Building and Deploying Reproducible Machine Learning Pipelines

Reproducibility of modeling is a problem that exists for any machine learning practitioner. The consequences of an irreproducible model can include significant financial costs, lost time, and even loss of personal reputation (if results prove unable to be replicated). This page is developed to share what I found necessary to create reproducible codes that enable smooth collaborations and onboarding.

1. Introduction to creating a reproducible ML/AI pipeline

The following articles provide an overview of the different stages involved in creating a reproducible ML/AI pipeline.

- Building and Deploying a Reproducible Machine Learning Pipeline - article
- Building a Reproducible Machine Learning Pipeline - long article
- Reproducible Machine Learning - presentation, Kaggle
- The Machine Learning Reproducibility Crisis - article, by Google developer
- A systems perspective to reproducibility in Production Machine Learning
- Hidden technical debt in machine learning systems - Google

1.1 Scikit-Learn and Sklearn Pipeline (Recommended)

The following articles provide a step-by-step guidance on creating a reproducible pipeline based on widely popular Scikit-Learn and Sklearn Pipeline.

- Introduction to Scikit-Learn
- Six reasons why I recommend Scikit-Learn
- Why you should learn Scikit-Learn
- Deep dive into SKlearn pipelines from Kaggle
- SKlearn pipeline tutorial from Kaggle
- Managing Machine Learning workflows with Sklearn pipelines
- A simple example of pipeline in Machine Learning using SKlearn

Scikit-Learn and sklearn pipeline

Advantages

- Can be tested, versioned, tracked and controlled
- Can build future models on top
- Good software developer practice
- Leverages the power of acknowledged API
- Data scientists familiar with Pipeline use, reduced over-head
- Engineering steps can be packaged and re-used in future ML models

Disadvantages

- Requires team of software developers to build and maintain
- Overhead for software developers to familiarise with code for sklearn API ⇒ difficulties debugging

1.2 Facebook and Uber Piplelines (FYI)

The following articles provides other architectures used in alternative organizations (I have not used them personally):

Introducing FBLearner Flow: Facebook's AI backbone

Scaling Machine Learning as a service: Uber's pipeline

2. Links to threads regarding reproducibility in setting the Seed for Keras

To ensure you have reproducible results after using Keras for training stage, please follow the following guidelines.

- Keras documentation
- How to get reproducible results in keras, StackOverflow
- Any way to note down or control the random seeds? in git Keras issues
- mnist_cnn.py does not give reproducible results, in git Keras issues