Machine Learning Modeling Notes

Initial notes

Going into the first session of our Machine learning modeling class, the fundamentals we learned during our Exploratory Data Analysis proved to be consistent with the workflow in SageMaker.

Workflow

- Data Exploration
  - preprocessing
  - visualization
- Distributing data into training and testing sets
- Model training and deployment
  - straightforward approach
  - batch transform
- Model Evaluation

Key Takeaways

Which approach to adopt:

```
In [11]: from sagemaker import LinearLearner

         estimator = LinearLearner(role=role,
                                   instance_count=1,
                                   instance_type='ml.m5.xlarge',
                                   predictor_type='regressor',
                                   mini_batch_size=4)

In [12]: # upload into S3 behind the scenes
         record_set = estimator.record_set(train=X_train.reshape(-1,1).astype('float32'), labels=y_train.astype('float32'))

In [13]: record_set

Out[13]: (<class 'sagemaker.amazon.amazon_estimator.RecordSet'>, {'s3_data': 's3://sagemaker-us-east-1-305262579855/sagemaker-record-set
         s/LinearLearner-2021-05-08-07-51-08-424/.amazon.manifest', 'feature_dim': 1, 'num_records': 14, 's3_data_type': 'ManifestFile',
         'channel': 'train'})

In [14]: estimator.fit(record_set)

         Defaulting to the only supported framework/algorithm version: 1. Ignoring framework/algorithm version: 1.
         Defaulting to the only supported framework/algorithm version: 1. Ignoring framework/algorithm version: 1.

         2021-05-08 07:51:08 Starting - Starting the training job...
         2021-05-08 07:51:31 Starting - Launching requested ML instancesProfilerReport-1620460268: InProgress
         ......
         2021-05-08 07:52:38 Starting - Preparing the instances for training.........
         2021-05-08 07:54:09 Downloading - Downloading input data
         2021-05-08 07:54:09 Training - Downloading the training image...
         2021-05-08 07:54:39 Uploading - Uploading generated training model
         2021-05-08 07:54:39 Completed - Training job completed
         Docker entrypoint called with argument(s): train
         Running default environment configuration script
         [05/08/2021 07:54:28 INFO 139883830048576] Reading default configuration from /opt/amazon/lib/python3.7/site-packages/algorit
         hm/resources/default-input.json: {'mini_batch_size': '1000', 'epochs': '15', 'feature_dim': 'auto', 'use_bias': 'true', 'bina
         ry_classifier_model_selection_criteria': 'accuracy', 'f_beta': '1.0', 'target_recall': '0.8', 'target_precision': '0.8', 'num
         _models': 'auto', 'num_calibration_samples': '10000000', 'init_method': 'uniform', 'init_scale': '0.07', 'init_sigma': '0.0
         1', 'init_bias': '0.0', 'optimizer': 'auto', 'loss': 'auto', 'margin': '1.0', 'quantile': '0.5', 'loss_insensitivity': '0.0
         1', 'huber_delta': '1.0', 'num_classes': '1', 'accuracy_top_k': '3', 'wd': 'auto', 'l1': 'auto', 'momentum': 'auto', 'learnin
```

- simplest approach
- cheaper alternative
  - use when notebook instance is relatively weak in terms of performance and training data is large
  - mitigates the need to deploy a 'big' model just so we can process a large training set faster.
- setting hyperparameters is possible in SageMaker

Use Batch Transform

PDF | Kindle | RSS

Use batch transform when you need to do the following:

- Preprocess datasets to remove noise or bias that interferes with training or inference from your dataset.
- Get inferences from large datasets.
- Run inference when you don't need a persistent endpoint.
- Associate input records with inferences to assist the interpretation of results.

- SageMaker takes care of the heavy lifting for you since it manages all of the compute resources needed to acquire inferences.

HuggingFace Transformer Packages

- Python libraries that house pretrained models for Natural Language Processing tasks
- Enables more configurability to fine-tune a model's learning process through its hyperparameters

With regards to data

- data is always saved in S3
- manipulating data to be fed to different algorithms is crucial