Compulsory exercise 3

TMA4268 Statistical Learning V2019

Silje Anfindsen

15 April, 2020

Problem 1

```
library(ISLR)
library(keras)
set.seed(1)
College$Private = as.numeric(College$Private)
train.ind = sample(1:nrow(College), 0.5 * nrow(College))
college.train = College[train.ind, ]
college.test = College[-train.ind, ]
str(College)
```

```
## 'data.frame':    777 obs. of  18 variables:
##  $ Private    : num  2 2 2 2 2 2 2 2 2 2 ...
##  $ Apps       : num  1660 2186 1428 417 193 ...
##  $ Accept     : num  1232 1924 1097 349 146 ...
##  $ Enroll     : num  721 512 336 137 55 158 103 489 227 172 ...
##  $ Top10perc  : num  23 16 22 60 16 38 17 37 30 21 ...
##  $ Top25perc  : num  52 29 50 89 44 62 45 68 63 44 ...
##  $ F.Undergrad: num  2885 2683 1036 510 249 ...
##  $ P.Undergrad: num  537 1227 99 63 869 ...
##  $ Outstate   : num  7440 12280 11250 12960 7560 ...
##  $ Room.Board : num  3300 6450 3750 5450 4120 ...
##  $ Books      : num  450 750 400 450 800 500 500 450 300 660 ...
##  $ Personal   : num  2200 1500 1165 875 1500 ...
##  $ PhD        : num  70 29 53 92 76 67 90 89 79 40 ...
##  $ Terminal   : num  78 30 66 97 72 73 93 100 84 41 ...
##  $ S.F.Ratio  : num  18.1 12.2 12.9 7.7 11.9 9.4 11.5 13.7 11.3 11.5 ...
##  $ perc.alumni: num  12 16 30 37 2 11 26 37 23 15 ...
##  $ Expend     : num  7041 10527 8735 19016 10922 ...
##  $ Grad.Rate  : num  60 56 54 59 15 55 63 73 80 52 ...
```

a)

Preprocessing the data by applying feature-wise normalization to the predictors.

```r
#divide data into response and covariates
#training data
train_x <- college.train[,-9]
train_y <- college.train[,9]

#test data
test_x <- college.test[,-9]
test_y <- college.test[,9]

#we use the mean and std of the training data for both test and train set
mean <- apply(train_x , 2, mean)
std <- apply(train_x, 2, sd)
train_x <- scale(train_x, center = mean, scale = std)
test_x <- scale(test_x, center = mean, scale = std)
```

b)

The equation describing a network that predicts `Outstate`. The output layer has one node wich is numerical (the Out-of-state tuition), therefore we can choose betwenn ReLu and linear activation function for this layer, we choose the linear activation function.

$$\hat{y_1}(x) = \beta_{01} + \sum_{m=1}^{64} \beta_{m1} \max(\gamma_{0m} + \sum_{l=1}^{64} \gamma_{lm} \max(\alpha_{0l} + \sum_{j=1}^{17} \alpha_{jl} x_j, 0), 0)$$

skal det være med bias term eller ikke for hvert lag?

c)

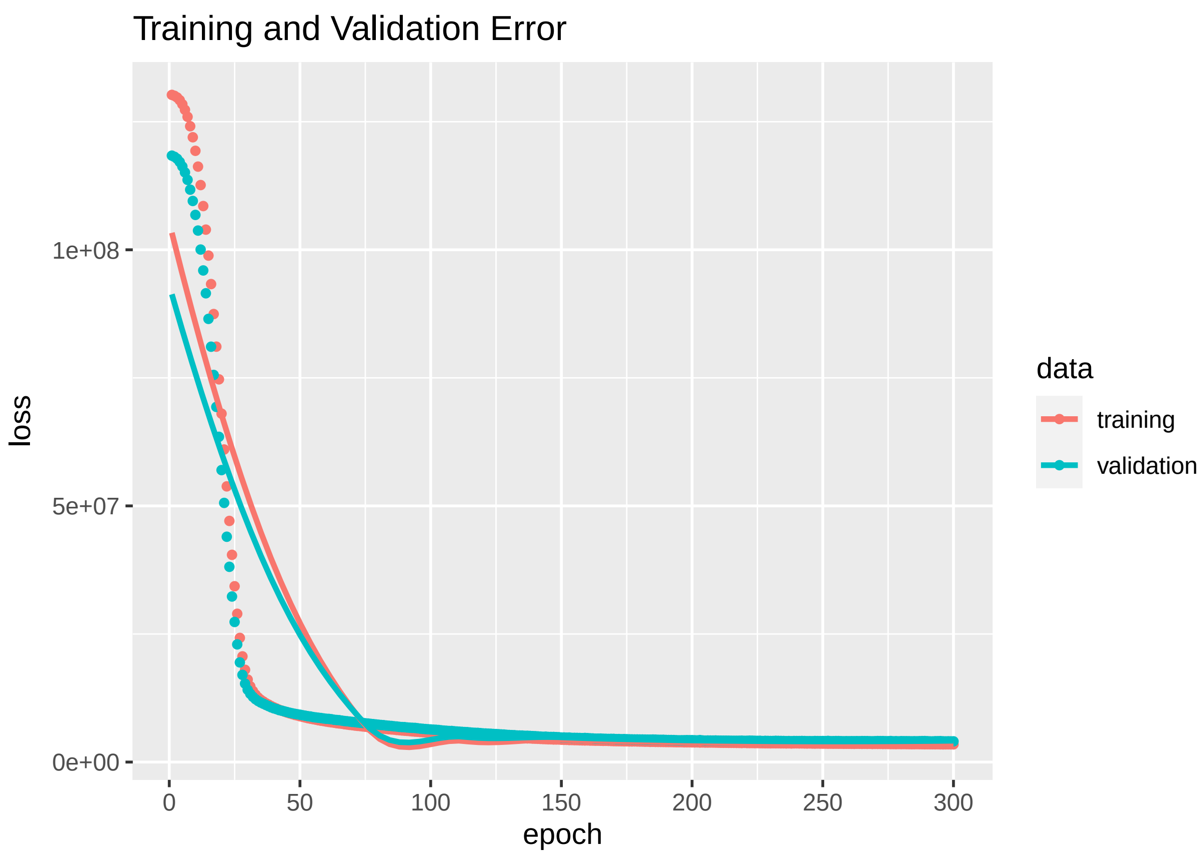
We will now train the network from b).

```r
set.seed(123)

#define the model
model = keras_model_sequential() %>%
  layer_dense(units = 64, activation = 'relu', input_shape = dim(train_x)[2]) %>%
  layer_dense(units = 64, activation = 'relu') %>%
  layer_dense(units = 1)

#compile
model %>% compile(optimizer = "rmsprop", loss = "mse")

#train
history = model %>% fit(train_x, train_y, epochs = 300, batch_size = 8,
    validation_split = 0.2) #20% of the training set as validation set
plot(history)+ggtitle("Training and Validation Error")
```




```r
#test MSE
mse <- model %>% evaluate(test_x, test_y)
```

The test MSE for this network is 3.6865554×10^6. From Compulsory 2, using the same dataset, the test MSE´s for Forward selection is 4.11268×10^6, for Lasso is 3.71702×10^6 and for Random Forest the MSE is 2.607985×10^6. The results

d)

```r
set.seed(123)
```

e)

Problem 2

a)

b)

c)

d)

Problem 3

a)

b)

c)

Problem 4

a)

b)

c)

d)

e)

Problem 5

a)

b)

c)

d)

e)

f)

References

James, G., D. Witten, T. Hastie, and R. Tibshirani. 2013. An Introduction to Statistical Learning with Applications in R. New York: Springer.