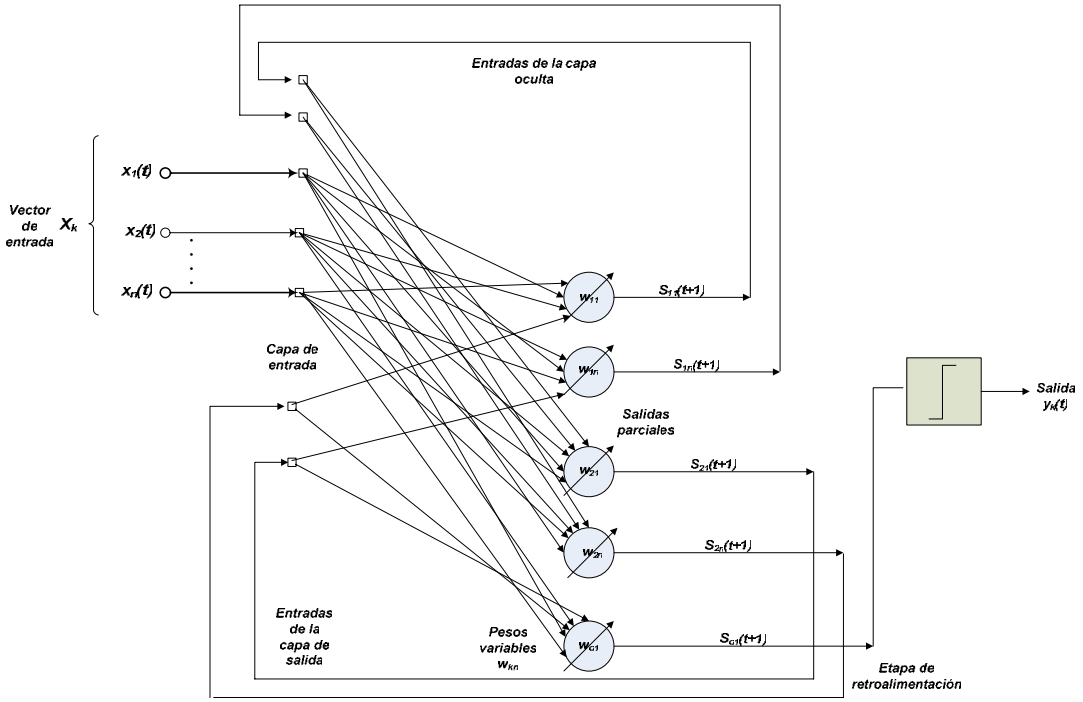

Figura 3.3. Red Elman, función de activación *signum*

Las redes Elman poseen varias características que las hacen superiores a las FFNN. En primer lugar, para una meta de desempeño dada con un margen de error pueden converger a dicho valor más rápido que las redes *feedforward.* Eso significa que el número de iteraciones que los datos de entrenamiento y posteriormente los datos de prueba deben realizar dentro de la red es menor. También, el porcentaje de efectividad de estas redes es significativamente menor al de las FFNN debido a que los caminos de retroalimentación generan un mejor comportamiento no-lineal durante el aprendizaje supervisado. Es menos probable que un IDS basado en redes Elman identifique un ataque cuando en realidad se trata de código normal, o que una vez identificado el ataque lo clasifique en una categoría errónea [2].

Sin embargo, estas ventajas tienen un costo importante: el tiempo de procesamiento. Las redes Elman tienen un tiempo de procesamiento mayor que el de las redes *feedforward*, y esta característica las hace más lentas en aplicaciones donde el número de neuronas es grande tanto en las capas de entrada y salida como en las capas ocultas. De hecho, el incluir más de una capa oculta en una red Elman para la resolución de un problema sólo debe ser contemplado si la complejidad del mismo es substancial. En conclusión, este tipo de redes son un compromiso entre desempeño y rapidez. No son tan eficaces como las redes neuronales recurrentes completamente conectadas, pero tampoco requieren tiempos de procesamiento tan alto como ellas. En cuanto a las redes *feedforward*, hay aplicaciones como las de reconocimiento de imágenes (lectores ópticos, reconocimiento de firmas, fotografía) en las que la no-linearidad del problema es un factor secundario y en el que excesivo tiempo de procesamiento sería perjudicial para la solución del problema.

3.4 REDES NEURONALES RECURRENTES

Las redes neuronales recurrentes completamente conectadas, a diferencia de las redes Elman, tienen caminos de retroalimentación entre todos los elementos que las conforman. Una sola neurona está entonces conectada a las neuronas posteriores en la siguiente capa, las neuronas pasadas de la capa anterior y a ella misma a través de vectores de pesos variables que sufren alteraciones en cada *epoch* con el fin de alcanzar los parámetros o metas de operación.

La complejidad de este tipo de redes es alta en comparación con una red *feedforward*, por ejemplo, ya que en esta última la red sólo es capaz de transmitir la información hacia las capas siguientes resultando en un efecto de propagación hacia atrás *en el tiempo.* Las redes neuronales recurrentes, en cambio, realizan el intercambio de información entre neuronas de una manera mucho más compleja y por sus características, dependiendo del tipo de algoritmo de entrenamiento que se elija, pueden propagar la información hacia delante en el tiempo, lo cual equivale a predecir eventos. Esta es una característica muy importante para ciertas aplicaciones, como los IDS, ya que la capacidad de predicción de eventos significativos (en este caso, ataques a la red) basada en las entradas anteriores al sistema le proporciona un beneficio importante a la seguridad del mismo [3,5].

La arquitectura básica de una RNN se muestra en la Figura 3.4. Una característica importante es la inclusión de *delays* (z^{-1}) a la salida de las neuronas en las capas intermedias; las salidas parciales $s_{mn}(t+1)$ se convierten en valores $s_{mn}(t)$, un instante de tiempo anterior, y así se retroalimentan a todos los componentes de la red, guardando información de instantes de tiempo anteriores. Puede observarse cómo todos los nodos están interconectados entre sí y también con los nodos anteriores a ellos a través de conexiones directas y también *delays* antes de cada capa, o memorias

temporales. El diagrama ha sido simplificado para no incurrir en excesiva complejidad, pero cada una de las capas está representada por cierto número de neuronas.

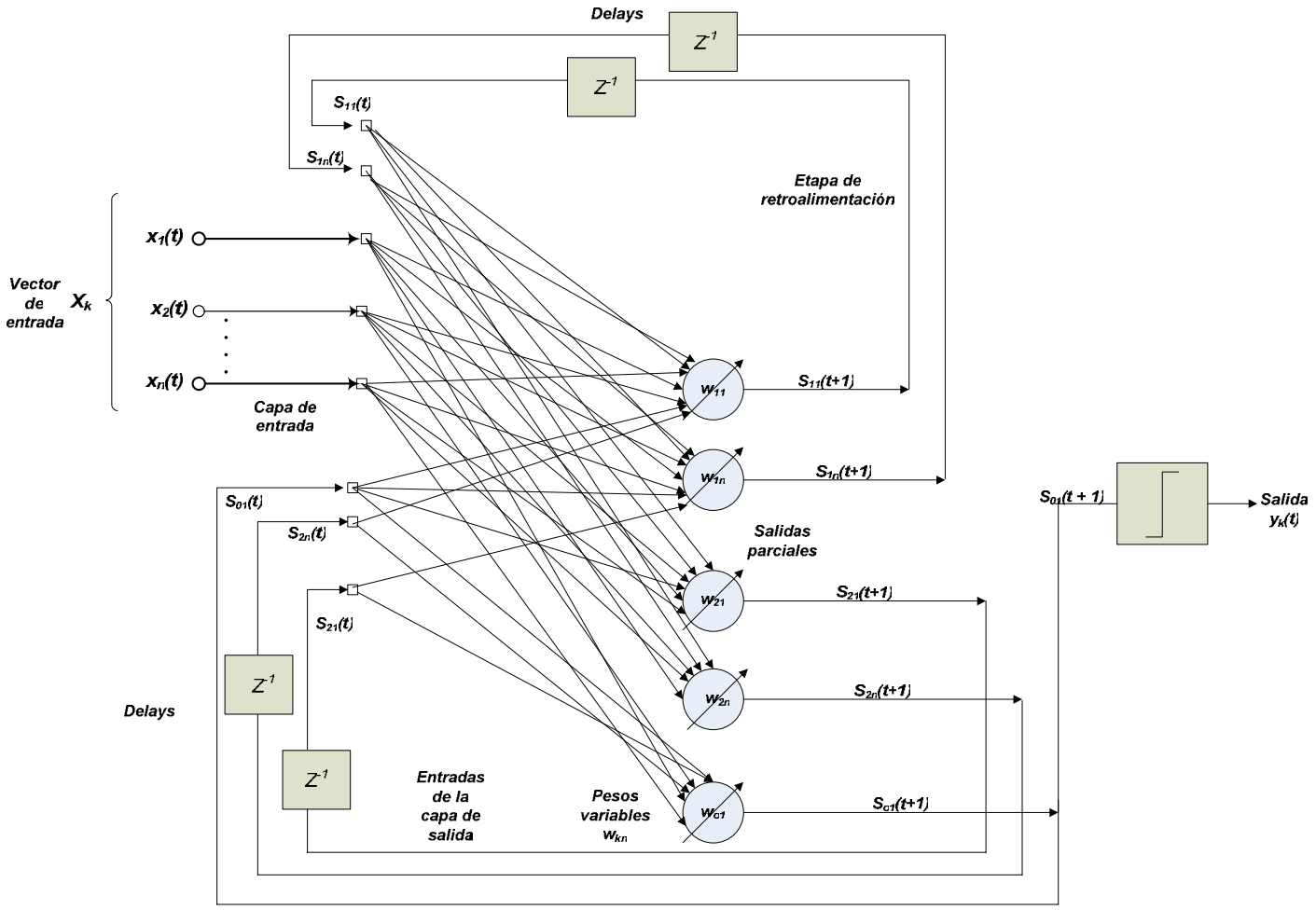

Figura 3.4. Red neuronal completamente conectada

Las redes neuronales recurrentes son más eficaces para resolver problemas con no-linealidades temporales significativas. Son especialmente útiles en aplicaciones tales como el reconocimiento de patrones secuenciales, cambiantes en el tiempo, ya que las capacidades de predicción y mapeo de las RNN así lo permiten. En la investigación [14, 15], se ha hecho énfasis en que, a diferencia de las redes *feedforward* en las que la meta es que la red converja a un valor de estados fijo en cierto periodo de tiempo, las RNN

tienen comportamiento variable en el tiempo y esto ofrece posibilidades de resolución de problemas diferentes a aquellos con una arquitectura tradicional.

En los sistemas biológicos, los cuales fueron las bases conceptuales de las redes neuronales, el número de interconexiones entre todas las neuronas es muy grande. Las RNN pueden aproximarse más a ese comportamiento que los demás tipos de redes, pero por lo general la complejidad intrínseca de éstas requiere tiempos de procesamiento muy superiores. Las características variables de sus estados internos a través del tiempo también hacen muy importante considerar en qué momento deben actualizarse los pesos: al final de cada *epoch* (*epoch-wise training)* o continuamente. En el segundo caso, encontrar el instante adecuado para la actualización es un reto significativo [5].

3.5 RESUMEN

En general, todos los sistemas basados en redes neuronales son diseñados con el fin de aprender, generalizar y en ciertos casos predecir patrones o estados deseados a través del entrenamiento y prueba de dichos sistemas. Las diferentes arquitecturas de las redes neuronales tienen diferentes propiedades dependiendo de la manera en que sus componentes, las neuronas, se interconectan y comparten información.

Para los problemas en los que se requiere que la red neuronal se estabilice en un estado final fijo para después someterla a pruebas con datos de entrenamiento, la arquitectura ideal es aquella que comprende a las redes *feedforward.* Este tipo de redes pueden compartir información con las capas inmediatas posteriores y de esta manera actualizan sus pesos al evaluarse la función de error al final de la red, pasando a una nueva iteración. En términos de tiempo de procesamiento, estas redes no demandan tantos cálculos y actualizaciones, por lo que son más rápidas al procesar pero más lentas al llegar al resultado deseado. Adicionalmente, su margen de error es mayor que de los

otros tipos de arquitectura y por lo tanto deben utilizarse sólo en aquellos problemas que no presenten no-linealidades significativas en el tiempo.

Las redes Elman son un paso intermedio entre las redes *feedforward* y las redes neuronales recurrentes completamente conectadas. Estas redes cuentan con caminos de retroalimentación simples entre capas, lo que le da a la red la capacidad de almacenar información del estado inmediato anterior. Es por esto que estas SRN tienen algunas de las ventajas de las RNN, pero su eficiencia y velocidad de convergencia son inferiores a éstas últimas. En términos del tiempo de procesamiento requerido para que este tipo de redes logren la meta, se le puede considerar como un término intermedio. No obstante, su arquitectura las hace óptimas solamente para aplicaciones muy específicas limitadas a predicciones un instante de tiempo adelante, y la mayoría de los problemas a los que se someten las redes recurrentes necesitan capacidades de procesamiento, almacenamiento y predicción superiores.

Las RNN, por otro lado, son ideales para problemas tales como los que se le presentan a un IDS durante su funcionamiento: la identificación y clasificación de patrones secuenciales con distintas probabilidades de ocurrir a través del tiempo. El comportamiento no lineal que las interconexiones de retroalimentación ocasionan en las RNN las hace óptimas para resolver estos problemas.

Dependiendo del tipo de algoritmo de entrenamiento que se utilice, es posible que las RNN requieran tiempos de procesamiento muy grandes. No obstante, a pesar de esto la velocidad con la que convergen al resultado esperado ignorando máximos y/o mínimos locales sobrepasa con creces esta desventaja, en especial en redes con un número de neuronas reducido. En cuanto a sus porcentajes de error, las RNN son más efectivas al detectar, clasificar y aprender nuevos patrones. Para un mismo conjunto de

datos de entrenamiento, entregarán mejores resultados que cualquiera de los otros dos tipos de redes neuronales.