



ESCUELA SUPERIOR DE INGENIERÍA

Aprendizaje Computacional

**Práctica 6:**  
**Redes Neuronales Recurrentes**

Rafael Gutiérrez Corrales

Finn-Henri Smidt

Luke Voss

# 1. Introducción

Las redes feed-forward no tienen memoria debido a su topología. Al admitir conexiones de entrada y salida de una capa a otra en ambas direcciones, logramos redes neuronales con memoria. Estas son las Redes Neuronales Recurrentes (RNNs: Recurrent Neural Networks). Gracias a que las conexiones de este tipo de redes pueden ser dirigidas o bidireccionales, podemos construir redes capaces de completar patrones incompletos o ruidosos.

En los últimos años se han vuelto populares las redes LSTM (Long/Short Term Memory), que responden a una arquitectura de RNN para diversas aplicaciones. Esta red introdujo el concepto de la celda de memoria, distanciándose de las arquitecturas de redes neuronales basadas en neuronas. La celda de memoria posee la capacidad de conservar su valor durante un tiempo más o menos extenso dependiendo de sus entradas, lo que permite a la celda no solo conservar el último valor procesado, sino mantener los valores más relevantes. Las redes LSTM revolucionaron el reconocimiento de voz, superando a los modelos tradicionales, por ello se emplean en sistemas que usamos diariamente, como los smartphones. También son excepcionales para el reconocimiento de escritura a mano y para la traducción automática.

Posteriormente se introdujo una simplificación de las redes LSTM, las redes GRU (Gated Recurrent Unit). Al ser más simples que las redes LSTM, pueden ser entrenadas con mayor velocidad y generalmente llegan a ser más eficientes. No obstante, las redes LSTM son más precisas, ofreciendo resultados mejores y más fiables. Sus principales campos de uso son el modelado de música polifónica y el modelado de señales de voz.

# 2. Metodología

Hemos llevado a cabo la implementación de una RNN Hopfield y de una RNN BAM. Para la comprobación de su correcto funcionamiento hemos diseñado una serie de patrones sencillos cuyos elementos pueden tomar valores 1 ó -1, también hemos implementado una función que transforma los patrones en matrices para, a continuación, dibujar los patrones en pantalla y así poder observar visualmente los resultados. En ambos casos se ha considerado un umbral de 0.

### 3. Resultados

Las figuras 1, 2 y 3 corresponden a los patrones guardados en la red Hopfield. La figura 4 muestra el patrón incompleto utilizado como input y el patrón completo reconocido que la red nos ha proporcionado.

Las figuras 5, 6 y 7 corresponden a las parejas de patrones guardadas en la red BAM. La figura 8 muestra el patrón tomado como input y su pareja correspondiente, resultado del procesamiento.

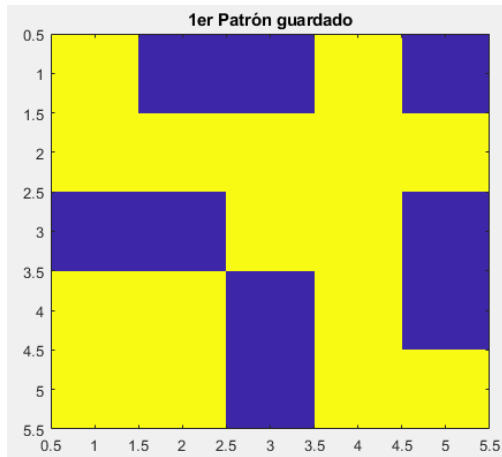


Figura 1: Primer patrón guardado.

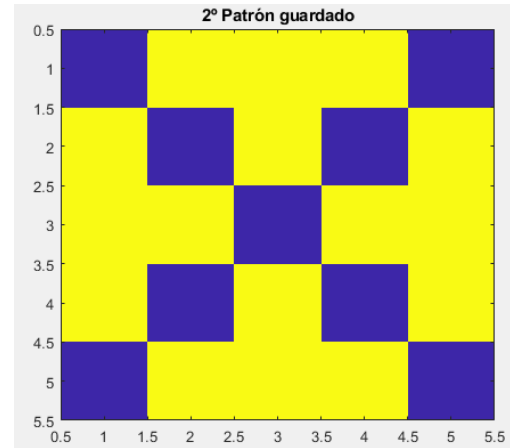


Figura 2: Segundo patrón guardado.

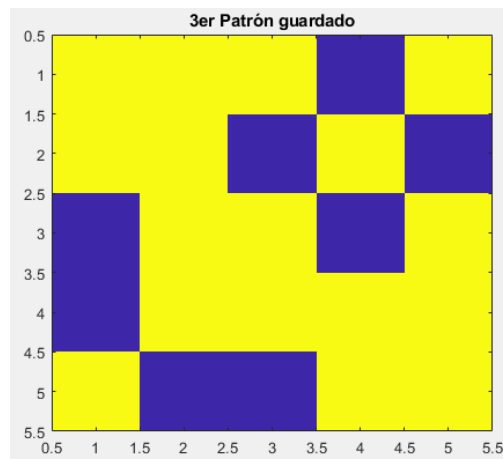


Figura 3: Tercer patrón guardado.

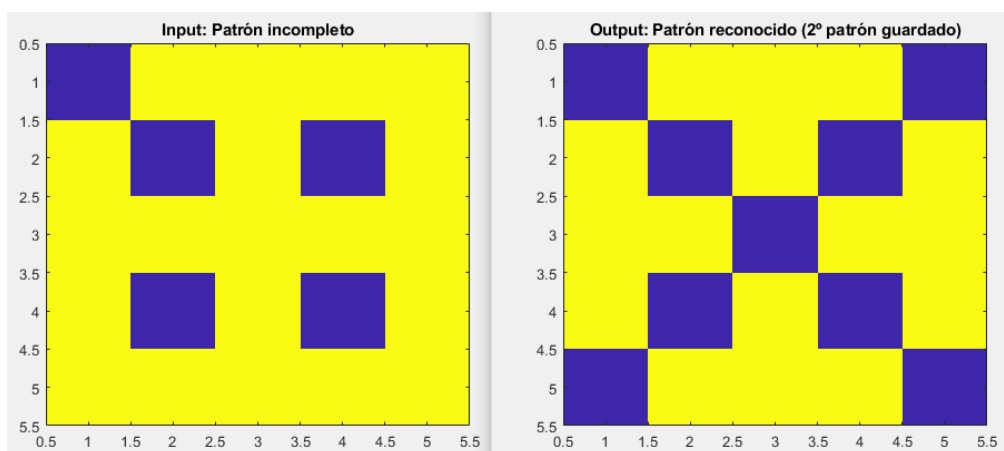


Figura 4: Patrón input incompleto (izqda.) y Patrón output reconocido (dcha.).

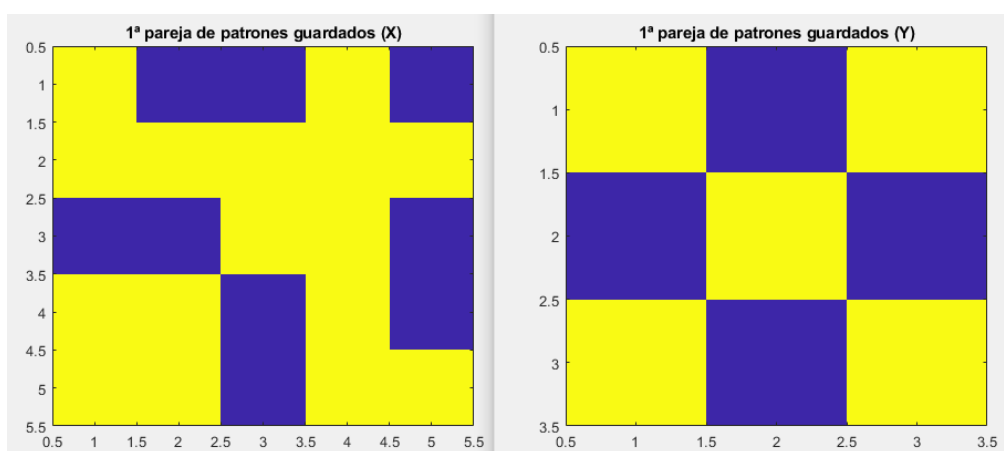


Figura 5: Primera pareja de patrones guardados.

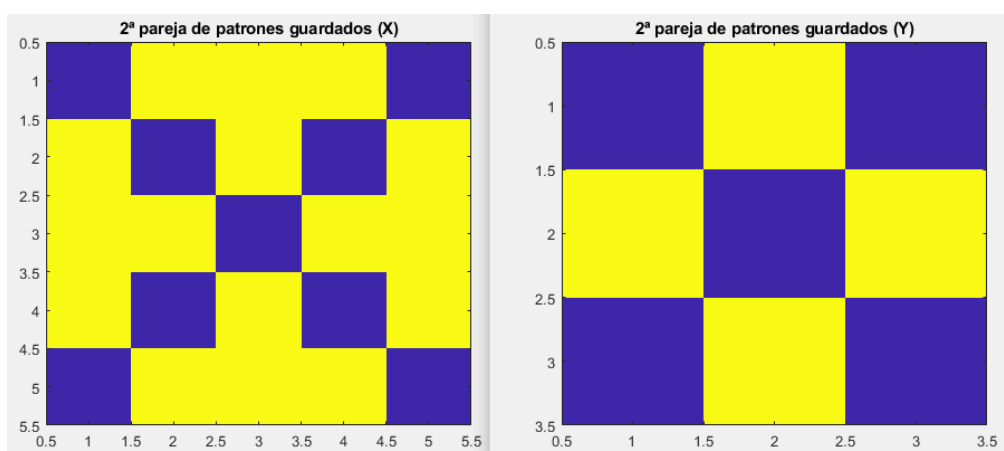


Figura 6: Segunda pareja de patrones guardados.

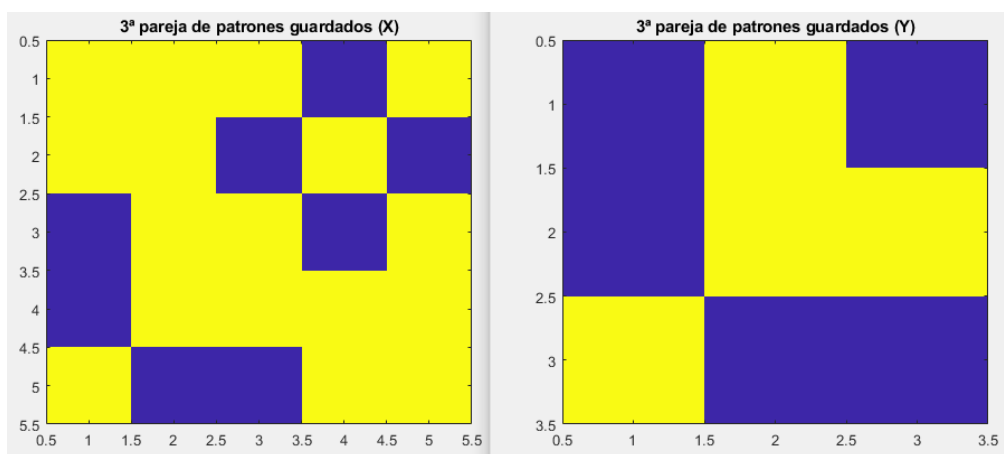


Figura 7: Tercera pareja de patrones guardados.

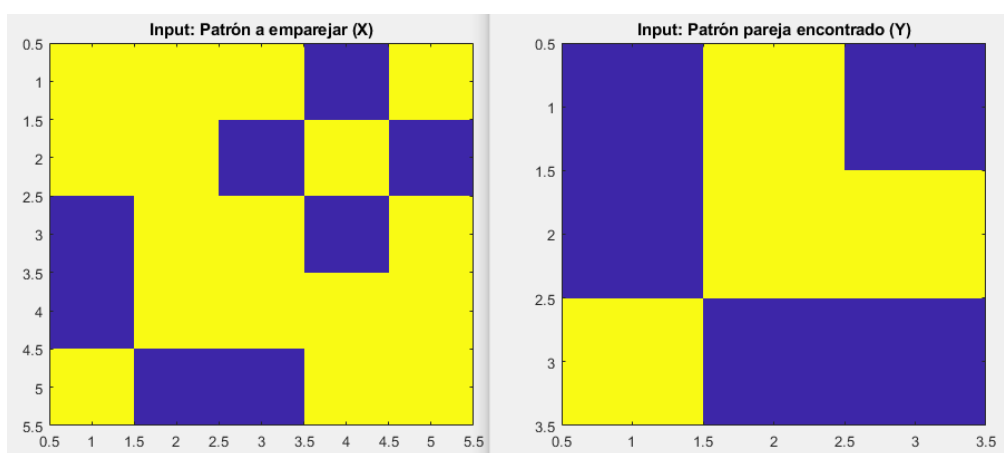


Figura 8: Patrón input a emparejar (izqda.) y Patrón output correspondiente (dcha.).

## 4. Conclusiones

Rafael Gutiérrez Corrales:

Utilizando distintos ejemplos he podido comprobar que no siempre se estabilizan las redes. A veces continúan su ejecución indefinidamente. El tiempo de ejecución está ligado al valor de los patrones que tiene guardados y al input que se desea procesar.

Finn-Henri Smidt:

.

Luke Voss:

.

## 5. Preguntas tipo test

Rafael Gutiérrez Corrales:

- ¿Cuál de estos tipos de redes neuronales es el adecuado para el reconocimiento de firmas?

Finn-Henri Smidt:

- ¿

Luke Voss: