ams580\_quiz7

Mustafa Yigit Isik

2023-03-27

#Question 1  
if (!requireNamespace("tidyverse")) install.packages('tidyverse')

## Loading required namespace: tidyverse

if (!requireNamespace("caret")) install.packages('caret')

## Loading required namespace: caret

if (!requireNamespace("neuralnet")) install.packages('neuralnet')

## Loading required namespace: neuralnet

if (!requireNamespace("tensorflow")) install.packages('tensorflow')

## Loading required namespace: tensorflow

library(tidyverse)

## ── Attaching packages  
## ───────────────────────────────────────  
## tidyverse 1.3.2 ──

## ✔ ggplot2 3.4.0 ✔ purrr 1.0.1   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.3.0 ✔ stringr 1.5.0   
## ✔ readr 2.1.4 ✔ forcats 1.0.0   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(caret)

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

library(neuralnet)

##   
## Attaching package: 'neuralnet'  
##   
## The following object is masked from 'package:dplyr':  
##   
## compute

#devtools::install\_github("rstudio/tensorflow")  
library(tensorflow)

##   
## Attaching package: 'tensorflow'  
##   
## The following object is masked from 'package:caret':  
##   
## train

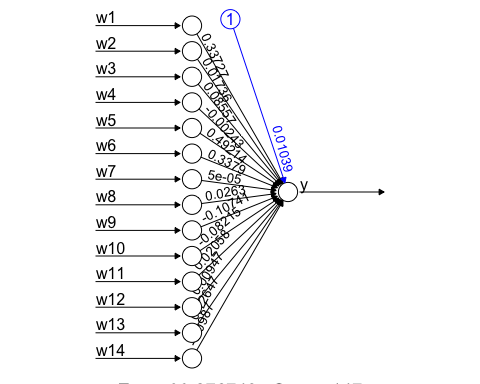
data <- read.csv('/Users/mustafayigitisik/Desktop/stuff/semesters/spring 2023/ams 580/quizzes/quiz7/QuestionMark.csv')  
#omit empty   
data <- na.omit(data)  
#change w4 column to numerical  
data$w4 <- ifelse(data$w4=='Y', 1, 0)  
#take mean and sd for future  
mean = mean(data$y)  
sd = sd(data$y)  
#normalize the data  
data <- data.frame(scale(data))  
  
set.seed(123)  
training.samples <- data$y %>%  
 createDataPartition(p = 0.75, list = FALSE)  
train.data <- data[training.samples, ]  
test.data <- data[-training.samples, ]  
str(train.data) # 1097 obs

## 'data.frame': 1097 obs. of 15 variables:  
## $ w1 : num -0.0718 0.6513 1.3743 -0.7949 1.3743 ...  
## $ w2 : num 2.179 -0.517 -0.517 -0.517 -0.517 ...  
## $ w3 : num -0.0919 0.0735 0.375 0.3605 -0.0434 ...  
## $ w4 : num 0.264 0.264 0.264 0.264 0.264 ...  
## $ w5 : num 0.2571 -0.6276 -0.0456 -0.9484 1.3745 ...  
## $ w6 : num -0.795 1.189 1.617 0.502 -0.795 ...  
## $ w7 : num 0.789 0.789 0.789 -1.026 0.789 ...  
## $ w8 : num -0.761 1.227 1.227 1.227 -0.761 ...  
## $ w9 : num 0.164 0.164 1.39 -2.288 0.164 ...  
## $ w10: num -0.211 -0.211 -0.211 -0.211 -0.211 ...  
## $ w11: num 0.6 0.6 0.6 -0.951 0.6 ...  
## $ w12: num 0.312 0.312 1.65 0.312 0.312 ...  
## $ w13: num -0.0607 0.6315 1.6979 0.0328 0.7625 ...  
## $ w14: num 0.39 -0.821 -0.724 -0.434 -1.015 ...  
## $ y : num 0.00729 0.53597 0.86954 -0.47734 1.58704 ...

str(test.data) # 363 obs

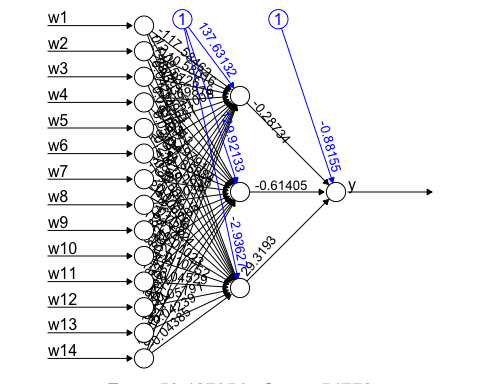
## 'data.frame': 363 obs. of 15 variables:  
## $ w1 : num 0.651 0.651 0.651 -0.795 -0.795 ...  
## $ w2 : num -0.517 -0.517 0.382 -0.517 0.382 ...  
## $ w3 : num -0.2071 -0.0969 -0.0135 0.0684 0.2456 ...  
## $ w4 : num 0.264 0.264 0.264 0.264 0.264 ...  
## $ w5 : num -0.793 -0.522 -0.144 -0.317 -0.648 ...  
## $ w6 : num 1.161 0.937 1.457 -0.795 -0.795 ...  
## $ w7 : num 0.789 -1.026 0.789 -1.026 -1.026 ...  
## $ w8 : num 1.227 -0.761 1.227 -0.761 -0.761 ...  
## $ w9 : num 0.164 0.164 0.164 0.164 -1.062 ...  
## $ w10: num -0.211 -0.211 -0.211 -0.211 -0.211 ...  
## $ w11: num -0.951 0.6 2.151 -0.951 -0.951 ...  
## $ w12: num 0.312 1.65 0.312 -1.027 -1.027 ...  
## $ w13: num 0.3509 0.7905 0.0515 -0.4162 -0.5658 ...  
## $ w14: num -0.87 0.632 0.632 0.971 1.117 ...  
## $ y : num 0.347 -0.515 0.24 -0.647 -0.465 ...

#Question 2  
set.seed(123)  
nn = neuralnet(y~., data = train.data, hidden = 0, err.fct = "sse", linear.output = T)  
plot(nn, rep = 'best')



pr.nn0 = predict(nn, test.data)  
# Test MSE  
MSE.nn.1 = RMSE(test.data$y\*sd+mean, pr.nn0\*sd+mean)^2

#Question 3  
set.seed(123)  
nn = neuralnet(y~., data = train.data, hidden = 3, err.fct = "sse", linear.output = T)  
plot(nn, rep = 'best')



pr.nn1 = predict(nn, test.data)  
# Test MSE  
(MSE.nn.2 = RMSE(test.data$y\*sd+mean, pr.nn1\*sd+mean)^2)

## [1] 2536116

#Question 4  
set.seed(123)  
mlr = lm(y~., data = train.data)  
summary(mlr)

##   
## Call:  
## lm(formula = y ~ ., data = train.data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.61873 -0.24252 -0.00817 0.19540 3.12751   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.039e-02 1.298e-02 0.800 0.424   
## w1 3.373e-01 2.147e-02 15.706 < 2e-16 \*\*\*  
## w2 1.737e-02 1.402e-02 1.239 0.216   
## w3 8.558e-02 1.585e-02 5.400 8.18e-08 \*\*\*  
## w4 -2.443e-03 1.428e-02 -0.171 0.864   
## w5 4.921e-01 2.241e-02 21.963 < 2e-16 \*\*\*  
## w6 3.379e-01 2.457e-02 13.753 < 2e-16 \*\*\*  
## w7 6.668e-05 2.016e-02 0.003 0.997   
## w8 2.632e-02 1.785e-02 1.474 0.141   
## w9 -1.074e-01 1.685e-02 -6.374 2.72e-10 \*\*\*  
## w10 -8.215e-02 1.431e-02 -5.742 1.22e-08 \*\*\*  
## w11 2.059e-02 1.598e-02 1.289 0.198   
## w12 9.457e-03 3.002e-02 0.315 0.753   
## w13 1.265e-01 2.884e-02 4.386 1.27e-05 \*\*\*  
## w14 -9.869e-02 1.738e-02 -5.677 1.76e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4297 on 1082 degrees of freedom  
## Multiple R-squared: 0.8251, Adjusted R-squared: 0.8228   
## F-statistic: 364.6 on 14 and 1082 DF, p-value: < 2.2e-16

pr.mlr = predict(mlr, test.data)  
# Test MSE  
(MSE.mlr = RMSE(test.data$y\*sd+mean, pr.mlr\*sd+mean)^2)

## [1] 954609.5

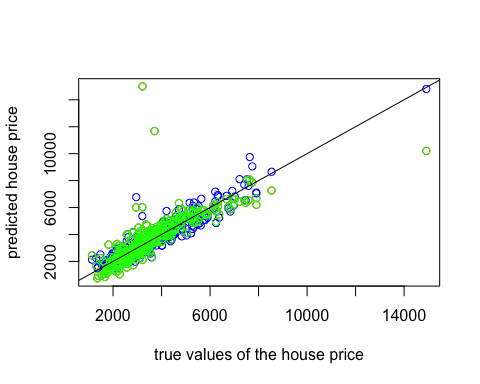
# Compare MSE  
print(paste(MSE.nn.1, MSE.nn.2, MSE.mlr))

## [1] "954640.567487749 2536116.29562727 954609.488203768"

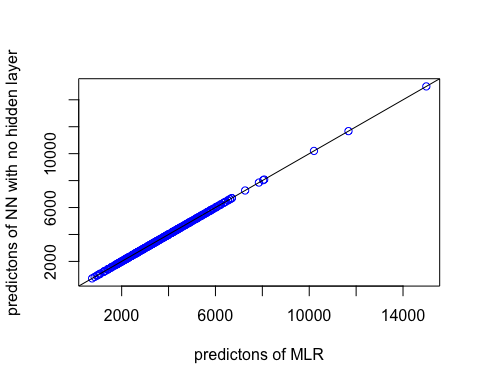
# Compare with multiple linear regression  
# summarize the predictions from different models  
final1 <- data.frame(predictions\_NN0=pr.nn0\*sd+mean, predictions\_NN1=pr.nn1\*sd+mean,predictions\_MLR =pr.mlr\*sd+mean, actual\_response=test.data$y\*sd+mean)  
knitr::kable(head(final1))

|  | predictions\_NN0 | predictions\_NN1 | predictions\_MLR | actual\_response |
| --- | --- | --- | --- | --- |
| 1 | 4191.307 | 3926.182 | 4191.321 | 4186 |
| 4 | 4138.596 | 4160.371 | 4138.525 | 2816 |
| 8 | 4714.537 | 4279.422 | 4714.564 | 4016 |
| 11 | 2229.462 | 2324.775 | 2229.457 | 2606 |
| 13 | 2175.766 | 2286.158 | 2175.769 | 2896 |
| 16 | 3324.269 | 3073.587 | 3324.308 | 2656 |

attach(final1)  
  
# NN model with no hidden layer, with one hidden layer with 3 neurons and MLR vs. the true values  
plot(actual\_response,predictions\_NN0,col="red", ylab = 'predicted house price', xlab = 'true values of the house price')  
points(actual\_response,predictions\_NN1,col="blue")  
points(actual\_response,predictions\_MLR,col="green")  
abline(a = 0, b = 1)



# NN model with one hidden layer with 3 neurons vs. MLR  
plot(predictions\_MLR,predictions\_NN0,col="blue", ylab = 'predictons of NN with no hidden layer', xlab = 'predictons of MLR')  
abline(a = 0, b = 1)



#Question 5  
library(keras)  
library(dplyr)  
library(cloudml)

## Loading required package: tfruns

train\_x = subset(train.data, select = -y)  
train\_x\_s = scale(train\_x)  
train\_y = as.matrix(subset(train.data, select = y))  
test\_x = subset(test.data, select = -y)  
test\_x\_s = scale(test\_x)  
test\_y = as.matrix(subset(test.data, select = y))  
  
set.seed(123)  
model <- keras\_model\_sequential()   
model %>% layer\_dense(units = 12, activation = 'relu', input\_shape = c(12)) %>%   
 layer\_dense(units = 3, activation = "relu") %>%  
 layer\_dense(units = 1, activation = "linear")  
model %>% compile(loss='mse',optimizer='adam',metrics='mse')  
summary(model)

## Model: "sequential"  
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
## Layer (type) Output Shape Param #   
## ================================================================================  
## dense\_2 (Dense) (None, 12) 156   
## dense\_1 (Dense) (None, 3) 39   
## dense (Dense) (None, 1) 4   
## ================================================================================  
## Total params: 199  
## Trainable params: 199  
## Non-trainable params: 0  
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#I kept getting tensorflow/python errors (probably due to environment, im using m1 mac pro)  
#lot of people mention the same situation here:  
#https://developer.apple.com/forums/thread/721619?page=2  
#that's why the remaining part of code couldn't be processed  
  
  
# history <- model %>% fit(train\_x\_s,train\_y, epochs=100, batch\_size = 8, validation\_split = 0.1)  
# plot(history)  
# preds <- predict(model, test\_x\_s)  
#   
# # test MSE  
# RMSE(test.data$y\*sd+mean, preds\*sd+mean)^2  
#   
# # Compare with multiple linear regression  
# final2 <- data.frame(predictions\_NN\_RELU=preds\*sd+mean,predictions\_MLR =pr.mlr\*sd+mean, actual\_response=test\_y\*sd+mean)  
# knitr::kable(head(final2))  
# attach(final2)  
# plot(actual\_response\*sd+mean,predictions\_NN\_RELU\*sd+mean,col="red", ylab = 'predictions', xlab = 'actual response')  
# points(actual\_response\*sd+mean,predictions\_MLR\*sd+mean,col="green")  
# abline(a = 0, b = 1)