Statistical Learning with Deep Artificial Neural Networks

tfruns, a package of tools for Tensorflow training runs

FRC-EVL

Master's degree in Statistics and Operations Research, UPC- UB, 2021/22

Table of Contents

Introduction

Overview

Use the **tfruns** package to:

- Track the hyperparameters, metrics, output, and source code of every training run.
- Compare hyperparameters and metrics across runs to find the best performing model.
- Automatically generate reports to visualize individual training runs or comparisons between runs.

For more information https://tensorflow.rstudio.com/tools/tfruns/overview/

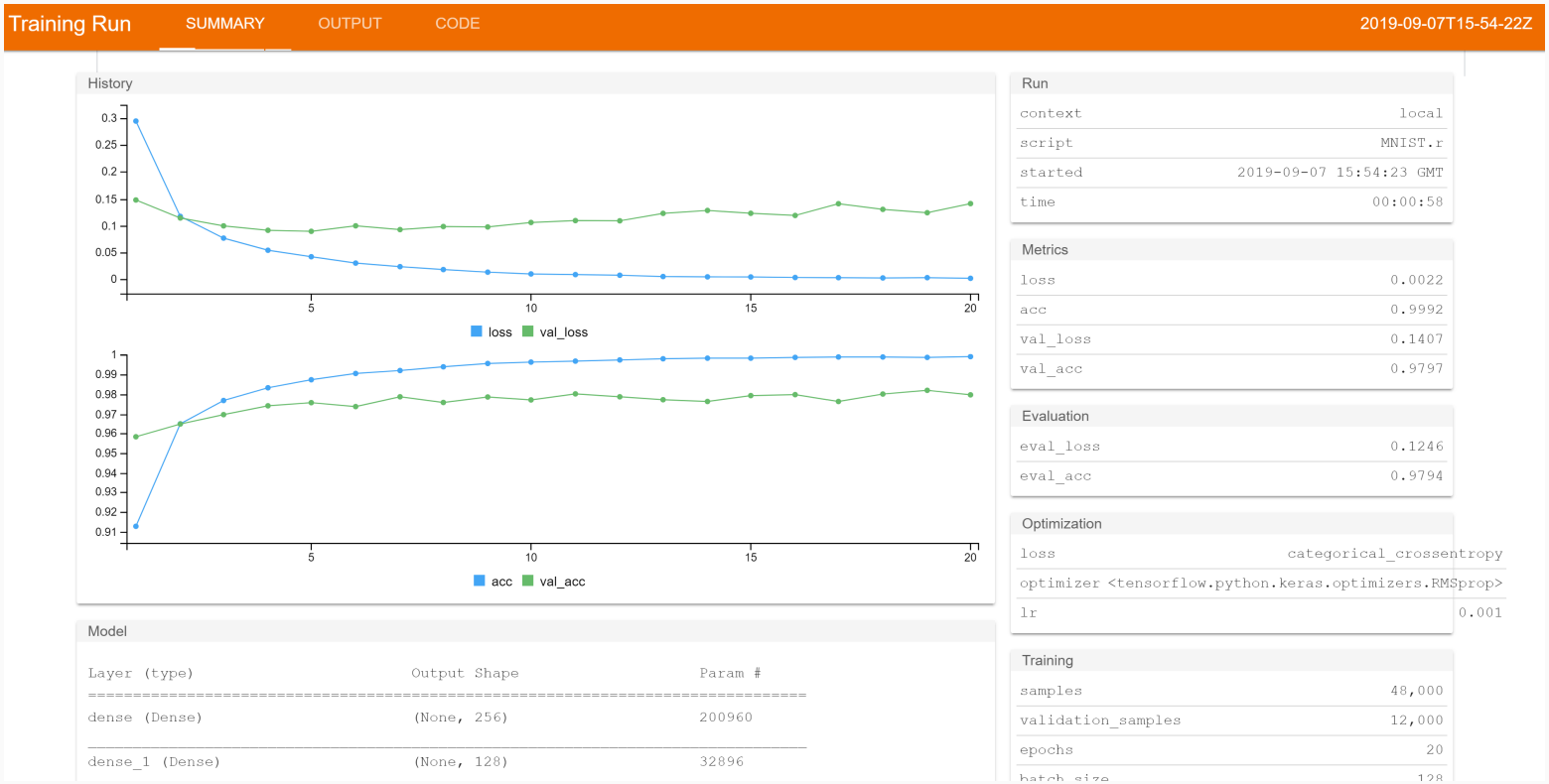
Training

Training (1/2)

To train a model with tfruns, just use the `training_run()` function to execute your R script. For example:

```
library(tfruns)
training_run("mnist.R")
```

Output:



Training (2/2)

The metrics and output of each run are automatically captured within a run directory which is unique for each run that you initiate. Some functions to view the results are:

- `latest_run()` shows the results of the last run.
- `view_run("runs/2022-03-06T16-28-09Z")` shows the results of a determined run.

From last version of Rstudio IDE there are tfruns functions into **Addins** menu to provides a quick access.

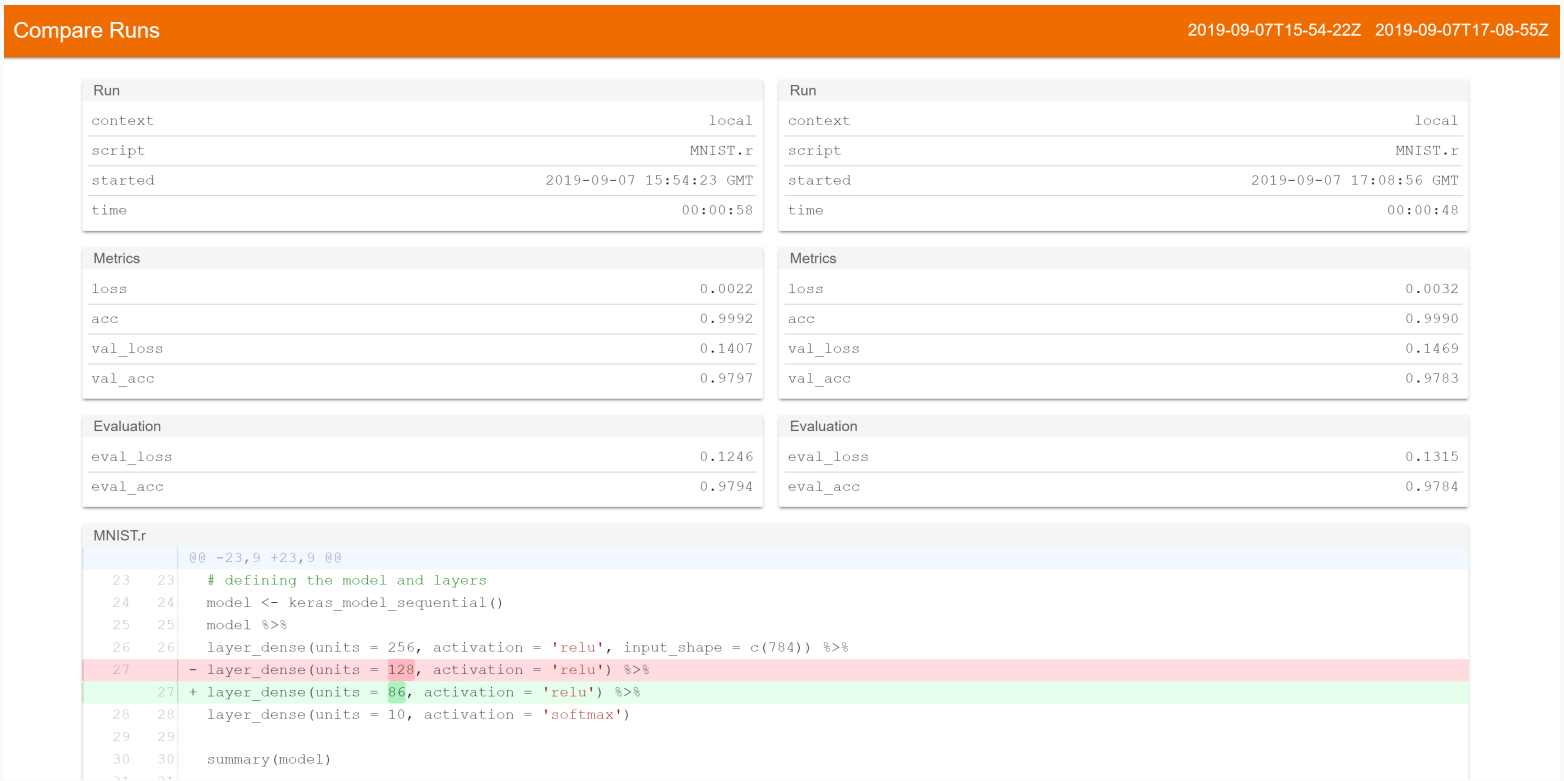
Comparing two training runs

Comparing runs

Some functions to comparing runs are:

- `compare_runs()` shows the comparison between this run and the previous one. However, you can pass any two run directories you like to be compared.

Output:



Tuning a model

Using flags (1/2)

Tuning a model often requires exploring the impact of changes to many hyperparameters. Then, we can define flags for key hyperparameters that you want to vary. **Step 1**: Modify r code.

```
# Define the flags
FLAGS <- flags(
  flag_numeric("hl1", 256),
  flag_numeric("hl2", 128)
)
# Use this flags in the model
model <- keras_model_sequential()
model %>%
layer_dense(units = FLAGS$hl1, activation = 'relu',
input_shape = c(784)) %>%
layer_dense(units = FLAGS$hl2, activation = 'relu') %>%
layer_dense(units = 10, activation = 'softmax')
```

Using flags (2/2)

Step 2: Pass alternate flag values with `training_run()`.

```
# simple case
training_run('mnist_flags.R',
  flags = c(hl1 = 200, hl2 = 100))

# loop case
for (hl1 in c(200, 256, 300))
  training_run('mnist_flags.R', flags = c(hl1 = hl1))
...
```

Comparing several training runs

Analyzing runs

You can use `ls_runs` function to compare many runs done.

Some examples of use:

```
# simple case, show all runs
ls_runs()
# Better presentation of results
View(ls_runs())
# show selection runs
ls_runs(metric_val_accuracy > 0.98,
        order = metric_val_accuracy)
# combine compare_runs() with ls_runs()
compare_runs(ls_runs(metric_val_accuracy > 0.98,
                     order = metric_val_accuracy))
```