

Práctica 12: red neuronal

Abraham Azael Morales Juárez 1422745

7 de mayo de 2019

1. Introducción

En esta práctica se trata de lo que se conoce como aprendizaje a máquina o una red neuronal, este es un proceso que permite a una computadora aprender a interpretar el comportamiento o naturaleza de un conjunto de datos a partir de observaciones o muestras [2]. El elemento básico de una red neuronal es un perceptrón que es un hiperplano que se coloca en la frontera de la separación de entradas verdaderas y falsas [3]. Con la dimensión del perceptrón d dada a lo largo de un vector x de entrada y representando el estado del perceptrón por w que tiene los pesos [3].

2. Objetivos

Paralelizar el código y estudiar de manera sistemática el desempeño de la red neuronal para los diez dígitos en función de las tres probabilidades asignadas (ngb) variando estos valores.

3. Resultados

El código base [3] se modificó para paralelizar utilizando como referencia el código de un compañero [1]. Además de la paralelización se midió el desempeño del código tomando en cuenta la varianza con respecto a la normal de los valores obtenidos. Los resultados se pueden observar en las figuras 1 y 2.

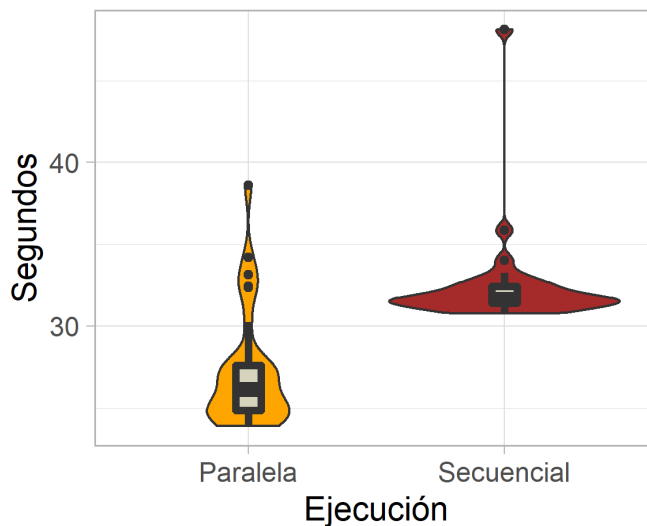


Figura 1: Se realizó la comparación de los métodos en secuencial y en paralelo, los tiempos arrojaron que el paralelo fue el mejor.

Se agrega un fragmento del código con el cual se obtuvo estos resultados.

```

1 suppressMessages(library(doParallel))
2 Mc <- makeCluster(detectedCores() - 1)
3 registerDoParallel(Mc)
4
5 prueba1 <- microbenchmark({
6   contadores <- matrix(rep(0, k*(k+1)), nrow=k, ncol=(k+1))
7   rownames(contadores) <- 0:tope
8   colnames(contadores) <- c(0:tope, NA)
9   conta <- foreach(t = 1:300, .combine = "rbind", .export = c("tope", "dim", "modelos", "neuronas", "
   pixeles")) %dopar% newr()
10  for (i in 1:300){
11    contadores[conta[i, 1], conta[i, 2]] <- contadores[conta[i, 1], conta[i, 2]] + 1
12  }
13 }
14 times = repe, unit = "s")
15 stopCluster(Mc)
16
17 entrenamientos <- cbind("Secuencial" = entrena$time)
18 pruebas <- cbind("Secuencial" = prueba$time, "Paralela" = prueba1$time)
19 library(ggplot2)
20 pruebas <- melt(pruebas)
21 ggplot(tests, aes(x=as.factor(Var2), y=value/10000000)) + geom_violin(aes(fill=as.factor(Var2))) +
22   geom_boxplot(fill="#d6d4bc", width=0.1, lwd =1.5) + scale_fill_manual(values=c("orange", "brown"))
23   +
24   labs(x = "Ejecucion", y = "Segundos") + theme_light(base_size = 14) + guides(fill=FALSE)
25 ggsave("Tiempo.png")

```

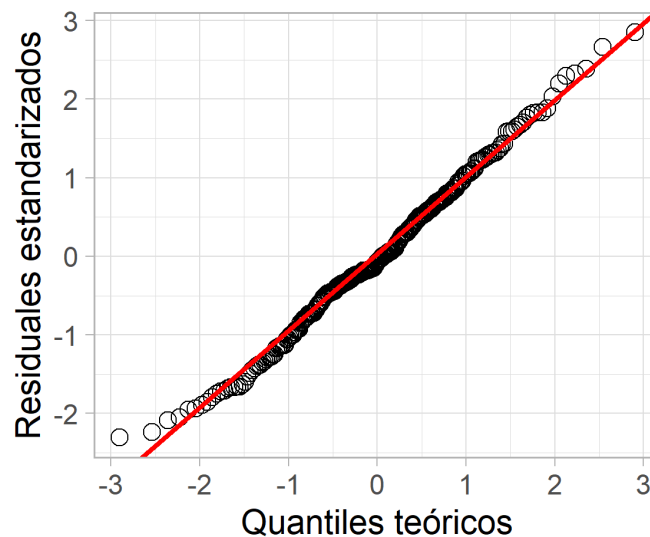


Figura 2: Normalidad de los resultados obtenidos

En la figura 2 se puede observar que el mayor porcentaje de los valores dados se encuentran muy cerca de la normalidad. Para obtener mayor información se realizó un análisis de varianza, cuyos resultados se presentan en la figura 3 se presenta una tabla, en la cual se observan que la mayor presencia del color de píxeles son el blanco y negro, del cual el píxel negro presente una mayor prevalencia.

Color píxel	Df	Sum Sq	Mean Sq	F value	Pr (>F)
Negro	2	4.78	2.39	65.53	0.0000
Gris	2	0.67	0.33	9.17	0.0001
Blanco	2	4.73	2.37	64.86	0.0000
Negro/Gris	4	0.32	0.08	2.20	0.0693
Negro/Blanco	4	0.48	0.12	3.27	0.0123
Gris/Blanco	4	0.05	0.01	0.34	0.8475
Negro/Gris/Blanco	8	0.47	0.06	1.60	0.1240
Residuales	243	8.87	0.04		

Figura 3: Resultados del análisis de varianza

También se presenta un fragmente del código del cual se uso para poder obtener la normalidad de los valores.

```
1 print(xtable(summary(av), type='latex'), file="Tabla_valores.txt")
2 ggQQ <- function(lm) {
3   # https://stackoverflow.com/questions/4357031/qnorm-and-qqline-in-ggplot2/19990107#19990107
4   d <- data.frame(std.resid = rstandard(lm))
5   y <- quantile(d$std.resid[!is.na(d$std.resid)], c(0.25, 0.75))
6   x <- qnorm(c(0.25, 0.75))
7   slope <- diff(y)/diff(x)
8   int <- y[1L] - slope * x[1L]
9
10  p <- ggplot(data=d, aes(sample=std.resid)) +
11    stat_qq(shape=1, size=3) + # open circles
12    labs(title=NULL, # plot title
13         x="Quantiles  $t_{f3}$ ricos", # x-axis label
14         y="Residuales estandarizados") + # y-axis label
15    geom_abline(slope = slope, intercept = int, col = "red", lwd = 1) # dashed reference line
16  p <- p + theme_light(base_size = 14)
17  return(p)
18 }
19 ggQQ(lm_model)
20 ggsave("Normalidad.png")
```

4. Conclusiones

Se redujo los tiempos del procesamiento al usar el método paralelo. Los datos arrojados por los experimentos muestran que los datos son normalmente distribuidos, pero no tienen la misma proporción de distribución porque las varianzas no son iguales a 1. Además, que los píxeles con mayor presencia con los blancos y negros, del cual resalta por poco el píxel negro.

Referencias

- [1] Ricardo Rosas Macías. Práctica 12: Red neuronal. 2018. URL <https://github.com/RicardoRosMac/Simulation/tree/master/HWP12>.
- [2] Eduardo Coca Reyna. Práctica 12: Red neuronal. *Universidad Autónoma de Nuevo León*, 2018.
- [3] Satu Elisa Schaeffer. Práctica 12: Red neuronal, 2019. URL <https://elisa.dyndns-web.com/teaching/comp/par/p10.html>.