Project:
Introducing Deep Learning Pipelines for Apache Spark

Prachi Sethi 19963

Table of Contents

- Introduction
- Overview and Philosophy
- Steps
- Cluster Setup
- Compatibility
- Tools and Capabilities
- Working with Images in Spark DataFrames
- Transfer Learning
- Applying Models at Scale
- Conclusion
- References

Introduction

Deep Learning Pipelines: A new library by Databricks.

Purpose: High-level APIs for scalable deep learning model application and transfer learning.

Integration: Combines popular deep learning libraries with MLlib Pipelines and Spark SQL.

Overview and Philosophy

Overview: For detailed usage examples, refer to the Deep Learning Pipelines README.

Philosophy: Check out the Databricks blog post for the philosophy behind the library.

Steps

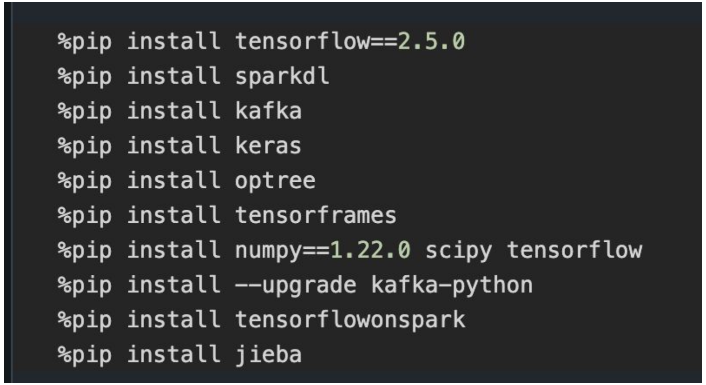
Step 1- https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/5669198905533692/3647723071348946/3983381308530741/latest.html

Step 2- Create a data bricks account

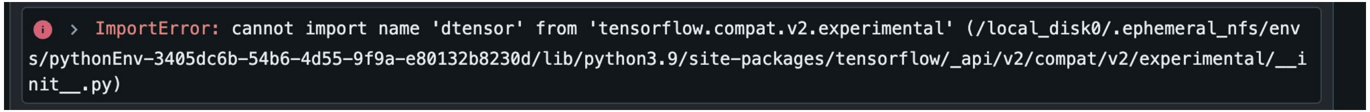
Step 3- Import the notebook and install packages

Step 4 - Create the cluster

Step 5- Run the cells one by one



```
%pip install tensorflow==2.5.0
%pip install sparkdl
%pip install kafka
%pip install keras
%pip install optree
%pip install tensorframes
%pip install numpy==1.22.0 scipy tensorflow
%pip install --upgrade kafka-python
%pip install tensorflowonspark
%pip install jieba
```



```
ImportError: cannot import name 'dtensor' from 'tensorflow.compat.v2.experimental' (/local_disk0/.ephemeral_nfs/env
s/pythonEnv-3405dc6b-54b6-4d55-9f9a-e80132b8230d/lib/python3.9/site-packages/tensorflow/_api/v2/compat/v2/experimental/__i
nit__.py)
```

Cluster Setup

Availability: Deep Learning Pipelines is available as a Spark Package.

Setup Steps:

Create a new library with Source option "Maven Coordinate".

Search Spark Packages and Maven Central for "spark-deep-learning".

Attach the library to a cluster.

Compatibility

Version: Works with spark-deep-learning release 0.1.0-spark2.1-s_2.11.

Future Releases: Check the project's GitHub page for the latest examples and docs.

Libraries: Also create and attach these libraries via PyPI: tensorflow, keras, h5py.

Spark Version: Compatible with Spark versions 2.0 or higher.

Instance Types: Works with any instance type (CPU or GPU).

Tools and Capabilities

Image Processing: Tools for working with images using deep learning.

Categories:

Working with Images: Natively in Spark DataFrames.

Transfer Learning: Leverage deep learning quickly.

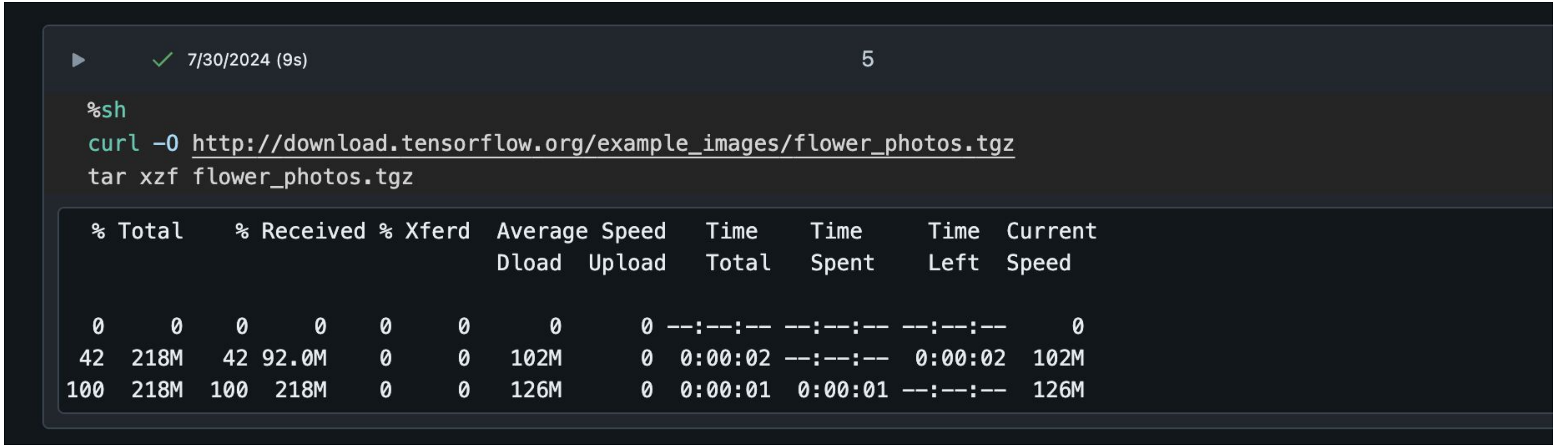
Model Application: Apply deep learning models at scale.

SQL Functions: Deploy models as SQL functions (coming soon).

Hyper-parameter Tuning: Distributed tuning via Spark MLlib Pipelines (coming soon).

Working with Images in Spark DataFrames

Capability: Load, transform, and analyze images directly within Spark DataFrames.



▶ ✓ 7/30/2024 (9s) 5

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

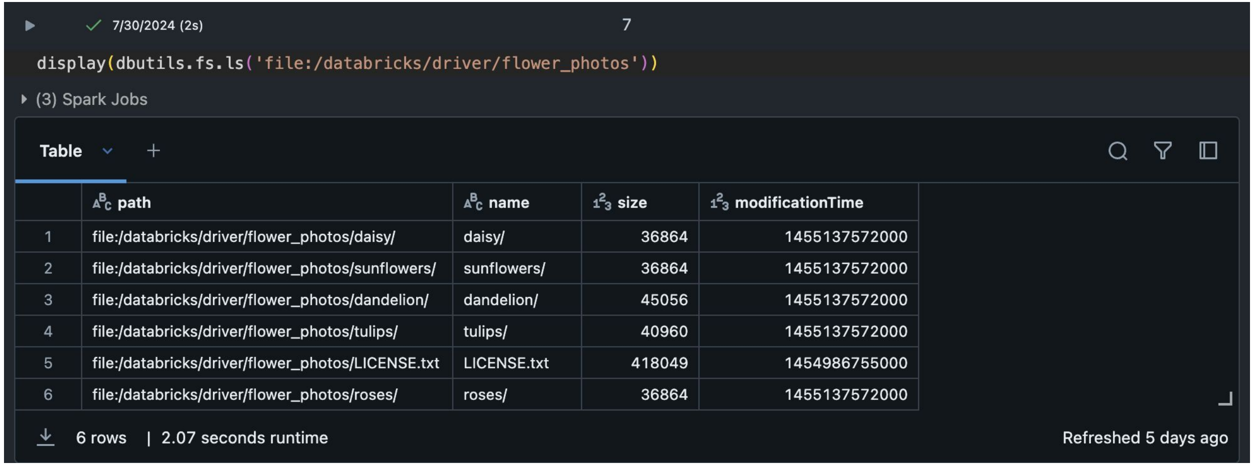
```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed

  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
 42  218M   42 92.0M    0     0   102M      0  0:00:02 --:--:--  0:00:02  102M
100  218M  100  218M    0     0   126M      0  0:00:01  0:00:01 --:--:--  126M
```

Transfer Learning

Quick Leverage: Utilize pre-trained models for various tasks with minimal effort.

Example: Apply a pre-trained model to classify images.



```
display(dbutils.fs.ls('file:/databricks/driver/flower_photos'))
```

	path	name	size	modificationTime
1	file:/databricks/driver/flower_photos/daisy/	daisy/	36864	1455137572000
2	file:/databricks/driver/flower_photos/sunflowers/	sunflowers/	36864	1455137572000
3	file:/databricks/driver/flower_photos/dandelion/	dandelion/	45056	1455137572000
4	file:/databricks/driver/flower_photos/tulips/	tulips/	40960	1455137572000
5	file:/databricks/driver/flower_photos/LICENSE.txt	LICENSE.txt	418049	1454986755000
6	file:/databricks/driver/flower_photos/roses/	roses/	36864	1455137572000

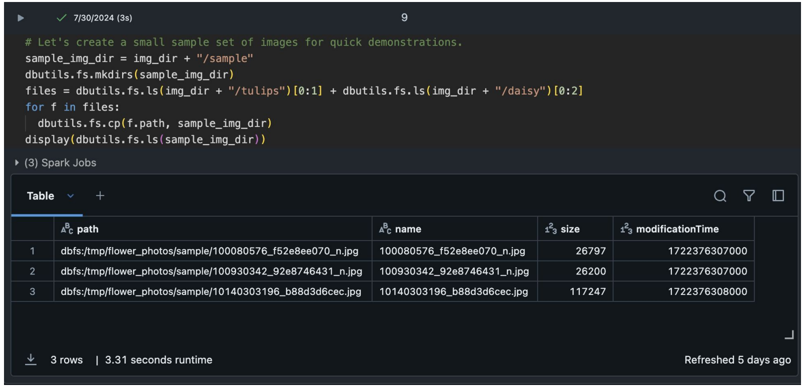
6 rows | 2.07 seconds runtime

Refreshed 5 days ago

Applying Models at Scale

Scalability: Apply your own or popular models to image data at scale.

Use Cases: Image classification, feature extraction, etc.

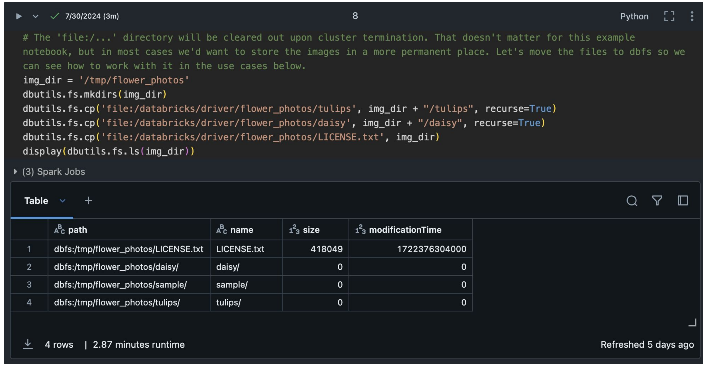


```python
# Let's create a small sample set of images for quick demonstrations.
sample_img_dir = img_dir + "/sample"
dbutils.fs.mkdirs(sample_img_dir)
files = dbutils.fs.ls(img_dir + "/tulips")[0:1] + dbutils.fs.ls(img_dir + "/daisy")[0:2]
for f in files:
  dbutils.fs.cp(f.path, sample_img_dir)
display(dbutils.fs.ls(sample_img_dir))
```

	path	name	size	modificationTime
1	dbfs:/tmp/flower_photos/sample/100080576_f52e8ee070_n.jpg	100080576_f52e8ee070_n.jpg	26797	1722376307000
2	dbfs:/tmp/flower_photos/sample/100930342_92e8746431_n.jpg	100930342_92e8746431_n.jpg	26200	1722376307000
3	dbfs:/tmp/flower_photos/sample/10140303196_b88d3d6cec.jpg	10140303196_b88d3d6cec.jpg	117247	1722376308000

3 rows | 3.31 seconds runtime

Refreshed 5 days ago



```python
# The 'file:/...' directory will be cleared out upon cluster termination. That doesn't matter for this example
notebook, but in most cases we'd want to store the images in a more permanent place. Let's move the files to dbfs so we
can see how to work with it in the use cases below.
img_dir = '/tmp/flower_photos'
dbutils.fs.mkdirs(img_dir)
dbutils.fs.cp('file:/databricks/driver/flower_photos/tulips', img_dir + "/tulips", recurse=True)
dbutils.fs.cp('file:/databricks/driver/flower_photos/daisy', img_dir + "/daisy", recurse=True)
dbutils.fs.cp('file:/databricks/driver/flower_photos/LICENSE.txt', img_dir)
display(dbutils.fs.ls(img_dir))
```

	path	name	size	modificationTime
1	dbfs:/tmp/flower_photos/LICENSE.txt	LICENSE.txt	418049	1722376304000
2	dbfs:/tmp/flower_photos/daisy/	daisy/	0	0
3	dbfs:/tmp/flower_photos/sample/	sample/	0	0
4	dbfs:/tmp/flower_photos/tulips/	tulips/	0	0

4 rows | 2.87 minutes runtime

Refreshed 5 days ago

Conclusion

- Deep Learning Pipelines: Facilitates scalable deep learning with Spark.
- Future Developments: Stay updated with the latest releases and features on GitHub.

References

https://github.com/Prachi1615/MachineLearning

https://community.cloud.databricks.com/?o=4454248012642049#notebook/914229071432723/command/914229071432726