Training ML models using GCP

Team MLoB

April 19, 2018

1 Introduction

This guide can be used for connecting with GCP (Google Cloud Platform) and for training your ML models using Cloud ML. We've added a folder with name 'root' which contains several files which you can use to train your models. This guide assumes that you're using Keras, so if you're using PyTorch or any other package, then you'll have to alter the contents of the mentioned folder.

There's a (big) chance that the tutorial is not explanatory enough. If you're in need of assistance or have any questions, then feel free to contact us (please try Google first as Google knows everything):

- **Mick van Hulst:** J.vanHulst@student.ru.nl
- **Dennis verheijden:** d.verheijden@student.ru.nl

2 GCP setup

The TA's of this course have created an unique project for each group on GCP. After logging into GCP, you first have to select your project (top of the page).

After you've selected your project you need to create a bucket[1]. You can use the search bar at the top of the screen to go to the storage page (this is where you can create a new bucket). After creating a new bucket, make sure to note the region you choose as you'll need this later.

Lastly, to work with GCP from your computer, you'll need to install their SDK (using the following *tutorial*). This tutorial will prompt you to select a project for which you'll have to select 'Pilot project 1'.

[1] Basically a folder which we'll use to save data files and trained models.

3 Upload training data

To train your model, you first have to send your training data to GCP. This can be done in a number of ways, most people will probably want to send a folder (with train, test and maybe validation data). This can be done using the following (terminal) command:

```
gsutil cp -r data_dir gs://bucket_name
```

Where `data_dir` is the local directory where the files are stored. You can also specify where the files have to be saved, by adjusting the destination url, for example to: `gs://bucket_name/data`.

For copying separate files you can remove the `-r` (recursive) parameter and specify file source and destination:

```
gsutil cp -r local_path_to_file/file.extension \
gs://bucket_name/file.extension
```

Downloading data can be done by switching the **gs://url** with the local folder.

4 Folder structure and templates

Use the following folder structure for training your model on GCP:

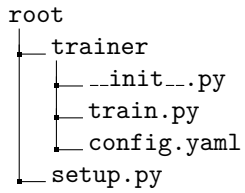

Figure 1: Folder structure GCP

4.1 Use cases files

The file `__init__.py` is an empty initialization file.

The file `train.py` is a template which can be used to combine your model with the structure needed to use GCP. The remainder of this tutorial assumes that you're using this template.

The file `config.yaml` is a file which can be used to adjust the amount of GPU's you want to use during training.

The file `setup.py` is a file that GCP uses to check which packages you want to use. Make sure to add all packages that you're using (excluding the standard Python packages) to this file.

5 Training your model

5.1 Local

Training your model locally with a small dataset is wise as it allows you to test your model after amending the changes which are described above. To test your model locally, use the following command:

```
gcloud ml-engine local train --module-name trainer.train \
--package-path path_to_root/trainer -- \
--train-file path_to_local_train_file/train_file.extension \
--job-dir ./tmp/test_script_gcp
```

5.2 Cloud

To test your model in the cloud use the following command, where the job name needs to be unique and where the region should correspond to the region of your bucket:

```
gcloud ml-engine jobs submit training job_name --stream-logs \
--runtime-version 1.4 --job-dir gs://bucket_name/job_name \
--package-path path_to_root/trainer --module-name trainer.train \
--region region_name  --config path_to_root/trainer/config.yaml \
-- \
--train-file gs://bucket_name/data_file.extension
```

After the 'empty' parameter (`--`), you may add your own parameters that can be passed to the trainer.

During training you can look monitor the progress of your model by going to the logging page on GCP. Make sure to apply a correct filter so that you only see logging data from the current job.

After training, you can use your model to make predictions. You'll have to download your model first and then make the predictions locally. Use the following command to download your model:

```
gsutil cp gs://bucket_name/job_name/model.h5 local_folder/
```