

# Red neuronal

Francisco Gerardo Meza Fierro

## 1. Introducción

En esta práctica se trabajará con una demostración a máquina donde se reconocerán los diez dígitos creados con una red neuronal en imágenes con escalas de blanco y negro. Estos dígitos se codifican pixel a pixel y la red neuronal intenta descifrar o leer los dígitos codificados para poder decir qué dígito la red cree haber leído.

El objetivo de esta práctica consistirá en paralelizar el código propuesto y ver las diferencias en los tiempos de ejecución que se tienen con respecto al código propuesto y el código paralelizado. Se analizará el desempeño de la red al variar las probabilidades de asignación de generación de los dígitos en la mencionada escala de blancos y negros por pixel; además se implementarán, además de los diez dígitos, doce símbolos nuevos a la red para que los intente descifrar.

## 2. Paralelización del código

El código propuesto cuenta con dos secciones importantes, por así decirlo: una donde se entrena a la red para que sea capaz de descifrar los dígitos (la cual no es posible paralelizar debido a que el valor de las neuronas va cambiando en el proceso, el cual no se realiza de manera independiente) y otra donde realizan las pruebas de la red neuronal (la cual se paralelizará).

Para la paralelización se creó una función donde de manera aleatoria se genera una imagen en pixeles de un dígito y la red lo leerá arrojando un valor verdadero si la red leyó sin problemas el dígito o un valor falso si la red no pudo interpretar el dígito.

Debido a que el proceso toma poco tiempo, se decidió aumentar la cantidad de datos a analizar para comparar si la versión paralelizada del código resultó ser más eficiente que la versión secuencial. La figura 1 muestra estas comparaciones.

Pese a que a primera instancia pareciera que el código secuencial era mejor (en cuanto a velocidad se refiere), a medida que se aumentaba la cantidad de datos a analizar, el código secuencial fue quedándose atrás en comparación al código paralelizado. Esto nos indica que la implementación paralela del código fue adecuada.

## 3. Probabilidades asignadas a la generación de los dígitos

Lo que se quiera hacer ahora es ver de qué manera las probabilidades asignadas a la generación de los dígitos afecta el porcentaje de datos reconocidos satisfactoriamente por la red. Para ello se modificaron independientemente las probabilidades de pintar un pixel ya sea en blanco, gris o negro creando veinte repeticiones para cada color. La figura 2 muestra cómo se vieron afectados estos porcentajes de datos correctamente identificados al variar cada una de las tres probabilidades.

Se tomaron probabilidades cercanas a las propuestas en el código propuesto para determinar si aumentar o disminuir esa probabilidad mostraría alguna tendencia en el incremento o decremento del porcentaje de datos correctamente identificados, pero no fue así ya que, como se observa en la figura 2 no se puede ver ese "patrón" de incremento o decremento salvo por la probabilidad del color gris que pareciera que mientras más alta sea su probabilidad, más alto el porcentaje de datos correctamente identificados; sin embargo al aumentar su probabilidad a un valor de 0,95 y 0,96, el porcentaje disminuyó.

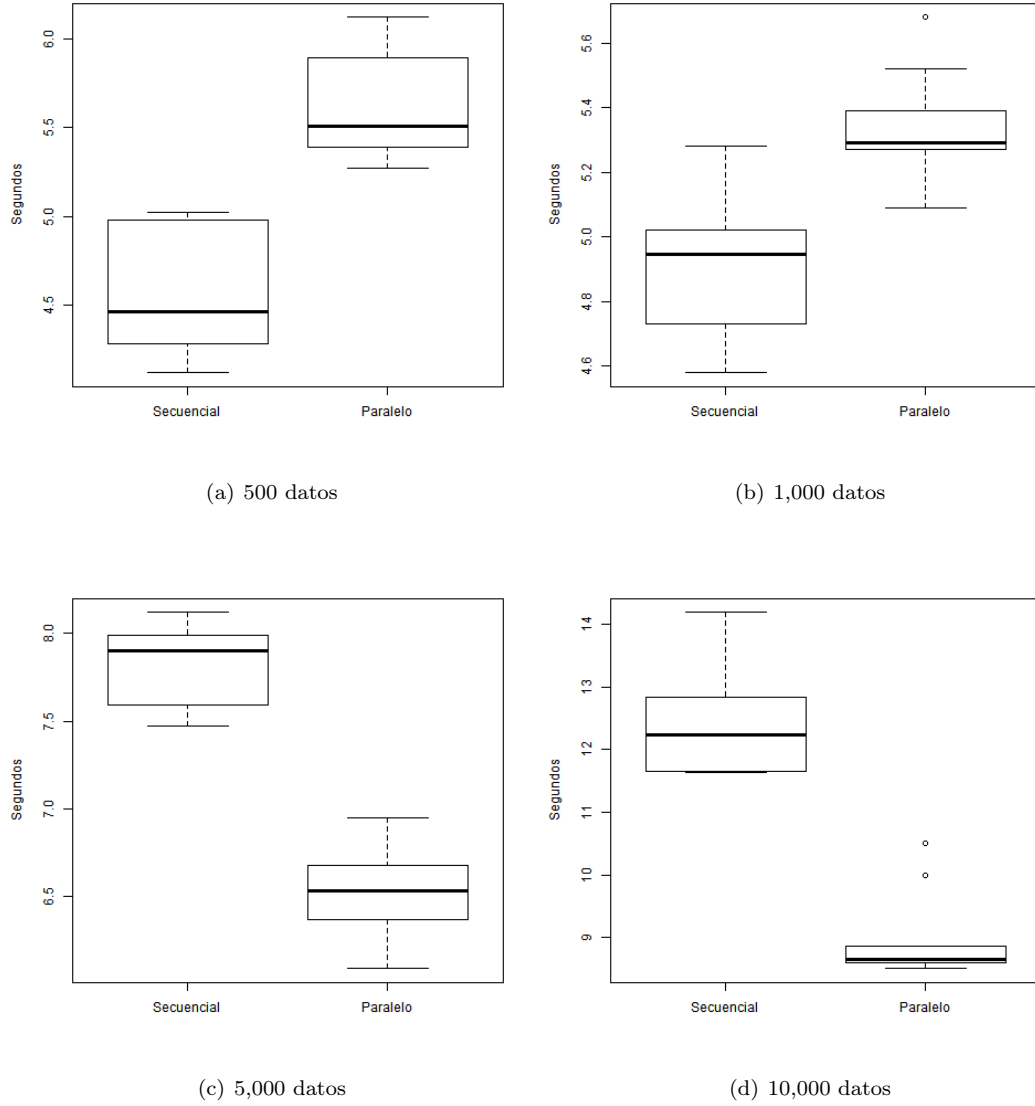


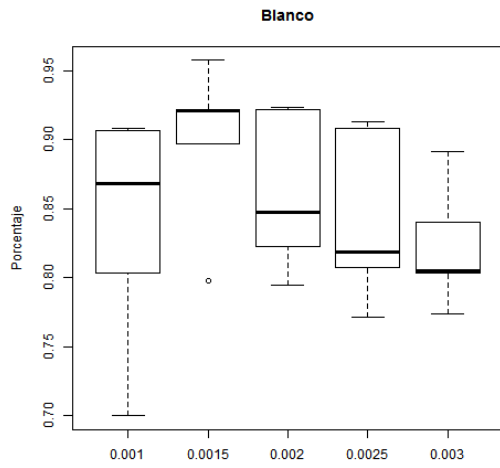
Figura 1: Comparaciones entre los tiempos de ejecución de los códigos

Las probabilidades obtenidas que mejoran el porcentaje fueron de 0,0015, 0,94 y de 0,994 para los colores blanco, gris y negro respectivamente y bajo estas probabilidades se tuvo en promedio un porcentaje de acierto de 91,18, ligeramente superior al porcentaje obtenido con los valores originales el cual rondaba en promedio en 88,67.

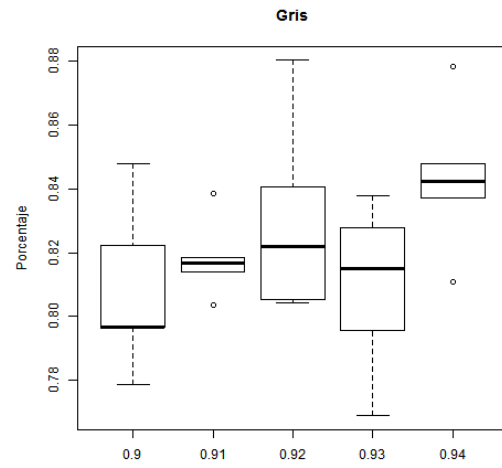
#### 4. Doce símbolos adicionales

Finalmente se entrenó a la red neuronal para poder identificar más símbolos, entre los cuales se encuentran los símbolos de la suma, resta, multiplicación, división, etcétera (se anexa archivo .csv al repositorio con la lista de los símbolos).

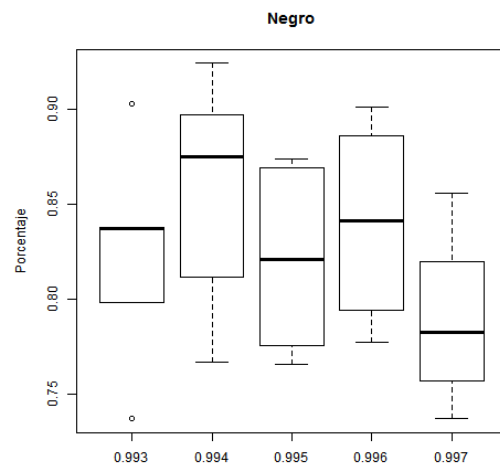
La figura 3 muestra la interpretación de la red sobre los nuevos símbolos.



(a) 500 datos



(b) 1,000 datos



(c) 5,000 datos

Figura 2: Comparaciones entre los tiempos de ejecución de los códigos

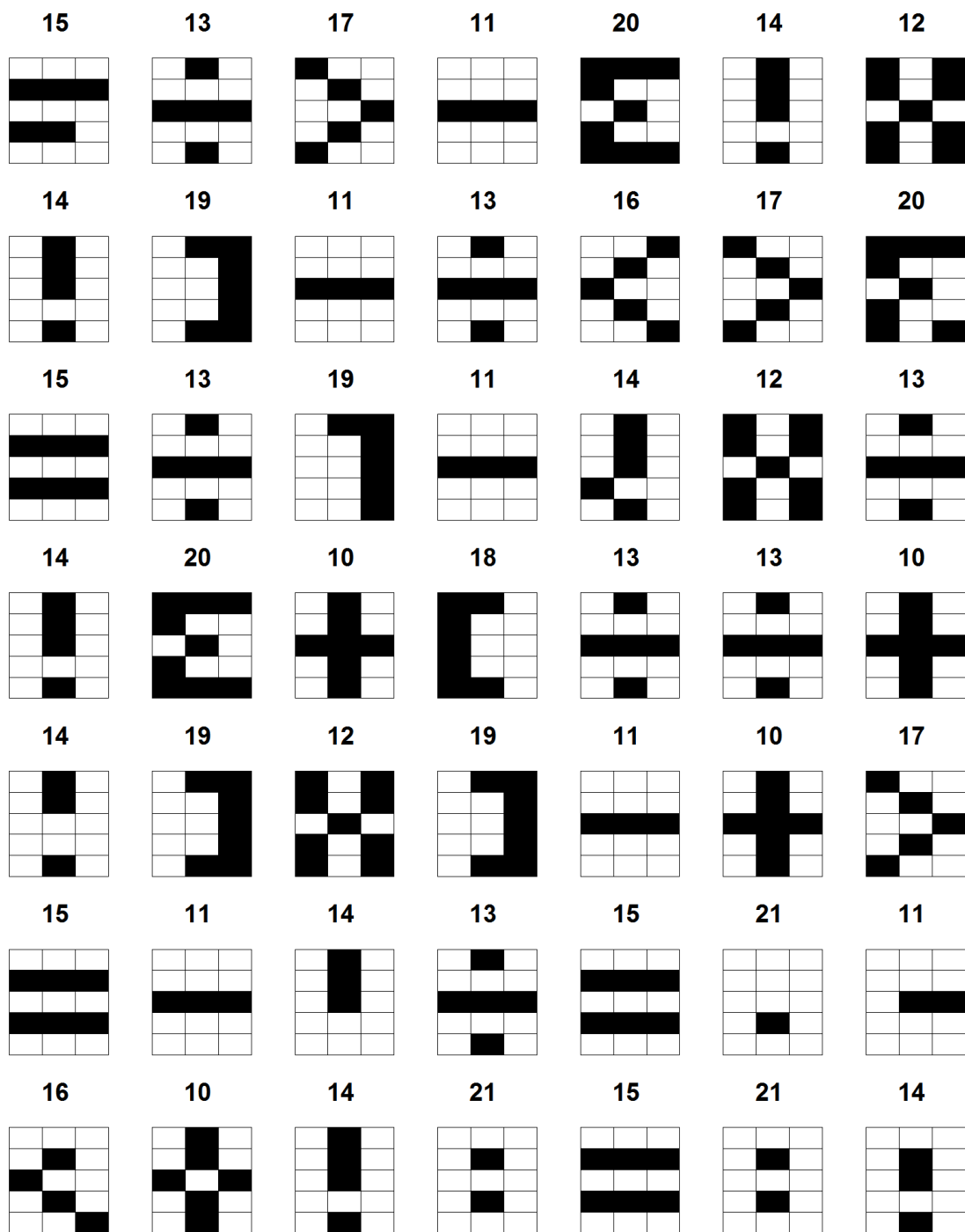


Figura 3: Generación de los nuevos doce símbolos