thesis

Rujia Yuan

3/3/2022

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.1.2

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tm)

## Warning: package 'tm' was built under R version 4.1.2

## Loading required package: NLP

library(stringr)

## Warning: package 'stringr' was built under R version 4.1.2

library(stringi)

## Warning: package 'stringi' was built under R version 4.1.2

library(tidyr)

## Warning: package 'tidyr' was built under R version 4.1.2

library(keras)

## Warning: package 'keras' was built under R version 4.1.3

library(tensorflow)

## Warning: package 'tensorflow' was built under R version 4.1.3

library(devtools)

## Warning: package 'devtools' was built under R version 4.1.3

## Loading required package: usethis

## Warning: package 'usethis' was built under R version 4.1.3

library(reticulate)

## Warning: package 'reticulate' was built under R version 4.1.3

getwd()

## [1] "C:/Users/Jadeyuan/Desktop/sta391"

#Setup the working directory:  
setwd("C:/Users/Jadeyuan/Desktop/sta391")

xtrain <- read.table("X\_train.txt", header = FALSE, sep = "")

ytrain <- read.table("y\_train.txt", header = FALSE, sep = "")

xtest <- read.table("X\_test.txt", header = FALSE, sep = "")

ytest <- read.table("y\_test.txt", header = FALSE, sep = "")

xtrain <- na.omit(xtrain)

sum(is.na(xtrain))

## [1] 0

dim(xtrain)

## [1] 7767 561

ytrain <- na.omit(ytrain)

sum(is.na(ytrain))

## [1] 0

dim(ytrain)

## [1] 7767 1

xtest <- na.omit(xtest)

sum(is.na(xtest))

## [1] 0

dim(xtest)

## [1] 3162 561

ytest <- na.omit(ytest)

sum(is.na(ytest))

## [1] 0

dim(ytest)

## [1] 3162 1

ytrain <- as.data.frame(ytrain)

dim(ytrain)

## [1] 7767 1

set.seed(1234)  
ynew = matrix(0,nrow=7767,ncol = 12)  
for (i in 1:7767){  
 ynew[i,ytrain[i,1]] = 1  
}

dim(ynew)

## [1] 7767 12

model <- keras\_model\_sequential()

## Loaded Tensorflow version 2.6.0

model %>%   
 layer\_dense(name = "DeepLayer1",  
 units = 256,  
 activation = "relu",  
 input\_shape = dim(xtrain)[2]) %>%   
 layer\_dense(name = "DeepLayer2",  
 units = 128,  
 activation = "relu") %>%   
 layer\_dense(name = "DeepLayer3",  
 units = 128,  
 activation = "relu") %>%   
 layer\_dense(name = "OutputLayer",  
 units = 12,  
 activation = "softmax")  
  
summary(model)

## Model: "sequential"  
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
## Layer (type) Output Shape Param #   
## ================================================================================  
## DeepLayer1 (Dense) (None, 256) 143872   
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
## DeepLayer2 (Dense) (None, 128) 32896   
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
## DeepLayer3 (Dense) (None, 128) 16512   
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
## OutputLayer (Dense) (None, 12) 1548   
## ================================================================================  
## Total params: 194,828  
## Trainable params: 194,828  
## Non-trainable params: 0  
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# compile the model  
# categorical\_crossentrophy is used when we have two or more categorical variables   
model %>% compile(loss = "categorical\_crossentropy",  
 optimizer = "adam",  
 metrics = c("accuracy"))

newx <- as.matrix(xtrain)

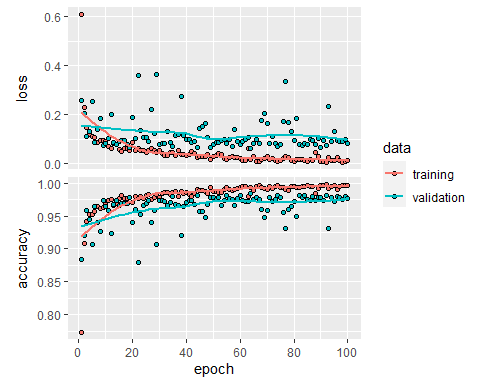
ncol(newx)

## [1] 561

history <- model %>%   
 fit(newx,  
 ynew,  
 epoch = 100,  
 batch\_size = 50,  
 validation\_split=0.1)

plot(history)

## `geom\_smooth()` using formula 'y ~ x'



set.seed(1234)  
ynewtest = matrix(0,nrow=3162,ncol = 12)  
for (i in 1:3162){  
 ynewtest[i,ytest[i,1]] = 1  
}

xnewtest <- as.matrix(xtest)

# evaluate model  
model %>%   
 evaluate(xnewtest,  
 ynewtest)

## loss accuracy   
## 0.4488342 0.9272612