

P12

Ismael Crespo

18 de mayo de 2022

1. Introducción

En esta práctica se presenta una demostración básica de aprendizaje a máquina: se busca reconocer dígitos de imágenes pequeñas en blanco y negro con una red neuronal. El elemento básico de una red neuronal es un perceptrón que esencialmente es un hiperplano (una línea si nos limitamos a dos dimensiones) que busca colocarse en la frontera que separa las entradas verdaderas y las entradas falsas. La dimensión d del perceptrón es el largo del vector x que toma como entrada, y su estado interno se representa con otro vector w que contiene sus pesos. Para responder a una salida proporcionada a ello, el perceptrón calcula el producto interno de $x * w$, es decir $\sum_{i=1}^d x_i * w_i$, y si esta suma es positiva, la salida del perceptrón es verdad, en otro caso es falso (E. Schaeffer [2]).

2. Objetivo

Estudiar de manera sistemática el desempeño de la red neuronal en términos de su puntaje F (F-score en inglés) para los diez dígitos en función de las tres probabilidades asignadas a la generación de los dígitos (**ngb**), variando a las tres en un experimento factorial adecuado.

3. Programación en R

La creación de los dígitos dentro de una pantalla (figura 1), por medio del encendido probabilístico a negro, gris y blanco (**ngb**), así como la aplicación de una red neuronal se realiza por medio de las rutinas elaboradas por E. Schaeffer [2], utilizando estas rutinas se añade una evaluación de tipo F. Score al desempeño de la red neuronal y se cambian las probabilidades de encendido **ngb** para elaborar un experimento factorial analizando estas tres instancias como factores y generando 5 replicas para cada experimento. La adición de la evaluación F. Score (ecuaciones 1, 2 y 3) y la variación de los tres factores se presentan en los códigos 1 y 2. Para consulta a detalle de los códigos consulte el repositorio de I. Hernández [1].

$$precision = \frac{TP}{TP + FP} \quad (1)$$

$$rellamado = \frac{TP}{TP + FN} \quad (2)$$

$$F_1 = 2 \times \frac{precision \times rellamado}{precision + rellamado} \quad (3)$$

Donde TP son los verdaderos positivos, FP los falsos positivos y FN los falsos negativos obtenidos del vector **contadores**.

Código 1: Variación de factores *ngb*.

```

1 neg=c(0.990,0.999,0.980)
2 gris=c(0.920,0.930,0.910)
3 blan=c(0.002,0.001,0.003)
4 n <- floor(log(k-1, 2)) + 1
5 neuronas <- matrix(runif(n * dim), nrow=n, ncol=dim) # perceptrones
6 for(ne in neg){
7   for(gr in gris){
8     for(bl in blan){
9       fscore=numeric()
10      modelos <- read.csv("digits.txt", sep=" ", header=FALSE, stringsAsFactors=F)
11      modelos[modelos=='n'] <- ne
12      modelos[modelos=='g'] <- gr
13      modelos[modelos=='b'] <- bl
14      for(i in 1:replicas){
15        .....
16      }}}}

```

Código 2: Calculo de F. Score

```

1 tp=numeric()
2 for (i in 1:10){
3   tp=c(contadores[i,i],tp)
4 }
5 tp=sum(tp)
6 fn=sum(contadores[,11])
7 tn=0
8 fp=abs(tp+fn-sum(contadores))
9 acuraccy=(tp+tn)/(tp+tn+fp+fn)
10 precision=tp/(tp+fp)
11 recall=tp/(tp+tn)
12 fscore=c(fscore,2*(precision*recall)/(precision+recall))
13 }
14 fscoremean=c(fscoremean,mean(fscore))

```

4. Resultados

4.1. Variación del número de funciones objetivo.

Las tablas 1, 2 y 3, presentan las medianas de los F. Score de cada experimento, separado para la combinación de cada factor, las variaciones de cada probabilidad se muestran en el código 1, el factor 1 (f_1) se refiere a la probabilidad de negro, el factor 2 (f_2) a la probabilidad de ser gris y el factor 3 (f_3) la probabilidad de ser blanco y los subíndices (1,2 y 3) a cada valor que toma cada factor.

Cuadro 1: f_{1_1} media= 0.8952291.

	f_{2_1}	f_{2_2}	f_{2_3}
f_{3_1}	0.8930	0.9061	0.8875
f_{3_2}	0.8984	0.9015	0.8913
f_{3_3}	0.8959	0.8940	0.8895

Cuadro 2: $f1_2$ media=0.9094911.

	$f2_1$	$f2_2$	$f2_3$
$f3_1$	0.9088	0.9126	0.9021
$f3_2$	0.9058	0.9135	0.9066
$f3_3$	0.9147	0.9095	0.9117

Cuadro 3: $f1_3$ media= 0.8846223.

	$f2_1$	$f2_2$	$f2_3$
$f3_1$	0.8826	0.8888	0.8711
$f3_2$	0.8759	0.8914	0.8950
$f3_3$	0.8774	0.8908	0.8886

La tabla 4 presenta la prueba ANOVA, realizada por medio de la función `aov`, la columna $Pr(> F)$ indica la significancia para cada factor y entre estos $f1$ resulto tener la mayor significancia seguido de $f2$ no se mostró significancia entre los factores.

Cuadro 4: Prueba ANOVA.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
f1	1.0000000	0.0027662	0.0027662	74.6404832	0.0000001
f2	1.0000000	0.0002326	0.0002326	6.2774170	0.0214907
f3	1.0000000	0.0000028	0.0000028	0.0750199	0.7871188
f1:f2	1.0000000	0.0000000	0.0000000	0.0006303	0.9802320
f1:f3	1.0000000	0.0000187	0.0000187	0.5055443	0.4857119
f2:f3	1.0000000	0.0000069	0.0000069	0.1856300	0.6714272
f1:f2:f3	1.0000000	0.0000288	0.0000288	0.7778320	0.3888266
Residuals	19.0000000	0.0007041	0.0000371		

5. Conclusiones

La utilización de una red neuronal para identificar los dígitos generados probabilísticamente a través de píxeles, según los valores de `ngb`, funciona con valores de F. score mayores a 0.9 para los mejores casos.

El análisis por medio de experimentos factoriales nos indica las mejores opciones para implementar el experimento de la red neuronal.

Referencias

- [1] I. Crespo. P12: red neuronal. 2022. URL <https://github.com/IsmaelHC/Simulacion-NANO-2022/tree/main/P12>.
- [2] E. Schaeffer. Simulación. 2022. URL <https://satuelisa.github.io/simulation>.

0806488095120769408930
1318689667168257677142
8080854609130898002575
0213761930574968032093
4295893113091044532201
2382278015269970101743
8778945172732814926689
8207183632820017222952
3781608672067580907649
1118079205241826800702
7685241743327037409634
5475674711522535788843
0413534118129281574652
5292285785316361613856
9819151685511734938861
2446556001935101606531
7125796168525357064682
8972570458458130857178
9140750203839145645018
2299684877132513033836
4184201986594449412353
4687963712645630105141
6425004250398883

Figura 1: Generación de dígitos con encendido de pixeles.