Kubeflow

Code

Code Structure

This model has the following steps: (a) extract data from BigQuery, transform it and write the transformed data to Cloud Storage. (b) Train a TensorFlow Estimator API model and do hyperparameter tuning of the model on the cloud. (c) Once the best learning rate, batch size, etc. are determined, train the model longer and on more data using those parameters (d) Deploy the trained model to Cloud ML Engine.

Deploying the notebook as a component of the pipeline

1. Top of the notebook accept Bucket(Storage), Project as parameters
2. Build a Docker image that is capable of executing the notebook. To execute a notebook, use the Python package papermill.
3. Use a .yaml file to pass parameters
4. Run the Docker Image

Disadvantages

- Has a lot of dependencies attached to Google Cloud Platform
- Cannot be downloaded with the free trial version of Google Cloud Platform since it costs more than 300$ (22,000 rupees) a year.
- Not many resources available online other than the official Kubeflow documentation and Google Cloud articles.
- More resources available on Kubeflow pipeline implemented with Python3 + Docker and fewer resources available with Jupyter Notebook.
- If you are using the GCP Free Tier or the 12-month trial period with $300 credit, note that you can’t run the default GCP installation of Kubeflow, because the free tier does not offer enough resources. You need to upgrade to a paid account.
- For more information, see the following issues:
  - kubeflow/website #1065 reports the problem.
  - kubeflow/kubeflow #3936 requests a Kubeflow configuration to work with a free trial project.

Advantages

- Functions with Azure and AWS
- Has more documentation than Nuclio
- Has a much more intuitive way of adding notebooks to the pipeline as compared to Nuclio
- Has been on the market longer and is more trusted than Nuclio
- Easier Coding practices and syntax