Homework 06

Vrunda Shah, Hansika Karkera

19/04/2020

```
library(readxl)
data_df <- read_excel("18 Toyota Corolla.xlsx" , sheet = 1)
data_df <- data_df[c(3,4,7,9,12,13,19)]

## normalizing the data
normalize <- function(x) {
   return ((x - min(x)) / (max(x) - min(x)))
}
data_df <- as.data.frame(lapply(data_df, normalize))
train=sample(row.names(data_df), dim(data_df)[1]*0.75)
val=setdiff(row.names(data_df), train)
train_df <- data_df[train,c(1,2,3,4,6)]
val_df<- data_df[val,c(1,2,3,4,6)]
```

Task 1(a)

```
# threshold value = 1
library(neuralnet)

## Warning: package 'neuralnet' was built under R version 3.6.3

th_1 <- neuralnet(Price~Age_08_04+KM+HP+cc,data=train_df,linear.output=FALSE,
threshold=1, algorithm ='rprop+')
train_pred1 <- neuralnet::compute(th_1, train_df[,-1])
train.pred1<- train_pred1$net.result
rmse_train1 <- sqrt(mean((train.pred1-train_df$Price)^2))
valid_pred1 <- neuralnet::compute(th_1, val_df[,-1])
valid.pred1<- valid_pred1$net.result
rmse_valid1 <- sqrt(mean((valid.pred1-val_df$Price)^2))

# threshold value = 0.1
th_0.1 <- neuralnet(Price~Age_08_04+KM+HP+cc,data=train_df,
linear.output=FALSE,threshold=0.1, algorithm ='rprop+')
train_pred0.1 <- neuralnet::compute(th_0.1, train_df[,-1])
train.pred0.1<- train_pred0.1$net.result
rmse_train0.1 <- sqrt(mean((train.pred0.1-train_df$Price)^2))
valid_pred0.1 <- neuralnet::compute(th_1, val_df[,-1])
valid.pred0.1<- valid_pred0.1$net.result
rmse_valid0.1 <- sqrt(mean((valid.pred0.1-val_df$Price)^2))

#threshold value = 0.05
```

```r
th_0.05 <- neuralnet(Price~Age_08_04+KM+HP+cc,data=train_df,
linear.output=FALSE,threshold=0.05, algorithm ='rprop+')
train_pred0.05 <- neuralnet::compute(th_0.05, train_df[,-1])
train.pred0.05<- train_pred0.05$net.result
rmse_train0.05 <- sqrt(mean((train.pred0.05-train_df$Price)^2))
valid_pred0.05 <- neuralnet::compute(th_0.05, val_df[,-1])
valid.pred0.05<- valid_pred0.05$net.result
rmse_valid0.05 <- sqrt(mean((valid.pred0.05-val_df$Price)^2))

# threshold value = 0.01
th_0.01 <- neuralnet(Price~Age_08_04+KM+HP+cc,data=train_df,
                 linear.output=FALSE,threshold=0.01, algorithm ='rprop+')
train_pred0.01 <- neuralnet::compute(th_0.01, train_df[,-1])
train.pred0.01<- train_pred0.01$net.result
rmse_train0.01 <- sqrt(mean((train.pred0.01-train_df$Price)^2))
valid_pred0.01 <- neuralnet::compute(th_0.01, val_df[,-1])
valid.pred0.01<- valid_pred0.01$net.result
rmse_valid0.01 <- sqrt(mean((valid.pred0.01-val_df$Price)^2))

# threshold value = 0.005
th_0.005 <-
neuralnet(Price~Age_08_04+KM+HP+cc,data=train_df,linear.output=FALSE,threshol
d=0.005, algorithm ='rprop+')
train_pred0.005 <- neuralnet::compute(th_0.005, train_df[,-1])
train.pred0.005<- train_pred0.005$net.result
rmse_train0.005 <- sqrt(mean((train.pred0.005-train_df$Price)^2))
valid_pred0.005 <- neuralnet::compute(th_0.005, val_df[,-1])
valid.pred0.005<- valid_pred0.005$net.result
rmse_valid0.005 <- sqrt(mean((valid.pred0.005-val_df$Price)^2))

#threshold value = 0.001
th_0.001 <- neuralnet(Price~Age_08_04+KM+HP+cc,data=train_df,
linear.output=FALSE,threshold=0.001, algorithm ='rprop+')
train_pred0.001 <- neuralnet::compute(th_0.001, train_df[,-1])
train.pred0.001<- train_pred0.001$net.result
rmse_train0.001 <- sqrt(mean((train.pred0.001-train_df$Price)^2))
valid_pred0.001 <- neuralnet::compute(th_0.001, val_df[,-1])
valid.pred0.001<- valid_pred0.001$net.result
rmse_valid0.001 <- sqrt(mean((valid.pred0.001-val_df$Price)^2))

#threshold value = 0.0001
th_0.0001 <- neuralnet(Price~Age_08_04+KM+HP+cc,data=train_df,
                 linear.output=FALSE,threshold=0.0001, algorithm ='rprop+')
train_pred0.0001 <- neuralnet::compute(th_0.0001, train_df[,-1])
train.pred0.0001<- train_pred0.0001$net.result
rmse_train0.0001 <- sqrt(mean((train.pred0.0001-train_df$Price)^2))
valid_pred0.0001 <- neuralnet::compute(th_0.0001, val_df[,-1])
valid.pred0.0001<- valid_pred0.0001$net.result
rmse_valid0.0001 <- sqrt(mean((valid.pred0.0001-val_df$Price)^2))
```

```
# RMS error value of training and validatio dataset covering differen
threshold value
rmse.trainvalues<-rbind(rmse_train1, rmse_train0.1, rmse_train0.05,
rmse_train0.01, rmse_train0.005, rmse_train0.001,rmse_train0.0001)
rmse.valvalues<-rbind(rmse_valid1, rmse_valid0.1, rmse_valid0.05,
rmse_valid0.01, rmse_valid0.005, rmse_valid0.001,rmse_valid0.0001)
# RMSE for training data
rmse.trainvalues

##                        [,1]
## rmse_train1      0.05854048
## rmse_train0.1    0.04954358
## rmse_train0.05   0.04817202
## rmse_train0.01   0.04819155
## rmse_train0.005  0.04811381
## rmse_train0.001  0.04807487
## rmse_train0.0001 0.04807345
```

As the value of threshold decreases the rms error also decreases for the training dataset.

Task 1(b)

```
rmse.valvalues

##                        [,1]
## rmse_valid1      0.04872714
## rmse_valid0.1    0.04872714
## rmse_valid0.05   0.04390349
## rmse_valid0.01   0.04404671
## rmse_valid0.005  0.04396724
## rmse_valid0.001  0.04389562
## rmse_valid0.0001 0.04388378
```

As the value of threshold decreases the rms error also decreases for the validation dataset.

Task 1(c)

```
# neural net model with 2 hidden layer in it
h2 <-
neuralnet(Price~Age_08_04+KM+HP+cc,data=train_df,linear.output=FALSE,threshol
d=1, algorithm ='rprop+', hidden = 2)
train_pred_h2 <- neuralnet::compute(h2, train_df[,-1])
train.pred_h2<- train_pred_h2$net.result
rmse_train_h2 <- sqrt(mean((train.pred_h2-train_df$Price)^2))
valid_pred_h2 <- neuralnet::compute(h2, val_df[,-1])
valid.pred_h2<- valid_pred_h2$net.result
rmse_valid_h2 <- sqrt(mean((valid.pred_h2-val_df$Price)^2))

# neural net model with 4 hidden layer in it
```

```r
h4 <- neuralnet(Price~Age_08_04+KM+HP+cc,data=train_df,
linear.output=FALSE,threshold=1, algorithm ='rprop+', hidden = 4)
train_pred_h4<- neuralnet::compute(h4, train_df[,-1])
train.pred_h4<- train_pred_h4$net.result
rmse_train_h4 <- sqrt(mean((train.pred_h4-train_df$Price)^2))
valid_pred_h4 <- neuralnet::compute(h4, val_df[,-1])
valid.pred_h4<- valid_pred_h4$net.result
rmse_valid_h4 <- sqrt(mean((valid.pred_h4-val_df$Price)^2))

# neural net model with 8 hidden layer in it
h8 <- neuralnet(Price~Age_08_04+KM+HP+cc,data=train_df,
linear.output=FALSE,threshold=1, algorithm ='rprop+', hidden = 8)
train_pred_h8<- neuralnet::compute(h8, train_df[,-1])
train.pred_h8<- train_pred_h8$net.result
rmse_train_h8 <- sqrt(mean((train.pred_h8-train_df$Price)^2))
valid_pred_h8 <- neuralnet::compute(h8, val_df[,-1])
valid.pred_h8<- valid_pred_h8$net.result
rmse_valid_h8 <- sqrt(mean((valid.pred_h8-val_df$Price)^2))

# rmse values for different hidden layers
rmse_hidden_train<-rbind(rmse_train_h2, rmse_train_h4 ,rmse_train_h8)
rmse_hidden_valid<-rbind(rmse_valid_h2, rmse_valid_h4, rmse_valid_h8)
rmse_hidden_train

##                     [,1]
## rmse_train_h2 0.13220716
## rmse_train_h4 0.05561547
## rmse_train_h8 0.05673036

rmse_hidden_train

##                     [,1]
## rmse_train_h2 0.13220716
## rmse_train_h4 0.05561547
## rmse_train_h8 0.05673036
```

With the increase in hidden layer nodes we observe a decrease in the rmse value for both training and validation data set.

Task 1(d)

```r
# changing of hidden layer from to 2 layers
h12 <- neuralnet(Price~Age_08_04+KM+HP+cc,data=train_df,
linear.output=FALSE,threshold=1, algorithm ='rprop+', hidden = c(1,2))
train_pred_h12 <- neuralnet::compute(h12, train_df[,-1])
train.pred_h12<- train_pred_h12$net.result
rmse_train_h12 <- sqrt(mean((train.pred_h12-train_df$Price)^2))
valid_pred_h12 <- neuralnet::compute(h12, val_df[,-1])
valid.pred_h12<- valid_pred_h12$net.result
```

```r
rmse_valid_h12 <- sqrt(mean((valid.pred_h12-val_df$Price)^2))
rmse_train_h12
```

```
## [1] 0.1303921
```

```r
rmse_valid_h12
```

```
## [1] 0.1059461
```

```r
# changing hidden layer to 1 layer
h21 <-
neuralnet(Price~Age_08_04+KM+HP+cc,data=train_df,linear.output=FALSE,threshol
d=1, algorithm ='rprop+', hidden = 2)
train_pred_h21 <- neuralnet::compute(h21, train_df[,-1])
train.pred_h21<- train_pred_h21$net.result
rmse_train_h21 <- sqrt(mean((train.pred_h21-train_df$Price)^2))
valid_pred_h21 <- neuralnet::compute(h21, val_df[,-1])
valid.pred_h21<- valid_pred_h21$net.result
rmse_valid_h21 <- sqrt(mean((valid.pred_h21-val_df$Price)^2))
rmse_train_h21
```

```
## [1] 0.1248887
```

```r
rmse_valid_h21
```

```
## [1] 0.1025943
```

The rms error is lower for a single layer network.

Task 1(e)

```r
l1 <- neuralnet(Price~Age_08_04+KM+HP+cc,data=train_df,
                 linear.output=FALSE,threshold=1, algorithm ='rprop+',
learningrate = 1)
train_pred_l1 <- neuralnet::compute(l1, train_df[,-1])
train.pred_l1<- train_pred_l1$net.result
rmse_train_l1 <- sqrt(mean((train.pred_l1-train_df$Price)^2))
valid_pred_l1 <- neuralnet::compute(l1, val_df[,-1])
valid.pred_l1<- valid_pred_l1$net.result
rmse_valid_l1 <- sqrt(mean((valid.pred_l1-val_df$Price)^2))
rmse_train_l1
```

```
## [1] 0.135157
```

```r
rmse_valid_l1
```

```
## [1] 0.1111427
```

```r
l2 <-
neuralnet(Price~Age_08_04+KM+HP+cc,data=train_df,linear.output=FALSE,threshol
d=0.5, algorithm ='rprop+', learningrate = 1)
train_pred_l2 <- neuralnet::compute(l2, train_df[,-1])
```

```
train.pred_l2<- train_pred_l2$net.result
rmse_train_l2 <- sqrt(mean((train.pred_l2-train_df$Price)^2))
valid_pred_l2 <- neuralnet::compute(l2, val_df[,-1])
valid.pred_l2 <- valid_pred_l2$net.result
rmse_valid_l2 <- sqrt(mean((valid.pred_l2-val_df$Price)^2))
rmse_train_l2

## [1] 0.0520004

rmse_valid_l2

## [1] 0.04520544
```

As learning rate decreases, the training process becomes slower, and the rmse value also decreases.