project2\_nn\_regression

EMVP

2023-05-02

#Q1  
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

library(MASS)

##   
## Attaching package: 'MASS'  
##   
## The following object is masked from 'package:dplyr':  
##   
## select

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

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

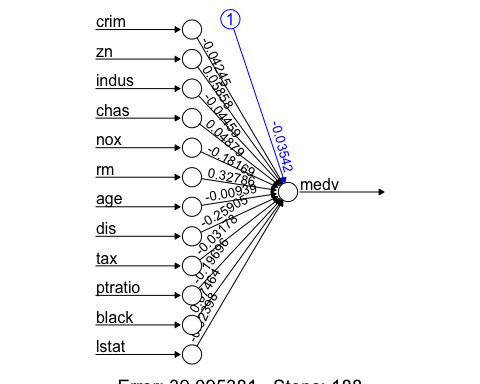
#install\_tensorflow()  
  
data("Boston")  
data = Boston  
data <- subset(data, select = -c(rad))  
# mean & standard deviation of the response  
mean = mean(data$medv)  
sd = sd(data$medv)  
# normalize the data  
data = data.frame(scale(data))  
  
set.seed(123)  
training.samples <- data$medv %>%  
 createDataPartition(p = 0.75, list = FALSE)  
train.data <- data[training.samples, ]  
test.data <- data[-training.samples, ]  
str(train.data) # 381 obs

## 'data.frame': 381 obs. of 13 variables:  
## $ crim : num -0.419 -0.417 -0.416 -0.412 -0.41 ...  
## $ zn : num 0.2845 -0.4872 -0.4872 -0.4872 0.0487 ...  
## $ indus : num -1.287 -0.593 -1.306 -1.306 -0.476 ...  
## $ chas : num -0.272 -0.272 -0.272 -0.272 -0.272 ...  
## $ nox : num -0.144 -0.74 -0.834 -0.834 -0.265 ...  
## $ rm : num 0.413 0.194 1.015 1.227 -0.388 ...  
## $ age : num -0.1199 0.3668 -0.8091 -0.5107 -0.0702 ...  
## $ dis : num 0.14 0.557 1.077 1.077 0.838 ...  
## $ tax : num -0.666 -0.986 -1.105 -1.105 -0.577 ...  
## $ ptratio: num -1.458 -0.303 0.113 0.113 -1.504 ...  
## $ black : num 0.441 0.441 0.416 0.441 0.426 ...  
## $ lstat : num -1.0745 -0.492 -1.3602 -1.0255 -0.0312 ...  
## $ medv : num 0.1595 -0.1014 1.1816 1.486 0.0399 ...

str(test.data) # 125 obs

## 'data.frame': 125 obs. of 13 variables:  
## $ crim : num -0.417 -0.417 -0.396 -0.394 -0.347 ...  
## $ zn : num -0.4872 -0.4872 0.0487 0.0487 -0.4872 ...  
## $ indus : num -0.593 -1.306 -0.476 -0.476 -0.437 ...  
## $ chas : num -0.272 -0.272 -0.272 -0.272 -0.272 ...  
## $ nox : num -0.74 -0.834 -0.265 -0.265 -0.144 ...  
## $ rm : num 1.281 0.207 -0.93 0.131 -0.478 ...  
## $ age : num -0.266 -0.351 1.116 0.914 -0.241 ...  
## $ dis : num 0.557 1.077 1.086 1.212 0.433 ...  
## $ tax : num -0.986 -1.105 -0.577 -0.577 -0.601 ...  
## $ ptratio: num -0.303 0.113 -1.504 -1.504 1.175 ...  
## $ black : num 0.396 0.41 0.328 0.393 0.441 ...  
## $ lstat : num -1.208 -1.042 2.419 1.092 -0.615 ...  
## $ medv : num 1.323 0.671 -0.656 -0.819 -0.232 ...

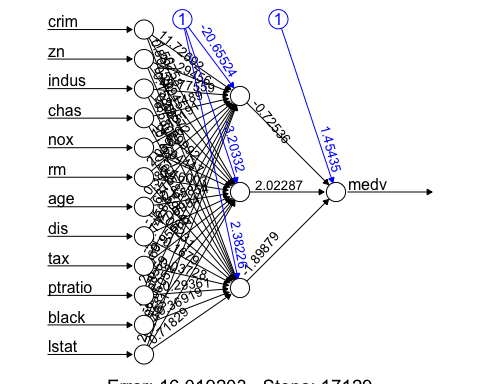
#Q2  
set.seed(123)  
nn = neuralnet(medv~., data = train.data, hidden = 0, err.fct = "sse", linear.output = T)  
plot(nn, rep = 'best')



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

## [1] 42.90577

#Q3  
set.seed(123)  
nn = neuralnet(medv~., 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$medv\*sd+mean, pr.nn1\*sd+mean)^2)

## [1] 37.01437

#Q4  
set.seed(123)  
mlr = lm(medv~., data = train.data)  
summary(mlr)

##   
## Call:  
## lm(formula = medv ~ ., data = train.data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.40013 -0.26214 -0.06769 0.17814 2.35738   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.035423 0.023695 -1.495 0.135789   
## crim -0.042440 0.031485 -1.348 0.178506   
## zn 0.058543 0.035236 1.661 0.097474 .   
## indus -0.044564 0.046408 -0.960 0.337545   
## chas 0.048801 0.026021 1.875 0.061518 .   
## nox -0.181812 0.050075 -3.631 0.000323 \*\*\*  
## rm 0.327836 0.032518 10.082 < 2e-16 \*\*\*  
## age -0.009319 0.042395 -0.220 0.826144   
## dis -0.259040 0.045304 -5.718 2.23e-08 \*\*\*  
## tax -0.031723 0.043656 -0.727 0.467900   
## ptratio -0.197012 0.030720 -6.413 4.38e-10 \*\*\*  
## black 0.074621 0.026127 2.856 0.004533 \*\*   
## lstat -0.324002 0.044322 -7.310 1.68e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4609 on 368 degrees of freedom  
## Multiple R-squared: 0.7626, Adjusted R-squared: 0.7549   
## F-statistic: 98.52 on 12 and 368 DF, p-value: < 2.2e-16

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

## [1] 42.90551

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

## [1] "42.9057695828218 37.0143749146895 42.9055116896362"

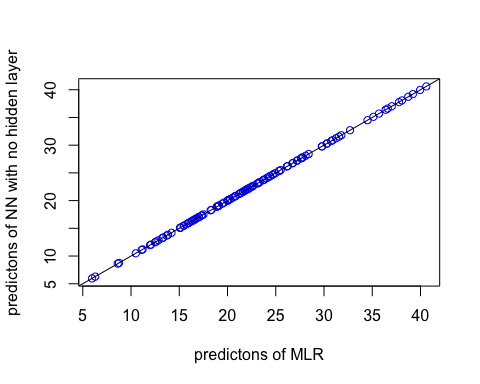
# Compare with multiple linear regression  
# summarize the predictions from different models  
final1 <- data.frame(predictions\_NN0=pr.nn0\*sd+mean, predictions\_NN1=pr.nn1\*sd+mean,predictions\_MLR =pr.mlr\*sd+mean, actual\_response=test.data$medv\*sd+mean)  
knitr::kable(head(final1))

|  | predictions\_NN0 | predictions\_NN1 | predictions\_MLR | actual\_response |
| --- | --- | --- | --- | --- |
| 3 | 30.73145 | 33.85545 | 30.73168 | 34.7 |
| 6 | 25.50940 | 22.44689 | 25.50949 | 28.7 |
| 9 | 13.32310 | 18.73894 | 13.32406 | 16.5 |
| 11 | 20.24198 | 19.76108 | 20.24290 | 15.0 |
| 14 | 20.10770 | 20.45413 | 20.10700 | 20.4 |
| 15 | 19.98503 | 20.33412 | 19.98474 | 18.2 |

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)



#Q5  
# devtools::install\_github('rstudio/cloudml')  
library(keras)  
library(dplyr)  
library(cloudml)

## Loading required package: tfruns

train\_x = subset(train.data, select = -medv)  
train\_x\_s = scale(train\_x)  
train\_y = as.matrix(subset(train.data, select = medv))  
test\_x = subset(test.data, select = -medv)  
test\_x\_s = scale(test\_x)  
test\_y = as.matrix(subset(test.data, select = medv))  
  
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  
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#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$medv\*sd+mean, preds\*sd+mean)^2

## [1] 260.2572

# 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))

|  | predictions\_NN\_RELU | predictions\_MLR | medv |
| --- | --- | --- | --- |
| 3 | 20.90953 | 30.73168 | 34.7 |
| 6 | 21.03648 | 25.50949 | 28.7 |
| 9 | 31.53971 | 13.32406 | 16.5 |
| 11 | 24.56861 | 20.24290 | 15.0 |
| 14 | 22.31025 | 20.10700 | 20.4 |
| 15 | 27.80966 | 19.98474 | 18.2 |

attach(final2)

## The following object is masked from final1:  
##   
## predictions\_MLR

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)

