Informe 6: Construyendo modelos

Carmen Lebrero Cia

Construcción de modelo de Machine Learning supervisado con datos de expresión génica

En este informe tenemos como objetivo construir un modelo de Machine Learning Supervisado utilizando los datos de Expresión génica transformados. Además utilizaremos la versión filtrada de estos datos por un lado, y la versión con únicamente los genes diferencialmente expresados por otro.

Obtención y procesado de los datos

Instalación y puesta en marcha de los paquetes necesarios

```
if (!requireNamespace("BiocManager", quietly=TRUE))
    install.packages("BiocManager")
BiocManager::install("TCGAbiolinks")
library(TCGAbiolinks)
library(dplyr)
library(SummarizedExperiment)
library(DT)
```

Descarga de los datos y preparación de objetos R

```
setwd("D:/datos")

QueryExpGenRSEM <- GDCquery(project = "TCGA-KIRC",
                  data.category = "Gene expression",
                  data.type = "Gene expression quantification",
                  experimental.strategy = "RNA-Seq",
                  file.type = "results",
                  legacy = TRUE)
GDCdownload(QueryExpGenRSEM)
ExpGenTCGA_KIRC_RawData <- GDCprepare(QueryExpGenRSEM)
```

Preprocesado

```
#subconjunto quedándonos con las filas con sumatorio de conteos mayores a 1 y guardado de este cambio e
keep <- rowSums(assay(ExpGenTCGA_KIRC_RawData)) > 1
ExpGenTCGA_KIRC_RawData <- ExpGenTCGA_KIRC_RawData[keep,]
nrow(assay(ExpGenTCGA_KIRC_RawData))
```

Normalización

```
#Downstream análisis usando datos de expresión génica de muestras dde TCGA de IlluminaHiSeq_RNASeqV2 co

ExpGenTCGA_KIRC_Norm <- TCGAanalyze_Normalization(tabDF = ExpGenTCGA_KIRC_RawData, geneInfo =  geneInfo
```

Transformación log2

```
ExpGenTCGA_KIRC_Norm_Trans<- log2(ExpGenTCGA_KIRC_Norm+1)
```

```
ExpGenTCGA_KIRC_Norm_Trans[1:10,1:3]
```

```
##          TCGA-B0-5694-01A-11R-1541-07 TCGA-CJ-4637-01A-02R-1325-07
## A1BG                         5.209453                     6.475733
## A2M                         16.117704                    15.538310
## NAT1                         6.807355                     8.209453
## NAT2                         2.584963                     6.988685
## SERPINA3                     8.724514                     9.330917
## AADAC                        0.000000                     6.285402
## AAMP                        12.147523                    12.221285
## AANAT                        1.000000                     0.000000
## AARS                        12.055282                    12.634584
## ABAT                         9.459432                     9.894818
##          TCGA-CZ-4860-01A-01R-1305-07
## A1BG                         5.930737
## A2M                         15.254106
## NAT1                         7.707359
## NAT2                         3.321928
## SERPINA3                    10.615630
## AADAC                        2.321928
## AAMP                        13.512000
## AANAT                        0.000000
## AARS                        14.873588
## ABAT                         8.243174
```

Obtención de los objetos de datos filtrados y DEGs

Filtrado

```
#quantile filter of genes
dim(ExpGenTCGA_KIRC_Norm)
ExpGenTCGA_KIRC_Norm_Filt75 <- TCGAanalyze_Filtering(tabDF = ExpGenTCGA_KIRC_Norm, method = "quantile",
dim(ExpGenTCGA_KIRC_Norm_Filt75)
```

Pasamos de tener 19586 genes a 4897 genes.

```
#quantile filter of genes
dim(ExpGenTCGA_KIRC_Norm_Trans)
ExpGenTCGA_KIRC_Norm_Trans_Filt75 <- TCGAanalyze_Filtering(tabDF = ExpGenTCGA_KIRC_Norm_Trans, method =
dim(ExpGenTCGA_KIRC_Norm_Trans_Filt75)
```

Análisis de expresión diferencial

```
ExpGenTCGA_KIRC_SampleName_DeadStatus <- subset(ExpGenTCGA_KIRC_RawData$barcode, ExpGenTCGA_KIRC_RawData
ExpGenTCGA_KIRC_SampleName_AliveStatus <- subset(ExpGenTCGA_KIRC_RawData$barcode, ExpGenTCGA_KIRC_RawDa
```

```
# Diff.expr.analysis (DEA)
ExpGenTCGA_KIRC_Norm_Filt75_DEGs <- TCGAanalyze_DEA(mat1 = ExpGenTCGA_KIRC_Norm_Filt75[,ExpGenTCGA_KIRC
                            mat2 = ExpGenTCGA_KIRC_Norm_Filt75[,ExpGenTCGA_KIRC_SampleName_AliveStatus]
                            Cond1type = "Dead",
                            Cond2type = "Alive",
                            fdr.cut = 0.10,
                            logFC.cut = 0.4,
                            method = "glmLRT")
```

```
# Diff.expr.analysis (DEA)
ExpGenTCGA_KIRC_Norm_Trans_Filt75_DEGs <- TCGAanalyze_DEA(mat1 = ExpGenTCGA_KIRC_Norm_Trans_Filt75[,Exp
                            mat2 = ExpGenTCGA_KIRC_Norm_Trans_Filt75[,ExpGenTCGA_KIRC_SampleName_AliveS
                            Cond1type = "Dead",
                            Cond2type = "Alive",
                            fdr.cut = 0.10,
                            logFC.cut = 0.4,
                            method = "glmLRT")
```

```
dim(ExpGenTCGA_KIRC_Norm_Filt75_DEGs)
```

```
## [1] 238   5
```

Vamos a hacer un subconjunto de nuestro set de Expresión génica con los 238 genes que hemos visto que tienen expresión diferencial tras el análisis.

```
#Obtención de índices de DEGs en ExpGenTCGA_KIRC_Norm_Filt75 (números de fila)
DEGsNames <- rownames(ExpGenTCGA_KIRC_Norm_Filt75_DEGs)
IndexDEGs_in_ExpGenTCGA_KIRC_Norm_Filt75 <- c()
for (i in DEGsNames){
  IndexDEGs_in_ExpGenTCGA_KIRC_Norm_Filt75 <- c(IndexDEGs_in_ExpGenTCGA_KIRC_Norm_Filt75, which(rowname
}
str(IndexDEGs_in_ExpGenTCGA_KIRC_Norm_Filt75)
```

```
# Subconjunto de toda la matriz ExpGenTCGA_KIRC_Norm_Filt75 de las filas de los DEGs

ExpGenTCGA_KIRC_Norm_Filt75_238DEG <- ExpGenTCGA_KIRC_Norm_Filt75[IndexDEGs_in_ExpGenTCGA_KIRC_Norm_Fil
dim(ExpGenTCGA_KIRC_Norm_Filt75_238DEG)
rownames(ExpGenTCGA_KIRC_Norm_Filt75_238DEG)
```

```
# Subconjunto de toda la matriz ExpGenTCGA_KIRC_Norm_Trans_Filt75 de las filas de los DEGs

ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG <- ExpGenTCGA_KIRC_Norm_Trans_Filt75[IndexDEGs_in_ExpGenTCGA_K
dim(ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG)
```

Construcción del modelo

Instalación de paquetes

```
library(keras)
```

```
##
## Attaching package: 'keras'

## The following object is masked from 'package:BiocGenerics':
##
##     normalize
```

```
install_keras()
```

Introducción al aprendizaje supervisado

El aprendizaje supervisado consiste en "mapear" unos datos de entrada a unas etiquetas *targets* (también conocidas como anotaciones), dados un conjunto de ejemplos anotados por humanos. Casi todas las aplicaciones de Deep Learning actuales pertenecen a esta categoría. Por lo que, necesitaremos unas etiquetas (*Labels*) para nuestros datos, en nuestro caso tomaremos la variable `vital_status`.

Preparación datos, conjuntos de test y train y etiquetas

Vamos a cambiar el array y vamos a poner las muestras como filas y las filas como muestras. Vamos a realizar la prueba del modelo de Deep Learning con dos tipos de datos:

- `ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG_Transposed`: Los datos de Expresion Génica de KIRC, normalizados, transformados, con filtrado de genes al 75% y con los 238 genes más diferencialmente expresados. Número de muestras = 606, número de genes = 238.
- `ExpGenTCGA_KIRC_Norm_Trans_Filt75_Transposed`: Los datos de Expresion Génica de KIRC, normalizados, transformados y con filtrado de genes al 75%. Número de muestras = 606, número de genes = 4897.

```
ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG_Transposed <- t(ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG)
ExpGenTCGA_KIRC_Norm_Trans_Filt75_Transposed <- t(ExpGenTCGA_KIRC_Norm_Trans_Filt75)
```

```
# Creación set train y test para ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG_Transposed

set.seed(231)
ExpGenTCGA_KIRC_Index_Training <- sample(1:nrow(ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG_Transposed), s
```

```
ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG_Transposed_Test <- ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG_Tra
ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG_Transposed_Train <- ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG_Tr

# Creación set train y test para ExpGenTCGA_KIRC_Norm_Trans_Filt75_Transposed

ExpGenTCGA_KIRC_Norm_Trans_Filt75_Transposed_Test <- ExpGenTCGA_KIRC_Norm_Trans_Filt75_Transposed[-ExpG
ExpGenTCGA_KIRC_Norm_Trans_Filt75_Transposed_Train <- ExpGenTCGA_KIRC_Norm_Trans_Filt75_Transposed[ExpG
```

```
# Obtener etiquetas de cada set para ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG_Transposed

ExpGenTCGA_KIRC_Norm_Trans_Filt75_Transposed_Test_Labels <- ExpGenTCGA_KIRC_RawData$vital_status[-ExpGen
ExpGenTCGA_KIRC_Norm_Trans_Filt75_Transposed_Train_Labels <- ExpGenTCGA_KIRC_RawData$vital_status[ExpGen


ExpGenTCGA_KIRC_Norm_Trans_Filt75_Transposed_Test_Labels_FactorNumb <- as.integer(factor(ExpGenTCGA_KIRC
ExpGenTCGA_KIRC_Norm_Trans_Filt75_Transposed_Test_Labels_FactorNumb[ExpGenTCGA_KIRC_Norm_Trans_Filt75_Tr


ExpGenTCGA_KIRC_Norm_Trans_Filt75_Transposed_Test_Labels_FactorNumb <- as.integer(factor(ExpGenTCGA_KIRC
ExpGenTCGA_KIRC_Norm_Trans_Filt75_Transposed_Test_Labels_FactorNumb[ExpGenTCGA_KIRC_Norm_Trans_Filt75_Tr

ExpGenTCGA_KIRC_Norm_Trans_Filt75_Transposed_Train_Labels_FactorNumb <- as.integer(factor(ExpGenTCGA_KIR
ExpGenTCGA_KIRC_Norm_Trans_Filt75_Transposed_Train_Labels_FactorNumb[ExpGenTCGA_KIRC_Norm_Trans_Filt75_


# Dado que las muestras son las mismas en ambas muestras, porque coinciden no hace falta obtener las et
```

Creación del modelo

Modelo 1

```
# Definición del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG
Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG <- keras_model_sequential() %>%
  layer_dense(units = 16, activation = "relu", input_shape = c(238)) %>%
  layer_dense(units = 16, activation = "relu") %>%
  layer_dense(units = 1, activation = "sigmoid")
```

```
summary(Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG)
```

```
## Model: "sequential"
## _________________________________________________________________________
## Layer (type)                 Output Shape              Param #
## =========================================================================
## dense_2 (Dense)              (None, 16)                3824
## _________________________________________________________________________
## dense_1 (Dense)              (None, 16)                272
## _________________________________________________________________________
## dense (Dense)                (None, 1)                 17
## =========================================================================
## Total params: 4,113
```

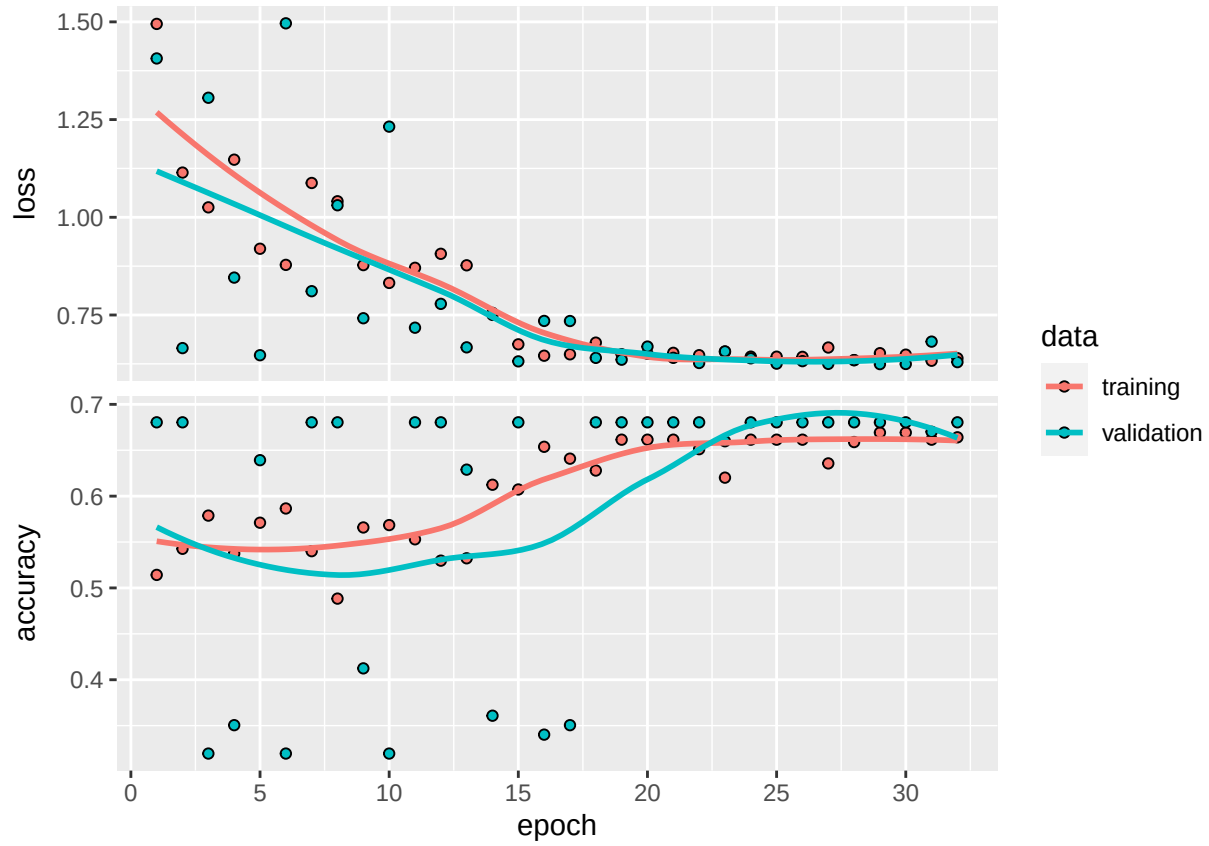
```
## Trainable params: 4,113
## Non-trainable params: 0
## _________________________________________________________________________________
```

```
# Compilación del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG

Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG  %>% compile(
  optimizer = "rmsprop",
  loss = "binary_crossentropy",
  metrics = c("accuracy")
)
```

```
# Entrenamiento del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG

history <- Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG %>% fit(ExpGenTCGA_KIRC_Norm_Trans_Filt75_238
                                                                  validation_split = 0.2)
```

```
plot(history)
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
# Evaluación del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG

Results_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG <- Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG %>% 
Results_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG
```

```
##      loss  accuracy
## 0.6201108 0.6721311
```

Buscando el overfitting del modelo

Crear un modelo que se sobreajuste es bastante fácil:

1. Añadir capas
2. Hacer las capas más grandes
3. Entrenar para más iteraciones

Modelo 2

```
# Definición del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG
Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG <- keras_model_sequential() %>%
  layer_dense(units = 200, activation = "relu", input_shape = c(238)) %>%
  layer_dense(units = 100, activation = "relu") %>%
  layer_dense(units = 50, activation = "relu") %>%
  layer_dense(units = 1, activation = "sigmoid")
```

```
summary(Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG)
```

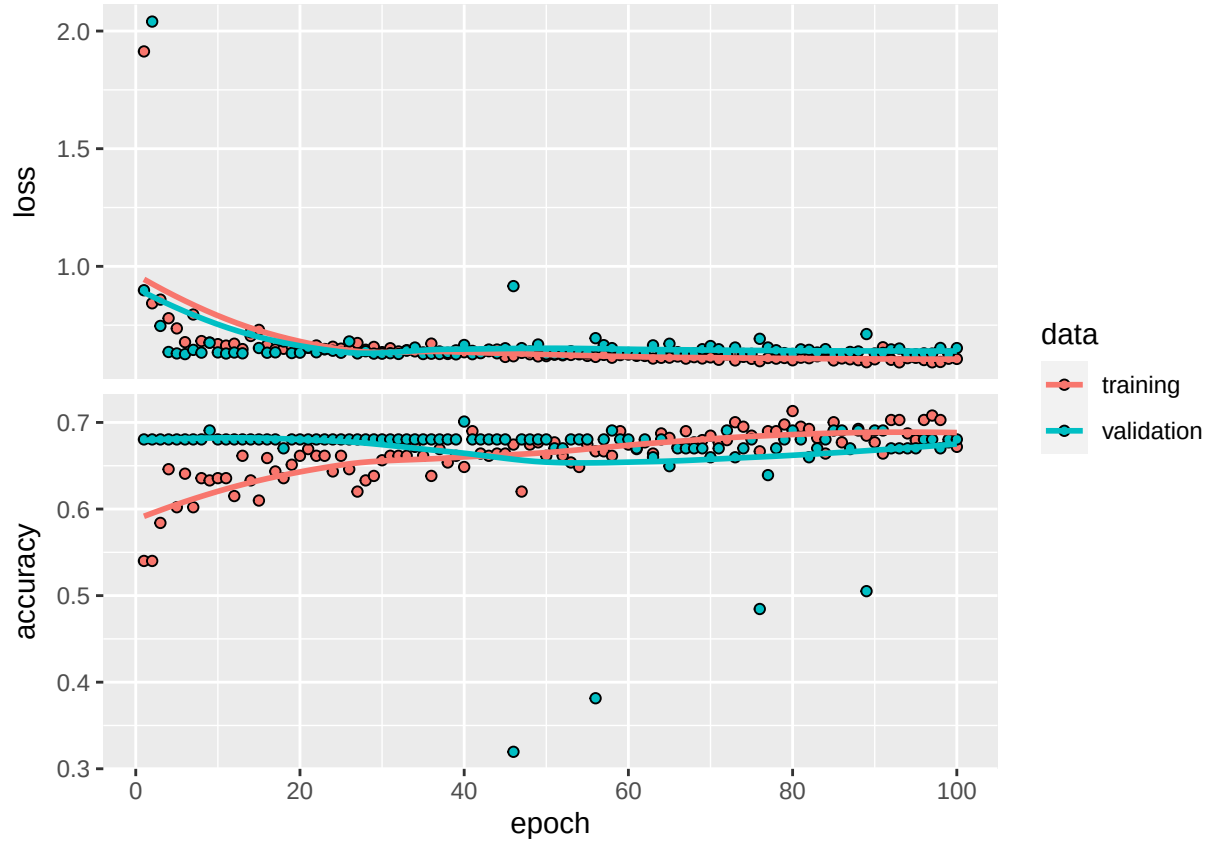
```
## Model: "sequential_1"
## ________________________________________________________________________________
## Layer (type)                        Output Shape                    Param #
## ================================================================================
## dense_6 (Dense)                     (None, 200)                     47800
## ________________________________________________________________________________
## dense_5 (Dense)                     (None, 100)                     20100
## ________________________________________________________________________________
## dense_4 (Dense)                     (None, 50)                      5050
## ________________________________________________________________________________
## dense_3 (Dense)                     (None, 1)                       51
## ================================================================================
## Total params: 73,001
## Trainable params: 73,001
## Non-trainable params: 0
## ________________________________________________________________________________
```

```
# Compilación del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG

Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG  %>% compile(
  optimizer = "rmsprop",
  loss = "binary_crossentropy",
  metrics = c("accuracy")
)
```

```
# Entrenamiento del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG

history <- Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG %>% fit(ExpGenTCGA_KIRC_Norm_Trans_Filt75_238
                                                                  validation_split = 0.2)
```

```
plot(history)
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
# Evaluación del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG

Results_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG <- Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG %>% 
Results_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG
```

```
##      loss  accuracy
## 0.6128544 0.6721311
```

Modelo 3

```
# Definición del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG
Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG <- keras_model_sequential() %>%
  layer_dense(units = 400, activation = "relu", input_shape = c(238)) %>%
```

```
  layer_dense(units = 200, activation = "relu") %>%
  layer_dense(units = 100, activation = "relu") %>%
  layer_dense(units = 1, activation = "sigmoid")
```

```
summary(Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG)
```

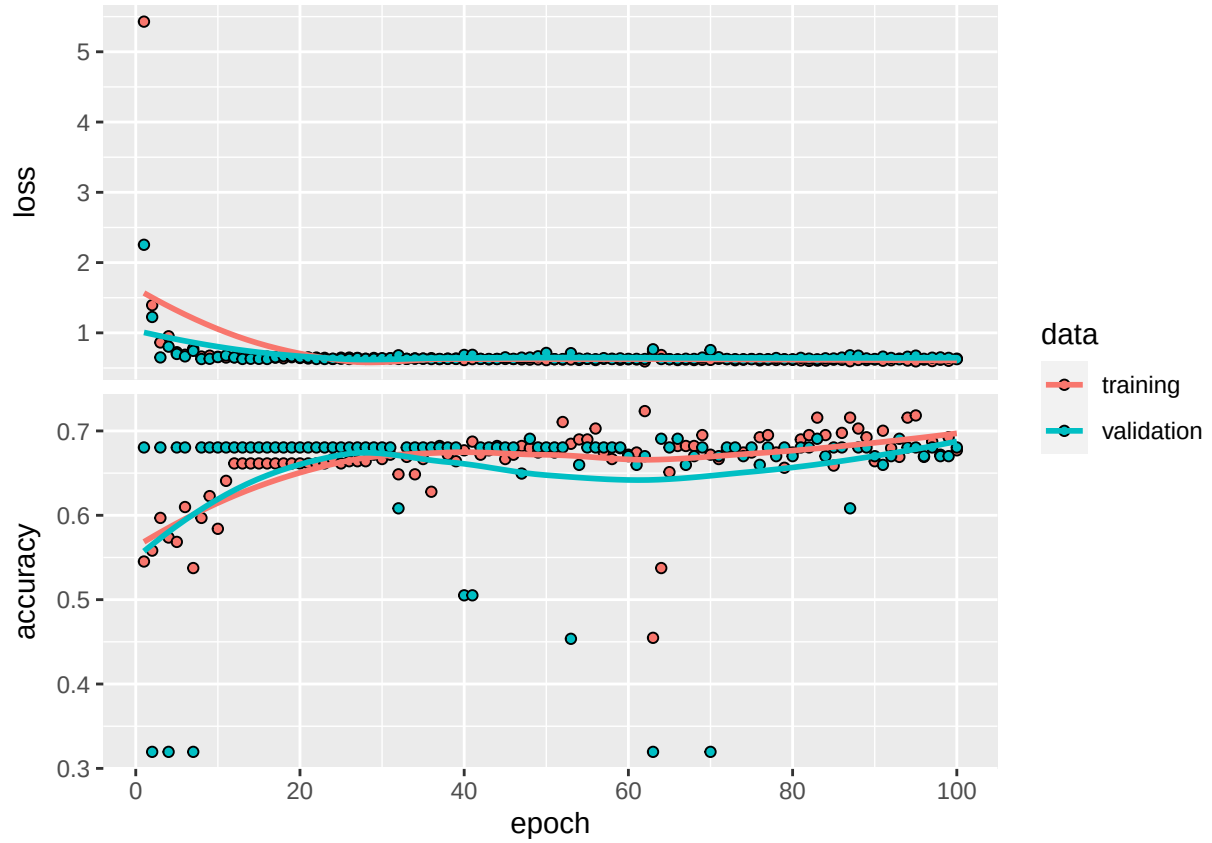
```
## Model: "sequential_2"
## _________________________________________________________________________
## Layer (type)                 Output Shape              Param #
## =========================================================================
## dense_10 (Dense)             (None, 400)               95600
## _________________________________________________________________________
## dense_9 (Dense)              (None, 200)               80200
## _________________________________________________________________________
## dense_8 (Dense)              (None, 100)               20100
## _________________________________________________________________________
## dense_7 (Dense)              (None, 1)                 101
## =========================================================================
## Total params: 196,001
## Trainable params: 196,001
## Non-trainable params: 0
## _________________________________________________________________________
```

```
# Compilación del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG

Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG  %>% compile(
  optimizer = "rmsprop",
  loss = "binary_crossentropy",
  metrics = c("accuracy")
)
```

```
# Entrenamiento del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG

history <- Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG %>% fit(ExpGenTCGA_KIRC_Norm_Trans_Filt75_238
                                                                  validation_split = 0.2)
```

```
plot(history)
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
# Evaluación del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG

Results_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG <- Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG %>% e
Results_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG
```

```
##      loss  accuracy
## 0.5909629 0.6803279
```

Modelo 4

```
# Definición del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG
Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG <- keras_model_sequential() %>%
  layer_dense(units = 400, activation = "relu", input_shape = c(238)) %>%
  layer_dense(units = 200, activation = "relu") %>%
  layer_dense(units = 100, activation = "relu") %>%
  layer_dense(units = 100, activation = "relu") %>%
  layer_dense(units = 100, activation = "relu") %>%
  layer_dense(units = 100, activation = "relu") %>%
  layer_dense(units = 1, activation = "sigmoid")
```

```
summary(Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG)
```

```
## Model: "sequential_3"
```

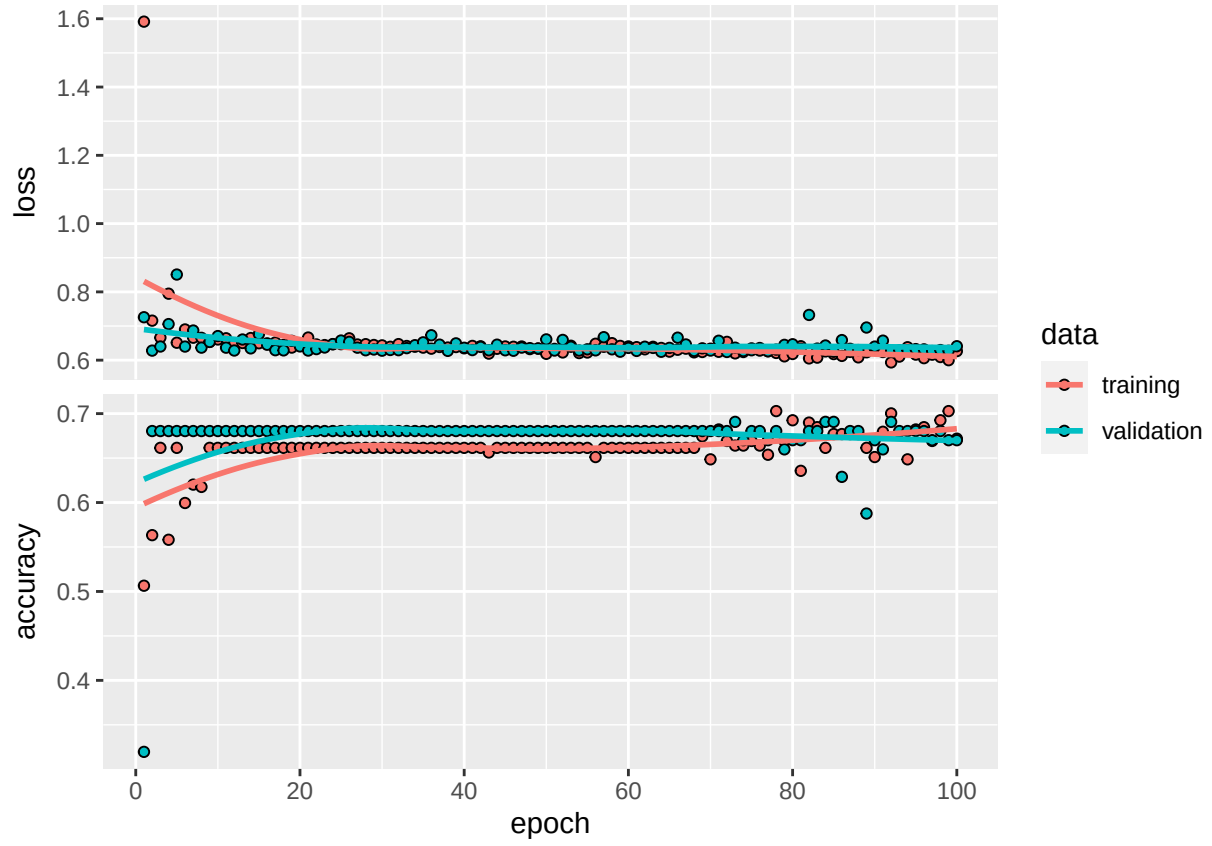
```
## ________________________________________________________________________________
## Layer (type)                        Output Shape                    Param #
## ================================================================================
## dense_17 (Dense)                    (None, 400)                     95600
## ________________________________________________________________________________
## dense_16 (Dense)                    (None, 200)                     80200
## ________________________________________________________________________________
## dense_15 (Dense)                    (None, 100)                     20100
## ________________________________________________________________________________
## dense_14 (Dense)                    (None, 100)                     10100
## ________________________________________________________________________________
## dense_13 (Dense)                    (None, 100)                     10100
## ________________________________________________________________________________
## dense_12 (Dense)                    (None, 100)                     10100
## ________________________________________________________________________________
## dense_11 (Dense)                    (None, 1)                       101
## ================================================================================
## Total params: 226,301
## Trainable params: 226,301
## Non-trainable params: 0
## ________________________________________________________________________________
```

```
# Compilación del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG

Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG  %>% compile(
  optimizer = "rmsprop",
  loss = "binary_crossentropy",
  metrics = c("accuracy")
)
```

```
# Entrenamiento del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG

history <- Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG %>% fit(ExpGenTCGA_KIRC_Norm_Trans_Filt75_238
                                                                  validation_split = 0.2)
```

```
plot(history)
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
# Evaluación del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG

Results_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG <- Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG %>% e
Results_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG
```

```
##      loss  accuracy
## 0.6019629 0.7049180
```

Modelo 5

```
# Definición del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG
Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG <- keras_model_sequential() %>%
  layer_dense(units = 200, activation = "relu", input_shape = c(238)) %>%
  layer_dense(units = 100, activation = "relu") %>%
  layer_dense(units = 50, activation = "relu") %>%
  layer_dense(units = 20, activation = "relu") %>%
  layer_dense(units = 1, activation = "sigmoid")
```

```
summary(Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG)
```

```
## Model: "sequential_4"
## ________________________________________________________________________________
```

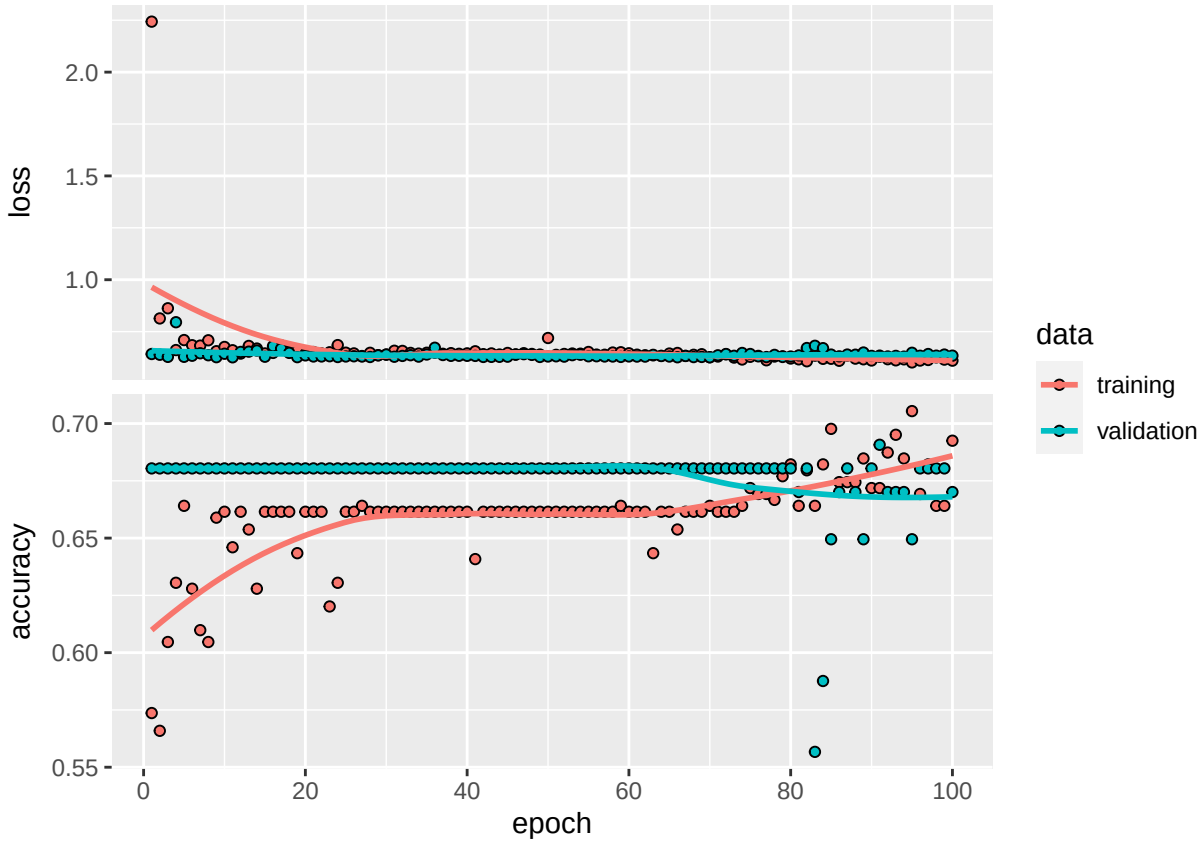
```
## Layer (type)                        Output Shape                      Param #
## ================================================================================
## dense_22 (Dense)                    (None, 200)                       47800
## ________________________________________________________________________________
## dense_21 (Dense)                    (None, 100)                       20100
## ________________________________________________________________________________
## dense_20 (Dense)                    (None, 50)                        5050
## ________________________________________________________________________________
## dense_19 (Dense)                    (None, 20)                        1020
## ________________________________________________________________________________
## dense_18 (Dense)                    (None, 1)                         21
## ================================================================================
## Total params: 73,991
## Trainable params: 73,991
## Non-trainable params: 0
## ________________________________________________________________________________
```

```
# Compilación del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG

Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG  %>% compile(
  optimizer = "rmsprop",
  loss = "binary_crossentropy",
  metrics = c("accuracy")
)
```

```
# Entrenamiento del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG

history <- Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG %>% fit(ExpGenTCGA_KIRC_Norm_Trans_Filt75_238
                                                                  validation_split = 0.2)
```

```
plot(history)
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
# Evaluación del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG

Results_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG <- Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG %>% e
Results_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG
```

```
##      loss  accuracy
## 0.5889738 0.6721311
```

Modelo 6

```
# Definición del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG
Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG <- keras_model_sequential() %>%
  layer_dense(units = 200, activation = "relu", input_shape = c(238)) %>%
  layer_dense(units = 100, activation = "relu") %>%
  layer_dense(units = 50, activation = "relu") %>%
  layer_dense(units = 20, activation = "relu") %>%
  layer_dense(units = 10, activation = "relu") %>%
  layer_dense(units = 5, activation = "relu") %>%
  layer_dense(units = 1, activation = "sigmoid")
```

```
summary(Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG)
```

```
## Model: "sequential_5"
```

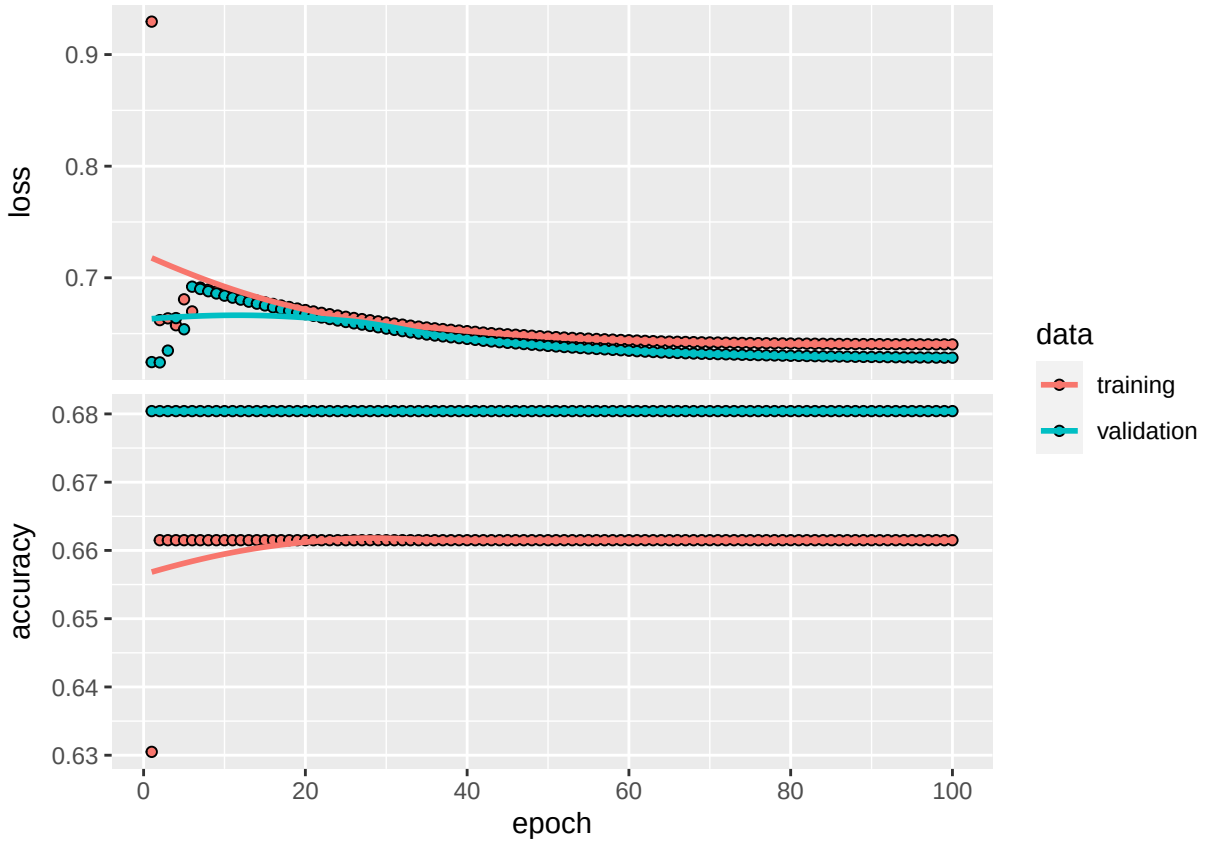
```
## _________________________________________________________________________
## Layer (type)                    Output Shape                  Param #
## =========================================================================
## dense_29 (Dense)                (None, 200)                   47800
## _________________________________________________________________________
## dense_28 (Dense)                (None, 100)                   20100
## _________________________________________________________________________
## dense_27 (Dense)                (None, 50)                    5050
## _________________________________________________________________________
## dense_26 (Dense)                (None, 20)                    1020
## _________________________________________________________________________
## dense_25 (Dense)                (None, 10)                    210
## _________________________________________________________________________
## dense_24 (Dense)                (None, 5)                     55
## _________________________________________________________________________
## dense_23 (Dense)                (None, 1)                     6
## =========================================================================
## Total params: 74,241
## Trainable params: 74,241
## Non-trainable params: 0
## _________________________________________________________________________
```

```
# Compilación del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG

Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG  %>% compile(
  optimizer = "rmsprop",
  loss = "binary_crossentropy",
  metrics = c("accuracy")
)
```

```
# Entrenamiento del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG

history <- Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG %>% fit(ExpGenTCGA_KIRC_Norm_Trans_Filt75_238
                                                                  validation_split = 0.2)
```

```
plot(history)
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
# Evaluación del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG

Results_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG <- Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG %>% e
Results_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG
```

```
##      loss  accuracy
## 0.6335880 0.6721311
```

Modelo 7

```
# Definición del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG
Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG <- keras_model_sequential() %>%
  layer_dense(units = 500, activation = "relu", input_shape = c(238)) %>%
  layer_dense(units = 300, activation = "relu") %>%
  layer_dense(units = 100, activation = "relu") %>%
  layer_dense(units = 50, activation = "relu") %>%
  layer_dense(units = 1, activation = "sigmoid")
```

```
summary(Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG)
```

```
## Model: "sequential_6"
## ________________________________________________________________________________
```

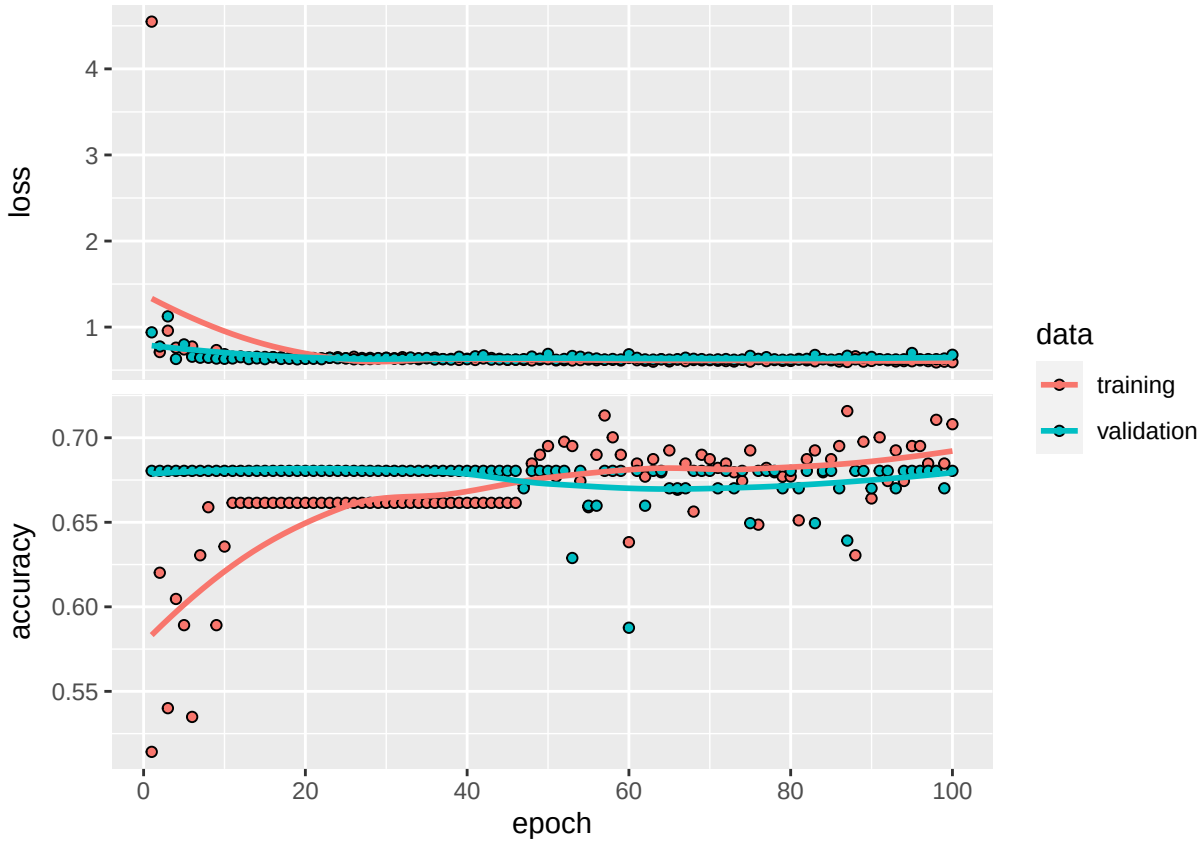
```
## Layer (type)                        Output Shape                      Param #
## ================================================================================
## dense_34 (Dense)                    (None, 500)                       119500
## ________________________________________________________________________________
## dense_33 (Dense)                    (None, 300)                       150300
## ________________________________________________________________________________
## dense_32 (Dense)                    (None, 100)                       30100
## ________________________________________________________________________________
## dense_31 (Dense)                    (None, 50)                        5050
## ________________________________________________________________________________
## dense_30 (Dense)                    (None, 1)                         51
## ================================================================================
## Total params: 305,001
## Trainable params: 305,001
## Non-trainable params: 0
## ________________________________________________________________________________
```

```
# Compilación del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG

Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG  %>% compile(
  optimizer = "rmsprop",
  loss = "binary_crossentropy",
  metrics = c("accuracy")
)
```

```
# Entrenamiento del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG

history <- Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG %>% fit(ExpGenTCGA_KIRC_Norm_Trans_Filt75_238
                                                                  validation_split = 0.2)
```

```
plot(history)
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
# Evaluación del modelo ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG

Results_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG <- Model_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG %>% e
Results_ExpGenTCGA_KIRC_Norm_Trans_Filt75_238DEG
```

```
##      loss  accuracy
## 0.6445902 0.6803279
```

tfruns: Track and Visualize Training Runs

Una mejor manera de comparar los entrenamientos es con el paquete tfruns.

- Hace un seguimiento de los hiperparámetros, métricas, resultados y código fuente de cada entrenamiento.
- Compara hiperparámetros y métricas de todos los runs y encuentra el mejor modelo.
- Genera automáticamente informes para visualizar indivualmente los entrenamientos o comparación entre runs.
- No se necesita cambiar el código fuente.

Instalación

```
install.packages("tfruns")
install.packages("tfestimators")
```

Entrenamiento

En las siguientes secciones se describirán las capacidades de `tfruns`. Nuestro script de entrenamiento `script.R` entrena un modelo Keras para clasificar pacientes entre vivos y muertos a partir de datos de Expresión génica.

Para entrenar la red con **tfruns** utiliza la función `training_run()` para ejecutar el script de R.

```
library(tfestimators)
library(tfruns)
training_run("script.R")
```

Cuando el entrenamiento se haya completado, un resumen de la ejecución aparecerá en pantalla si estás en una sesión interactiva de R.

Enlaces de interés

https://riuma.uma.es/xmlui/bitstream/handle/10630/13942/103060005.pdf;jsessionid=CBBC364B3A641B87C959E0E5C42
sequence=1 https://tensorflow.rstudio.com/tools/tfruns/overview/ https://tensorflow.rstudio.com/tools/
tensorboard/tensorboard/