Columbia University

AI and OR at Scale and Cloud

IEOR E4577

Assignment 4

Date – 02/23/2020

Group members:

Vibhor Malik (vm2599)
Sarthak Tiwari (st3284)
Curren Tipnis (cst2145)

Github link:

https://github.com/vibhormalik97/A.I.-CLOUD-4/tree/master

Pre-Processing ETL:

The preprocessing code from previous assignment required several changes such as integrating the code snippet provided in the assignment, making the end to end function, and removing np and nltk libraries.

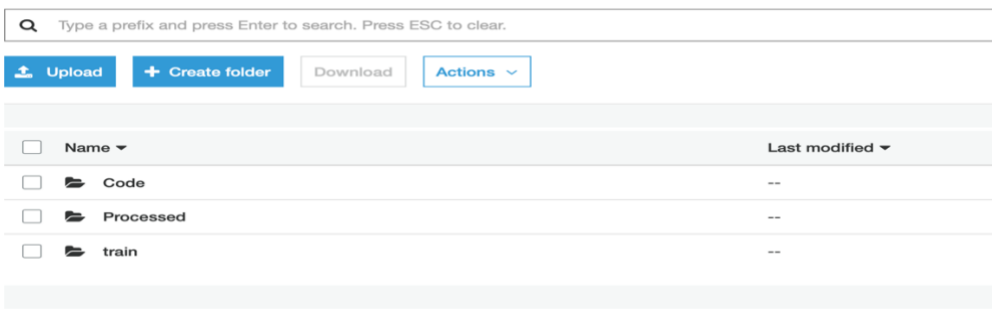
Preprocessing Code snippets are as follows:

```python
import os
from nltoolkit import TweetTokenizer
import re
import zipfile


class tweet:

    def __init__(self,max_length_tweet=20,max_length_dictionary=1000000):

        self.max_length_tweet = max_length_tweet
        self.max_lenght_dictionary = max_length_dictionary

        # Importing dictionary
        file_path = './Assignment_4.zip/dictionary.txt'
        archive_path = os.path.abspath(file_path)
        split = archive_path.split(".zip/")
        archive_path = split[0] + ".zip"
        path_inside = split[1]
        archive = zipfile.ZipFile(archive_path, "r")
        self.embeddings = archive.read(path_inside).decode("utf8").split("\n")
        self.embeddings = self.embeddings[:max_length_dictionary]

       #Importing Stopwords

        file_path = './Assignment_4.zip/english'
        archive_path = os.path.abspath(file_path)
        split = archive_path.split(".zip/")
        archive_path = split[0] + ".zip"
        path_inside = split[1]
        archive = zipfile.ZipFile(archive_path, "r")
        self.stopwords = archive.read(path_inside).decode("utf8").split("\n")
        self.tokenizer = TweetTokenizer()
```

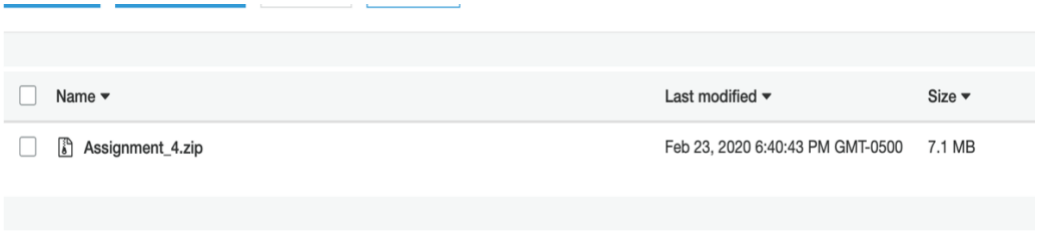
End to end function has been displayed below:

```python
    def e2e(self,text):
        "end to end function"
        clean =    self.clean_text(text)
        tokenize = self.tokenize_text(clean)
        index =    self.token_to_index(tokenize)
        emb_pad =  self.pad_sequence(index)

        return emb_pad
```

Data in S3 Bucket:

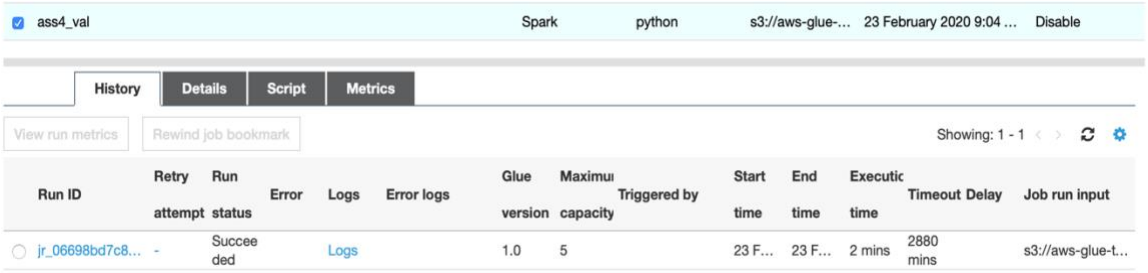


Preprocessing library in S3:

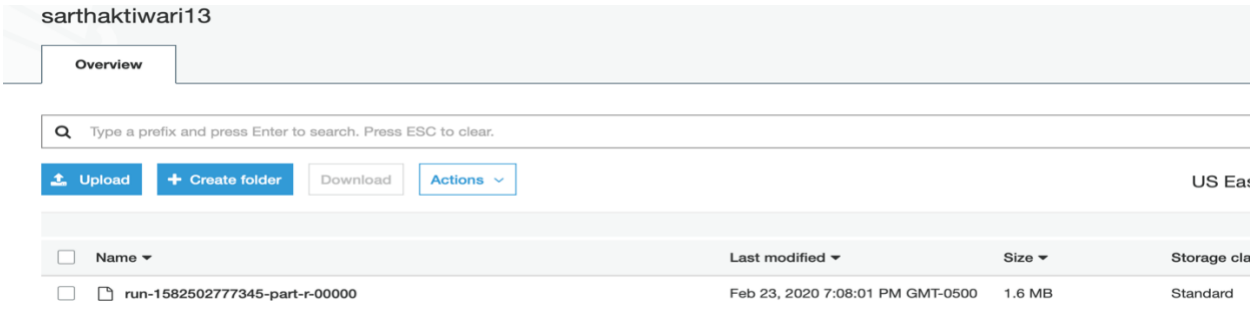


ETL jobs (one for each dataset : train,eval,dev):

```
import Class_ass4
from Class_ass4 import tweet
prep = tweet()
## @params: [JOB_NAME]
args = getResolvedOptions(sys.argv, ['JOB_NAME'])
sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args['JOB_NAME'], args)
## @type: DataSource
## @args: [database = "ass_4", table_name = "eval", transformation_ctx = "datasource0"]
## @return: datasource0
## @inputs: []
datasource0 = glueContext.create_dynamic_frame.from_catalog(database = "ass_4", table_name = "eval", transformation_ctx = "datasource0")
## @type: ApplyMapping
## @args: [mapping = [("col1", "string", "sentiment", "string"), ("col2", "string", "id", "string"), ("col6", "string", "tweet", "string")], t
## @return: applymapping1
## @inputs: [frame = datasource0]
applymapping1 = ApplyMapping.apply(frame = datasource0, mappings = [("col1", "string", "sentiment", "string"), ("col2", "string", "id", "strin
def map_function(dynamicRecord):
    tweet = dynamicRecord["tweet"]
    features = prep.e2e(tweet)
    dynamicRecord["features"] = features
    return dynamicRecord
map_processed = Map.apply(frame = applymapping1, f = map_function, transformation_ctx = "map_processed")
## @type: DataSink
## @args: [connection_type = "s3", connection_options = {"path": "s3://sarthaktiwari13/Ass_4/Processed/eval"}, format = "json", transformation
## @return: datasink2
## @inputs: [frame = applymapping1]
datasink2 = glueContext.write_dynamic_frame.from_options(frame = map_processed, connection_type = "s3", connection_options = {"path": "s3://sa
job.commit()
```



JSON file output in S3 :



3: Tensorflow model:

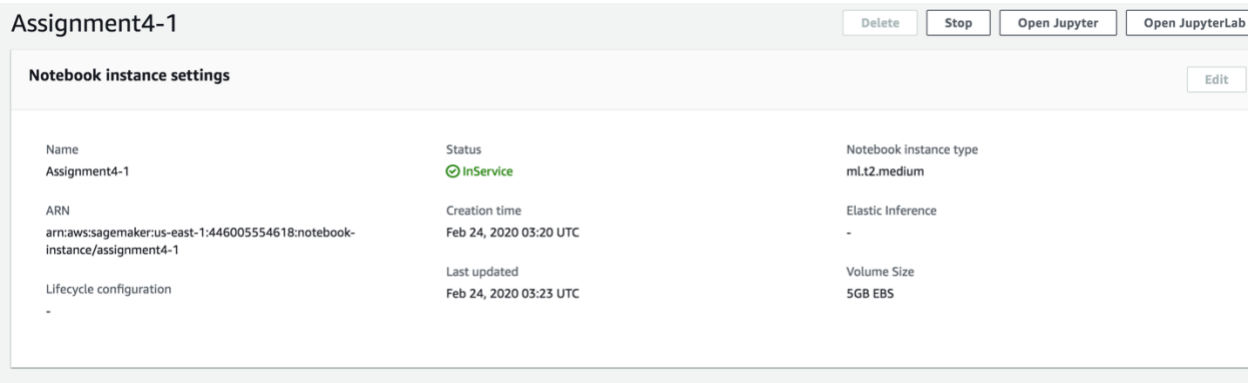
Model development code:

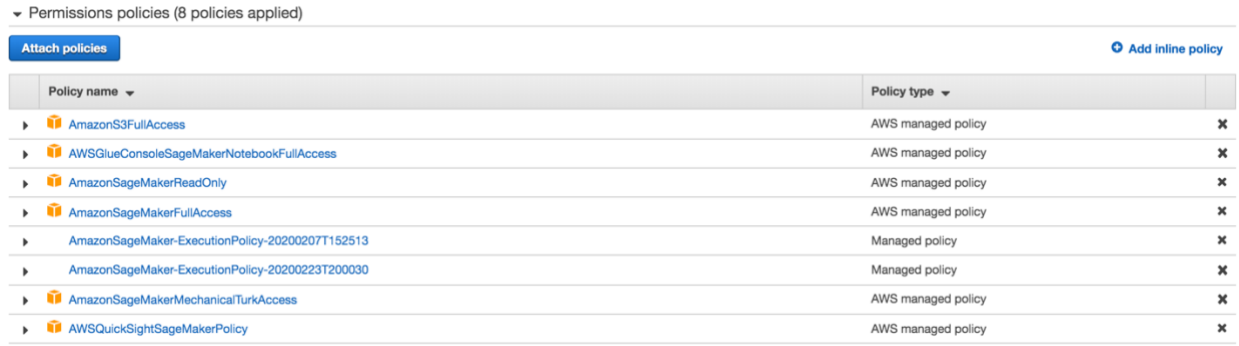
```
import os
import tensorflow as tf
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Embedding
from tensorflow.keras.layers import Conv1D
from tensorflow.keras.layers import GlobalMaxPool1D
from tensorflow.keras.optimizers import Adam

def keras_model_fn(_, config):
    """
    Creating a CNN model for sentiment modeling
    """
    # load the whole embedding into memory
    embeddings_index = dict()
    f = open(config["embeddings_path"], encoding="utf-8")
    for line in f:
        values = line.split()
        word = values[0]
        coefs = np.array(values[1:], dtype='float32')
        embeddings_index[word] = coefs
    f.close()
    print('Loaded %s word vectors.' % len(embeddings_index))

    vocab_size = config["embeddings_dictionary_size"]

    n = len(embeddings_index.keys())
    m = len(embeddings_index['the'])

    embedding_matrix = np.zeros((n,m))
    for index, key in zip(range(0, n), embeddings_index.keys()):
        embedding_matrix[index] = embeddings_index[key]

    # define model
    model = Sequential()
```

Code output as run on Terminal:
Test accuracy: 0.61
Number of epochs: 10

```
Loaded 1193514 word vectors.
2020-02-23 21:43:25.960119: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: SSE4.1 SSE4.2 AVX AVX2 FMA
2020-02-23 21:43:25.960308: I tensorflow/core/common_runtime/process_util.cc:69] Creating new thread pool with default inter op setting: 4. Tune using inter_op_parallelism_threads for best performance.
Defined model
Starting training...
/anaconda3/lib/python3.6/site-packages/tensorflow/python/ops/gradients_impl.py:112: UserWarning: Converting sparse IndexedSlices to a dense Tensor of unknown shape. This may consume a large amount of memo
ry.
  "Converting sparse IndexedSlices to a dense Tensor of unknown shape. "
Epoch 1/10
85/85 [==============================] - 66s 781ms/step - loss: 0.6915 - acc: 0.5387 - val_loss: 0.6762 - val_acc: 0.5660
Epoch 2/10
85/85 [==============================] - 66s 779ms/step - loss: 0.6633 - acc: 0.5929 - val_loss: 0.6639 - val_acc: 0.5830
Epoch 3/10
85/85 [==============================] - 68s 795ms/step - loss: 0.6285 - acc: 0.6533 - val_loss: 0.6519 - val_acc: 0.6110
Epoch 4/10
85/85 [==============================] - 67s 785ms/step - loss: 0.5855 - acc: 0.6872 - val_loss: 0.6533 - val_acc: 0.6250
Epoch 5/10
85/85 [==============================] - 65s 771ms/step - loss: 0.5282 - acc: 0.7387 - val_loss: 0.6485 - val_acc: 0.6300
Epoch 6/10
85/85 [==============================] - 65s 770ms/step - loss: 0.4672 - acc: 0.7791 - val_loss: 0.6937 - val_acc: 0.6170
Epoch 7/10
85/85 [==============================] - 66s 781ms/step - loss: 0.4032 - acc: 0.8168 - val_loss: 0.6786 - val_acc: 0.6500
Epoch 8/10
85/85 [==============================] - 66s 777ms/step - loss: 0.3418 - acc: 0.8519 - val_loss: 0.7477 - val_acc: 0.6420
Epoch 9/10
85/85 [==============================] - 68s 801ms/step - loss: 0.2824 - acc: 0.8829 - val_loss: 0.7769 - val_acc: 0.6410
Epoch 10/10
85/85 [==============================] - 68s 794ms/step - loss: 0.2392 - acc: 0.9029 - val_loss: 0.8154 - val_acc: 0.6490
Test loss:0.8695541858673096
Test accuracy:0.6100000023841858
Model successfully saved at: ./sentiment_model.h5
dyn-160-39-196-123:Assignment4 currentipnis$ cd ..
```

Sagemaker :





Some extra policies were tried to run the sagemaker

Code:

```
import pandas as pd
import sagemaker
import boto3
from time import gmtime, strftime
from sagemaker.tensorflow import TensorFlow
from sagemaker.tuner import IntegerParameter, CategoricalParameter, ContinuousParameter, HyperparameterTuner
```

99]:
```
import warnings
warnings.filterwarnings("ignore")
```

00]:
```
role = sagemaker.get_execution_role()
```

01]:
```
region = boto3.Session().region_name
smclient = boto3.Session().client('sagemaker')
```

04]:
```
estimator = TensorFlow(base_job_name='a4', \
                       entry_point='sentiment_training.py', \
                       source_dir='s3://twittertextdata/Assignment4.tar.gz', \
                       role=role, \
                       framework_version='1.14.0', py_version='py3', \
                       hyperparameters={'num_epoch' : 10, 'config_file': 'training_config.json'}, \
                       train_instance_count=1, train_instance_type='ml.m4.xlarge')
```

05]:
```
estimator.fit({'train' : 's3://twittertextdata/train', \
               'validation' : 's3://twittertextdata/dev', \
               'eval' : 's3://twittertextdata/eval'})
```

```
---------------------------------------------------------------------------
ClientError                               Traceback (most recent call last)
<ipython-input-105-a78989ea9466> in <module>()
```

Error:

```
~/anaconda3/envs/python3/lib/python3.6/site-packages/sagemaker/session.py in train(self, input_mode, input_config, ro
le, job_name, output_config, resource_config, vpc_config, hyperparameters, stop_condition, tags, metric_definitions,
enable_network_isolation, image, algorithm_arn, encrypt_inter_container_traffic, train_use_spot_instances, checkpoint
_s3_uri, checkpoint_local_path, experiment_config, debugger_rule_configs, debugger_hook_config, tensorboard_output_co
nfig, enable_sagemaker_metrics)
    567         LOGGER.info("Creating training-job with name: %s", job_name)
    568         LOGGER.debug("train request: %s", json.dumps(train_request, indent=4))
--> 569         self.sagemaker_client.create_training_job(**train_request)
    570
    571     def process(

~/anaconda3/envs/python3/lib/python3.6/site-packages/botocore/client.py in _api_call(self, *args, **kwargs)
    274                     "%s() only accepts keyword arguments." % py_operation_name)
    275             # The "self" in this scope is referring to the BaseClient.
--> 276             return self._make_api_call(operation_name, kwargs)
    277
    278         _api_call.__name__ = str(py_operation_name)

~/anaconda3/envs/python3/lib/python3.6/site-packages/botocore/client.py in _make_api_call(self, operation_name, api_p
arams)
    584             error_code = parsed_response.get("Error", {}).get("Code")
    585             error_class = self.exceptions.from_code(error_code)
--> 586             raise error_class(parsed_response, operation_name)
    587         else:
    588             return parsed_response

ClientError: An error occurred (AccessDeniedException) when calling the CreateTrainingJob operation: User: arn:aws:st
s::446005554618:assumed-role/AmazonSageMaker-ExecutionRole-20200223T200030/SageMaker is not authorized to perform: sa
gemaker:CreateTrainingJob on resource: arn:aws:sagemaker:us-east-1:446005554618:training-job/a4-2020-02-24-03-50-47-1
28 with an explicit deny
```

Many efforts were made to run the SageMaker successfully, for example : adding extra policies, changing the code, searching up the internet for suggestions but we were not able to solve this error.

END OF REPORT
THANK-YOU!!!