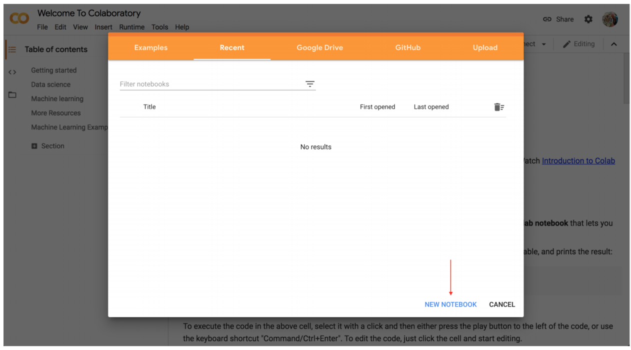
Assignment Zero: MNIST Classification

In this assignment, you will implement Python code to perform classification on the MNIST dataset using the deep learning framework Keras. There are two parts to this assignment. In **Part I,** your main goal will be to simply get the code running. Hopefully this will build your confidence to move on to the next part. In **Part II**, you will "tease" apart the code from **Part I**, section by section, with the objective being to better understand how the code works.

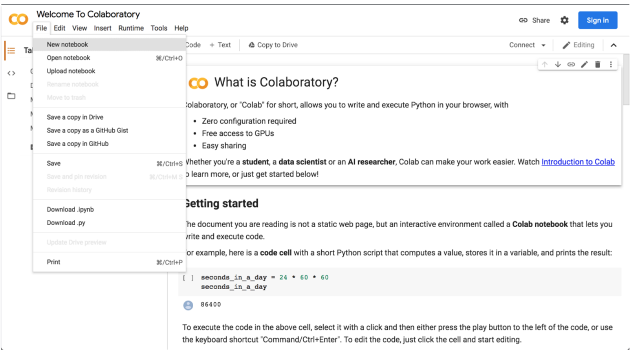
Part I: Create and Run a Python Notebook

1. Navigate to `https://colab.research.google.com/` and select "New Notebook" as in the following figures.



To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut "Command/Ctrl+Enter". To edit the code, just click the cell and start editing.

Figure 1: Signed-in to Google



What is Colaboratory?

olaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Vhether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch Introduction to Colab o learn more, or just get started below!

etting started

he document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you rite and execute code.

or example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

```
[ ] seconds_in_a_day = 24 * 60 * 60
    seconds_in_a_day
```

```
86400
```

To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut "Command/Ctrl+Enter". To edit the code, just click the cell and start editing.

Figure 2: Not Signed-in to Google

2. Navigate to `https://keras.io/examples/mnist_cnn/`. Copy and paste the code into the first "cell" of the Colab Notebook.

3. On the first line, add in the statement

   import numpy as np

4. Immediately following the line containing **epochs = 12**, add the line

   np.random.seed(0)

   Keras gets its source of randomness from numpy's random number generator. To ensure reproducible results, this starts random computation off at the same place every run.

5. Immediately **following** the line containing model.add(Dense(num_classes, activation=’softmax’)) and **before** the line containing model.compile..., add the line

   print(model.summary())

   This will print a summary of your model’s architecture and parameters.

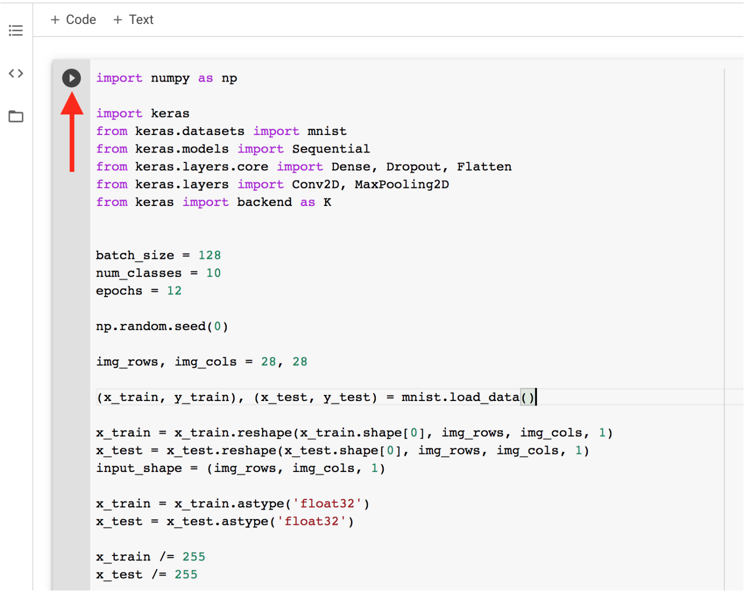
6. Change the line that has model.fit(... to

   history = model.fit(...

7. Delete the last three lines of code, i.e. score = ..., print(’Test..., and print(’Test... and replace them with

   print([(key, round(value[0], 4)) for key, value in history.history.items()])

8. To run the code, press the Play ▶ button as shown in **Figure 3**.



```
import numpy as np

import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers.core import Dense, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D
from keras import backend as K


batch_size = 128
num_classes = 10
epochs = 12

np.random.seed(0)

img_rows, img_cols = 28, 28

(x_train, y_train), (x_test, y_test) = mnist.load_data()

x_train = x_train.reshape(x_train.shape[0], img_rows, img_cols, 1)
x_test = x_test.reshape(x_test.shape[0], img_rows, img_cols, 1)
input_shape = (img_rows, img_cols, 1)

x_train = x_train.astype('float32')
x_test = x_test.astype('float32')

x_train /= 255
x_test /= 255
```

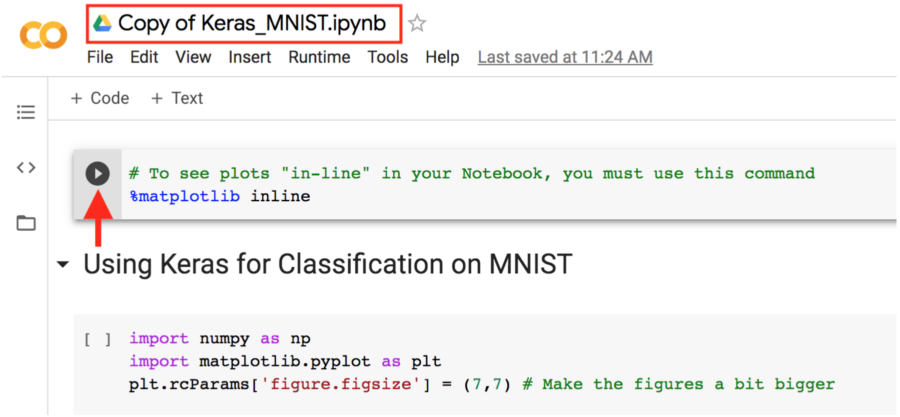
Figure 3. Run Colab Notebook Code

9. If your code runs, congratulations!! Your output should look like the Output.txt file that is with this document. If your code doesn’t work or your output doesn’t match, try re-tracing your steps and making sure you did everything correctly . If that doesn’t work, try Googling the error message(s) you are receiving.

Part II: Understanding Part I

In **Part I** you successfully ran Python code which uses a Convolutional Neural Network to classify the images in the MNIST dataset. For many of you, that probably doesn't make a lot of sense. So our goal in this part of the assignment will be to take a closer look at what's happening in the code from **Part I**. I have added some additional lines to the code you will be examining. These serve only as aides to better illustrate the concepts therein.

To get started, you will need to be signed in to your Google account. Once signed in, navigate to the Colab Notebook at `https://bit.ly/2Yqiqjn` and select the "Save a copy in Drive" option under the "File" Menu. Once you have done that, the file name should now be "Copy of Keras_MNIST.ipynb" and you can run the first cell, as shown in **Figure 4** below.


Copy of Keras_MNIST.ipynb
File Edit View Insert Runtime Tools Help Last saved at 11:24 AM
+ Code + Text
[ ]

```
# To see plots "in-line" in your Notebook, you must use this command
%matplotlib inline
```

Using Keras for Classification on MNIST

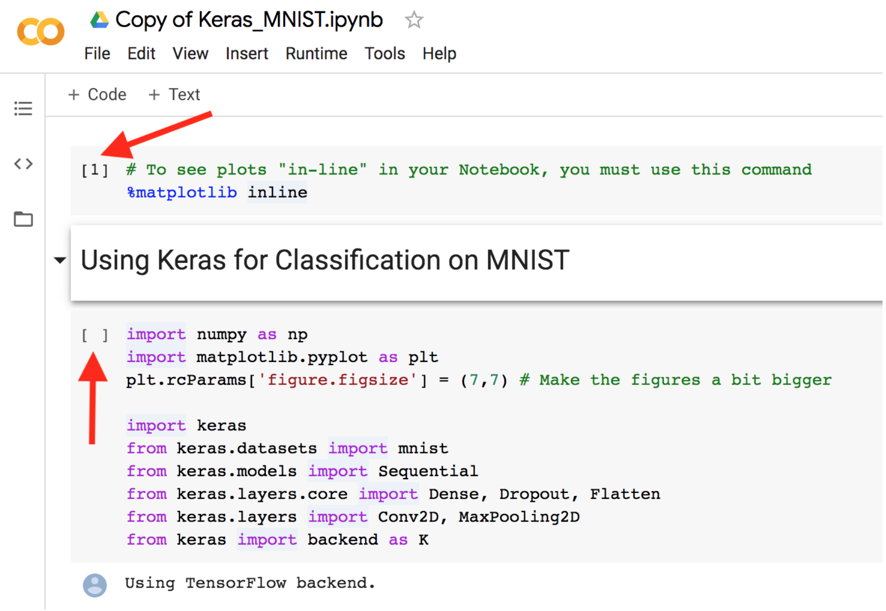
```
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = (7,7) # Make the figures a bit bigger
```

Figure 4: Save a Copy and Run First Cell

Once you've run this cell, the empty square brackets at the left of the cell should now read [1] (see **Figure 5**), as opposed to [ ] before we ran the cell (e.g. look at cell 2 in **Figure 5**, it's [ ] because it hasn't been run yet).

From here, you will simply go cell by cell, running each and observing the output. I have included some notes and code block titles to try and help with the break down. I would suggest reading each line in an individual cell and try to make sense of what it is doing. The shortcut to run a cell is Shift + Enter, which should make it a little smoother to work your way through.

Lastly, don't forget, you made a copy of this Notebook, so feel free to make as **many changes** as you want. If it helps, you can add print statements after every single line! Good luck and have fun!


Copy of Keras_MNIST.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text

```
[1] # To see plots "in-line" in your Notebook, you must use this command
    %matplotlib inline
```

Using Keras for Classification on MNIST

```
[ ] import numpy as np
    import matplotlib.pyplot as plt
    plt.rcParams['figure.figsize'] = (7,7) # Make the figures a bit bigger

    import keras
    from keras.datasets import mnist
    from keras.models import Sequential
    from keras.layers.core import Dense, Dropout, Flatten
    from keras.layers import Conv2D, MaxPooling2D
    from keras import backend as K

Using TensorFlow backend.
```

Figure 5: After Running Cell 1