Reporte práctica 12

Red neuronal

Introducción

Para la última práctica se realizará una simulación de aprendizaje a máquina, donde se van a reconocer dígitos de imágenes pequeñas en blanco y negro con una red neuronal. Se usarán quince pixeles por digito, tres de ancho y cinco de altura. Se crearon las imágenes de dígitos de una manera probabilística a partir de plantillas, donde los pixeles que son negros en la plantilla serán puestos casi siempre, los grises de ocasional y los blancos solamente con poca frecuencia.

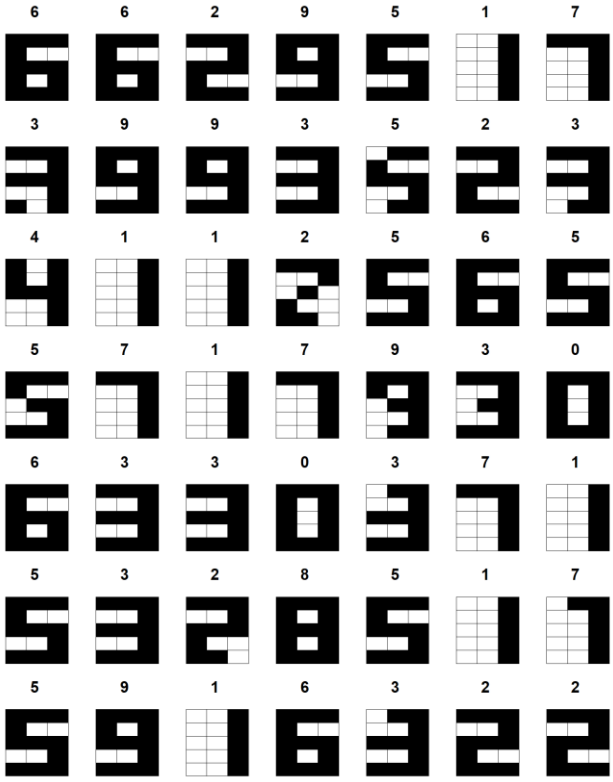

Figura 1. Ejemplo de plantilla de las imágenes de dígitos.

Objetivos

Paralelizar la red neuronal y estudiar el efecto en el tiempo de ejecución con pruebas estadísticas y visualizaciones pertinentes.

Simulación y Resultados

Para poder implementar el uso de clúster y realizar los procedimientos de modo paralelo, se acude a la librería *doParallel*, la cual ya ha sido utilizada anteriormente. Y se crearon dos funciones para agilizar los tiempos de ejecución, las cuales fueron *binario* y *decimal*.

Para conseguir la lectura de nuestra plantilla, fue necesario codificarla en un archivo de datos separados por comas (.csv) que llamamos *digitos.csv*.

Para la demostración del beneficio de paralelizar la simulación, se optó por utilizar múltiples números de pruebas que van desde 500 hasta 2500, teniendo un número de repeticiones de 10 para cada cantidad anterior.

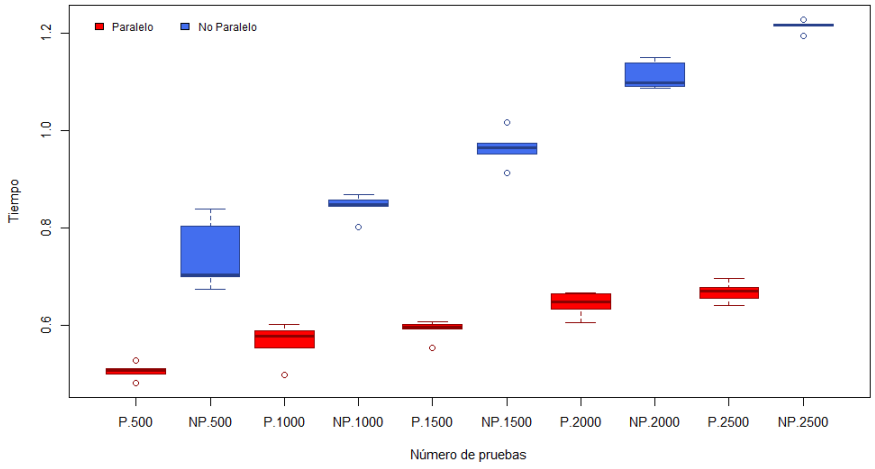

Figura 2. Efecto en el tiempo de ejecución al incrementar el número de pruebas.

Como se puede observar en la figura 2, la simulación en paralelo mantiene una gran ventaja en los tiempos de ejecución llegando en el valor máximo de pruebas alrededor de 0.7 segundos, mientras que si lo comparamos con el código original, este alcanzó los 1.2 segundos en la misma cantidad de pruebas.

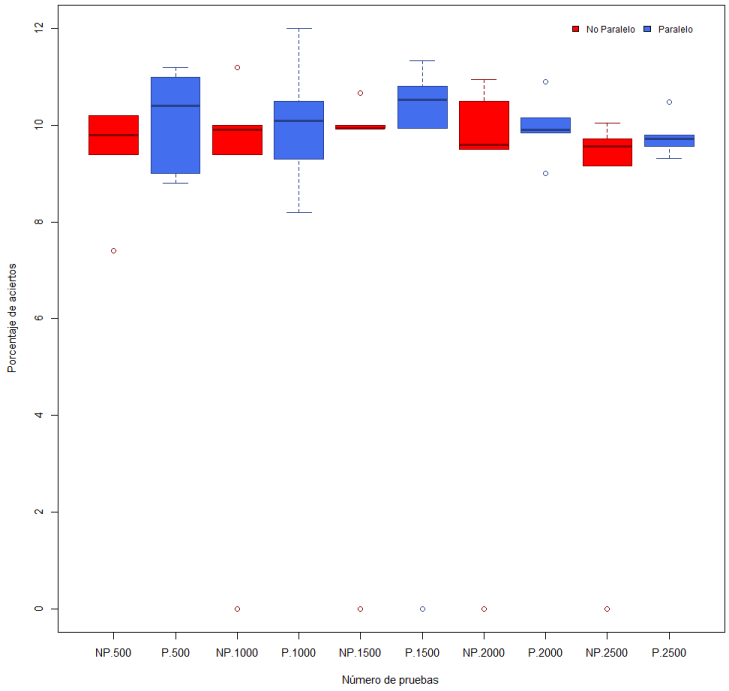

Figura 3. Porcentaje de aciertos de la simulación incrementando el número de pruebas.

En la figura 3, se aprecia que ambos tipos de código tienden básicamente al mismo resultado, aunque es notorio que para cantidades de 1500 de pruebas o menos, el código sin paralelizar muestra ser más constante que el paralelo, mientras que al pasar a 2000 pruebas o más, los papeles cambian, mostrando mayor constancia por parte del código en paralelo.

Conclusiones

Para finalizar se puede decir que se logró modificar el código de forma que se consiguió paralelizar los procedimientos necesarios para que disminuya considerablemente el tiempo de ejecución de cada corrida. Mostrando siempre una gran ventaja el código en paralelo sobre el código original.

Mientras que si verificamos los resultados en base al porcentaje de aciertos, existe una ligera ventaja del cogido original sobre el paralelo, pero solo si se toman cantidades de número de pruebas de 1500 o menores, ya que al aumentar la ventaja la obtiene el código en paralelo.