DSE6211 Module 02 Lab 01

Joseph Annand

2024-01-25

## Load Libraries

library(reticulate)

## Warning: package 'reticulate' was built under R version 4.3.2

library(keras)

## Warning: package 'keras' was built under R version 4.3.2

library(tensorflow)

## Warning: package 'tensorflow' was built under R version 4.3.2

use\_virtualenv("my\_tf\_workspace", required = TRUE)

## Neural Network

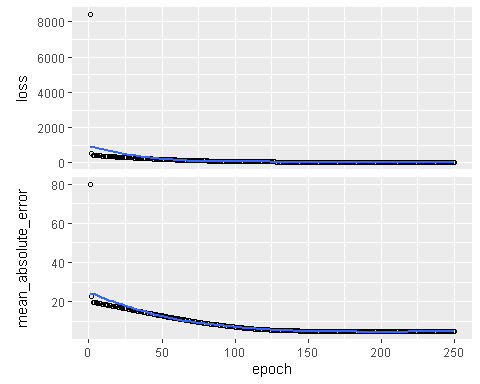
mtcars <- mtcars  
  
mtcars\_x <- mtcars[, c("cyl", "disp", "hp")]  
mtcars\_x <- array(data = unlist(mtcars\_x),  
 dim = c(32, 3),  
 dimnames = list(rownames(mtcars\_x),  
 colnames(mtcars\_x)))  
  
mtcars\_y <- mtcars[, "mpg"]  
  
nn\_model <- keras\_model\_sequential() %>%  
 layer\_dense(units = 1, input\_shape = 3, activation = "linear")  
  
nn\_model

## Model: "sequential"  
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
## Layer (type) Output Shape Param #   
## ================================================================================  
## dense (Dense) (None, 1) 4   
## ================================================================================  
## Total params: 4 (16.00 Byte)  
## Trainable params: 4 (16.00 Byte)  
## Non-trainable params: 0 (0.00 Byte)  
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# 4 weights in the output layer: 3 for units of the input layer and 1 for the bias  
  
nn\_model <- keras\_model\_sequential() %>%  
 layer\_dense(units = 2, input\_shape = 3, activation = "relu") %>%  
 layer\_dense(units = 1, activation = "linear")  
  
nn\_model

## Model: "sequential\_1"  
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
## Layer (type) Output Shape Param #   
## ================================================================================  
## dense\_2 (Dense) (None, 2) 8   
## dense\_1 (Dense) (None, 1) 3   
## ================================================================================  
## Total params: 11 (44.00 Byte)  
## Trainable params: 11 (44.00 Byte)  
## Non-trainable params: 0 (0.00 Byte)  
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# 8 weights for the first layer: 6 for the connections between the input layer with   
# 3 units and the hidden layer with 2 units (3x2 = 6) and two for the bias  
  
nn\_model %>% compile(optimizer = optimizer\_adam(learning\_rate = 0.2),  
 loss = "mean\_squared\_error",  
 metrics = "mean\_absolute\_error")  
  
nn\_model\_training <- nn\_model %>% fit(x = mtcars\_x,  
 y = mtcars\_y,  
 epochs = 250,  
 verbose = FALSE)  
  
plot(nn\_model\_training)



get\_weights(nn\_model)

## [[1]]  
## [,1] [,2]  
## [1,] -1.8021748 0.4014661  
## [2,] -0.7321286 0.1197203  
## [3,] -1.7549776 -0.4472417  
##   
## [[2]]  
## [1] -1.284455 0.000000  
##   
## [[3]]  
## [,1]  
## [1,] 0.1615062  
## [2,] 0.5990912  
##   
## [[4]]  
## [1] 19.81955

## Exercsises

1.)

prediction <- predict(nn\_model, array(c(8, 250, 200), dim = c(1,3)))

## 1/1 - 0s - 63ms/epoch - 63ms/step

prediction

## [,1]  
## [1,] 19.81955

This vehicle has a predicted miles per gallon of 19.81955.

2.) The loss function that is specified in the neural network above is mean squared error.

3.) The rank of a tensor is the number of axes that the tensor possesses. The rank of a matrix is 2 because it has two axes.

4.) Vector data is a rank 1 tensor because it has one axis.

5.) A tensor defined as: array(1, dim=c(500, 256, 256, 3)) has a rank of 4. Its shape is (500, 256, 256, 3). The dimension of the second axis is 256. If the tensor represented a collection of color images, there would be 500 images in the tensor.