

Predicting Financial Time Series using Deep Learning

Module1. Google Colaboratory

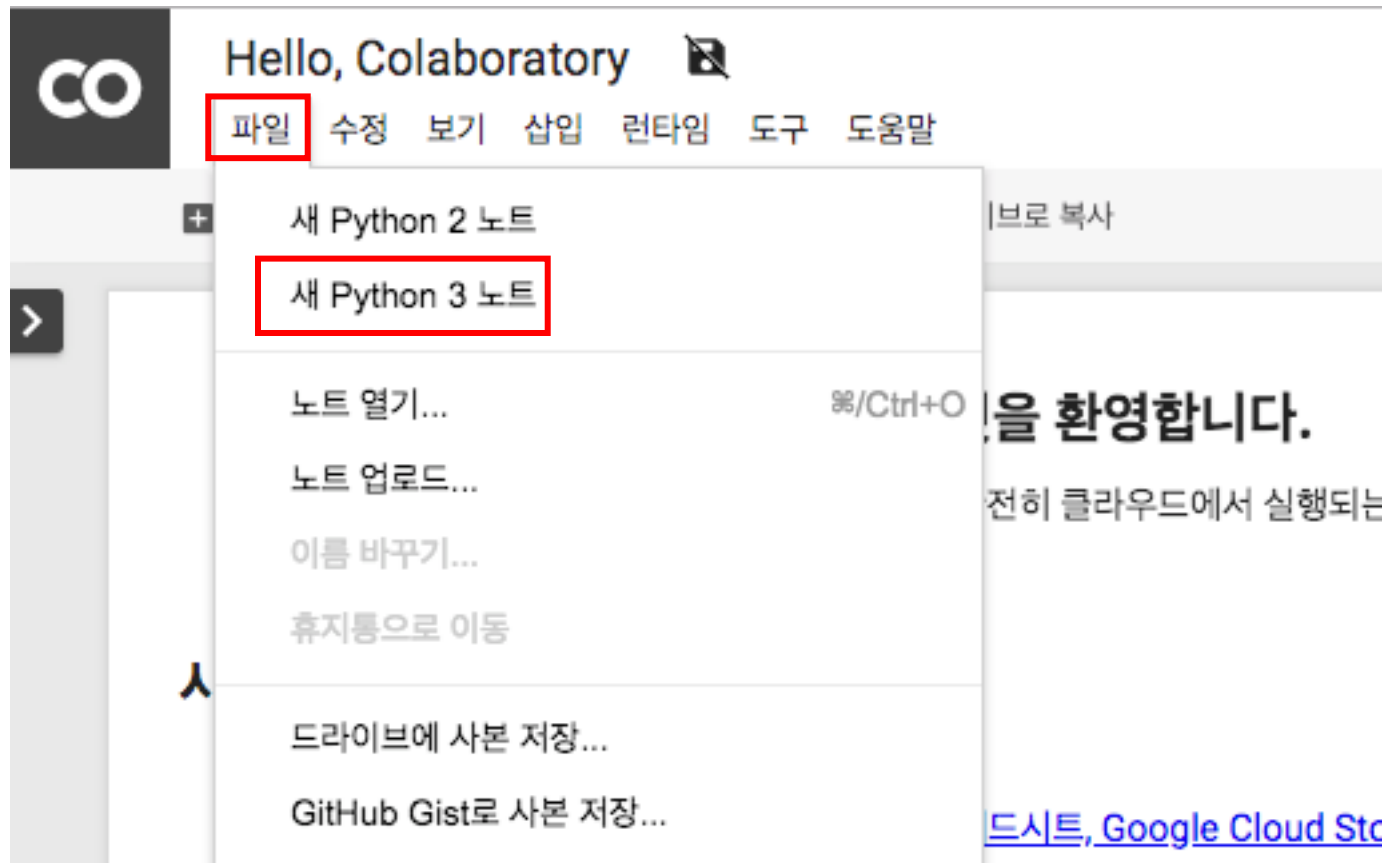
Jongho Kim

NICE Pricing & Information Inc.

Fall, 2018

“Hello World” on Colab

Access to URL: <https://colab.research.google.com/notebooks/welcome.ipynb#recent=true>



“Hello World” on Colab



A screenshot of a Google Colab code cell. The top bar is grey with a play button icon on the left. The code area shows the text `print("Hello World")` in a monospace font, with the opening quote in blue and the rest in red. Below the code area, the output is displayed as `Hello World` in a monospace font, preceded by a copy icon.

- `print("Hello World")`
- Click button or type “CTRL + ENTER”

“File Upload” on Colab

Access to URL: <https://colab.research.google.com/notebooks/io.ipynb>

```
▶ from google.colab import files  
uploaded = files.upload()  
  
for fn in uploaded.keys():  
    print('User uploaded file "{name}" with length {length} bytes'.format(  
        name=fn, length=len(uploaded[fn])))
```

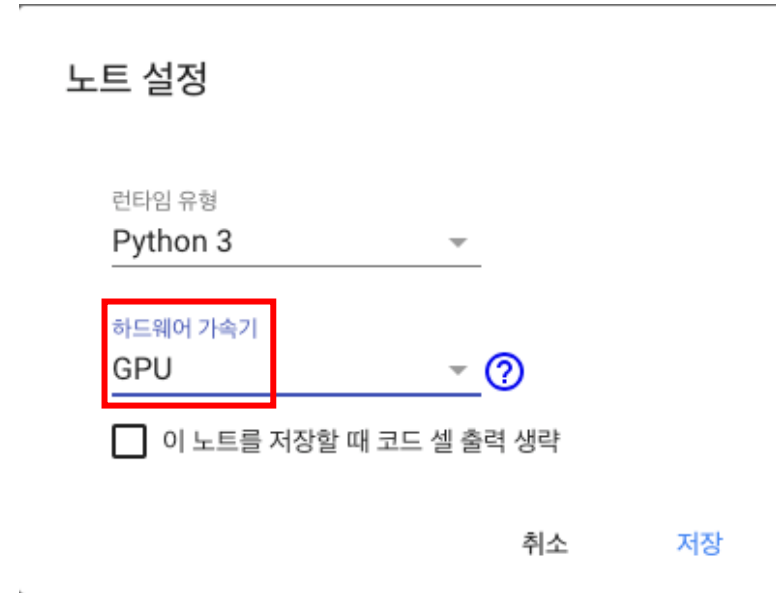


파일 선택

선택된 파일 없음

Cancel upload

“Set GPU” on Colab



Run Keras on Google Colab

Code URL: https://github.com/jonghkim/financial-time-series-prediction/blob/master/Module1/Hands-on-Labs/Lab1_Keras.ipynb

This is keras tutorial code from pythonprogramming

- <https://pythonprogramming.net/introduction-deep-learning-python-tensorflow-keras/>

```
In [ ]: import tensorflow as tf # deep learning library. Tensors are just multi-dimensional arrays
import matplotlib.pyplot as plt
%matplotlib inline
```

keras MNIST data load

```
In [ ]: mnist = tf.keras.datasets.mnist # mnist is a dataset of 28x28 images of handwritten digits and their labels
(x_train, y_train), (x_test, y_test) = mnist.load_data() # unpacks images to x_train/x_test and labels to y_train/y_test
```

```
In [ ]: x_train = tf.keras.utils.normalize(x_train, axis=1) # scales data between 0 and 1
x_test = tf.keras.utils.normalize(x_test, axis=1) # scales data between 0 and 1
```

Fully Connected Neural Net Model

```
In [ ]: model = tf.keras.models.Sequential() # a basic feed-forward model
model.add(tf.keras.layers.Flatten()) # takes our 28x28 and makes it 1x784
model.add(tf.keras.layers.Dense(128, activation=tf.nn.relu)) # a simple fully-connected layer, 128 units, relu activation
model.add(tf.keras.layers.Dense(128, activation=tf.nn.relu)) # a simple fully-connected layer, 128 units, relu activation
```

Google Colaboratoy Useful Shortcuts

Actions	Colab	Jupyter
show keyboard shortcuts	Ctrl/Cmd M H	H
Insert code cell above	Ctrl/Cmd M A	A
Insert code cell below	Ctrl/Cmd M B	B
Delete cell/selection	Ctrl/Cmd M D	DD
Interrupt execution	Ctrl/Cmd M I	II
Convert to code cell	Ctrl/Cmd M Y	Y
Convert to text cell	Ctrl/Cmd M M	M
Split at cursor	Ctrl/Cmd M -	Ctrl Shift -

Thank you ☺

Contact Info: quantic.jh@gmail.com