Guia 1

Grupo 3

9/28/2019

Ejercicio 1:

Realice un programa que permita el entrenamiento y prueba de un perceptrón simple con una cantidad variable de entradas.

Resolución del problema OR

- Lectura de los patrones de entrenamiento

```
OR_trn <- read_csv("../PUBLICO/Encuentro 1/Práctica/data/OR_trn.csv", col_names = FALSE)
OR_tst <- read_csv("../PUBLICO/Encuentro 1/Práctica/data/OR_tst.csv", col_names = FALSE)
```

- Selección de parámetros y entrenamiento de perceptrón

Utilizamos las funciones implementadas en el archivo "PerceptronSimple.R"

```
pesos <- entrenarPerceptron(OR_trn, maxEpocas = 10, critFinalizacion = 0.8)
```

```
## Epoca: 1 - Tasa: 1
```

```
# Modelo obtenido
pesos
```

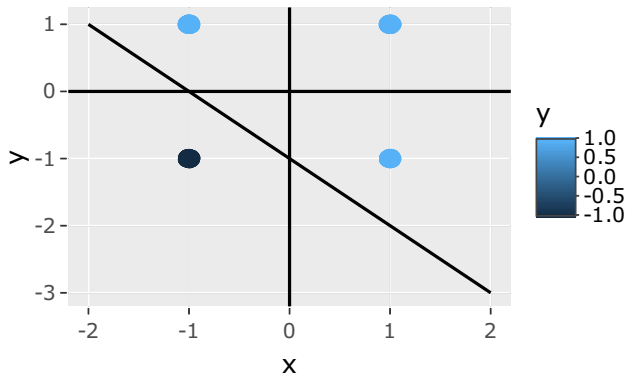
```
## [1] 1 1 1
```

El entrenamiento nos devuelve la tasa obtenida en cada época y al final mostramos los valores de los pesos w que definen la recta del modelo.

$ pesos = [w_0, w_1, w_2, ... w_n] $

- Graficas

```
graficarRectaSeparacion(pesos, OR_trn)
```



En la gráfica vemos como la recta del modelo separa el dominio según la clase.

- Prueba con datos de test

```
test <- aplicarPerceptron(pesos, OR_tst)
test$tasa
```

```
## [1] 1
```

Por ser este un problema sencillo donde las clases son separables por una recta, la tasa de aciertos en test nos da un 100%.

Resolución del problema XOR

- Lectura de los patrones de entrenamiento

```
XOR_trn <- read_csv("../PUBLICO/Encuentro 1/Práctica/data/XOR_trn.csv", col_names = FALSE)
XOR_tst <- read_csv("../PUBLICO/Encuentro 1/Práctica/data/XOR_tst.csv", col_names = FALSE)
```

- Selección de parámetros y entrenamiento de perceptrón

```
pesos <- entrenarPerceptron(XOR_trn, maxEpocas = 10, critFinalizacion = 0.8)
```

```
## Epoca: 1 - Tasa: 0.4965
## Epoca: 2 - Tasa: 0.4965
## Epoca: 3 - Tasa: 0.4965
## Epoca: 4 - Tasa: 0.4965
## Epoca: 5 - Tasa: 0.4965
## Epoca: 6 - Tasa: 0.4965
## Epoca: 7 - Tasa: 0.4965
## Epoca: 8 - Tasa: 0.4965
## Epoca: 9 - Tasa: 0.4965
## Epoca: 10 - Tasa: 0.4965
```

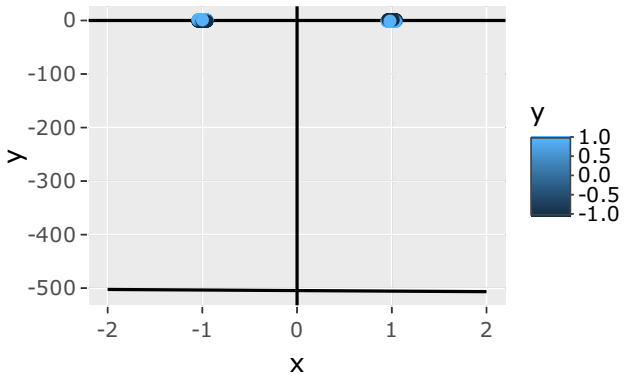
```
pesos
```

```
## [1] 504.50000  -2.75460  -2.85718
```

Vemos que luego de 10 épocas el perceptrón simple no logra resolver el problema del XOR, ni tampoco mejorar su tasa de aciertos.

- Graficas

```
graficarRectaSeparacion(pesos, XOR_trn)
```



En la gráfica vemos la recta obtenida que deja todos los patrones clasificados de igual manera.

- Prueba con datos de test

```
test <- aplicarPerceptron(pesos, XOR_tst)
test$tasa
```

```
## [1] 0.44
```

Este problema no pudo ser resuelto por le perceptrón simple, la tasa de aciertos obtenida es peor que el azar.

Preguntas:

- ¿Pueden ser resueltos ambos problemas (OR y XOR) empleando un perceptrón simple? Justifique su respuesta.

En los resultados obtenidos verificamos que no se puede resolver el problema XOR con el perceptrón simple porque no es posible dividir la clase con una recta que separe el dominio en dos.

- ¿Qué efecto tiene la tasa de aprendizaje en el entrenamiento del perceptrón? Explique cómo este parámetro afecta a la actualización de la frontera de decisión

La tasa de aprendizaje nos permite variar cuanto se actualizan los valores de los pesos. Una tasa muy baja puede generar que el algoritmo demore mucho en llegar al mínimo, y una tasa muy alta puede generar que el algoritmo diverja y nunca llegue a un mínimo. En nuestro caso, se implementó una función que posee una tasa de aprendizaje de 0,05 (nu=0.05) por defecto y no fue necesario cambiarlo.

Ejercicio 2:

Punto a:

- Lectura de datos

```
spheres1d10 <- read_csv("../PUBLICO/Encuentro 1/Práctica/data/spheres1d10.csv", col_names = FALSE)
#Nombre de las columnas
colnames(spheres1d10)
```

```
## [1] "X1" "X2" "X3" "X4"
```

```
#Dimenciones de los datos
dim(spheres1d10)
```

```
## [1] 1000    4
```

- Grafica

```
plot_ly(x=spheres1d10$X1, y=spheres1d10$X2, z=spheres1d10$X3, type="scatter3d",
        mode="markers", color=spheres1d10$X4)
```

- Generamos las particiones

Utilizamos la función implementada en el archivo “Particiones.R”

```
particion_spheres1d10 <- generarNParticionesPorID(dataset = spheres1d10, nroParticiones = 5,
                                                  porcEntrenamiento = 0.8, semilla = 1,
                                                  clase = "X4")
```

Esta función nos devuelve un listado de 5 listas de ID para entrenamiento, de aproximadamente 800 elementos, y 5 listas de ID para prueba, de aproximadamente 200 elementos.

- Generamos los modelos con el perceptrón simple.

```
for (n in seq(1,5)) {
  print(glue::glue("Partición: {n}"))
  #Entrenamiento
  pesos <- entrenarPerceptron(spheres1d10[particion_spheres1d10$trn[[n]],],
                                    maxEpocas = 10, critFinalizacion = 0.8)
  print(glue::glue("Pesos: {pesos}"))
```

```r
  #Prueba
  test <- aplicarPerceptron(pesos, spheres1d10[particion_spheres1d10$tst[[n]],])
  print(glue::glue("Tasa de aciertos en test: {test$tasa}"))
}
```

```
## Partición: 1
## Epoca: 1 - Tasa: 0.53125
## Epoca: 2 - Tasa: 0.53125
## Epoca: 3 - Tasa: 0.53125
## Epoca: 4 - Tasa: 0.53125
## Epoca: 5 - Tasa: 0.53125
## Epoca: 6 - Tasa: 0.53125
## Epoca: 7 - Tasa: 0.53125
## Epoca: 8 - Tasa: 0.53125
## Epoca: 9 - Tasa: 0.53125
## Epoca: 10 - Tasa: 0.53125
## Pesos: 187.800000000008
## Pesos: -106.131447
## Pesos: 4.64245900000002
## Pesos: -7.95994049999996
## Tasa de aciertos en test: 0.53
## Partición: 2
## Epoca: 1 - Tasa: 0.53125
## Epoca: 2 - Tasa: 0.53125
## Epoca: 3 - Tasa: 0.53125
## Epoca: 4 - Tasa: 0.53125
## Epoca: 5 - Tasa: 0.53125
## Epoca: 6 - Tasa: 0.53125
## Epoca: 7 - Tasa: 0.53125
## Epoca: 8 - Tasa: 0.53125
## Epoca: 9 - Tasa: 0.53125
## Epoca: 10 - Tasa: 0.53125
## Pesos: 187.800000000008
## Pesos: -118.1787755
## Pesos: 14.6540814999999
## Pesos: -16.9216049999999
## Tasa de aciertos en test: 0.53
## Partición: 3
## Epoca: 1 - Tasa: 0.53125
## Epoca: 2 - Tasa: 0.53125
## Epoca: 3 - Tasa: 0.53125
## Epoca: 4 - Tasa: 0.53125
## Epoca: 5 - Tasa: 0.53125
## Epoca: 6 - Tasa: 0.53125
## Epoca: 7 - Tasa: 0.53125
## Epoca: 8 - Tasa: 0.53125
## Epoca: 9 - Tasa: 0.53125
## Epoca: 10 - Tasa: 0.53125
## Pesos: 187.400000000008
## Pesos: -108.1191785
## Pesos: -1.5604655
## Pesos: -14.224063
## Tasa de aciertos en test: 0.53
```

```
## Partición: 4
## Epoca: 1 - Tasa: 0.530586766541823
## Epoca: 2 - Tasa: 0.530586766541823
## Epoca: 3 - Tasa: 0.530586766541823
## Epoca: 4 - Tasa: 0.530586766541823
## Epoca: 5 - Tasa: 0.530586766541823
## Epoca: 6 - Tasa: 0.530586766541823
## Epoca: 7 - Tasa: 0.530586766541823
## Epoca: 8 - Tasa: 0.530586766541823
## Epoca: 9 - Tasa: 0.530586766541823
## Epoca: 10 - Tasa: 0.530586766541823
## Pesos: 188.450000000009
## Pesos: -111.6222595
## Pesos: 6.21698749999997
## Pesos: -14.0072609999999
## Tasa de aciertos en test: 0.532663316582915
## Partición: 5
## Epoca: 1 - Tasa: 0.530663329161452
## Epoca: 2 - Tasa: 0.530663329161452
## Epoca: 3 - Tasa: 0.530663329161452
## Epoca: 4 - Tasa: 0.530663329161452
## Epoca: 5 - Tasa: 0.530663329161452
## Epoca: 6 - Tasa: 0.530663329161452
## Epoca: 7 - Tasa: 0.530663329161452
## Epoca: 8 - Tasa: 0.530663329161452
## Epoca: 9 - Tasa: 0.530663329161452
## Epoca: 10 - Tasa: 0.530663329161452
## Pesos: 187.900000000008
## Pesos: -104.4862115
## Pesos: -5.546697
## Pesos: -13.4374634999999
## Tasa de aciertos en test: 0.532338308457711
```

Punto b:

- Lectura de datos

```
spheres2d10 <- read_csv("../PUBLICO/Encuentro 1/Práctica/data/spheres2d10.csv", col_names = FALSE)
spheres2d50 <- read_csv("../PUBLICO/Encuentro 1/Práctica/data/spheres2d50.csv", col_names = FALSE)
spheres2d70 <- read_csv("../PUBLICO/Encuentro 1/Práctica/data/spheres2d70.csv", col_names = FALSE)
```

- Generamos las particiones

Llamamos a la función dos veces con dos semillas para obtener las 10 particiones. La función implementada genera las particiones sin reemplazo, con una relación 80/20 no se pueden generar más de 5 particiones.

```
# 10%
particion_spheres2d10 <- generarNParticionesPorID(dataset = spheres2d10, nroParticiones = 5,
                                                  porcEntrenamiento = 0.8, semilla = 1,
                                                  clase = "X4")
particion_spheres2d10$trn <- c(particion_spheres2d10$trn, generarNParticionesPorID(
  dataset = spheres2d10, nroParticiones = 5, porcEntrenamiento = 0.8, semilla = 12,
```

```
  clase = "X4")$trn)
particion_spheres2d10$tst <- c(particion_spheres2d10$tst, generarNParticionesPorID(
  dataset = spheres2d10, nroParticiones = 5, porcEntrenamiento = 0.8, semilla = 12,
  clase = "X4")$tst)

# 50%
particion_spheres2d50 <- generarNParticionesPorID(dataset = spheres2d50, nroParticiones = 5,
                                                  porcEntrenamiento = 0.8, semilla = 1,
                                                  clase = "X4")
particion_spheres2d50$trn <- c(particion_spheres2d50$trn, generarNParticionesPorID(
  dataset = spheres2d50, nroParticiones = 5, porcEntrenamiento = 0.8, semilla = 12,
  clase = "X4")$trn)
particion_spheres2d50$tst <- c(particion_spheres2d50$tst, generarNParticionesPorID(
  dataset = spheres2d50, nroParticiones = 5, porcEntrenamiento = 0.8, semilla = 12,
  clase = "X4")$tst)

# 70%
particion_spheres2d70 <- generarNParticionesPorID(dataset = spheres2d70, nroParticiones = 5,
                                                  porcEntrenamiento = 0.8, semilla = 1,
                                                  clase = "X4")
particion_spheres2d70$trn <- c(particion_spheres2d70$trn, generarNParticionesPorID(
  dataset = spheres2d70, nroParticiones = 5, porcEntrenamiento = 0.8, semilla = 12,
  clase = "X4")$trn)
particion_spheres2d70$tst <- c(particion_spheres2d70$tst, generarNParticionesPorID(
  dataset = spheres2d70, nroParticiones = 5, porcEntrenamiento = 0.8, semilla = 12,
  clase = "X4")$tst)
```

- Generamos los modelos con el perceptrón simple.

– Datos con desviaciones del 10%

```
plot_ly(x=spheres2d10$X1, y=spheres2d10$X2, z=spheres2d10$X3, type="scatter3d",
        mode="markers", color=spheres2d10$X4)
```

WebGL is not supported by your browser - visit https://get.webgl.org for more info

```
tasaMedia <- 0
for (n in seq(1,10)) {
  print(glue::glue("Partición: {n}"))
  #Entrenamiento
  pesos <- entrenarPerceptron(spheres2d10[particion_spheres2d10$trn[[n]],],
                              maxEpocas = 5, critFinalizacion = 0.79)
  print(glue::glue("Pesos: {pesos}"))
  #Pruba
  test <- aplicarPerceptron(pesos, spheres2d10[particion_spheres2d10$tst[[n]],])
  print(glue::glue("Tasa de aciertos en test: {test$tasa}"))
  tasaMedia <- tasaMedia + test$tasa
}
```

```
## Partición: 1
## Epoca: 1 - Tasa: 0.768
## Epoca: 2 - Tasa: 0.78075
## Epoca: 3 - Tasa: 0.78425
## Epoca: 4 - Tasa: 0.7875
## Epoca: 5 - Tasa: 0.79
## Pesos: 222.700000000016
## Pesos: -219.029914999999
## Pesos: 7.65209900000001
## Pesos: -7.60560449999999
## Tasa de aciertos en test: 0.792
```

```
## Partición: 2
## Epoca: 1 - Tasa: 0.7685
## Epoca: 2 - Tasa: 0.77925
## Epoca: 3 - Tasa: 0.78375
## Epoca: 4 - Tasa: 0.7855
## Epoca: 5 - Tasa: 0.78675
## Pesos: 224.500000000017
## Pesos: -220.723124999999
## Pesos: 7.1609235
## Pesos: -7.48625199999997
## Tasa de aciertos en test: 0.789
## Partición: 3
## Epoca: 1 - Tasa: 0.77725
## Epoca: 2 - Tasa: 0.78675
## Epoca: 3 - Tasa: 0.79
## Epoca: 4 - Tasa: 0.791
## Pesos: 174.350000000005
## Pesos: -171.550399
## Pesos: 5.7034205
## Pesos: -5.89705049999999
## Tasa de aciertos en test: 0.782
## Partición: 4
## Epoca: 1 - Tasa: 0.766308422894276
## Epoca: 2 - Tasa: 0.781554611347163
## Epoca: 3 - Tasa: 0.78530367408148
## Epoca: 4 - Tasa: 0.788052986753312
## Epoca: 5 - Tasa: 0.789052736815796
## Pesos: 221.150000000016
## Pesos: -217.529987
## Pesos: 6.97034999999998
## Pesos: -7.62000599999998
## Tasa de aciertos en test: 0.791791791791792
## Partición: 5
## Epoca: 1 - Tasa: 0.77569392348087
## Epoca: 2 - Tasa: 0.785196299074769
## Epoca: 3 - Tasa: 0.788697174293573
## Epoca: 4 - Tasa: 0.790447611902976
## Pesos: 179.250000000007
## Pesos: -176.2329115
## Pesos: 6.246016
## Pesos: -5.78665799999998
## Tasa de aciertos en test: 0.788211788211788
## Partición: 6
## Epoca: 1 - Tasa: 0.77
## Epoca: 2 - Tasa: 0.78125
## Epoca: 3 - Tasa: 0.7855
## Epoca: 4 - Tasa: 0.7875
## Epoca: 5 - Tasa: 0.78875
## Pesos: 221.900000000016
## Pesos: -218.283639499999
## Pesos: 7.06003300000001
## Pesos: -7.45113349999997
## Tasa de aciertos en test: 0.786
## Partición: 7
```

```
## Epoca: 1 - Tasa: 0.76825
## Epoca: 2 - Tasa: 0.77925
## Epoca: 3 - Tasa: 0.7835
## Epoca: 4 - Tasa: 0.78575
## Epoca: 5 - Tasa: 0.78775
## Pesos: 222.950000000016
## Pesos: -219.351822999999
## Pesos: 7.45062699999999
## Pesos: -7.42106949999998
## Tasa de aciertos en test: 0.799
## Partición: 8
## Epoca: 1 - Tasa: 0.77275
## Epoca: 2 - Tasa: 0.784
## Epoca: 3 - Tasa: 0.7865
## Epoca: 4 - Tasa: 0.78775
## Epoca: 5 - Tasa: 0.7905
## Pesos: 220.050000000016
## Pesos: -216.367657499999
## Pesos: 7.38539349999999
## Pesos: -7.39522749999999
## Tasa de aciertos en test: 0.783
## Partición: 9
## Epoca: 1 - Tasa: 0.775306173456636
## Epoca: 2 - Tasa: 0.784053986503374
## Epoca: 3 - Tasa: 0.787303174206448
## Epoca: 4 - Tasa: 0.789052736815796
## Epoca: 5 - Tasa: 0.790302424393902
## Pesos: 218.200000000015
## Pesos: -214.778701499999
## Pesos: 7.327286
## Pesos: -7.73471149999999
## Tasa de aciertos en test: 0.794794794794795
## Partición: 10
## Epoca: 1 - Tasa: 0.770192548137034
## Epoca: 2 - Tasa: 0.782445611402851
## Epoca: 3 - Tasa: 0.786946736684171
## Epoca: 4 - Tasa: 0.790197549387347
## Pesos: 177.800000000006
## Pesos: -174.7530875
## Pesos: 6.0350335
## Pesos: -5.84073549999999
## Tasa de aciertos en test: 0.782217782217782
```

```
print(glue::glue("Tasa de aciertos media en test: {tasaMedia/10}"))
```

```
## Tasa de aciertos media en test: 0.788801615701616
```

– Datos con desviaciones del 50%

```
plot_ly(x=spheres2d50$X1, y=spheres2d50$X2, z=spheres2d50$X3, type="scatter3d", mode="markers", color=sp
```

WebGL is not supported by your browser - visit https://get.webgl.org for more info

```
tasaMedia <- 0
for (n in seq(1,10)) {
  print(glue::glue("Partición: {n}"))
  #Entrenamiento
  pesos <- entrenarPerceptron(spheres2d50[particion_spheres2d50$trn[[n]],],
                              maxEpocas = 5, critFinalizacion = 0.79)
  print(glue::glue("Pesos: {pesos}"))
  #Pruba
  test <- aplicarPerceptron(pesos, spheres2d50[particion_spheres2d50$tst[[n]],])
  print(glue::glue("Tasa de aciertos en test: {test$tasa}"))
  tasaMedia <- tasaMedia + test$tasa
}
```

```
## Partición: 1
## Epoca: 1 - Tasa: 0.7835
## Epoca: 2 - Tasa: 0.78725
## Epoca: 3 - Tasa: 0.78675
## Epoca: 4 - Tasa: 0.78675
## Epoca: 5 - Tasa: 0.78675
## Pesos: 212.950000000014
## Pesos: -196.5935165
## Pesos: 17.616281
## Pesos: -21.197985
## Tasa de aciertos en test: 0.8
```

```
## Partición: 2
## Epoca: 1 - Tasa: 0.787053236690827
## Epoca: 2 - Tasa: 0.79005248687828
## Pesos: 87.1499999999972
## Pesos: -80.0293034999999
## Pesos: 7.69552100000001
## Pesos: -9.41939549999998
## Tasa de aciertos en test: 0.784784784784785
## Partición: 3
## Epoca: 1 - Tasa: 0.788197049262316
## Epoca: 2 - Tasa: 0.789947486871718
## Epoca: 3 - Tasa: 0.790447611902976
## Pesos: 127.699999999995
## Pesos: -117.8163755
## Pesos: 11.58034
## Pesos: -13.0146915
## Tasa de aciertos en test: 0.786213786213786
## Partición: 4
## Epoca: 1 - Tasa: 0.785803549112722
## Epoca: 2 - Tasa: 0.790552361909523
## Pesos: 84.8499999999973
## Pesos: -77.8731169999999
## Pesos: 7.30685
## Pesos: -9.36183249999999
## Tasa de aciertos en test: 0.78978978978979
## Partición: 5
## Epoca: 1 - Tasa: 0.784696174043511
## Epoca: 2 - Tasa: 0.78944736184046
## Epoca: 3 - Tasa: 0.790947736934234
## Pesos: 124.999999999995
## Pesos: -115.358573
## Pesos: 10.105714
## Pesos: -12.8431495
## Tasa de aciertos en test: 0.788211788211788
## Partición: 6
## Epoca: 1 - Tasa: 0.78725
## Epoca: 2 - Tasa: 0.7915
## Pesos: 86.2499999999972
## Pesos: -79.1619959999999
## Pesos: 7.32983300000001
## Pesos: -9.27587249999998
## Tasa de aciertos en test: 0.783
## Partición: 7
## Epoca: 1 - Tasa: 0.785553611597101
## Epoca: 2 - Tasa: 0.79005248687828
## Pesos: 86.7999999999972
## Pesos: -79.6983764999999
## Pesos: 7.08412100000002
## Pesos: -9.34603799999999
## Tasa de aciertos en test: 0.78978978978979
## Partición: 8
## Epoca: 1 - Tasa: 0.784196049012253
## Epoca: 2 - Tasa: 0.787696924231058
## Epoca: 3 - Tasa: 0.788947236809202
```

```
## Epoca: 4 - Tasa: 0.788947236809202
## Epoca: 5 - Tasa: 0.789197299324831
## Pesos: 210.000000000014
## Pesos: -193.9144705
## Pesos: 16.746861
## Pesos: -20.7784565
## Tasa de aciertos en test: 0.788211788211788
## Partición: 9
## Epoca: 1 - Tasa: 0.790302424393902
## Pesos: 44.2999999999996
## Pesos: -40.714496
## Pesos: 3.87243950000001
## Pesos: -4.87139549999999
## Tasa de aciertos en test: 0.792792792792793
## Partición: 10
## Epoca: 1 - Tasa: 0.78494623655914
## Epoca: 2 - Tasa: 0.789197299324831
## Epoca: 3 - Tasa: 0.789697424356089
## Epoca: 4 - Tasa: 0.789947486871718
## Epoca: 5 - Tasa: 0.789947486871718
## Pesos: 209.450000000013
## Pesos: -193.2499555
## Pesos: 18.045507
## Pesos: -21.7199505
## Tasa de aciertos en test: 0.796203796203796
```

```
print(glue::glue("Tasa de aciertos media en test: {tasaMedia/10}"))
```

```
## Tasa de aciertos media en test: 0.789899831599832
```

– Datos con desviaciones del 70%

```
plot_ly(x=spheres2d70$X1, y=spheres2d70$X2, z=spheres2d70$X3, type="scatter3d", mode="markers", color=sp
```

WebGL is not supported by your browser - visit https://get.webgl.org for more info

```
tasaMedia <- 0
for (n in seq(1,10)) {
  print(glue::glue("Partición: {n}"))
  #Entrenamiento
  pesos <- entrenarPerceptron(spheres2d70[particion_spheres2d70$trn[[n]],],
                              maxEpocas = 10, critFinalizacion = 0.77)
  print(glue::glue("Pesos: {pesos}"))
  #Pruba
  test <- aplicarPerceptron(pesos, spheres2d70[particion_spheres2d70$tst[[n]],])
  print(glue::glue("Tasa de aciertos en test: {test$tasa}"))
  tasaMedia <- tasaMedia + test$tasa
}
```

```
## Partición: 1
## Epoca: 1 - Tasa: 0.7715
## Pesos: 48.0499999999994
## Pesos: -42.293065
## Pesos: 5.106986
## Pesos: -6.93356350000001
## Tasa de aciertos en test: 0.767
## Partición: 2
## Epoca: 1 - Tasa: 0.772056985753562
## Pesos: 47.2499999999995
## Pesos: -41.7873925000001
```

```
## Pesos: 5.08585699999999
## Pesos: -6.636121
## Tasa de aciertos en test: 0.767767767767768
## Partición: 3
## Epoca: 1 - Tasa: 0.770692673168292
## Pesos: 46.7999999999995
## Pesos: -41.362348
## Pesos: 4.839579
## Pesos: -7.2536615
## Tasa de aciertos en test: 0.792207792207792
## Partición: 4
## Epoca: 1 - Tasa: 0.775806048487878
## Pesos: 46.8499999999995
## Pesos: -41.2957785
## Pesos: 4.8841245
## Pesos: -7.0935385
## Tasa de aciertos en test: 0.762762762762763
## Partición: 5
## Epoca: 1 - Tasa: 0.774193548387097
## Pesos: 47.0499999999995
## Pesos: -41.607631
## Pesos: 4.41336199999999
## Pesos: -7.262367
## Tasa de aciertos en test: 0.773226773226773
## Partición: 6
## Epoca: 1 - Tasa: 0.77975
## Pesos: 46.4499999999995
## Pesos: -41.0160120000001
## Pesos: 4.57087449999999
## Pesos: -7.443204
## Tasa de aciertos en test: 0.767
## Partición: 7
## Epoca: 1 - Tasa: 0.771307173206698
## Pesos: 47.7999999999994
## Pesos: -42.2340885000001
## Pesos: 5.362159
## Pesos: -6.699302
## Tasa de aciertos en test: 0.774774774774775
## Partición: 8
## Epoca: 1 - Tasa: 0.770942735683921
## Pesos: 47.4999999999994
## Pesos: -41.9607505
## Pesos: 4.77894999999999
## Pesos: -7.018755
## Tasa de aciertos en test: 0.776223776223776
## Partición: 9
## Epoca: 1 - Tasa: 0.776555861034741
## Pesos: 46.9999999999995
## Pesos: -41.445653
## Pesos: 4.55273999999999
## Pesos: -7.43597599999999
## Tasa de aciertos en test: 0.772772772772773
## Partición: 10
## Epoca: 1 - Tasa: 0.770692673168292
```

```
## Pesos: 47.6499999999994
## Pesos: -42.0656860000001
## Pesos: 5.040652
## Pesos: -6.95709750000001
## Tasa de aciertos en test: 0.775224775224775
```

```
print(glue::glue("Tasa de aciertos media en test: {tasaMedia/10}"))
```

```
## Tasa de aciertos media en test: 0.77289611949612
```

Preguntas:

- ¿Qué beneficio supone el uso de validación cruzada?

El uso de validación cruzada nos permite usar todos los datos para test y poder evaluar de mejor manera si el método utilizado para la generación del modelo es bueno.

- ¿Qué ocurre con la tasa de acierto del perceptrón para los diferentes datasets del ejercicio 2b? Analice el desempeño al incrementarse la dispersión de los datos.

Al aumentar la dispersión de los datos, disminuye la tasa de aciertos porque el plano no permite separar correctamente las clases.