

Neural Network

Claudius Taylor

1/9/2019

```
# loading data
rawdata <- fread('./Peptide1.csv',stringsAsFactors = FALSE)

output_column <- c('#Spec')
target <- rawdata %>%
  select(output_column) %>%
  rename("spec" = "#Spec") # avoid using special character

input_columns <- c('-10lgP', 'Mass', 'Length', 'ppm', 'm/z', 'RT', '1/KO Range')
input_data <- rawdata %>%
  select(input_columns) %>%
  rename( "lgP" = "-10lgP", "mz" = 'm/z', "KRange" = '1/KO Range')

input_data <- input_data %>%
  separate( col = "KRange", into = c('min', 'max'), sep = '-') %>%
  mutate( min = as.numeric(min), max = as.numeric(max), range = max-min)

min_max_normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
} # min/max normalize function output between 0 and 1

input_data <-input_data %>%
  lapply(min_max_normalize) %>%
  as.data.table()

processed_data <- cbind(input_data, target)

n = nrow(processed_data)
train_sample <- runif(n, 0, 1) > 1 - 0.75 # boolean cover

train <- processed_data[train_sample,]
test <- processed_data[!train_sample,]

model <- neuralnet(formula = spec ~ lgP + Mass + Length + ppm + mz + RT + min + max + range,
  data = train,
  hidden = 1,
  threshold = 0.4,
  stepmax = 1e5,
  algorithm = "rprop+") # allow more iterations to converge

model_results <- compute(model, test[, 1:9])
predicted <- model_results$net.result

mae <- function(actual, predicted) {
  mean(abs(actual - predicted))
}
```

```
cor(predicted, test$spec)
```

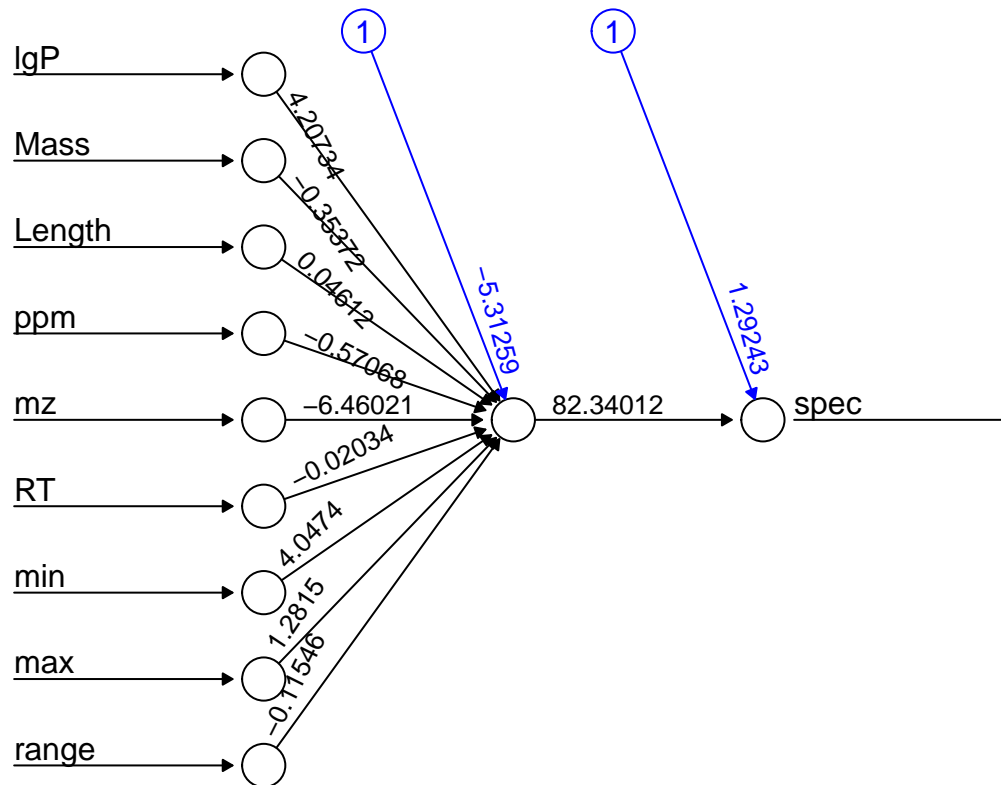
```
##           [,1]
```

```
## [1,] 0.2845872283
```

```
mae(predicted, test$spec)
```

```
## [1] 0.8979620886
```

```
plot(model, rep="best")
```



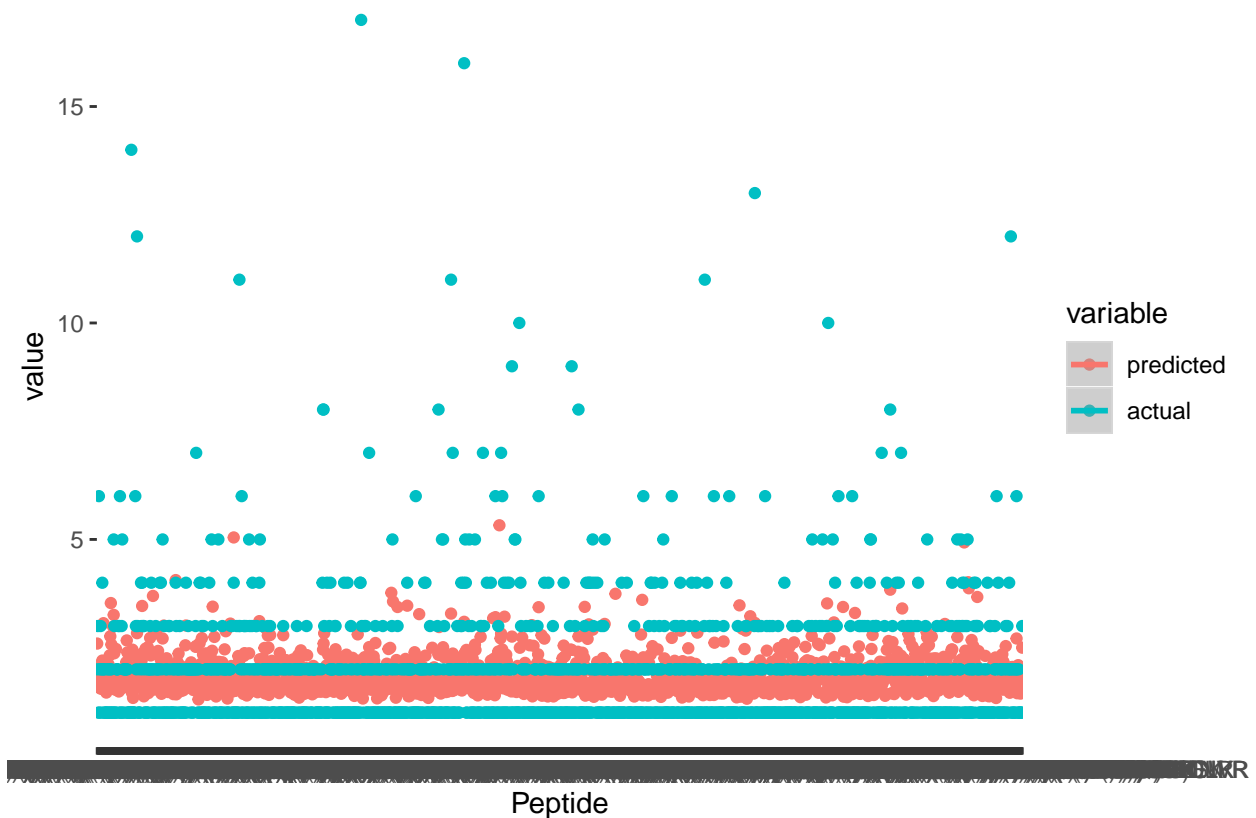
Error: 4788.599482 Steps: 10428

```
plot_data <- data.frame( peptide = rawdata[!train_sample, "Peptide"],  
                          predicted = predicted,  
                          actual = test$spec) %>%  
  melt()
```

```
## Using Peptide as id variables
```

```
ggplot(plot_data, aes(x = Peptide, y = value, color = variable)) +  
  geom_point() +  
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



```

model5 <- neuralnet(formula = spec ~ lgP + Mass + Length + ppm + mz + RT + min + max + range,
  data = train,
  hidden = 3,
  threshold = 0.4,
  stepmax = 1e5,
  algorithm = "rprop+")

model5_results <- compute(model5, test[, 1:9])
predicted5 <- model5_results$net.result

cor(predicted5, test$spec)

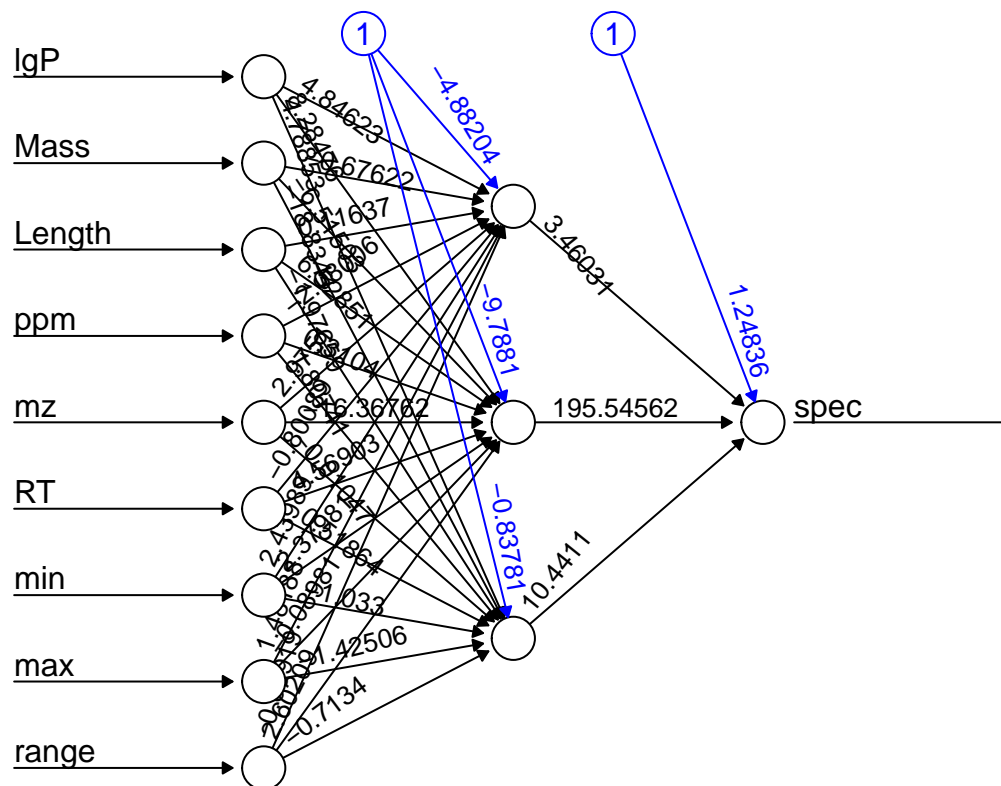
##           [,1]
## [1,] 0.3921995444

mae(predicted5, test$spec)

## [1] 0.8834841502

plot(model5, rep="best")

```



Error: 4454.504346 Steps: 4905

```
plot_data5 <- data.frame( peptide = rawdata[!train_sample, "Peptide"],
  predicted = predicted5,
  actual = test$spec) %>%
  melt()
```

```
## Using Peptide as id variables
```

```
ggplot(plot_data5,aes( x = Peptide, y = value, color = variable)) +
  geom_point() +
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

