

```
!git clone https://github.com/bellevue-university/dsc650.git
```

```
Cloning into 'dsc650'...
remote: Enumerating objects: 120326, done.
remote: Counting objects: 100% (128/128), done.
remote: Compressing objects: 100% (50/50), done.
remote: Total 120326 (delta 57), reused 98 (delta 46), pack-reused 120198
Receiving objects: 100% (120326/120326), 360.60 MiB | 13.18 MiB/s, done.
Resolving deltas: 100% (7340/7340), done.
Updating files: 100% (114699/114699), done.
```

```
from datetime import datetime
```

```
with open('/content/dsc650/dsc650/assignments/assignment01/logs/keras-mnist.log', 'a') as f:
```

```
    from tensorflow import keras
    from keras.datasets import mnist
    from keras.models import Sequential
    from keras.layers import Dense, Dropout
    from keras.optimizers import RMSprop
```

```
    batch_size = 128
    num_classes = 10
    epochs = 20
```

```
    # the data, split between train and test sets
    (x_train, y_train), (x_test, y_test) = mnist.load_data()
```

```
    x_train = x_train.reshape(60000, 784)
    x_test = x_test.reshape(10000, 784)
    x_train = x_train.astype('float32')
    x_test = x_test.astype('float32')
    x_train /= 255
    x_test /= 255
    print(x_train.shape[0], 'train samples')
    print(x_test.shape[0], 'test samples')
```

```
    # convert class vectors to binary class matrices
    y_train = keras.utils.to_categorical(y_train, num_classes)
    y_test = keras.utils.to_categorical(y_test, num_classes)
```

```
    model = Sequential()
    model.add(Dense(512, activation='relu', input_shape=(784,)))
    model.add(Dropout(0.2))
    model.add(Dense(512, activation='relu'))
    model.add(Dropout(0.2))
    model.add(Dense(num_classes, activation='softmax'))
```

```
    model.summary()
```

```
    model.compile(loss='categorical_crossentropy',
                  optimizer=RMSprop(),
                  metrics=['accuracy'])
```

```
    history = model.fit(x_train, y_train,
                        batch_size=batch_size,
                        epochs=epochs,
                        verbose=1,
                        validation_data=(x_test, y_test))
```

```
    score = model.evaluate(x_test, y_test, verbose=0)
```

```
f.write(datetime.now().strftime('%Y-%m-%d %H:%M:%S') + "\n" )
f.write('Test loss :' + str(score[0]) + "\n")
f.write('Test Accuracy :' + str(score[1]) + "\n")
```

| Layer (type) | Output Shape | Param # |
|----------------------|--------------|---------|
| dense_24 (Dense) | (None, 512) | 401920 |
| dropout_16 (Dropout) | (None, 512) | 0 |
| dense_25 (Dense) | (None, 512) | 262656 |
| dropout_17 (Dropout) | (None, 512) | 0 |
| dense_26 (Dense) | (None, 10) | 5130 |

```
=====
Total params: 669,706
Trainable params: 669,706
Non-trainable params: 0
```

```
Epoch 1/20
469/469 [=====] - 4s 6ms/step - loss: 0.2500 - accuracy: 0.9226 - val_loss:
Epoch 2/20
469/469 [=====] - 2s 4ms/step - loss: 0.1004 - accuracy: 0.9693 - val_loss:
Epoch 3/20
469/469 [=====] - 2s 4ms/step - loss: 0.0731 - accuracy: 0.9778 - val_loss:
Epoch 4/20
469/469 [=====] - 2s 4ms/step - loss: 0.0583 - accuracy: 0.9824 - val_loss:
Epoch 5/20
469/469 [=====] - 2s 4ms/step - loss: 0.0521 - accuracy: 0.9842 - val_loss:
Epoch 6/20
469/469 [=====] - 2s 4ms/step - loss: 0.0429 - accuracy: 0.9871 - val_loss:
Epoch 7/20
469/469 [=====] - 3s 6ms/step - loss: 0.0375 - accuracy: 0.9888 - val_loss:
Epoch 8/20
469/469 [=====] - 2s 5ms/step - loss: 0.0321 - accuracy: 0.9903 - val_loss:
Epoch 9/20
469/469 [=====] - 2s 4ms/step - loss: 0.0303 - accuracy: 0.9912 - val_loss:
Epoch 10/20
469/469 [=====] - 2s 5ms/step - loss: 0.0275 - accuracy: 0.9916 - val_loss:
Epoch 11/20
469/469 [=====] - 2s 4ms/step - loss: 0.0261 - accuracy: 0.9922 - val_loss:
Epoch 12/20
469/469 [=====] - 2s 4ms/step - loss: 0.0242 - accuracy: 0.9931 - val_loss:
Epoch 13/20
469/469 [=====] - 3s 6ms/step - loss: 0.0229 - accuracy: 0.9935 - val_loss:
Epoch 14/20
469/469 [=====] - 2s 5ms/step - loss: 0.0222 - accuracy: 0.9939 - val_loss:
Epoch 15/20
469/469 [=====] - 2s 4ms/step - loss: 0.0219 - accuracy: 0.9943 - val_loss:
Epoch 16/20
469/469 [=====] - 2s 4ms/step - loss: 0.0201 - accuracy: 0.9945 - val_loss:
Epoch 17/20
469/469 [=====] - 2s 4ms/step - loss: 0.0182 - accuracy: 0.9950 - val_loss:
Epoch 18/20
469/469 [=====] - 2s 4ms/step - loss: 0.0194 - accuracy: 0.9949 - val_loss:
Epoch 19/20
469/469 [=====] - 3s 6ms/step - loss: 0.0183 - accuracy: 0.9946 - val_loss:
Epoch 20/20
469/469 [=====] - 2s 5ms/step - loss: 0.0182 - accuracy: 0.9954 - val_loss:
```

```
!pip install pyspark
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting pyspark
  Downloading pyspark-3.3.2.tar.gz (281.4 MB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 281.4/281.4 MB 5.5 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting py4j==0.10.9.5
  Downloading py4j-0.10.9.5-py2.py3-none-any.whl (199 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 199.7/199.7 KB 13.8 MB/s eta 0:00:00
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.3.2-py2.py3-none-any.whl size=281824028 sha256=0852b66a
  Stored in directory: /root/.cache/pip/wheels/6c/e3/9b/0525ce8a69478916513509d43693511463c6468db0de237
Successfully built pyspark
Installing collected packages: py4j, pyspark
  Attempting uninstall: py4j
    Found existing installation: py4j 0.10.9.7
    Uninstalling py4j-0.10.9.7:
      Successfully uninstalled py4j-0.10.9.7
Successfully installed py4j-0.10.9.5 pyspark-3.3.2
```

```
from datetime import datetime
import sys
from random import random
from operator import add

from pyspark.sql import SparkSession

if __name__ == "__main__":
    """
    Usage: pi [partitions]
    """
    spark = SparkSession\
        .builder\
        .appName("PythonPi")\
        .getOrCreate()

    partitions = 2

    #partitions = int(sys.argv[1]) if len(sys.argv) > 1 else 2
    n = 100000 * partitions

    def f(_):
        x = random() * 2 - 1
        y = random() * 2 - 1
        return 1 if x ** 2 + y ** 2 <= 1 else 0

    count = spark.sparkContext.parallelize(range(1, n + 1), partitions).map(f).reduce(add)
    print("Pi is roughly %f" % (4.0 * count / n))

    spark.stop()
    with open('/content/dsc650/dsc650/assignments/assignment01/logs/spark-pi.log', 'a') as f:
        f.write (datetime.now().strftime('%Y-%m-%d %H:%M:%S') + "\n" )
        f.write ('Pi is : ' + str((4.0 * count / n)) + "\n")
```

Pi is roughly 3.135080

```
!apt-get install tree
```

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  tree
0 upgraded, 1 newly installed, 0 to remove and 23 not upgraded.
Need to get 43.0 kB of archives.
After this operation, 115 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal/universe amd64 tree amd64 1.8.0-1 [43.0 kB]
Fetched 43.0 kB in 0s (661 kB/s)
Selecting previously unselected package tree.
(Reading database ... 128276 files and directories currently installed.)
Preparing to unpack .../tree_1.8.0-1_amd64.deb ...
Unpacking tree (1.8.0-1) ...
Setting up tree (1.8.0-1) ...
Processing triggers for man-db (2.9.1-1) ...
```

✓ 4s completed at 9:00 PM

● ✕