Project: Deep Learning Pipelines for Apache Spark

Emily Weng 20016

Tools:

- Google Colab
- Databrick was pending for in google cloud platform

Step 1

- In Colab, download Tensorflow, hadoop, and pyspark

```
50s
!pip install pyspark tensorflow
```

```
!apt-get install openjdk-8-jdk-headless -qq > /dev/null
!wget -q https://archive.apache.org/dist/spark/spark-3.1.2/spark-3.1.2-bin-hadoop2.7.tgz
!tar xf spark-3.1.2-bin-hadoop2.7.tgz
!pip install -q findspark
```

```
[10] import os
     os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
     os.environ["SPARK_HOME"] = "/content/spark-3.1.2-bin-hadoop2.7"
```

```
[11] import findspark
     findspark.init('/content/spark-3.1.2-bin-hadoop2.7')
     from pyspark.sql import SparkSession
     spark = SparkSession.builder.master("local[*]").getOrCreate()
```

Step 2:

- Get the flower dataset:

```
%sh
curl -O http://download.tensorflow.org/example_images/flower_photos.tgz
tar xzf flower_photos.tgz
```

Step 3:

- Make directories for the flower:

```
import os
img_dir = '/content/flower_photos'
os.makedirs(img_dir + "/tulips", exist_ok=True)
os.makedirs(img_dir + "/daisy", exist_ok=True)
```

Step 4:

Working with images in Spark

- The first step to applying deep learning on images is the ability to load the images.
- Since it was done in colab, I used shutil instead

```
Successfully copied from /content/flower_photos/tulips to content/photos/tulips
Successfully copied from /content/flower_photos/daisy to content/photos/daisy
Successfully copied LICENSE.txt
```

Step 5:

Transfer learning

- Deep Learning Pipelines provides utilities to perform transfer learning on images.
- Results:

```
Copied /content/content/photos/tulips/100930342_92e8746431_n.jpg to /content/content/photos/sample
Copied /content/content/photos/daisy/100080576_f52e8ee070_n.jpg to /content/content/photos/sample
Copied /content/content/photos/daisy/10140303196_b88d3d6cec.jpg to /content/content/photos/sample
Contents of sample_img_dir:
100930342_92e8746431_n.jpg
10140303196_b88d3d6cec.jpg
100080576_f52e8ee070_n.jpg
```

See how well it does:

Accuracy is at 0.812

(Example was 0.97

```
36/36 [==============================] - 288s 8s/step
9/9 [==============================] - 73s 8s/step
Test set accuracy = 0.812
/usr/local/lib/python3.10/dist-packages/sklearn/linear_mod
  warnings.warn(
```

Applying popular image models

- This part keeps having errors, so it wasn't done properly

```
Image paths: []
Results: []
RangeIndex(start=0, stop=0, step=1)
Empty DataFrame
Columns: []
Index: []
```

Output

```
                                                  filePath
0  /content/content/photos/sample/100930342_92e87...
1  /content/content/photos/sample/10140303196_b88...
2  /content/content/photos/sample/100080576_f52e8...
```

Clean up afterwards

```
# Define your directories
img_dir = '/content/content/photos'  # Update with your image directory path
dbfs_model_path = '/content/content/model'  # Update with your model path

# Remove directories
remove_dir(img_dir)
remove_dir(dbfs_model_path)
```

```
Removed directory: /content/content/photos
Directory does not exist: /content/content/model
```

Github link

https://github.com/emilywengster/sfbu/tree/9a8bd031c7bc51c778a64beafcd0bab900685992/Cloud%20Computing/Machine%20Learning/Apache%20Spark%20%2B%20Deep%20Learning