Neural Network Example: House Price

We will use House Price example again for Neural Network practice. The HousePrices.csv data set includes prices and characteristics of n = 128 houses in a major US metropolitan area. The variables include Price (sale price in dollars), SqFt (size in square feet), Bedrooms (number of), Bathrooms (number of), Offers (number of offers the house has received while on the market), Brick (whether it is brick construction; Yes/No), and Neighborhood (East/North/West). The objective is to explain the sale price of a house as a function of its characteristics.

1. Import the data into R and change variables Brick and Neighborhood into factor variables. Eliminate variable Home ID from the original data frame.

**Ans:**

'data.frame': 128 obs. of 8 variables:

$ HomeID : int 1 2 3 4 5 6 7 8 9 10 ...

$ Price : int 114300 114200 114800 94700 119800 114600 151600 150700 119200 104000 ...

$ SqFt : int 1790 2030 1740 1980 2130 1780 1830 2160 2110 1730 ...

$ Bedrooms : int 2 4 3 3 3 3 3 4 4 3 ...

$ Bathrooms : int 2 2 2 2 3 2 3 2 2 3 ...

$ Offers : int 2 3 1 3 3 2 3 2 3 3 ...

$ Brick : chr "No" "No" "No" "No" ...

$ Neighborhood: chr "East" "East" "East" "East" ...

'data.frame': 128 obs. of 7 variables:

$ Price : int 114300 114200 114800 94700 119800 114600 151600 150700 119200 104000 ...

$ SqFt : int 1790 2030 1740 1980 2130 1780 1830 2160 2110 1730 ...

$ Bedrooms : int 2 4 3 3 3 3 3 4 4 3 ...

$ Bathrooms : int 2 2 2 2 3 2 3 2 2 3 ...

$ Offers : int 2 3 1 3 3 2 3 2 3 3 ...

$ Brick : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 1 1 1 ...

$ Neighborhood: Factor w/ 3 levels "East","North",..: 1 1 1 1 1 2 3 3 1 1 ...

**Code:**

house <- read.csv(file.choose(), stringsAsFactors = FALSE)

str(house)

house$Brick <- factor(ifelse(house$Brick=="No",0,1),

levels=c(0,1), labels=c("No", "Yes"))

str(house)

head(house$Brick)

house$Neighborhood <- factor(house$Neighborhood)

str(house)

head(house$Neighborhood)

house <- house[-1]

str(house)

1. Use aggregate function to inspect the mean of Price grouped by variable Neighborhood. What is the average price of houses in North neighborhood? What is the average price of houses in West neighborhood?

Hint: aggregate(x=dt[1], by=dt[7], FUN=mean)

Or aggregate(dt[1],dt[7],mean)

**Ans:**

#North 110154.5

#West 159294.9

**Code:**

aggregate(house[1],house[7],mean)

1. Overwrite variables Brick and Neighborhood into dummy variables. If Brick = “Yes”, we recode the variable value into 1, or else we recode it into 0. For Neighborhood variable, if neighborhood is west, we recode the variable value into 1, or else we recode it into 0.

**Ans:**

Price SqFt Bedrooms Bathrooms Offers Brick Neighborhood

1 114300 1790 2 2 2 0 0

2 114200 2030 4 2 3 0 0

3 114800 1740 3 2 1 0 0

4 94700 1980 3 2 3 0 0

5 119800 2130 3 3 3 0 0

6 114600 1780 3 2 2 0 0

**Code:**

house$Brick<-ifelse(house$Brick=="Yes",1,0)

house$Neighborhood<-ifelse(house$Neighborhood=="West",1,0)

head(house)

1. Generate a normalization function and apply this function to all variables in the House data frame. What are maximal and minimal values of SqFt? Confirm the range is now between zero and one.

**Ans:**

maximal values of SqFt = 1

minimal values of SqFt = 0

Yes, the range is between 0 and 1

**Code:**

normalize <- function(house, exclude = c("")){

i<-0

for(j in names(house)){

i <- i + 1

if (identical((j %in% exclude), TRUE) || identical((i %in% exclude), TRUE)) next;

if(is.numeric(house[[j]])) {

house[j] <- (house[j]-min(house[j])) / (max(house[j])-min(house[j]))

}

}

return(house)

}

house <- normalize(house)

summary(house)

1. Randomize the normalized data frame and create training and testing samples on randomized data frame. Select the first 70% of houses as training sample (you can round the number using floor() or other functions) and the rest 30% of houses as testing sample.

**Ans:**

'data.frame': 89 obs. of 7 variables:

$ Price : num 0.795 0.367 0.577 0.403 0.402 ...

$ SqFt : num 0.702 0.596 0.57 0.614 0.667 ...

$ Bedrooms : num 0.667 0.333 0.333 0.333 0.333 ...

$ Bathrooms : num 0.5 0 0 0.5 0.5 0.5 0 0 0 0 ...

$ Offers : num 0.4 0.4 0.4 0.8 0.6 0.2 0.4 0.2 0.4 0.8 ...

$ Brick : num 1 0 1 1 0 0 1 0 0 0 ...

$ Neighborhood: num 1 0 0 0 0 1 0 0 0 0 ...

data.frame': 39 obs. of 7 variables:

$ Price : num 0.318 0.317 0.322 0.574 0.246 ...

$ SqFt : num 0.298 0.509 0.254 0.623 0.246 ...

$ Bedrooms : num 0 0.667 0.333 0.667 0.333 ...

$ Bathrooms : num 0 0 0 0 0.5 0 0.5 0.5 0 0.5 ...

$ Offers : num 0.2 0.4 0 0.2 0.4 0.4 0.6 0.6 0 0.2 ...

$ Brick : num 0 0 0 0 0 1 1 0 1 1 ...

$ Neighborhood: num 0 0 0 1 0 0 0 0 0 1 ...

**Code:**

set.seed(123)

train.index <- sample(1:nrow(house), floor(0.7\*nrow(house)), replace = FALSE)

house.train <- house[train.index,]

house.test <- house[-train.index,]

str(house.train)

str(house.test)

1. Train a neuralnet model on training sample. Use 5 hidden neurons with Price as dependent variable and SqFt, Bedrooms, Bathrooms, Offers, recoded Brick and recoded Neighborhood as independent variables.   
   NOTE: in formula Price ~. may not work. You may have to write down all columns on the right side, ie Price ~ SqFT + Bedrooms +…

**Code:**

install.packages("neuralnet")

library(neuralnet)

model<- neuralnet(Price~.,data=house.train,hidden = 5)

model

1. Visualize the network topology by plotting the model. How many hidden nodes are included in the model? How many steps were used to build the model? What is the error value?

**Ans:**

2 layers of hidden nodes are there in the model.

Number of steps used to build the model = 465

Error value=0.134367

**Diagram

Description automatically generated**

**Code:**

predicted <- compute(model,house.train[-1])

head(predicted$net.result)

cor(predicted$net.result, house.train$Price)

plot(model)

1. Obtain the model prediction by applying the model on testing sample. Get the predictions of house price and examine the correlation between prediction and actual house price from testing sample. What is the correlation between these two variables?

**Ans:**

Correlation prediction and house price = 0.9143699

**Code:**

model.test<- neuralnet(Price~.,data=house.test,hidden = 5)

predicted <- compute(model.test,house.test[-1])

head(predicted$net.result)

cor(predicted$net.result, house.test$Price)

1. Make a residual plot (i.e. residuals on y-axis and fitted values on x-axis). Do you see any trend?   
   Hint: residuals<- dt.test$Price - predicted$net.result

**Ans:**

Yes there is a trend, The residuals tend to fall at first, with lowest point in the median of the fitted value, then it starts increasing .

**Code:**

residuals<- house.test$Price - predicted$net.result

plot(predicted$net.result,residuals,main="Residuals vs fitted values",xlab="Fitted values",ylab="Residuals**")**

1. Now run the model using SVM on training dataset and use kernel='vanilladot'. Use this model to predict Price using testing dataset. Examine the correlation between prediction and actual house price from testing sample. What is the correlation between these two variables?

**Ans:**

**Correlation - 0.9039639**

**Code:**

install.packages("e1071")

library(e1071)

install.packages("kernlab")

library(kernlab)

model.svm <- ksvm(Price~., data=house.train, kernel="vanilladot")

pred.svm <- predict(model.svm, house.test[-1])

head(pred.svm)

cor(pred.svm, house.test$Price)