

Test Docker

November 14, 2021

This document is a test of Docker. The code is drawn from Section 10.9.1 of ISLR2. Please click [Knit](#) to verify that you can successfully compile this document.

First let's load some libraries (all of which are packaged in this Docker container).

```
library (ISLR2)      # for Hitters data
library(kableExtra)  # for nice tables
library(glmnet)      # for lasso
library(keras)       # for deep learning
library(tidyverse)   # for everything else
```

First let's run the linear model:

```
# -----
# test linear model
# -----

# create data; sample test observations
Gitters <- na.omit (Hitters)
n <- nrow (Gitters)
set.seed (13)
ntest <- trunc (n / 3)
testid <- sample (1:n, ntest)

# fit linear model and compute prediction error
lfit <- lm(Salary ~ ., data = Gitters[-testid , ])
lpred <- predict (lfit , Gitters[testid , ])
lm_error = with (Gitters[testid , ], mean (abs (lpred - Salary)))
```

Next let's run the lasso:

```
# -----
# test glmnet
# -----

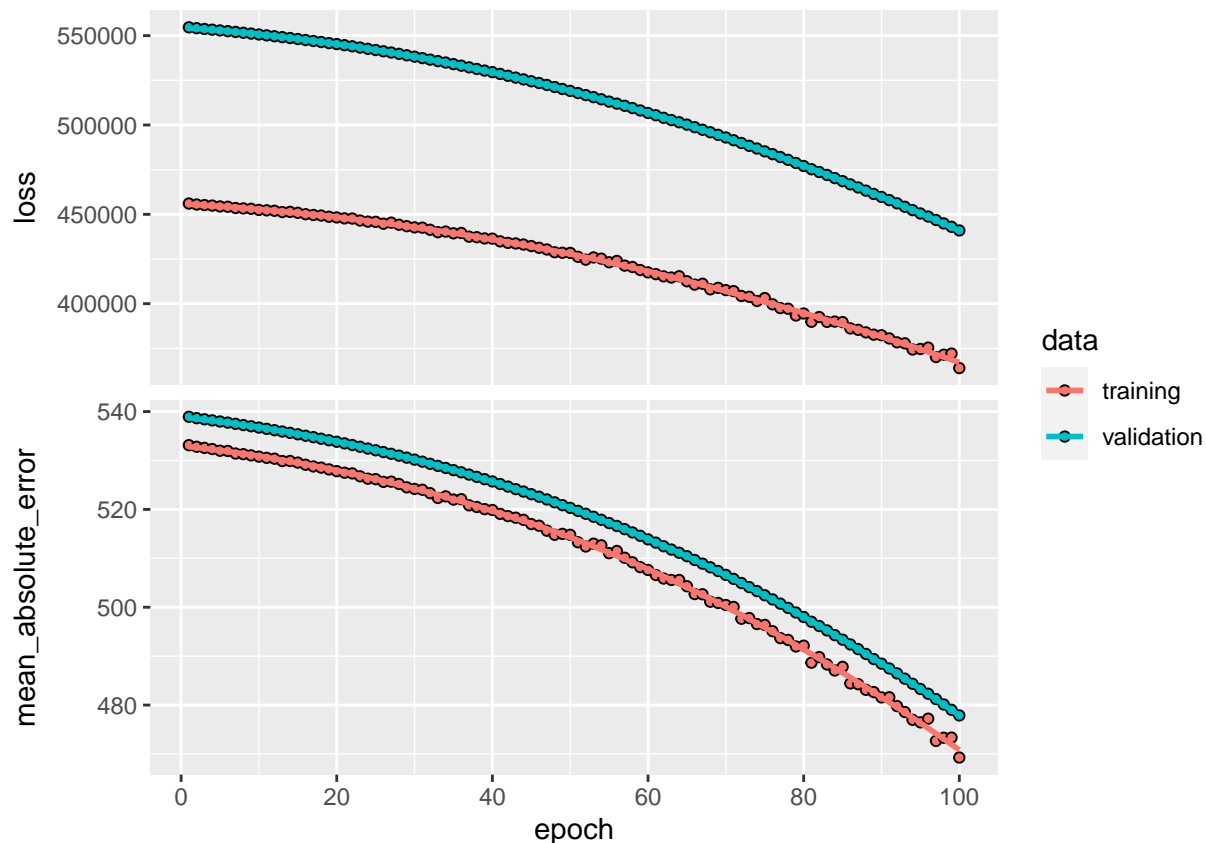
# load glmnet library for lasso
library (glmnet)

# fit lasso
x <- scale(model.matrix(Salary ~ . - 1, data = Gitters))
y <- Gitters$Salary
cvfit <- cv.glmnet (x[-testid , ], y[-testid],
                   type.measure = "mae")

# compute prediction error
cpred <- predict (cvfit , x[testid , ], s = "lambda.min")
lasso_error = mean(abs (y[testid] - cpred))
```

Finally we train a neural network:

```
# -----  
# test keras  
# -----  
  
# load keras library for deep learning  
library(keras)  
  
# define neural network structure  
modnn <- keras_model_sequential() %>%  
  layer_dense(units = 50, activation = "relu",  
              input_shape = ncol(x)) %>%  
  layer_dropout(rate = 0.4) %>%  
  layer_dense(units = 1)  
  
## Loaded Tensorflow version 2.5.0  
  
# compile the model  
modnn %>% compile(loss = "mse",  
                 optimizer = optimizer_rmsprop(),  
                 metrics = list("mean_absolute_error"))  
  
# fit the model  
history <- modnn %>% fit(x[-testid, ],  
                        y[-testid],  
                        epochs = 100,  
                        batch_size = 32,  
                        validation_data = list(x[testid, ], y[testid]))  
  
# plot training progress  
plot(history)  
  
## `geom_smooth()` using formula 'y ~ x'
```



```
# evaluate test error
npred <- predict(modnn, x[testid, ])
deep_learning_error = mean(abs(y[testid] - npred))
```

Let's compare the prediction errors:

```
tibble(Method = c("Linear Model", "Lasso", "Neural Network"),
  `Mean absolute error` = c(lm_error, lasso_error, deep_learning_error)) %>%
  kable(format = "latex", row.names = NA,
    booktabs = TRUE,
    digits = 2,
    caption = "Comparing the test errors of three different
      prediction methods.") %>%
  kable_styling(position = "center") %>%
  kable_styling(latex_options = "HOLD_position")
```

Table 1: Comparing the test errors of three different prediction methods.

Method	Mean absolute error
Linear Model	254.67
Lasso	252.30
Neural Network	477.88

Table 1 shows the three prediction errors. Here the neural network does poorly, but this is a reflection that we did not use enough epochs.