04/26/2020

Carlos Alberto Cordero Robles

Párrafo de la tarea

Utilizar los vistos en clase para generar una aplicación de redes neuronales, generar un reporte.

Introducción

En esta tarea se intentará desarrollar un código en R para la implementación de redes neuronales. El objetivo del modelo a desarrollar es crear un sistema que tome como entrada un dataset con especificaciones de características de tumores mamarios y en base a dicho dataset entrenado determinar si el tumor es benigno o cancerígeno. Se espera una eficiencia mayor al 90%.

Desarrollo

El dataset es de la universidad de Irvine california [1] he incluye datos relacionados con tumores mamarios y fueron clasificados si fueron cancerígenos o no, los datos de entrada son los siguientes:

* "Clump\_Thickness"
* "Uniformity\_Cell\_Size"
* "Uniformity\_Cell\_Shape"
* "Marginal\_Adhesion"
* "Single\_Epithelial\_Cell\_Size"
* "Bare\_Nuclei"
* "Bland\_Chromatin"
* "Normal\_Nucleoli"
* "Mitoses"

La clasificación esta en la columna "CellClass\_2\_benign4\_malignant" y como el nombre lo dice un 2 sera benigno y un 4 maligno.

El primer paso es leer la información y verificar que no falta algún dato.

|  |
| --- |
| getwd()  currentPath <- dirname(rstudioapi::getSourceEditorContext()$path)  setwd(currentPath)  dataset.data <- read.csv("breast-cancer.csv")  dim(dataset.data)  ################  ########Clean Dataset  ################  for (row in 1:nrow(dataset.data)){  if(10 !=length(dataset.data[row,])){  print(row)  }  } |

Como no imprimió nada entonces esta completa la información.

Ahora sigue limpiar la información. Los datos deben ser de 1 a 10 y en el caso de la clasificación solo 2 o 4. Haciendo un análisis con summary vi que los datos de entrada son correctos, pero en la clasificación hay datos no reconocidos como 21 o 41, dichas muestras fueron removidas del estudio.

|  |
| --- |
| for (row in 1:nrow(dataset.data)){  if(10 !=length(dataset.data[row,])){  print(row)  }  }  #Data is complete  summary(dataset.data)  # We have some errors in CellClass\_2\_benign4\_malignant it should be just 2 or 4 and we have 41  # lets clean it, all the ones with value diferent to 2 or 4 will be removed  dirty.data <- c()  for (row in 1:nrow(dataset.data)){  if((4 != dataset.data[row,"CellClass\_2\_benign4\_malignant"]) && (2 != dataset.data[row,"CellClass\_2\_benign4\_malignant"])){  dirty.data = c(dirty.data,row)  }  } |

El siguiente paso fue checar la correlación, donde se checo que si la correlación de dos valores era mayor a 80% entonces se discriminarían.

|  |
| --- |
| data.correlation <- cor(dataset.data.clean)  data.Names <- colnames(data.correlation)  for (row in 1:nrow(data.correlation)){  for (column in 1:ncol(data.correlation)){  if((0.8 < data.correlation[row,column]) && (row != column)){  print("Hight correlation in:")  print(data.Names[row])  print(data.Names[column])  }  }  } |

El resultado fue el siguiente:

|  |
| --- |
| "Hight correlation in:"  "Uniformity\_Cell\_Size"  "Uniformity\_Cell\_Shape"  "Hight correlation in:"  "Uniformity\_Cell\_Size"  "CellClass\_2\_benign4\_malignant"  "Hight correlation in:"  "Uniformity\_Cell\_Shape"  "Uniformity\_Cell\_Size"  "Hight correlation in:"  "Uniformity\_Cell\_Shape"  "CellClass\_2\_benign4\_malignant"  "Hight correlation in:"  "Bare\_Nuclei"  "CellClass\_2\_benign4\_malignant"  "Hight correlation in:"  "CellClass\_2\_benign4\_malignant"  "Uniformity\_Cell\_Size"  "Hight correlation in:"  "CellClass\_2\_benign4\_malignant"  "Uniformity\_Cell\_Shape"  "Hight correlation in:"  "CellClass\_2\_benign4\_malignant"  "Bare\_Nuclei" |

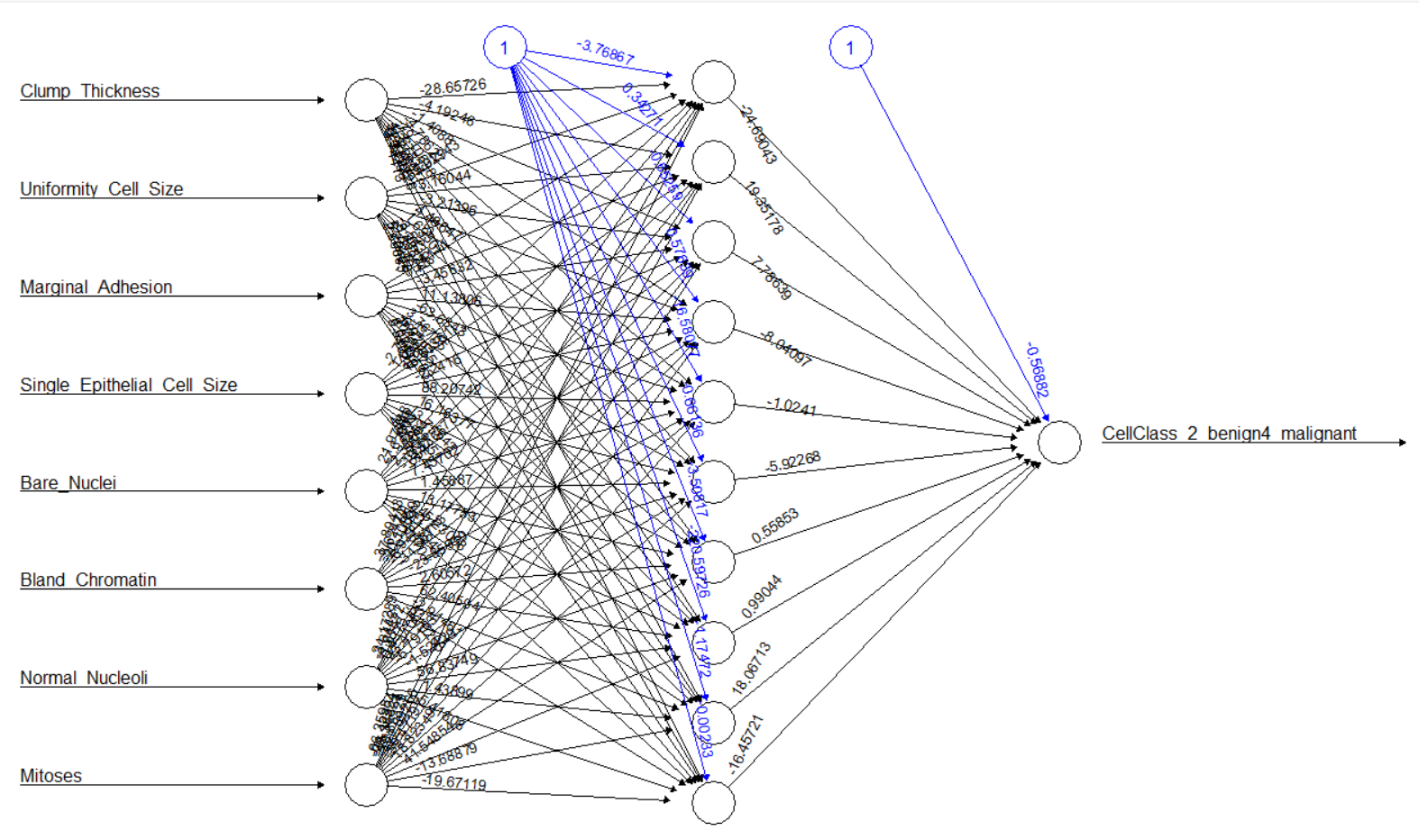
Las correlaciones con la salida se ignorarán, pero "Uniformity\_Cell\_Size" y "Uniformity\_Cell\_Shape" tienen alta correlación así que ignoraremos en esta ocasión la shape.

Ahora normalizaremos y separaremos en set de entrenamiento y de pruebas. Para normalizar dividimos simplemente entre 10 ya que es el valor máximo en todos los campos. Para crear los sets será aleatorio tomando el 20% para validación y 80% para entrenamiento.

|  |
| --- |
| dataset.data.clean.norm = dataset.data.clean/10  for (row in 1:nrow(dataset.data.clean.norm)){  if(0.2 == dataset.data.clean.norm[row,"CellClass\_2\_benign4\_malignant"]){  dataset.data.clean.norm[row,"CellClass\_2\_benign4\_malignant"] = 0  }  else  {  dataset.data.clean.norm[row,"CellClass\_2\_benign4\_malignant"] = 1  }  }  # We are going to take 20% for validation  set.seed(1234)  val.Index <- sample(1:nrow(dataset.data.clean.norm), 140)  val.sample <- dataset.data.clean.norm[val.Index,]  dataset.data.clean.norm.train <- dataset.data.clean.norm[-val.Index,]  dim(val.sample)  dim(dataset.data.clean.norm.train)  head(dataset.data.clean.norm.train)  summary(dataset.data.clean.norm.train) |

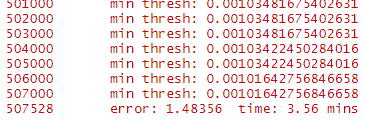
El siguiente paso será entrenar donde se usará la librería neuralnet.

|  |
| --- |
| formula.cancer <- CellClass\_2\_benign4\_malignant~Clump\_Thickness+Uniformity\_Cell\_Size+Marginal\_Adhesion+Single\_Epithelial\_Cell\_Size+Bare\_Nuclei+Bland\_Chromatin+Normal\_Nucleoli+Mitoses  ?neuralnet  #set.seed(1234)  #nn.cancer <- neuralnet(formula=formula.cancer,  # data = dataset.data.clean.norm.train,  # hidden=10, threshold=0.001,  # linear.output = T,  # stepmax=1e+06,  # lifesign = "full")  #plot(nn.cancer)  saveRDS(nn.cancer, file="modelweights.rds")  nn.cancer <- readRDS("modelweights.rds")  plot(nn.cancer) |



Como podemos notar se sacó "Uniformity\_Cell\_Shape" del análisis, se usaron 10 células de capa intermedia, y se espera llegar a un error de 0.001

El resultado fue el siguiente



El entrenamiento duro aproximadamente 4 minutos.

Se guardarán los pesos para evitar la necesidad de volver a entrenar.

Por último, se evaluarán los resultados con el set de validación. Para esto usamos el modelo generado y vemos las salidas que da con el dataset de validación. A ese resultado si es mayor a 0.5 lo redondearemos a 1 y si es menor a 0.5 lo redondearemos a 0.

Al comparar la predicción con el resultado nos da 134 aciertos y 6 fallos que significa una eficiencia del 95%.

|  |
| --- |
| cancer.prediction.results <- compute(nn.cancer, val.sample[,c("Clump\_Thickness","Uniformity\_Cell\_Size","Marginal\_Adhesion","Single\_Epithelial\_Cell\_Size","Bare\_Nuclei","Bland\_Chromatin","Normal\_Nucleoli","Mitoses")])  results <- cancer.prediction.results$net.result  for(element in 1:length(results)){  if( 0.5 < results[element,1]){  results[element,1] = 1  }  else  {  results[element,1] = 0  }  }  comparacion <- results == val.sample[,"CellClass\_2\_benign4\_malignant"]  comparacion.num <- as.numeric(comparacion)  sum(comparacion.num)  eficient.factor <- sum(comparacion.num) / 140 |



Conclusiones

\* En un dataset real casi siempre es necesario limpiar el dataset antes de usarlo.

\* Los datos con alta correlación es bueno sacarlos del estudio ya que aportan poca información adicional.

\* Es bastante útil normalizar la información.

\* El entrenamiento puede durar bastante tiempo.

\* Una vez hecho el modelo es bueno guardarlo para no tener que ejecutar cada vez que se haga el análisis.

\* Las redes neuronales resultaron de ser muy eficientes para este problema en particular de clasificación.

Referencias

[1]« Breast Cancer Data Set» **[En línea]. Disponible en:**

<http://mlr.cs.umass.edu/ml/datasets/Breast+Cancer> [Accedido: 26-Abl-2020].

[2]« W.H. Wolberg, W.N. Street, and O.L. Mangasarian. Machine learning techniques to diagnose breast cancer from fine-needle aspirates. Cancer Letters 77 (1994) 163-171. »

[3]« Saving and loading a model in R**» [En línea]. Disponible en:**

<https://stackoverflow.com/questions/14761496/saving-and-loading-a-model-in-r> [Accedido: 26-Abl-2020].