

Práctica 12

Jose David Martinez Ruiz

31 de octubre de 2017

1. Introducción

En esta práctica lo que buscamos es realizar una paralelización de un programa que trabaja con una red neuronal para la identificación de números a partir de un modelo previamente proporcionado. Al implementar la paralelización se comparó el tiempo que hace la implementación secuencial y la paralela. Se espera que la implementación en paralelo resulte ser más rápida.

Para el primer reto se modificó los valores de las probabilidades de los 3 colores que tenemos como referencia para los números que se leen como modelo y verificar la efectividad del sistema neuronal a partir de las modificaciones a estas probabilidades.

2. Desarrollo

Lo que el programa realiza es, a partir de un archivo que recibe como modelo para la lectura de los números, en este archivo tiene quince celdas por cada número, dichas celdas representan un numero en un espacio de cinco por tres. En ese espacio esta coloreado para así formar los números. El color que tienen es negro, gris y blanco, donde los negros y los grises forman el número y los blancos son los “huecos” que presenta un número. Ya una vez leído ese archivo con los modelos, a cada color se le asigna una probabilidad, para los negros y los grises una probabilidad muy cercana a uno, mientras que al blanco una muy cercana al cero.

En el archivo hay modelos para diez números, del cero al nueve, a partir de eso se creó una red neuronal para que, dado un numero aleatorio entre cero y nueve, cree una “imagen” al analizar una muestra devuelva el “código” binario el número que la red cree que sea. La red neuronal consta de cuatro neuronas, una por cada fila de la matriz y quince columnas, son cuatro filas ya que para poder representar en números binarios hasta el número nueve son necesarios cuatro dígitos y son quince columnas debido a que son quince los componentes de cada número que se lee. Cada neurona se encarga de regresar un dígito de la presentación binaria de un número, por ejemplo, si la red neuronal cree que está leyendo un tres, regresara FALSE FALSE TRUE TRUE que en código binario es 0011 y esto representa el número tres.

Debido a que la red neuronal se crea, en un principio, de manera aleatoria, ésta debe de ser “entrenada” para que así a la hora de identificar los números lo haga de una forma satisfactoria. Para ello, se dejó a la red intentar decir si una “imagen” dada representa el número que debe de ser. En caso de que una de las neuronas se equivoque queriendo dar su representación del número, es decir el dígito que le toca asignarle, se le hace un ajuste a la neurona, de manera que después arroje un resultado correcto.

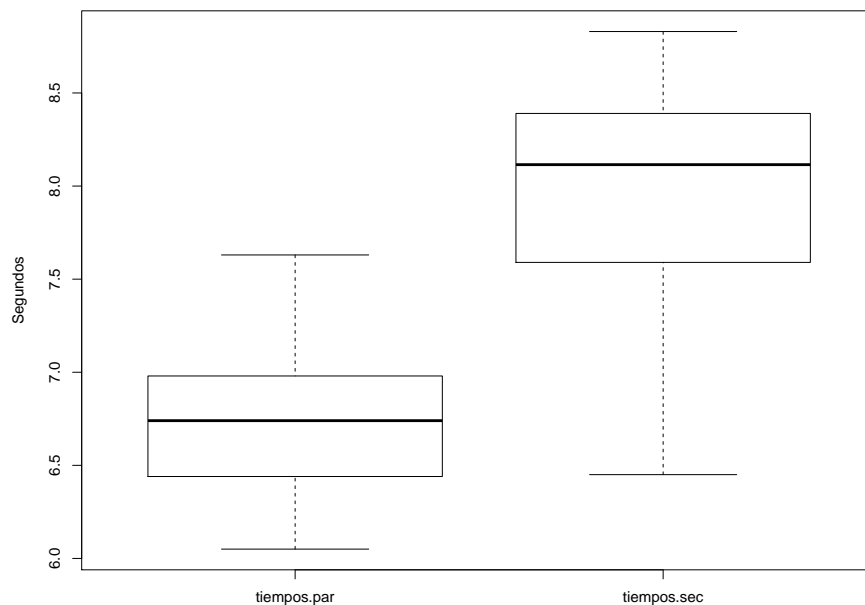
El código primero entrena la red neuronal para después realizar la prueba y determinar que tan buenos resultados da la red neuronal. Como nos interesa paralelizar se buscó las áreas del código donde se pueda implementar la paralelización. Los ciclos que el código tiene están en la fase de entrenamiento y la fase de la prueba, debido a que en la fase de entrenamiento se va modificando los valores de las neuronas, el proceso no se realiza de manera independiente por lo que no es posible paralelizar, así que nos enfocamos en la fase donde se realiza la prueba de la red neuronal.

Se creó una función donde se selecciona un numero al azar y genera una “imagen” que la red

neuronal leerá y tratará de identificar de que numero se trata. Si la red neuronal identifica correctamente el número, la función regresará un valor verdadero, y en caso de que no lo identifique correctamente regresará un valor falso. Esta función fue la que se paralelizó de manera que al final tengamos en un vector que tenga los resultados a la hora de identificar los números.

Para poder saber la efectividad de la red neuronal una vez ya efectuada la fase de la prueba, se suman los resultados obtenidos y se divide entre la cantidad de pruebas que se hicieron y así obtenemos el porcentaje de efectividad de la red neuronal.

Al haber realizado la paralelización nos importa mejorar los tiempos a la hora de ejecutar el programa. Originalmente la prueba se realizaba trescientas veces, pero para poder hacer más notoria la diferencia de los tiempos entre ambas implementaciones se aumentó la duración de la prueba para que ahora se hiciera la prueba a diez mil. Se calcularon diez veces los tiempos de ambas y se realizó una caja de bigotes para mostrar los tiempos obtenidos. Dicha caja se muestra a continuación.

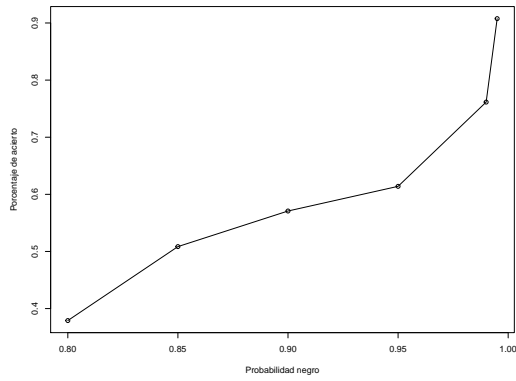


Como se puede observar en la caja de bigotes se puede observar que el tiempo de la implementación en paralelo es más pequeño que la implementación secuencial. Ahora haremos una prueba estadística para poder confirmar si la diferencia que hay es significativa o no. Se realizó una prueba estadística de Wilcoxon para comprobar esta diferencia. La prueba estadística arrojó un p-valor por debajo del cinco por ciento, por lo que si hay una diferencia significativa entre los tiempos de ambas implementaciones.

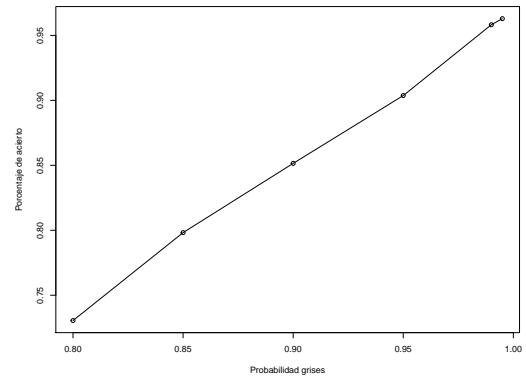
3. Reto 1

Para este reto lo que se busca es analizar el efecto que tiene el cambio de la probabilidad de los colores negro, gris y blanco con la efectividad de la red neuronal. Se le dieron valores ciertos valores similares a los valores predefinidos para cada color y se guardó la efectividad de la red neuronal para cada caso. Se graficó la relación que hubo entre la probabilidad dada y el desempeño de la red neuronal. a continuación, se muestran las gráficas del desempeño versus probabilidad para cada caso.

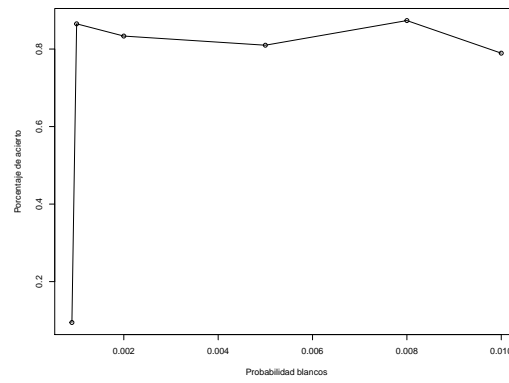
Las anteriores gráficas muestran el comportamiento solo cambiando las probabilidades de los respectivos colores. También nos interesa saber el efecto del cambio de estas probabilidades en con-



(a) Negros

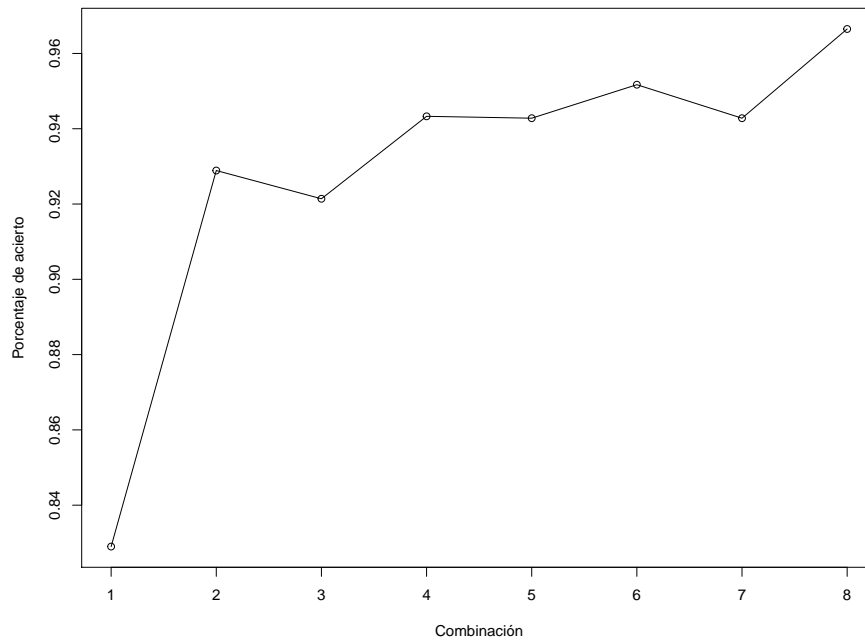


(b) Grises



(c) Blancos

junto, por lo que para poder ver este efecto se tomaron los valores de la probabilidad en cada caso que arrojaran los mejores resultados, estas probabilidades fueron de 0.99 y 0.995 tanto para el color negro y el color gris, y para el color blanco fueron de 0.008 y 0.002. estos valores se combinaron generando así ocho combinaciones diferentes, se calculó la efectividad para cada combinación y los resultados se muestran en la siguiente gráfica.



4. Conclusiones

De la tarea base se concluye que la implementación en paralelo resulta ser beneficiosa cuando los procesos son más largos. Si la paralelización se hace adecuadamente se puede ahorrar significativamente el tiempo a la hora de ejecutar el programa.

Del primer reto se concluye que el modificar de cierta forma las probabilidades se cambia drásticamente la efectividad de la red neuronal, como pasó en una de las modificaciones de la probabilidad del color blanco. al modificar individualmente cada probabilidad, en ciertos casos se obtuvo resultados bastante beneficiosos con porcentajes de acierto arriba del ochenta por ciento, pero al combinar las probabilidades de los tres colores se obtuvieron resultados aún mejor que los obtenidos modificando individualmente, por lo que la combinación de los tres colores genera un porcentaje de acierto mayor que los valores predefinidos y también que los valores que fueron estudiados.