MachineLearning

Babatunde Oguntade

October 8, 2019

## Set Up

Download R from <https://www.r-project.org/>

Download RStudio from <https://rstudio.com/products/rstudio/download/>

## Data Collection

For this exercise, we will utilize data on the compressive strength of concrete donated to the UCI Machine Learning Data Repository (<http://archive.ics.uci.edu/ml>) by I-Cheng Yeh. As he found success using neural networks to model these data, we will attempt to replicate his work using a simple neural network model in R.

For more information on Yeh’s approach to this learning task, refer to: **Yeh IC. Modeling of strength of high performance concrete using artificial neural networks. Cement and Concrete Research. 1998; 28:1797-1808**.

According to the website, the concrete dataset contains 1,030 examples of concrete with eight features describing the components used in the mixture. These features are thought to be related to the final compressive strength and they include the amount (in kilograms per cubic meter) of cement, slag, ash, water, superplasticizer, coarse aggregate, and fine aggregate used in the product in addition to the aging time (measured in days).

concrete <- read.csv("concrete.csv")  
str(concrete)

## 'data.frame': 1030 obs. of 9 variables:  
## $ cement : num 141 169 250 266 155 ...  
## $ slag : num 212 42.2 0 114 183.4 ...  
## $ ash : num 0 124.3 95.7 0 0 ...  
## $ water : num 204 158 187 228 193 ...  
## $ superplastic: num 0 10.8 5.5 0 9.1 0 0 6.4 0 9 ...  
## $ coarseagg : num 972 1081 957 932 1047 ...  
## $ fineagg : num 748 796 861 670 697 ...  
## $ age : int 28 14 28 28 28 90 7 56 28 28 ...  
## $ strength : num 29.9 23.5 29.2 45.9 18.3 ...

summary(concrete)

## cement slag ash water   
## Min. :102.0 Min. : 0.0 Min. : 0.00 Min. :121.8   
## 1st Qu.:192.4 1st Qu.: 0.0 1st Qu.: 0.00 1st Qu.:164.9   
## Median :272.9 Median : 22.0 Median : 0.00 Median :185.0   
## Mean :281.2 Mean : 73.9 Mean : 54.19 Mean :181.6   
## 3rd Qu.:350.0 3rd Qu.:142.9 3rd Qu.:118.30 3rd Qu.:192.0   
## Max. :540.0 Max. :359.4 Max. :200.10 Max. :247.0   
## superplastic coarseagg fineagg age   
## Min. : 0.000 Min. : 801.0 Min. :594.0 Min. : 1.00   
## 1st Qu.: 0.000 1st Qu.: 932.0 1st Qu.:731.0 1st Qu.: 7.00   
## Median : 6.400 Median : 968.0 Median :779.5 Median : 28.00   
## Mean : 6.205 Mean : 972.9 Mean :773.6 Mean : 45.66   
## 3rd Qu.:10.200 3rd Qu.:1029.4 3rd Qu.:824.0 3rd Qu.: 56.00   
## Max. :32.200 Max. :1145.0 Max. :992.6 Max. :365.00   
## strength   
## Min. : 2.33   
## 1st Qu.:23.71   
## Median :34.45   
## Mean :35.82   
## 3rd Qu.:46.13   
## Max. :82.60

normalize <- function(x) {  
return((x - min(x)) / (max(x) - min(x)))  
}

concrete\_norm <- as.data.frame(lapply(concrete, normalize))  
summary(concrete\_norm)

## cement slag ash water   
## Min. :0.0000 Min. :0.00000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.2063 1st Qu.:0.00000 1st Qu.:0.0000 1st Qu.:0.3442   
## Median :0.3902 Median :0.06121 Median :0.0000 Median :0.5048   
## Mean :0.4091 Mean :0.20561 Mean :0.2708 Mean :0.4774   
## 3rd Qu.:0.5662 3rd Qu.:0.39775 3rd Qu.:0.5912 3rd Qu.:0.5607   
## Max. :1.0000 Max. :1.00000 Max. :1.0000 Max. :1.0000   
## superplastic coarseagg fineagg age   
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.00000   
## 1st Qu.:0.0000 1st Qu.:0.3808 1st Qu.:0.3436 1st Qu.:0.01648   
## Median :0.1988 Median :0.4855 Median :0.4654 Median :0.07418   
## Mean :0.1927 Mean :0.4998 Mean :0.4505 Mean :0.12270   
## 3rd Qu.:0.3168 3rd Qu.:0.6640 3rd Qu.:0.5770 3rd Qu.:0.15110   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.00000   
## strength   
## Min. :0.0000   
## 1st Qu.:0.2664   
## Median :0.4001   
## Mean :0.4172   
## 3rd Qu.:0.5457   
## Max. :1.0000

Following Yeh’s precedent in the original publication, we will partition the data into a training set with 75 percent of the examples and a testing set with 25 percent. The CSV file we used was already sorted in random order, so we simply need to divide it into two portions:

concrete\_train <- concrete\_norm[1:773, ]  
concrete\_test <- concrete\_norm[774:1030, ]

We’ll use the training dataset to build the neural network and the testing dataset to evaluate how well the model generalizes to future results. As it is easy to overfit a neural network, this step is very important.

To model the relationship between the ingredients used in concrete and the strength of the finished product, we will use a multilayer feedforward neural network. The **neuralnet** package by *Stefan* *Fritsch* and *Frauke* *Guenther* provides a standard and easy-to-use implementation of such networks. It also offers a function to plot the network topology. For these reasons, the **neuralnet** implementation is a strong choice for learning more about neural networks, though this is not to say that it cannot be used to accomplish real work as well-it’s quite a powerful tool, as you will soon see.

#install.packages("neuralnet")  
library(neuralnet)

concrete\_model <- neuralnet(strength ~ cement + slag + ash + water + superplastic + coarseagg + fineagg + age, data = concrete\_train,hidden = 0)

plot(concrete\_model)

model\_results <- compute(concrete\_model, concrete\_test[1:8])  
predicted\_strength <- model\_results$net.result  
cor(predicted\_strength, concrete\_test$strength)

## [,1]  
## [1,] 0.7506367

concrete\_model2 <- neuralnet(strength ~ cement + slag + ash + water + superplastic + coarseagg + fineagg + age, data = concrete\_train, hidden = 5)

plot(concrete\_model2)

model\_results2 <- compute(concrete\_model2, concrete\_test[1:8])  
predicted\_strength2 <- model\_results2$net.result  
cor(predicted\_strength2, concrete\_test$strength)

## [,1]  
## [1,] 0.9343708

Interestingly, in the original publication, Yeh reported a mean correlation of **0.885** using a very similar neural network. This means that with relatively little effort, we were able to match the performance of a subject-matter expert. If you’d like more practice with neural networks, you might try applying the principles learned earlier in this chapter to see how it impacts model performance.