Neural Network Regression Practice

Installing packages

```
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
```

```
library(tidyverse)
```

```
## Warning: package 'tidyr' was built under R version 4.2.3

## Warning: package 'readr' was built under R version 4.2.3

## Warning: package 'dplyr' was built under R version 4.2.3

## Warning: package 'stringr' was built under R version 4.2.3

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(caret)
```

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

```
library(keras)
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
```

Reading data

```
data("Boston")
data <- Boston
data <- subset(data, select = -(rad)) #exclude RAD
mean <- mean(data$medv)
sd <- sd(data$medv)
data <- data.frame(scale(data)) # normalize the data
```

Split train and test 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)
```

```
## '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)
```

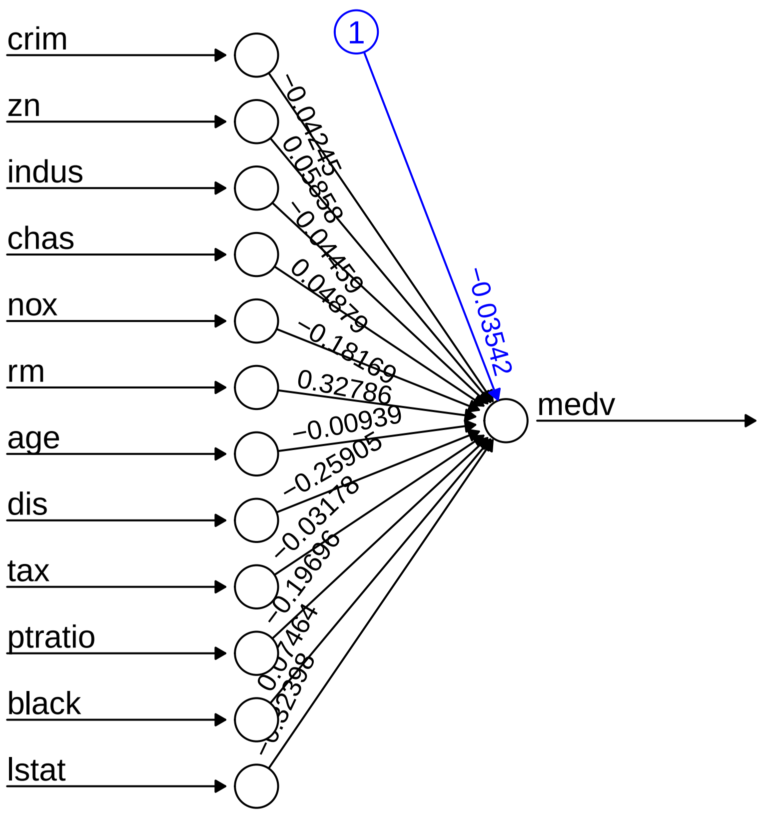
```
## 'data.frame':    125 obs. of  13 variables:
##  $ crim   : num  -0.417 -0.417 -0.396 -0.394 -0.347 ...
##  $ zn     : num  -0.4872 -0.4872 0.0487 0.0487 -0.4872 ...
##  $ indus  : num  -0.593 -1.306 -0.476 -0.476 -0.437 ...
##  $ chas   : num  -0.272 -0.272 -0.272 -0.272 -0.272 ...
##  $ nox    : 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 ...
```

Predictive model 1

(i): no hidden layers (ii): the default loss function of “sse” (iii): the default activation function of “identity”

```
set.seed(123)
nn <- neuralnet(medv~., data=train.data, hidden=0, err.fct="sse",linear.output=T)
plot(nn,rep="best")
```



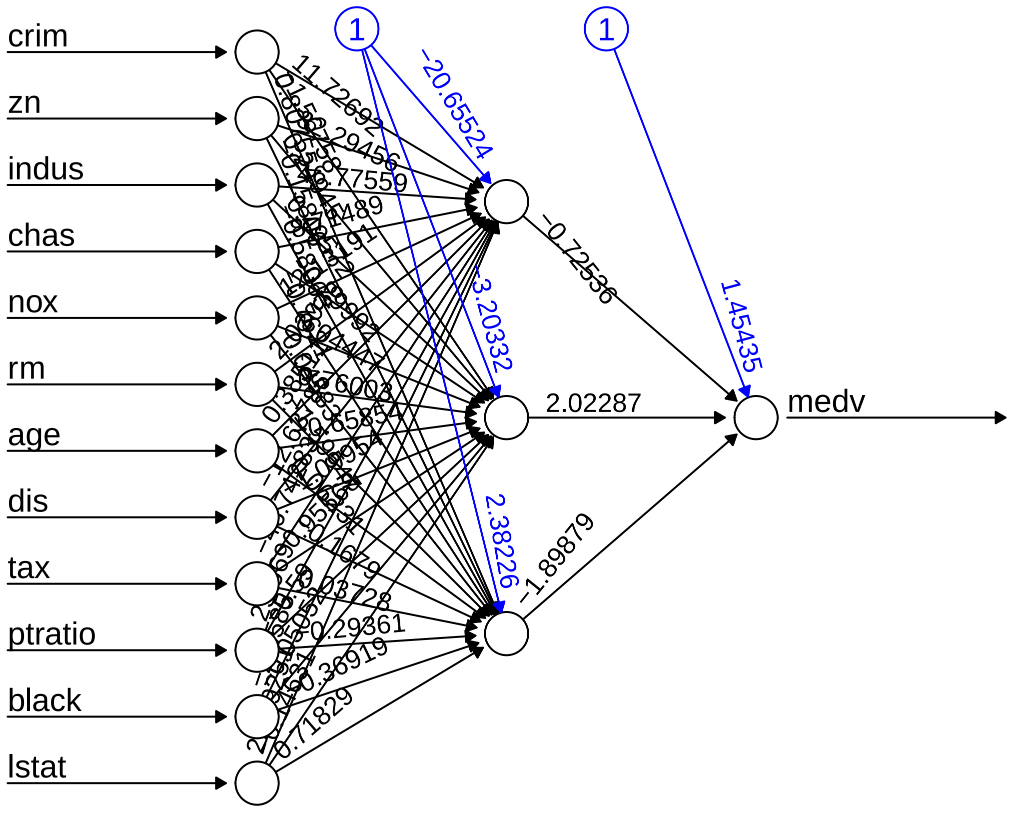

```
pr.nn0 <- predict(nn, test.data)
(MSE.nn.1 <- RMSE(test.data$medv*sd+mean,pr.nn0*sd+mean)^2)
```

```
## [1] 42.90577
```

Predictive model 2

(i): one hidden layer with 3 neurons (ii): the default loss function of “sse” (iii): the default activation function of “identity”

```r
set.seed(123)
nn <- neuralnet(medv~.,data=train.data,hidden=3,err.fct="sse",linear.output=T)
plot(nn, rep="best")
```




```r
pr.nn1 <- predict(nn, test.data)
(MSE.nn.2 = RMSE(test.data$medv*sd+mean,pr.nn1*sd+mean)^2)
```

```
## [1] 37.01437
```

Multiple regression model

```r
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)
(MSE.mlr <- RMSE(test.data$medv*sd+mean,pr.mlr*sd+mean)^2)
```

```
## [1] 42.90551
```

MLE comparision

```
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.r
knitr::kable(head(final1))
```

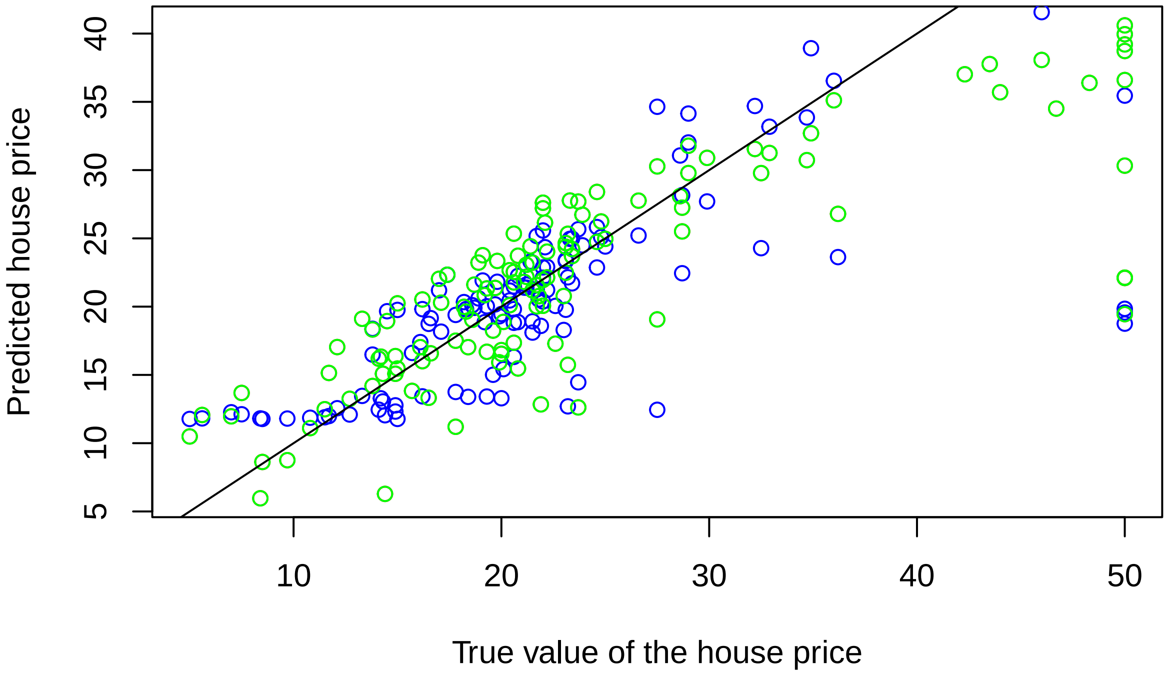
	predictions_NN0	predictions_NN1	predictions_MLR	actual_response
3	30.73145	33.85545	30.73168	34.7
6	25.50940	22.44689	25.50949	28.7
9	13.32310	18.73894	13.32406	16.5
11	20.24198	19.76108	20.24290	15.0
14	20.10770	20.45413	20.10700	20.4
15	19.98503	20.33412	19.98474	18.2

Plot 3 models vs. true values

```
attach(final1)

plot(actual_response, predictions_NN0,col="red",ylab="Predicted house price",xlab="True value of the hou
points(actual_response,predictions_NN1,col="blue")
```

```
points(actual_response,predictions_MLR,col="green")
abline(a=0,b=1)
```



NN Model with no hidden layer with 3 neurons vs. MLR

```
plot(predictions_MLR,predictions_NN0,col="blue",ylab="predictions of NN with no hidden layer", xlab="pre
abline(a=0,b=1)
```