

Práctica de redes neuronales para predicción

Daniel González Alonso

16 de enero de 2017

Resumen

En este documento se describen las fases hacia adelante y hacia atrás de las redes de Elman y Jordan, así como la evolución de la tasa de aciertos en la predicción de la cotización de Iberdrola para la práctica de redes neuronales para predicción de la asignatura Minería de Datos de Ingeniería Informática, Universidad de Valladolid.

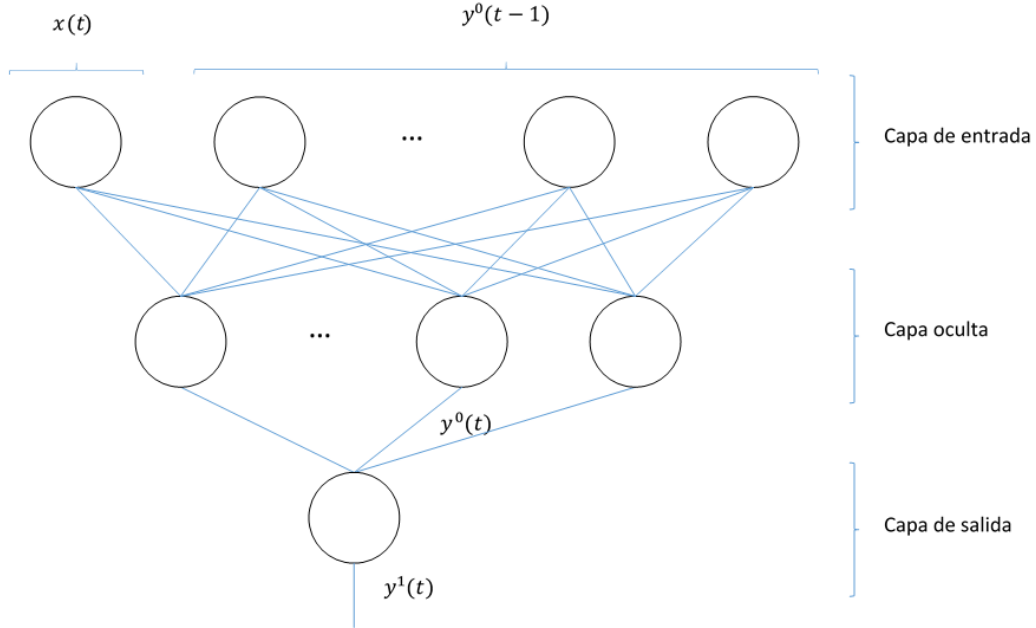
1. INTRODUCCIÓN

En este documento primero se describirá la Red de Elman y después la Red de Jordan, en ambos casos se incluirá apartados con la fase hacia adelante y hacia atrás así como los resultados obtenidos en la implementación de estas redes con Octave.

Antes de empezar voy a aclarar las posibles dudas que haya con la notación utilizada:

- Para la función de activación emplearé siempre el símbolo $F(x)$, en mi caso se ha empleado la función sigmoide $F(x) = \frac{1}{1+e^{-x}}$ tanto para la capa de salida como para la capa oculta.
- El vector $y(t)$ siempre se va a referir a los valores obtenidos por una de las capas de la red neuronal en el instante t , mientras que $x(t)$ representará a los valores de entrada a la red neuronal en el instante t .
- NI representará el número de neuronas de entrada, $N0$ el número de neuronas de la capa oculta, y $N1$ representará el número de neuronas de la capa de salida.
- $w_{i,j}$ se refiere al peso de la conexión entre una neurona j a la neurona i .
- Un superíndice 1, ya sea en $y(t)$ o en $w_{i,j}$, siempre se referirá a la capa de salida, mientras que un superíndice 0 se referirá a la capa oculta.

2. RED DE ELMAN



2.1. FASE HACIA DELANTE

En esta fase se calculará las salidas de la red neuronal de Elman en el instante t . Hay que tener en cuenta de que tratamos de una serie temporal en la que dependemos de los τ valores anteriores y del valor actual de $x_i(t)$ (solo un valor de x) para predecir el siguiente valor de $y_i^1(t)$ (solo un valor de y^1). Como en la red implementada solo hay dos capas tal y como se ve en el dibujo anterior, voy a dividir los cálculos de esta fase en la capa de salida y en la capa oculta.

2.1.1. CAPA DE SALIDA

Los valores obtenidos por la capa de salida en la iteración t se calculan con la siguiente fórmula:

$$y_i^1(t) = F(u_i^1(t)) \quad (1)$$

Donde $y_i^1(t)$ representa el valor obtenido por la neurona i de la capa de salida en el instante t . Por otro lado $u_i^1(t)$ representa el sumatorio de los valores de entrada de la neurona i después de aplicar sus respectivos pesos, el cual se calcula mediante la siguiente fórmula:

$$u_i^1(t) = \sum_{j=1}^{N0} w_{i,j}^1 \cdot y_j^0(t) + w_{i,N0+1}^1 \quad (2)$$

Donde $w_{i,j}^1$ representa el peso de la conexión de la neurona j de la capa oculta a la neurona i de la capa de salida, e $y_j^0(t)$ representa la salida de la neurona j de la capa oculta en el instante t . El último término $w_{i,N0+1}^1$ representa el peso del factor *bias* de la red neuronal.

2.1.2. CAPA OCULTA

Los valores obtenidos por la capa oculta en la iteración t se calculan con la siguiente fórmula:

$$y_j^0(t) = F(u_j^0(t)) \quad (3)$$

Donde $y_j^0(t)$ representa el valor obtenido por la neurona j de la capa oculta en el instante t , $u_j^0(t)$ es la suma de los valores de entrada de la neurona j junto con los valores obtenidos en la iteración anterior después de aplicar sus respectivos pesos, el cual se calcula mediante la siguiente fórmula:

$$u_j^0(t) = \sum_{k=1}^{NI} w_{j,k}^0 \cdot x_k(t) + \sum_{k=NI+1}^{NI+N0} w_{j,k}^0 \cdot y_{k-NI}^0(t-1) + w_{j,NI+N0+1} \quad (4)$$

Donde $w_{j,k}^0$ en el primer sumatorio representa el peso de la conexión de la neurona k de la capa de entrada a la neurona j de la capa oculta, $x_k(t)$ representa la entrada en el instante actual, $w_{j,k}^0$ en el segundo sumatorio es el peso de la conexión entre la neurona k con la neurona j , ambas de la capa oculta, y por último $y_{k-NI}^0(t-1)$ representa la salida de la neurona $k - NI$ de la capa oculta en el instante $t - 1$ (la iteración anterior).

2.2. FASE HACIA ATRÁS

El objetivo de esta fase es calcular las variaciones que han de llevarse a cabo sobre cada peso de la red neuronal a partir de una función de error $E(t)$. Esta función de error en mi caso la he calculado como la distancia euclídea entre la salida esperada para una entrada y la predicha en el instante anterior, con lo cual la función es la siguiente:

$$E(t) = \frac{1}{2} \sum_{i=1}^{N1} (x_i(t) - y_i^1(t-1))^2 \quad (5)$$

Para el calculo de de la variación que ha de sufrir cada peso, se emplea la siguiente fórmula:

$$\Delta w_{i,j}(t) = -\alpha \cdot \frac{\partial E(t)}{\partial w_{i,j}} \quad (6)$$

Donde α es el ratio de aprendizaje elegido. Para calcular $\Delta w_{i,j}(t)$ vamos a dividir esta fase de nuevo en la capa de salida y la capa oculta.

2.2.1. CAPA DE SALIDA

Para calcular la variación en los pesos de la neurona i de la capa de salida, $\Delta w_{i,j}^1(t)$, primero hemos de calcular $\frac{\partial E(t)}{\partial w_{i,j}^1}$. Aplicando la regla de la cadena se obtiene:

$$\frac{\partial E(t)}{\partial w_{i,j}^1} = \frac{\partial E(t)}{\partial y_i^1(t-1)} \cdot \frac{\partial y_i^1(t-1)}{\partial u_i^1(t-1)} \cdot \frac{\partial u_i^1(t-1)}{\partial w_{i,j}^1} \quad (7)$$

Donde el primer factor se calcula como:

$$\frac{\partial E(t)}{\partial y_i^1(t-1)} = \frac{\partial}{\partial y_i^1(t-1)} \left(\frac{1}{2} (x_i(t) - y_i^1(t-1))^2 \right) = -(x_i(t) - y_i^1(t-1)) \quad (8)$$

El segundo lo dejamos indicado como la derivada $F'(u_i^1(t-1))$. Combinando tanto el primer factor como este segundo, tenemos como resultado la función:

$$\delta_i^1(t) = -(x_i(t) - y_i^1(t-1)) \cdot F'(u_i^1(t-1)) \quad (9)$$

Por último, el tercer factor es:

$$\frac{\partial u_i^1(t-1)}{\partial w_{i,j}^1} = \begin{cases} y_j^0(t-1) & \text{si } j = 1, \dots, N0 \\ 1 & \text{si } j = N0 + 1 \end{cases} \quad (10)$$

Finalmente, para obtener la variación que hemos de llevar a cabo sobre el peso, $\Delta w_{i,j}^1$, mediante los resultados anteriores y la fórmula 6, obtenemos:

$$\Delta w_{i,j}^1(t) = -\alpha \cdot \frac{\partial E(t)}{\partial w_{i,j}^1} = \begin{cases} -\alpha \cdot \delta_i^1(t) \cdot y_j^0(t-1) & \text{si } j = 1, \dots, N0 \\ -\alpha \cdot \delta_i^1(t) & \text{si } j = N0 + 1 \end{cases} \quad (11)$$

2.2.2. CAPA OCULTA

Para calcular la variación en los pesos de la neurona j de la capa oculta, $\Delta w_{j,k}^0(t)$, primero hemos de calcular $\frac{\partial E(t)}{\partial w_{j,k}^0}$. Una vez más, aplicando la regla de la cadena se obtiene:

$$\frac{\partial E(t)}{\partial w_{j,k}^0} = \frac{\partial E(t)}{\partial y_j^0(t-1)} \cdot \frac{\partial y_j^0(t-1)}{\partial u_j^0(t-1)} \cdot \frac{\partial u_j^0(t-1)}{\partial w_{j,k}^0} \quad (12)$$

El primer factor lo obtenemos aplicando de nuevo la regla de la cadena:

$$\frac{\partial E(t)}{\partial y_j^0(t-1)} = \frac{\partial E(t)}{\partial y_i^1(t-1)} \cdot \frac{\partial y_i^1(t-1)}{\partial u_i^1(t-1)} \cdot \frac{\partial u_i^1(t-1)}{\partial y_j^0(t-1)} \quad (13)$$

Donde el primer y segundo factor se habían calculado anteriormente en la capa de salida como $\delta_i^1(t)$ y el último da simplemente $w_{i,j}^1$.

Por otro lado, el segundo factor de la fórmula 12 da como resultado $F'(u_j^0(t-1))$, si combinamos todo lo calculado hasta ahora tenemos:

$$\delta_j^0(t) = \delta_i^1(t) \cdot w_{i,j}^1 \cdot F'(u_j^0(t-1)) \quad (14)$$

Para el último factor de la fórmula 12, tenemos que:

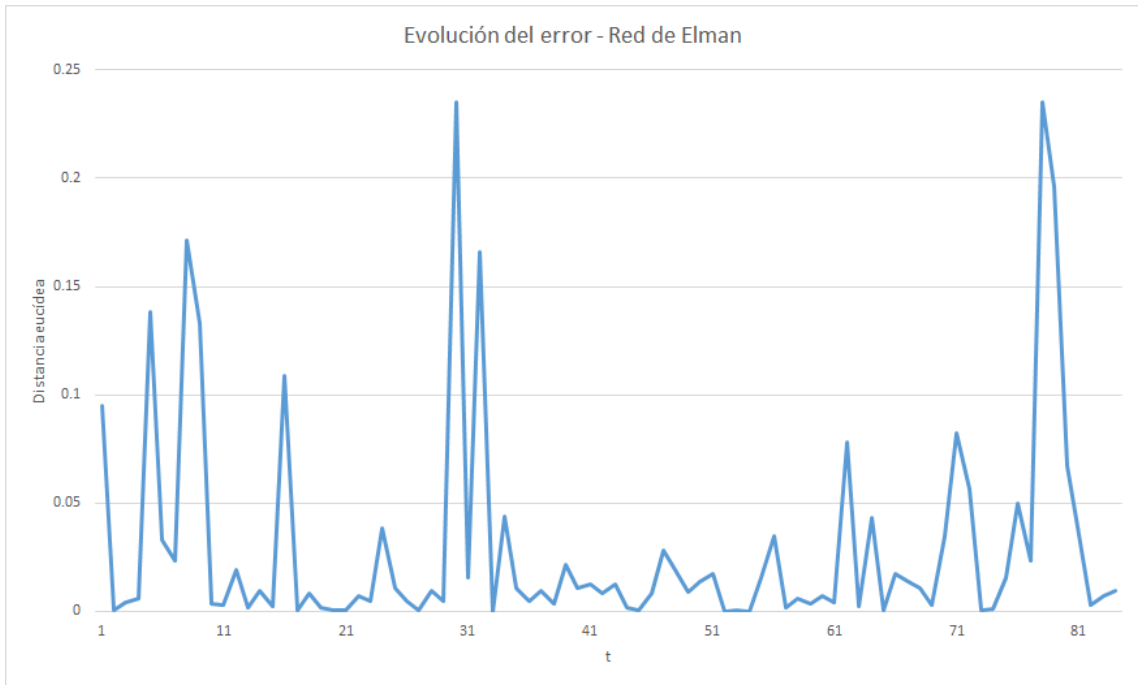
$$\frac{\partial u_j^0(t-1)}{\partial w_{j,k}^0} = \begin{cases} x_k(t-1) & \text{si } k = 1, \dots, NI \\ y_{k-NI}^0(t-2) & \text{si } k = NI + 1, \dots, NI + N0 \\ 1 & \text{si } k = NI + N0 + 1 \end{cases} \quad (15)$$

Finalmente, si lo juntamos todo para obtener la variación que hay que realizar sobre el peso $\Delta w_{j,k}^0$, obtenemos:

$$\Delta w_{j,k}^0 = -\alpha \cdot \frac{\partial E}{\partial w_{j,k}^0} = \begin{cases} -\alpha \cdot \delta_j^0(t) \cdot x_k(t-1) & \text{si } k = 1, \dots, NI \\ -\alpha \cdot \delta_j^0(t) \cdot y_{k-NI}^0(t-2) & \text{si } k = NI + 1, \dots, NI + N0 \\ -\alpha \cdot \delta_j^0(t) & \text{si } k = NI + N0 + 1 \end{cases} \quad (16)$$

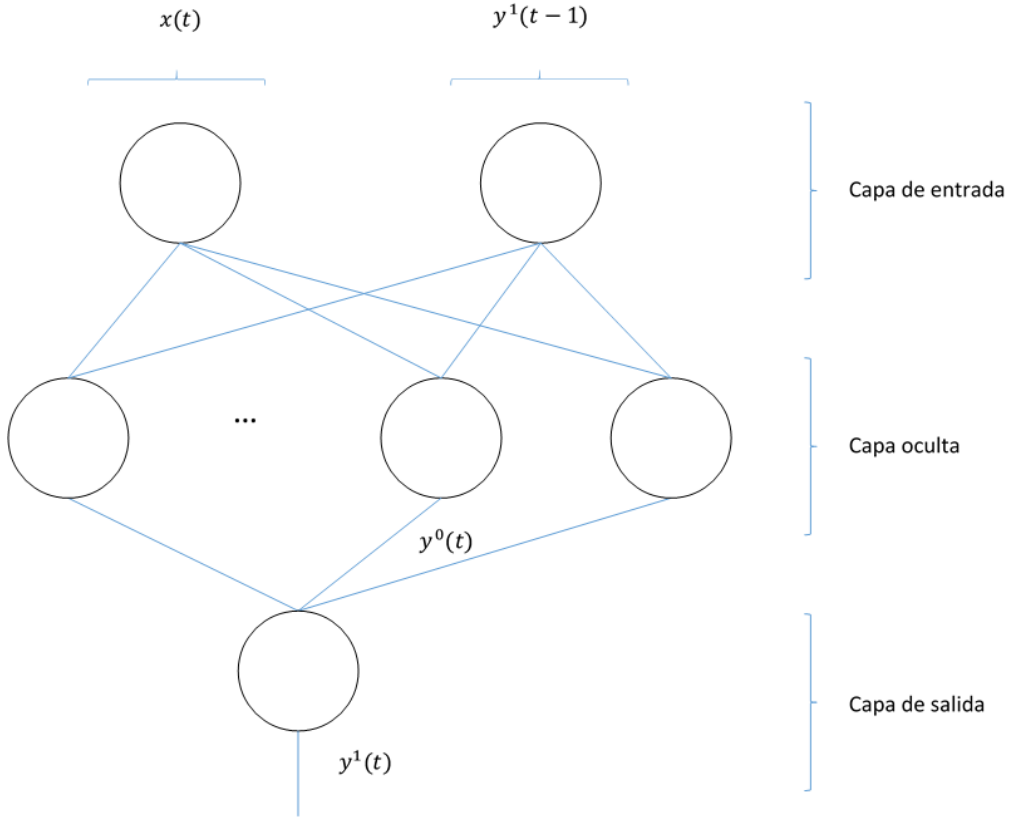
2.3. RESULTADOS OBTENIDOS

El código de la red de Elman está implementado en el archivo `red_elman.m` que acompaña a este texto, este fichero fue programado en Octave 4.0.3. Los datos empleados se encuentran en el archivo `iberdrola_Nov15-Dic16.csv` y para la implementación se dividieron mediante *Hold-Out* (2/3 entrenamiento - 1/3 prueba). La evolución de los resultados que se han ido obteniendo para el conjunto de prueba a lo largo de las iteraciones ha sido:



En este caso, teniendo en cuenta que un valor predicho se consideraba acierto si el error era inferior a 0.05, podemos concluir que más de un 75% de los valores predichos son aciertos.

3. RED DE JORDAN



3.1. FASE HACIA DELANTE

En esta fase se calculará las salidas de la red neuronal de Jordan en el instante t . En este caso tratamos de una serie temporal en la que dependemos únicamente del valor anterior $y_i^1(t-1)$ y de el valor actual de $x_i(t)$ (solo un valor de x) para predecir el siguiente valor de $y_i^1(t)$. Al igual que en la red de Elman, como en la red implementada solo hay dos capas tal y como se ve en el dibujo anterior, voy a dividir los cálculos de esta fase en la capa de salida y en la capa oculta.

3.1.1. CAPA DE SALIDA

Los valores obtenidos por la capa de salida en la iteración t se calculan con la siguiente fórmula:

$$y_i^1(t) = F(u_i^1(t)) \quad (17)$$

Donde $y_i^1(t)$ representa el valor obtenido por la neurona i de la capa de salida en el instante t . Por otro lado $u_i^1(t)$ representa el sumatorio de los valores de entrada de la neurona i después de aplicar sus respectivos pesos, el cual se calcula mediante la siguiente fórmula:

$$u_i^1(t) = \sum_{j=1}^{N0} w_{i,j}^1 \cdot y_j^0(t) + w_{i,N0+1}^1 \quad (18)$$

Donde $w_{i,j}^1$ representa el peso de la conexión de la neurona j de la capa oculta a la neurona i de la capa de salida, e $y_j^0(t)$ representa la salida de la neurona j de la capa oculta en el instante t . El último término $w_{i,N0+1}^1$ representa el peso del factor *bias* de la red neuronal.

3.1.2. CAPA OCULTA

Los valores obtenidos por la capa oculta en la iteración t se calculan con la siguiente fórmula:

$$y_j^0(t) = F(u_j^0(t)) \quad (19)$$

Donde $y_j^0(t)$ representa el valor obtenido por la neurona j de la capa oculta en el instante t , $u_j^0(t)$ es la suma de los valores de entrada de la neurona j junto los valores obtenidos en la iteración anterior en la capa de salida después de aplicar sus respectivos pesos, el cual se calcula mediante la siguiente fórmula:

$$u_j^0(t) = \sum_{k=1}^{NI} w_{j,k}^0 \cdot x_k(t) + \sum_{k=NI+1}^{NI+N1} w_{j,k}^0 \cdot y_{k-NI}^1(t-1) + w_{j,NI+N0+1} \quad (20)$$

Donde $w_{j,k}^0$ en el primer término representa el peso de la conexión de la neurona k de la capa de entrada a la neurona j de la capa oculta, $x_k(t)$ representa la entrada en el instante actual, en el segundo sumatorio, $w_{j,k}^0$ es el peso de la conexión entre la neurona k de la capa de salida con la neurona j de la capa oculta, y por último $y_{k-NI}^1(t-1)$ representa la salida de la neurona $k - NI$ de la capa de salida en el instante $t - 1$ (la iteración anterior).

3.2. FASE HACIA ATRÁS

El objetivo de esta fase es calcular las variaciones que han de llevarse a cabo sobre cada peso de la red neuronal a partir de una función de error $E(t)$. Esta función de error se ha vuelto a calcular como la distancia euclídea entre la salida esperada para una entrada y la predicha en el instante anterior, con lo cual la función es la siguiente:

$$E(t) = \frac{1}{2} \sum_{i=1}^{N1} (x_i(t) - y_i^1(t-1))^2 \quad (21)$$

Para el cálculo de la variación que ha de sufrir cada peso, se emplea la siguiente fórmula:

$$\Delta w_{i,j}(t) = -\alpha \cdot \frac{\partial E(t)}{\partial w_{i,j}} \quad (22)$$

Donde α es el ratio de aprendizaje elegido. Para calcular $\Delta w_{i,j}(t)$ vamos a dividir esta fase de nuevo en la capa de salida y la capa oculta.

3.2.1. CAPA DE SALIDA

Para calcular la variación en los pesos de la neurona i de la capa de salida, $\Delta w_{i,j}^1(t)$, primero hemos de calcular $\frac{\partial E(t)}{\partial w_{i,j}^1}$. Aplicando la regla de la cadena se obtiene:

$$\frac{\partial E(t)}{\partial w_{i,j}^1} = \frac{\partial E(t)}{\partial y_i^1(t-1)} \cdot \frac{\partial y_i^1(t-1)}{\partial u_i^1(t-1)} \cdot \frac{\partial u_i^1(t-1)}{\partial w_{i,j}^1} \quad (23)$$

Donde el primer factor se calcula como:

$$\frac{\partial E(t)}{\partial y_i^1(t-1)} = \frac{\partial}{\partial y_i^1(t-1)} \left(\frac{1}{2} (x_i(t) - y_i^1(t-1))^2 \right) = -(x_i(t) - y_i^1(t-1)) \quad (24)$$

El segundo lo dejamos indicado como la derivada $F'(u_i^1(t-1))$. Combinando tanto el primer factor como este segundo, tenemos como resultado la función:

$$\delta_i^1(t) = -(x_i(t) - y_i^1(t-1)) \cdot F'(u_i^1(t-1)) \quad (25)$$

Por último, el tercer factor es:

$$\frac{\partial u_i^1(t-1)}{\partial w_{i,j}^1} = \begin{cases} y_j^0(t-1) & \text{si } j = 1, \dots, N0 \\ 1 & \text{si } j = N0 + 1 \end{cases} \quad (26)$$

Finalmente, para obtener la variación que hemos de llevar a cabo sobre el peso, $\Delta w_{i,j}^1$, mediante los resultados anteriores y la fórmula 22, obtenemos:

$$\Delta w_{i,j}^1(t) = -\alpha \cdot \frac{\partial E(t)}{\partial w_{i,j}^1} = \begin{cases} -\alpha \cdot \delta_i^1(t) \cdot y_j^0(t-1) & \text{si } j = 1, \dots, N0 \\ -\alpha \cdot \delta_i^1(t) & \text{si } j = N0 + 1 \end{cases} \quad (27)$$

3.2.2. CAPA OCULTA

Para calcular la variación en los pesos de la neurona j de la capa oculta, $\Delta w_{j,k}^0(t)$, primero hemos de calcular $\frac{\partial E(t)}{\partial w_{j,k}^0}$. Una vez más, aplicando la regla de la cadena se obtiene:

$$\frac{\partial E(t)}{\partial w_{j,k}^0} = \frac{\partial E(t)}{\partial y_j^0(t-1)} \cdot \frac{\partial y_j^0(t-1)}{\partial u_j^0(t-1)} \cdot \frac{\partial u_j^0(t-1)}{\partial w_{j,k}^0} \quad (28)$$

El primer factor lo obtenemos aplicando de nuevo la regla de la cadena:

$$\frac{\partial E(t)}{\partial y_j^0(t-1)} = \frac{\partial E(t)}{\partial y_i^1(t-1)} \cdot \frac{\partial y_i^1(t-1)}{\partial u_i^1(t-1)} \cdot \frac{\partial u_i^1(t-1)}{\partial y_j^0(t-1)} \quad (29)$$

Donde el primer y segundo factor se habían calculado anteriormente en la capa de salida como $\delta_i^1(t)$ y el último da simplemente $w_{i,j}^1$.

Por otro lado, el segundo factor de la fórmula 28 da como resultado $F'(u_j^0(t-1))$, si combinamos todo lo calculado hasta ahora tenemos:

$$\delta_j^0(t) = \delta_i^1(t) \cdot w_{i,j}^1 \cdot F'(u_j^0(t-1)) \quad (30)$$

Para el último factor de la fórmula 28, tenemos que:

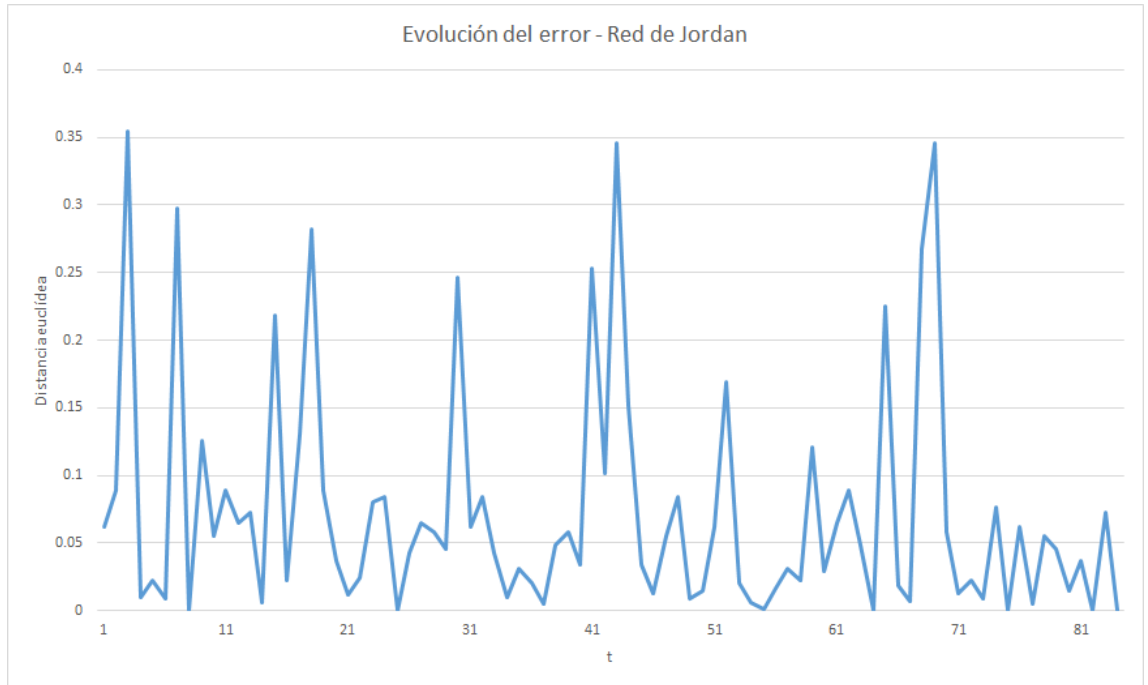
$$\frac{\partial u_j^0(t-1)}{\partial w_{j,k}^0} = \begin{cases} x_k(t-1) & \text{si } k = 1, \dots, NI \\ y_{k-NI}^1(t-2) & \text{si } k = NI + 1, \dots, NI + N1 \\ 1 & \text{si } k = NI + N1 + 1 \end{cases} \quad (31)$$

Finalmente, si lo juntamos todo para obtener la variación que hay que realizar sobre el peso $\Delta w_{j,k}^0$, obtenemos:

$$\Delta w_{j,k}^0 = -\alpha \cdot \frac{\partial E}{\partial w_{j,k}^0} = \begin{cases} -\alpha \cdot \delta_j^0(t) \cdot x_k(t-1) & \text{si } k = 1, \dots, NI \\ -\alpha \cdot \delta_j^0(t) \cdot y_{k-NI}^1(t-2) & \text{si } k = NI + 1, \dots, NI + N1 \\ -\alpha \cdot \delta_j^0(t) & \text{si } k = NI + N1 + 1 \end{cases} \quad (32)$$

3.3. RESULTADOS OBTENIDOS

El código de la red de Jordan está implementado en el archivo `red_jordan.m` que acompaña a este texto, este fichero fue programado en Octave 4.0.3. Los datos empleados fueron los mismos que para la red de Elman, y también fueron divididos mediante *Hold-Out*. La evolución de los resultados que se han ido obteniendo para el conjunto de prueba a lo largo de las iteraciones ha sido:



En este caso, teniendo en cuenta que un valor predicho se consideraba acierto si el error era inferior a 0.05, podemos concluir que aproximadamente un 50 % de los valores predichos son aciertos.