



## **Estrategia de búsqueda de dispositivos basada en el historial de conexiones utilizando redes neuronales**

J. A. Abad Padilla, J. L. Gorricho Moreno

**Resumen:** La movilidad es una de las principales características de las redes de comunicación actuales y produce cambios en su estructura, que en muchas ocasiones no son advertidos por la totalidad de los dispositivos que la conforman, principalmente por la distancia entre los dispositivos y el rango de transmisión. Por ello, la comunicación entre dos dispositivos se convierte en un problema de enrutamiento, el cual se define como la búsqueda de trayectorias mediante una adecuada estrategia. Para abordar esta situación, planteamos una estrategia de búsqueda basada en el historial de conexiones del dispositivo móvil. Por naturaleza, una persona tiende a exhibir comportamientos repetitivos, por lo que, observando estos patrones conductuales podremos, con cierta certeza, ubicarlo en un espacio y tiempo específico. Tomando en consideración lo anterior, si utilizamos el historial de conexiones de un dispositivo y mediante las apropiadas herramientas estocásticas, se podría lograr una visión de la estructura de la red y de esta forma predecir la secuencia de dispositivos que formarían la trayectoria para la comunicación entre dos dispositivos. Utilizaremos como herramienta de predicción de secuencias redes neuronales y analizaremos su contribución en el diseño de un algoritmo de búsqueda de dispositivos móviles.

Palabras Clave: estrategia de búsqueda, historial de conexiones, predicción, redes neuronales.

**Abstract:** Mobility is one of the main features of actual communication networks. This mobility causes changes in the network structure, which often are not noticed by all the devices in the network, mainly because of the distance between the devices and the transmission

range. Therefore, communication between two devices becomes a routing problem, which is defined as finding paths through a proper strategy. To address this situation, we propose a search strategy based on connections history of mobile devices. We know that, by nature, a person tends to exhibit repetitive behaviors, so analyzing these behaviors patterns we can probably establish his location. Considering the above, if we use the mobile device's connections history and an appropriate stochastic tool, we could achieve a vision of the network structure and predict the sequence of devices that form the path for setup a new connection between the intended two devices. As a tool for predicting sequences we will use neural networks, and also analyze their contributions in the construction of paths to search for devices.

Keywords: search strategy, connections history, prediction, neural networks.

### **Introducción**

Las redes actuales de comunicación están caracterizadas por la presencia de dispositivos móviles autónomos que se comunican entre sí, a través de enlaces inalámbricos, no cuentan con una infraestructura de red fija y la gestión se realiza de manera descentralizada aumentando su complejidad a medida que se incrementa su densidad.

En estos entornos, la colaboración entre los dispositivos o nodos presentes se convierte en la clave fundamental para la operación correcta de la red. Los nodos toman decisiones, realizan acciones encaminadas a la gestión de propia red y de servicios y toman parte importante en la operación de los algoritmos de encaminamiento y de búsqueda. De aquí la importancia, para cada nodo, de conocer la ubicación del resto de los elementos de la red para el momento en que intente contactarlos.

Sin embargo, el dinamismo de estos entornos hace titánica la tarea de los nodos para almacenar y manipular la gran cantidad de información que se requiere para tener una visión completa de la topología

---

José-Alfredo Abad Padilla, Facultad de Ingeniería, UABC, Campus Mexicali I, Blvd. Benito Juárez s/n, 21280, Mexicali, México, jaabad@entel.upc.edu.

Juan-Luis Gorricho Moreno, Departamento de Telemática, c/ Jordi Girona 1-3, 08034, Barcelona, España, juanluis@entel.upc.edu.

Este trabajo ha sido parcialmente financiado por la CICYT con registro: TSI2007-66637-C02 y por la Facultad de Ingeniería de la UABC.



de la red y con ello tener localizado a cada nodo en todo momento. Por esta razón, es indispensable una adecuada estrategia de búsqueda que contrarreste estas complejidades.

Por otra parte, sabemos que las personas por naturaleza tienden a exhibir comportamientos repetitivos, por ejemplo, cogen el autobús en el mismo sitio, siguen los mismos caminos de casa a la oficina y viceversa, toman café a determinada hora y en determinado lugar, se comunican un mismo grupo de personas, utilizan los mismos dispositivos de comunicación, entre otras. Por ello, haciendo un seguimiento de una persona, por un tiempo largo, poniendo atención en las actividades que realiza en el día a día y mediante el análisis de esta información podemos detectar patrones que caracterizan sus hábitos.

Tomando estas consideraciones, podemos asumir que, con cierta certeza, encontraremos a un usuario utilizando un dispositivo de comunicación, en la misma ubicación en un mismo espacio de tiempo a lo largo de distintos días, y lo mismo pasara con el resto de los usuarios de la red de comunicación, con esto, podríamos suponer que la estructura de la red tendría similitudes. Por lo tanto, tratando de explotar estas peculiaridades podríamos abordar el problema de la comunicación entre dispositivos en redes móviles y presentar una posible solución.

En este artículo de investigación presentamos como solución para este problema, una estrategia de búsqueda basada en la predicción de trayectorias de búsqueda aplicando para ello redes neuronales.

### **Información general**

Una red neuronal trata de simular las propiedades de una red neuronal biológica mediante una modelación matemática. Esta construida de neuronas que forman capas. Cada capa recibe una entrada y produce una salida que es enviada a la capa a la cual está conectada. Las conexiones entre neuronas de las distintas capas son llamadas sinapsis.

Debido a que una neurona aislada dispone de poca potencia de cálculo, las capas de la red neuronal están formadas por varias neuronas trabajando en paralelo, estas redes generalmente son nombradas redes neuronales multicapas.

Existen muchos problemas que pueden ser resueltos utilizando redes neuronales. Sin embargo, estas redes son utilizadas comúnmente para abordar problemas específicos. Entre estos están los

relacionados con: la clasificación de información, la predicción, el reconocimiento de patrones, y la optimización [1]. Por ello, consideramos a las redes neuronales como la herramienta estocástica viable para abordar el desarrollo de una estrategia de búsqueda, si queremos utilizar para este fin la predicción.

La utilización de redes neuronales en la solución de cualquier problema involucra 3 pasos [2]: 1. Entrenamiento, donde se actualizan los pesos de las conexiones entre las neuronas; 2. Verificación, cuando los valores de los pesos son constantes y se comprueba que la salida sea la misma que en la fase de entrenamiento; y 3. Generalización, la salida de la red es evaluada con datos que no fueron usados para entrenar la red. Una buena generalización es deseable a fin de conseguir una red preparada para procesar datos que podrían ocurrir en el futuro.

Existe literatura en la que se presentan solución a problemas de diversas índoles, utilizando redes neuronales con la finalidad de aprovechar su capacidad de predicción, en [3] se diagnostica anomalías en reactores nucleares y daños relacionados con los motores en una planta nuclear. En [4] se realiza una predicción de la tendencia del precio de las acciones. Por su parte en [5] se predice el lugar al que se desplazo una persona, cuando se encuentra fuera de su lugar habitual. Por su parte en [6] su aplicación está en la meteorología. En el área de las redes de comunicación en [2] se afronta a los ataques a las redes, en [7] se trabaja con el tema de la calidad de servicio para las redes locales inalámbricas y en [8] con tráfico en las redes.

### **Propuesta para la estrategia de búsqueda**

El problema de la comunicación entre nodos en una red inalámbrica consiste en la optimización de las trayectorias a seguir por uno de los nodos para alcanzar al nodo objetivo. Basado en los patrones de los usuarios y en específico, en el historial de conexiones de un dispositivo, presentamos como solución al problema de localización o búsqueda de dispositivos, un algoritmo basado en la creación de trayectorias de búsqueda mediante la predicción. La predicción normalmente tiene sus bases en la construcción de modelos estocásticos que predicen valores subsecuentes en el tiempo dada una historia de valores anteriores [8]. Y esto se logra a través del uso de modelos lineales y redes neuronales.



Para que un nodo pueda establecer una conexión con otro nodo se requiere encontrar la secuencia de nodos que lo conducirán a la localización de este nodo destino. Para este fin, utilizaremos redes neuronales de propagación hacia delante (Feedforward) y redes recurrentes, locales a cada nodo, esto es, el entrenamiento de la red neuronal se llevará a cabo con la información obtenida y almacenada por el nodo y las predicciones que de estas redes neuronales se obtengan serán sólo significativas para el nodo que las entreno. El papel de la red neuronal es indicar a través de una predicción un nodo, la secuencia de nodos predichos representa la trayectoria de búsqueda del nodo objetivo dentro de la red de comunicaciones.

La idea básica es plantear la estrategia de búsqueda considerando la siguiente hipótesis: El movimiento que describen los nodos es tal que, con el tiempo, se vuelve repetitivo, por lo que en una red de comunicaciones la secuencia de nodos que forman la trayectoria de búsqueda, desde un nodo X hacia un nodo Y, será la misma secuencia, cualquier día, para el mismo periodo del día. Por lo que, tomando el historial de conexiones de los dispositivos, podremos predecir trayectorias de búsquedas si conocemos trayectorias de búsquedas previas.

#### **Escenario de operación**

Con la finalidad de simplificar el análisis tomaremos el área de cobertura de la red como una rejilla. Consideraremos una rejilla de dos dimensiones, de 10 por 10 celdas, como el área sobre la cual se encuentran ubicados todos los nodos o dispositivos móviles.

Se establece un rango de transmisión de una celda, por lo que un nodo sólo será conciente de la presencia de los nodos que se encuentran dentro de su vecindad, ignorando los cambios en la estructura de la red provocados por los nodos ubicados fuera de esta área.

El número de nodos con los que trabajaremos será 20 nodos, y solo un nodo está presente en cada celda a la vez, distribuidos de manera arbitraria, sin embargo se tendrá que cumplir con la siguiente restricción: cada nodo tendrá como mínimo un nodo vecino. Lo anterior es con la finalidad que cualquier nodo sea alcanzable por cualquiera del resto de los nodos de la red de comunicación. En la figura 1 se muestra la topología de la red en el instante  $t$ .

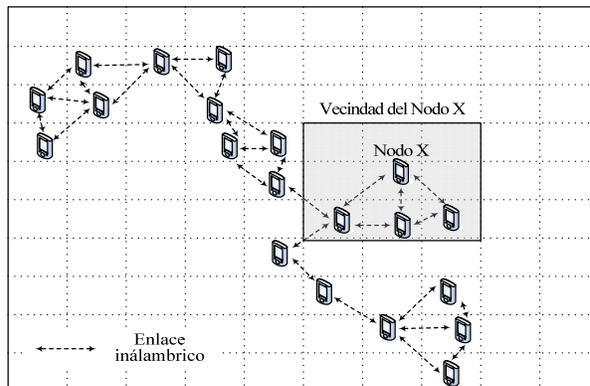


Fig. 1 Estructura de la red en el instante  $t$ .

#### **Codificación de los datos del historial**

Para el entrenamiento, verificación y generalización de la red neuronal utilizaremos los siguientes datos: la hora, el nodo buscado y los nodos de la trayectoria seguida. Estos datos serán extraídos del historial de conexiones del dispositivo, para el cual se entrena la red neuronal.

Los nodos están identificados como *Nodo  $n$* , donde  $n = 1, 2, \dots, 20$ . El *Nodo  $n$*  será codificado como vector de 20 elementos, donde cada elemento del vector tendrá valor de 0 excepto para el elemento de la posición igual a  $n$ , cuyo valor será igual a 1. Para manejar la hora haremos una división de ésta en 6 periodos del día, cada periodo agrupa 2 horas del día de una jornada de observación, que empieza a partir de las 8:00 horas y termina a las 20:00 horas, lo que implica que codificaremos la hora como un vector de 6 elementos, donde los valores de los elementos serán similares a la codificación de los nodos, es decir, pondremos un 1 en el primer elemento del vector y 0 en el resto de los elementos para codificar el primer periodo, para el segundo periodo ponemos 1 en la segunda posición y 0 en el resto de los elementos y así para cada uno de los periodos.

#### **Estructura de la red neuronal**

Para la predicción de los nodos para formar las trayectorias utilizaremos tres tipos distintos de redes neuronales multicapas: la red Feedforward, la red neuronal recurrente tipo Elman y la red neuronal recurrente tipo Jordan.

Las arquitecturas que manejamos para las redes son las siguientes. La red neuronal Feedforward está compuesta por: una capa oculta y una capa de salida.



Por otra parte, las redes recurrentes, además de contar con la capa oculta y la capa de salida, maneja una capa de contexto. Las neuronas de la capa de contexto tienen conexión directa con las neuronas de alguna de las capas de la red neuronal, es una retroalimentación, en el caso de la red Elman, la retroalimentación proviene de las neuronas de la capa oculta, mientras que en la red Jordan provienen de la capa de salida. La arquitectura de una red Feedforward se muestra en la figura 2 y en la figura 3 se muestra la de la red Elman. La red Jordan tiene la misma arquitectura de la figura 3, con la salvedad que la realimentación proviene de la capa de salida.

Para determinar el número de neuronas en cada capa de las redes neuronales que utilizaremos, se debe tomar en cuenta la forma de codificación de los datos y los datos que se manejan como entrada y salida de la red neuronal.

Manejaremos dos casos para analizar con cada una de las redes neuronales. En el caso 1 tendremos como vector de entrada  $X(t) = \{x_i\}$  para  $i = 1, 2, \dots, 66$ , de los cuales, 20 elementos del vector  $X$  son utilizados para representar el nodo a alcanzar (objetivo), recordamos que cada nodo está codificado como un vector de 20 elementos, 40 elementos representan un historial de 2 nodos ( $2 \times 20$ ) de la trayectoria y 6 elementos para el periodo del día. Como salida tendremos  $Y = \{y_k\}$  para  $k = 1, 2, \dots, 20$ , donde  $Y$  es la predicción del siguiente nodo que formara parte de la trayectoria. La capa oculta contiene 67 neuronas, ya que, de acuerdo con el estudio hecho en [5] se recomienda tener tantas neuronas como entradas se tenga más 1 neurona.

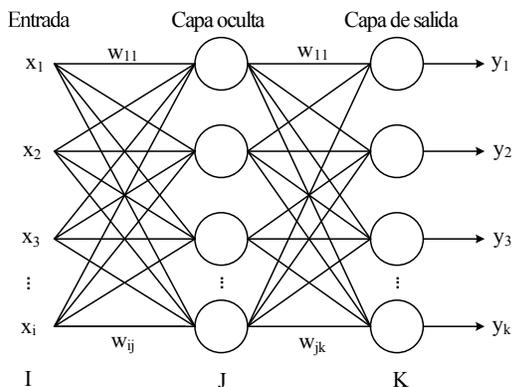


Fig. 2 Red neuronal de propagación hacia adelante (Feedforward) multicapas.

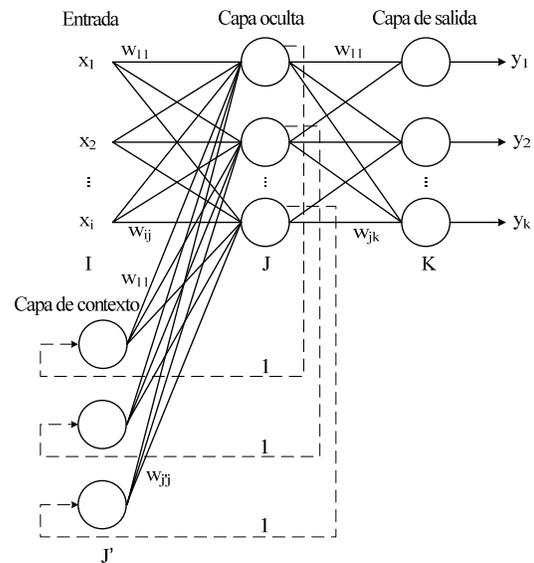


Fig. 3 Red neuronal recurrente Elman

Por tanto, tenemos para la red Feedforward los siguientes valores:  $J = 67$  y  $K = 20$ , donde  $J$  es el número de neuronas de la capa oculta y  $K$  el número de neuronas de la capa de salida. Para la red Elman  $J = 67$ ,  $K = 20$  y  $J' = 67$ , donde  $J'$  es el número de neuronas de la capa de contexto y que para la red Elman debe ser igual al número de neuronas de la capa oculta. Y para la red Jordan  $J = 67$ ,  $K = 20$  y  $J' = 20$ , donde  $J'$  debe ser igual a  $K$ .

Si consideramos las trayectorias como series en el tiempo, podemos representar la trayectoria como una secuencia de duplas:  $[t_1, n_1], [t_2, n_2], \dots, [t_s, n_s]$ , donde  $t_i$  representa el tiempo, y  $n_i$  representa el nodo en una trayectoria. Para el caso 1, el historial de 2 nodos representa el hecho de entrar en la red el nodo del tiempo  $t_1$  y el nodo del tiempo  $t_2$  para tener como salida el nodo del tiempo  $t_3$ .

En el caso 2 tenemos como vector de entrada  $X(t) = \{x_i\}$  para  $i = 1, 2, \dots, 46$ , y está distribuido de la siguiente forma: 20 elementos para el nodo destino, 20 para el nodo de la secuencia en el tiempo  $t$ , y 6 para el periodo del día. Aquí el nodo de la secuencia en el tiempo  $t$ , es manejado como la acumulación de nodos que preceden al nodo del tiempo  $t$ , con ello tendremos que, para predecir el nodo en el tiempo  $t_3$  utilizaremos como entrada para el nodo de la secuencia el acumulado de todos los nodos desde  $t_1$  a  $t_4$ , esta forma es utilizada también en [2] para hacer predicciones.



El número de neuronas para cada capa de la red neuronal Feedforward es  $J = 47$  y  $K = 20$ . Para la red Elman  $J = 47$ ,  $K = 20$  y  $J' = 47$ . En cuanto a la red Jordan  $J = 47$ ,  $K = 20$  y  $J' = 20$ .

### Entrenamiento

Para la utilización de la red neuronal para la estrategia de búsqueda, es necesario entrenarla para su aprendizaje y que adquiera capacidad de predicción, de aquí que, sea necesario el uso de algún algoritmo de entrenamiento. Para el entrenamiento de nuestras redes utilizaremos el algoritmo de aprendizaje supervisado ampliamente conocido, el algoritmo de propagación hacia atrás (Back-Propagation, BP por sus siglas en inglés) para la red Feedforward y BP en el tiempo (BP through time, BPTT) para las redes Elman y Jordan.

El algoritmo BP es un algoritmo de aprendizaje supervisado, por lo que será necesario proveer a la red con pares de datos (entrada y salida deseada), estas salidas esperadas serán comparadas con las salidas arrojadas por la red neuronal, se calculará el error total de la red y se ajustarán los pesos asociados con las conexiones, estos ajustes de pesos se realizan partiendo de la capa de salida hacia la capa de entrada, esto es en el sentido contrario a como entran los datos, este algoritmo es completado cuando el error dado por la red es menor que el umbral establecido, esto es:

$$E(W) = \frac{1}{2} \sum_{k=1}^K (\hat{y}_k - y_k)^2 \leq u \quad (1)$$

Donde  $u$  es el umbral de error establecido,  $\hat{y}_k$  es el valor esperado como salida. A continuación se plantean las ecuaciones para el algoritmo BP. Para cada neurona de la capa de salida tenemos como salida:

$$y_k = f\left(\sum_{j=1}^J (w_{jk}^{(2)} h_j + b_k)\right) \quad (2)$$

Donde  $b_k$  es la polarización o Bias que tiene un valor constante 1 y un peso  $b$ , y  $f(\cdot)$  es la función de activación. Para cada neurona de la capa oculta tenemos como salida:

$$h_j = f\left(\sum_{i=1}^I (w_{ij}^{(1)} x_i + b_j)\right) \quad (3)$$

Para cada neurona de la capa de salida el término de error se calcula como:

$$\delta_k = y_k (1 - y_k) (\hat{y}_k - y_k) \quad (4)$$

Para cada neurona de la capa oculta el error es calculado como:

$$\delta_j = h_j (1 - h_j) \sum_{k=1}^K (w_{jk}^{(2)} \delta_k) \quad (5)$$

Para actualizar los pesos de cada una de las conexiones hacemos lo siguiente:

$$w_{ij}^{(1)}(n+1) = w_{ij}^{(1)}(n) + \eta \delta_j x_{ij}, \quad (6)$$

$$w_{jk}^{(2)}(n+1) = w_{jk}^{(2)}(n) + \eta \delta_k h_{jk} \quad (7)$$

Donde  $\eta$  es la razón de aprendizaje. Para BPTT las ecuaciones dependen del tiempo, se utilizan las mismas ecuaciones, con modificaciones para el cálculo de la función de activación de la capa oculta y la actualización de los pesos para las conexiones. La función de activación de la capa oculta se ve afectada por la capa de contexto, por lo que ahora tenemos:

$$h_j(t) = f\left(\sum_{i=1}^I (w_{ij}^{(1)} x_i(t)) + \sum_{j'}^{J'} (w_{j'j}^{(3)} c_{j'}(t) + b_j)\right) \quad (8)$$

La salida que producen las neuronas de la capa de contexto es:

$$c_{j'}(t) = h_j(t-1) \quad (9)$$

Donde  $j' = j$  que corresponde al número de neuronas de la capa de contexto y de la capa oculta, respectivamente. La actualización de los pesos está dada por:

$$w_{ij}^{(1)}(n+1) = w_{ij}^{(1)}(n) + \sum_{l=1}^L (\eta \delta_j(l) x_{ij}(l)), \quad (10)$$

$$w_{jk}^{(2)}(n+1) = w_{jk}^{(2)}(n) + \sum_{l=1}^L (\eta \delta_k(l) h_{jk}(l)), \quad (11)$$



$$w_{jj}^{(3)}(n+1) = w_{jj}^{(3)}(n) + \sum_{l=1}^L (\eta \delta_j(l) c_{jj}(l)) \quad (12)$$

Donde  $L$  es la longitud de la secuencia.

### Simulación

Para la simulación tomamos como referente al nodo 1, por tanto, el historial de conexiones utilizado contiene trayectorias formadas entre el nodo 1 con el resto de los nodos de la red de comunicaciones, creamos las tres redes neuronales y preparamos los datos acordes a cada uno de los dos casos. Para cada red neuronal se utilizan dos formas distintas de vector de entrada.

El historial de conexiones contiene información recogida en un periodo de 4 semanas, de lunes a viernes, de las 8:00 horas a las 20:00 horas, de la cual se utilizó el 80% de la información para el entrenamiento de la red y posteriormente para la verificación de la misma y el 20% restante fue utilizada para generalizar la red.

Para el entrenamiento de la red se tomo una razón de aprendizaje  $\eta = 0.1$ , valor utilizado en la mayoría de la literatura consultada, y una función de activación sigmoid. También se consideró un número máximo de iteraciones para el entrenamiento, aplicándose cuatro distintos valores para ello, que fueron: 20, 100, 500 y 1000 iteraciones. Cada iteración corresponde a introducir en la red neuronal el total de datos de entrenamiento, esto es el 80% de la información del historial del nodo 1. Cabe recordar los datos de entrenamiento se manejan en pares: entrada y salida deseada, ya que manejamos entrenamiento supervisado.

Para minimizar el error entre la salida obtenida y la deseada, en la red Feedforward el ajuste de sus parámetros (pesos, Bias) se lleva a cabo por cada par de entrenamiento introducido, mientras que en las redes Elman y Jordan este ajuste se lleva a cabo cuando se introduce la secuencia completa, es decir con el último par de entrenamiento de una secuencia cualquiera. Una secuencia corresponde a una trayectoria.

Analizamos a continuación los resultados, con la finalidad de comparar el desempeño de las redes para cada uno de los dos casos propuestos. De acuerdo con [9] la red Elman presenta una alta precisión en la predicción comparada con las redes Feedforward, y por su parte en [2] se sostiene que la red Elman presenta menor precisión en contraste con al red Jordan.

### Resultados experimentales y discusiones

Para determinar que el predictor nos proporciona el número correspondiente al nodo, que será el siguiente nodo a formar parte de la trayectoria, es necesario que por lo menos una de las neuronas de la capa de salida se dispare, esto es, que alguna de las neuronas de como salida un valor mayor que cero; correspondiendo entonces la posición de la neurona que se disparó con el número que identifica al nodo. Cuando varias neuronas se disparan, tomaremos la de valor mayor para determinar el número que corresponde al nodo siguiente.

Estas redes neuronales fueron analizadas bajo dos perspectivas: como predictores de nodos siguientes y como predictores de secuencias.

Como predictor de nodos siguientes, sin considerarlos como elementos de una trayectoria, obtuvimos los siguientes resultados. El resumen de los resultados obtenidos en la etapa de verificación se muestra en la tabla 1. Mientras que los resultados para la etapa de generalización de cada red se resumen en la tabla 2.

Observando estos resultados encontramos que la red Elman ofrece los mejores resultados, lográndose una precisión en la predicción superior al 85%.

**Tabla 1:** Precisión en las predicciones de las redes neuronales multicapas durante la etapa de verificación

Predictor	20	100	500	1000
Feedforward caso1	0.75	0.82	0.83	0.83
Feedforward caso2	0.70	0.77	0.79	0.80
Elman caso 1	0.77	0.83	0.85	0.86
Elman caso 2	0.74	0.81	0.84	0.85
Jordan caso 1	0.76	0.83	0.84	0.83
Jordan caso 2	0.72	0.80	0.81	0.81

**Tabla 2:** Precisión en las predicciones de las redes neuronales multicapas durante la etapa de generalización

Predictor	20	100	500	1000
Feedforward caso1	0.69	0.73	0.73	0.73
Feedforward caso2	0.63	0.67	0.67	0.67
Elman caso 1	0.69	0.73	0.73	0.74
Elman caso 2	0.67	0.70	0.71	0.72
Jordan caso 1	0.69	0.73	0.73	0.72
Jordan caso 2	0.65	0.68	0.69	0.68



Otra observación importante es que la información manejada tal como se describe en el caso 1 (utilizando dos nodos de la secuencia para predecir el siguiente) proporciona mejores resultados para todas las redes que con el acumulado descrito en el caso 2.

Encontramos que se mejora la precisión al pasar del entrenamiento con 20 iteraciones al de 500 iteraciones hasta un 10%, sin embargo más allá de las 500 iteraciones la mejora se vuelve mínima o no la hay. Esto nos lleva a concluir que se ha logrado un ajuste adecuado de los parámetros de la red a las 500 iteraciones.

Por otra parte, desde la perspectiva de la predicción de secuencias a través de las redes, esto es, lograr predecir tanto los nodos como el orden correcto de los nodos a seguir para formar la trayectoria de búsqueda, se logran los siguientes resultados. En la tabla 3 y 4 se resumen los resultados conseguidos.

Los mejores resultados se obtienen nuevamente de las redes recurrentes, con mejores resultados obtenidos a través de la red Elman, pasando el 65% en la precisión de predicción.

Cabe mencionar que los resultados que se obtienen en la etapa de generalización están por debajo de los resultados obtenidos en la etapa de verificación, esto se debe a que en la etapa de generalización utilizamos información para la cual no fue entrenada la red neuronal. Sin embargo, para nuestro estudio, tiene mayor importancia los resultados obtenidos durante la verificación, ya que, esta estrategia de búsqueda de trayectorias es aplicada a escenarios donde los nodos tienden a repetir acciones, por lo que, será más probable que se trate de predecir una situación que ya se presentó con anterioridad. Un nodo se vuelve predecible a medida que su comportamiento se parece cada vez más al paso del tiempo. Por ello, cuando la red nos da una predicción, bajo condiciones similares a las de entrenamiento, la probabilidad de que esta predicción sea precisa es alta.

Respecto al tiempo requerido para el entrenamiento de las redes neuronales, encontramos que el tiempo empleado por la red recurrente Elman es mayor que el requerido por la Feedforward y por la red Jordan, principalmente se debe a la capa de contexto, que incrementa el número de neuronas en la red. Sin embargo, nuestro interés está centrado primordialmente en la capacidad de predicción, y para solventar el problema relacionado con el tiempo, el entrenamiento podría ser llevado a cabo en los tiempos muertos del

dispositivo, en modo *background* o en un servidor y posteriormente el dispositivo obtener la red ya entrenada. En la tabla 5 se resumen los tiempos requeridos para el entrenamiento de las distintas redes.

Finalmente, es importante indicar que, de la misma manera que las personas tienen comportamientos repetitivos, también con el paso del tiempo suelen cambiar estos comportamientos, por lo que es necesario detectar cuando las redes dejan de ser útiles en sus predicciones, ya que cuando la precisión en las predicciones disminuye implica que el nodo está cambiando sus hábitos, por lo que se deberá plantear un nuevo entrenamiento de la red para adecuar las predicciones a estos nuevos cambios.

**Tabla 3:** Precisión en la predicción de secuencias de las redes neuronales multicapas durante la etapa de verificación

Predictor	20	100	500	1000
Feedforward caso1	0.55	0.63	0.63	0.64
Feedforward caso2	0.51	0.57	0.59	0.60
Elman caso 1	0.56	0.64	0.65	0.67
Elman caso 2	0.55	0.62	0.65	0.66
Jordan caso 1	0.56	0.63	0.65	0.65
Jordan caso 2	0.52	0.60	0.62	0.62

**Tabla 4:** Precisión en la predicción de secuencias de las redes neuronales multicapas durante la etapa de generalización

Predictor	20	100	500	1000
Feedforward caso1	0.50	0.54	0.55	0.55
Feedforward caso2	0.47	0.50	0.51	0.51
Elman caso 1	0.50	0.55	0.55	0.56
Elman caso 2	0.49	0.52	0.55	0.56
Jordan caso 1	0.50	0.54	0.54	0.55
Jordan caso 2	0.46	0.51	0.52	0.53

**Tabla 5:** Comparación del tiempo requerido para el entrenamiento de las redes neuronales, tiempo expresado en segundos

Predictor	20	100	500	1000
Feedforward caso1	29.30	151.30	796.21	1,477.63
Feedforward caso2	17.46	85.42	467.54	916.69
Elman caso 1	37.46	187.73	1,036.26	1,884.47
Elman caso 2	20.05	99.11	518.13	1,378.96
Jordan caso 1	28.49	154.17	770.72	1,445.68
Jordan caso 2	16.53	82.32	447.36	877.19



### Conclusiones y trabajos futuros

Los nuevos entornos de comunicación están caracterizados por la ausencia de infraestructuras fijas debido al constante movimiento de los nodos, por lo que la comunicación entre dos dispositivos móviles se vuelve un problema de búsqueda de trayectorias.

Si consideramos que estos nodos pueden ser dispositivos manipulados por personas y que por naturaleza, una persona tiende a exhibir comportamientos repetitivos, el movimiento de estos nodos, entonces, describirá ciertos patrones, que con el tiempo podrían ser predecibles, por ello la estrategia de búsqueda puede ser atacada a través de la explotación de los historiales generados por los propios nodos y con herramientas adecuadas realizar predicciones con esta información para construir las trayectorias.

De acuerdo a los resultados obtenidos de una capacidad de hasta 85% en la predicción podemos decir que es posible la utilización de las redes neuronales para el diseño de estrategias de búsqueda de trayectoria en entornos dinámicos. Por sus resultados, proponemos utilizar las redes recurrentes con ventana de tiempo de 2 nodos.

Para el tiempo de entrenamiento, como trabajos futuros, podríamos considerar el tener una red neuronal para cada nodo con el cual se ha establecido conexión. Ya que se requiere re-entrenar una red neuronal cuando se cuenta con nueva información o que la precisión en la predicción sea tan baja que no contribuye en la optimización de la búsqueda de trayectorias de comunicación.

### Referencias

- [1] J. Heaton, *Introduction to Neural Networks with Java*, Second Edition, Heaton Research Inc., 2008.
- [2] J. Skaruz and F. Seredynski, "Recurrent neural networks towards detection of SQL attacks," *Parallel and Distributed Processing Symposium*, 2007. IPDPS 2007, IEEE International, pp. 1-8, 26-30 March 2007.
- [3] S. Şeker, E. Ayaz and E. Türkcan, "Elman's recurrent neural network applications to condition monitoring in nuclear power plant and rotatoring machinery," *Engineering Applications of Artificial Intelligence*, vol. 16, pp. 647-656, 2003.
- [4] Y. Kwon, S. Choi and B. Moon, "Stock prediction based on financial correlation," In *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation* (Washington DC, USA, June 25 - 29, 2005). H. Beyer, Ed. GECCO '05. ACM, New York, NY, pp. 2061-2066, 2005.
- [5] L. Vintan, A. Gellert, J. Petzold and T. Ungerer, "Person Movement Prediction Using Neural Networks," *Workshop on*

*Modeling and Retrieval of Context 2004*, MRC 2004, located at the 27th German Conference on Artificial Intelligence, Ulm, Germany, September 2004.

- [6] I. M. Galván and P. Isasi, "Multi-step Learning Rule for Recurrent Neural Models: An Application to Time Series Forecasting," *Neural Process. Letters*, vol. 13, no. 2, pp. 115-133, April 2001
- [7] A. Nogueira, P. Salvador, and R. Valadas, "Predicting the quality of service of wireless LANs using neural networks," In *Proceedings of the 9th ACM international Symposium on Modeling Analysis and Simulation of Wireless and Mobile Systems* (Terromolinos, Spain, October 02 - 06, 2006). MSWiM '06. ACM, New York, NY, pp. 52-60, 2006.
- [8] V. Alarcon-Aquino and J. A. Barria, "Multiresolution FIR neural-network-based learning algorithm applied to network traffic prediction," *Systems, Man, and Cybernetics, Part C: Applications and Reviews*, IEEE Transactions on, vol. 36, no. 2, pp. 208-220, March 2006.
- [9] J. Petzold, F. Bagci, T. Wolfgang and T. Ungerer, "Comparison of Different Methods for Next Location Prediction," *European Conference on Parallel Computing*, Euro-Par 2006, Dresden, Germany, August/September 2006.

### Currículo corto de los autores

*José-Alfredo Abad Padilla* es Ingeniero en Computación graduado de la Universidad Autónoma de Baja California en 1995, obtuvo la especialidad en Electrónica y Telecomunicaciones en 2000. Obtuvo el grado de Maestría en Ciencias en Electrónica y Telecomunicaciones en el Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE) en 2001. Actualmente es candidato al grado de Doctor en Ingeniería Telemática en la Universidad Politécnica de Cataluña. Sus áreas de interés actualmente están enfocadas en diseño de servicios telemáticos, sistemas móviles, redes neuronales, ontologías.

*Juan-Luis Gorricho Moreno* es Ingeniero de Telecomunicaciones por la Universidad Politécnica de Cataluña desde 1994, obtuvo el grado de Doctor Ingeniero por la misma Universidad en 1999. Sus áreas de investigación se centran en los aspectos relacionados con el diseño de nuevos servicios telemáticos, con especial interés en los servicios basados en el uso de dispositivos móviles y orientados hacia la explotación de la información sensible al contexto de operación del usuario.