Homework1

Quang Loc Lam

16 February 2019

library(keras)  
  
imdb <- dataset\_imdb(num\_words = 10000)  
c(c(train\_data, train\_labels), c(test\_data, test\_labels)) %<-% imdb

str(train\_data[[1]])

## int [1:218] 1 14 22 16 43 530 973 1622 1385 65 ...

train\_labels[[1]]

## [1] 1

max(sapply(train\_data, max))

## [1] 9999

# word\_index is a dictionary mapping words to an integer index  
word\_index <- dataset\_imdb\_word\_index()  
# We reverse it, mapping integer indices to words  
reverse\_word\_index <- names(word\_index)  
names(reverse\_word\_index) <- word\_index  
# We decode the review; note that our indices were offset by 3  
# because 0, 1 and 2 are reserved indices for "padding", "start of sequence", and "unknown".  
decoded\_review <- sapply(train\_data[[1]], function(index) {  
 word <- if (index >= 3) reverse\_word\_index[[as.character(index - 3)]]  
 if (!is.null(word)) word else "?"  
})

cat(decoded\_review)

## ? this film was just brilliant casting location scenery story direction everyone's really suited the part they played and you could just imagine being there robert ? is an amazing actor and now the same being director ? father came from the same scottish island as myself so i loved the fact there was a real connection with this film the witty remarks throughout the film were great it was just brilliant so much that i bought the film as soon as it was released for ? and would recommend it to everyone to watch and the fly fishing was amazing really cried at the end it was so sad and you know what they say if you cry at a film it must have been good and this definitely was also ? to the two little boy's that played the ? of norman and paul they were just brilliant children are often left out of the ? list i think because the stars that play them all grown up are such a big profile for the whole film but these children are amazing and should be praised for what they have done don't you think the whole story was so lovely because it was true and was someone's life after all that was shared with us all

vectorize\_sequences <- function(sequences, dimension = 10000) {  
 # Create an all-zero matrix of shape (len(sequences), dimension)  
 results <- matrix(0, nrow = length(sequences), ncol = dimension)  
 for (i in 1:length(sequences))  
 # Sets specific indices of results[i] to 1s  
 results[i, sequences[[i]]] <- 1  
 results  
}  
  
# Our vectorized training data  
x\_train <- vectorize\_sequences(train\_data)  
# Our vectorized test data  
x\_test <- vectorize\_sequences(test\_data)

str(x\_train[1,])

## num [1:10000] 1 1 0 1 1 1 1 1 1 0 ...

# Our vectorized labels  
y\_train <- as.numeric(train\_labels)  
y\_test <- as.numeric(test\_labels)

library(keras)  
  
model <- keras\_model\_sequential() %>%   
 layer\_dense(units = 5, activation = "relu", input\_shape = c(10000)) %>%   
 layer\_dense(units = 5, activation = "relu") %>%   
 layer\_dense(units = 5, activation = "relu") %>%   
 layer\_dense(units = 1, activation = "sigmoid")

model %>% compile(  
 optimizer = "rmsprop",  
 loss = "mse",  
 metrics = c("accuracy")  
)

model %>% compile(  
 optimizer = optimizer\_rmsprop(lr=0.001),  
 loss = "mse",  
 metrics = c("accuracy")  
)

model %>% compile(  
 optimizer = optimizer\_rmsprop(lr = 0.001),  
 loss = loss\_mean\_squared\_error,  
 metrics = metric\_binary\_accuracy  
)

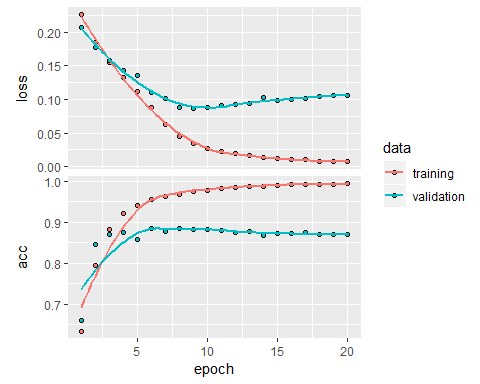
val\_indices <- 1:10000  
  
x\_val <- x\_train[val\_indices,]  
partial\_x\_train <- x\_train[-val\_indices,]  
  
y\_val <- y\_train[val\_indices]  
partial\_y\_train <- y\_train[-val\_indices]

model %>% compile(  
 optimizer = "rmsprop",  
 loss = "mse",  
 metrics = c("accuracy")  
)  
  
history <- model %>% fit(  
 partial\_x\_train,  
 partial\_y\_train,  
 epochs = 20,  
 batch\_size = 512,  
 validation\_data = list(x\_val, y\_val)  
)

str(history)

## List of 2  
## $ params :List of 8  
## ..$ batch\_size : int 512  
## ..$ epochs : int 20  
## ..$ steps : NULL  
## ..$ samples : int 15000  
## ..$ verbose : int 1  
## ..$ do\_validation : logi TRUE  
## ..$ metrics : chr [1:4] "loss" "acc" "val\_loss" "val\_acc"  
## ..$ validation\_samples: int 10000  
## $ metrics:List of 4  
## ..$ val\_loss: num [1:20] 0.208 0.177 0.158 0.143 0.135 ...  
## ..$ val\_acc : num [1:20] 0.66 0.845 0.869 0.876 0.857 ...  
## ..$ loss : num [1:20] 0.226 0.185 0.156 0.133 0.111 ...  
## ..$ acc : num [1:20] 0.634 0.795 0.884 0.922 0.942 ...  
## - attr(\*, "class")= chr "keras\_training\_history"

plot(history)



model <- keras\_model\_sequential() %>%   
 layer\_dense(units = 5, activation = "relu", input\_shape = c(10000)) %>%   
 layer\_dense(units = 5, activation = "relu") %>%   
 layer\_dense(units = 5, activation = "relu") %>%   
 layer\_dense(units = 1, activation = "sigmoid")  
  
model %>% compile(  
 optimizer = "rmsprop",  
 loss = "mse",  
 metrics = c("accuracy")  
)  
  
model %>% fit(x\_train, y\_train, epochs = 4, batch\_size = 512)  
results <- model %>% evaluate(x\_test, y\_test)

results

## $loss  
## [1] 0.08445174  
##   
## $acc  
## [1] 0.89084

model %>% predict(x\_test[1:10,])

## [,1]  
## [1,] 0.31560174  
## [2,] 0.99880910  
## [3,] 0.83304143  
## [4,] 0.70225352  
## [5,] 0.89829332  
## [6,] 0.64159095  
## [7,] 0.99684250  
## [8,] 0.04561223  
## [9,] 0.94962174  
## [10,] 0.95083147