

# Deep Learning Based Recommendation (Part B)

Topic 9B

# Deep learning

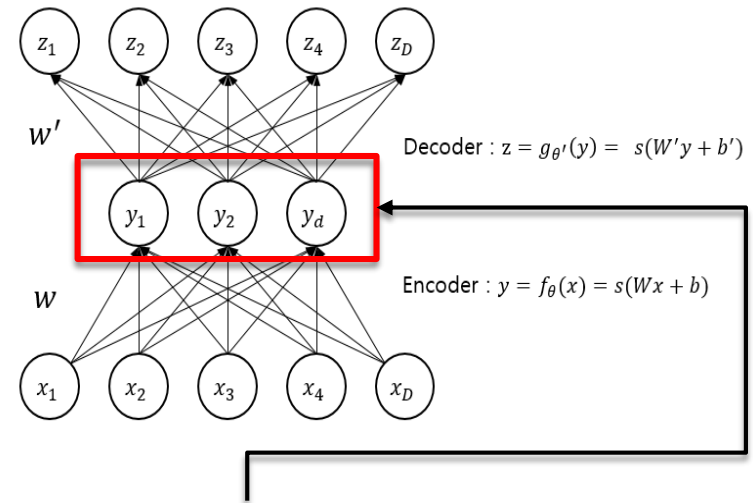
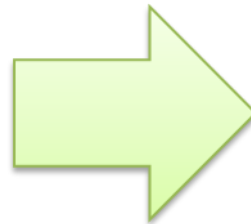
---

- Autoencoder for recommender system
  - ▣ Autoencoder example in Keras
  - ▣ Alternative DL Framework
  - ▣ Google Colab
  - ▣ Running an example in Google Colab

# Autoencoder for recommender system

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	0	3	0	3	0
User 2	4	0	0	2	0
User 3	0	0	3	0	0
User 4	3	0	4	0	3
User 5	4	3	0	4	0

A matrix of user/item ratings



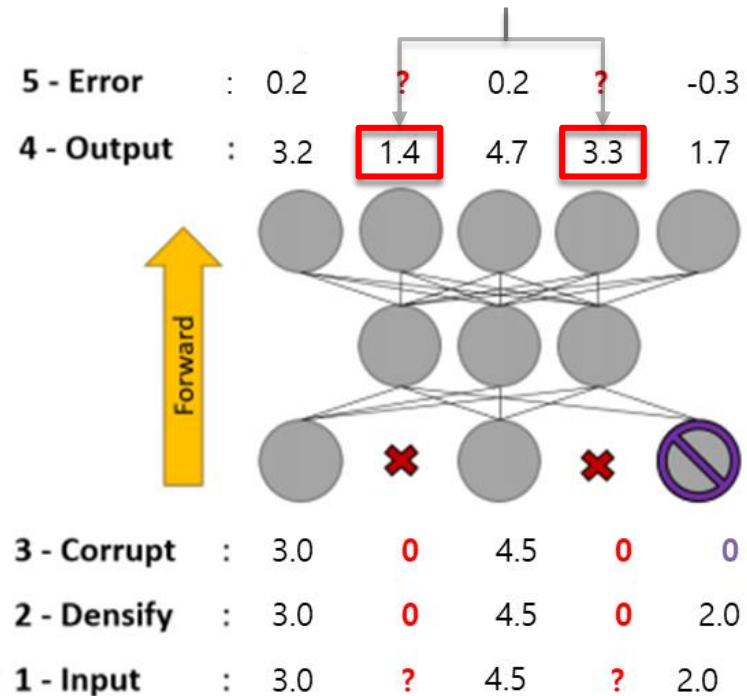
In global context, the latent representation in encoding layer can be expected to embrace users' preferences for items, or vice versa.

# Autoencoder for recommender system

	$i_1$	$i_2$	$i_3$	$i_4$	$i_5$
$u_1$		4.5	2.0		
$u_2$	4.0		3.5		
$u_3$	3.0		4.5		2.0
$u_4$		3.5	4.0		1.0

Rating Matrix( $R$ )

These can be considered as prediction



$$\operatorname{argmin}_{\{W_l\}\{b_l\}} = \underbrace{\|\tilde{X} - \hat{X}\|_F^2}_{\text{Training error on observed ratings}} + \lambda \underbrace{\sum_l^L (\|W_l\|_F^2 + \|b_l\|_F^2)}_{\text{Regularization to avoid overfitting}}$$

Training error on  
observed ratings

Regularization to  
avoid overfitting

# Autoencoder example in Keras

## Autoencoder for MNIST (Hand written images)

```
1  ## Build Autoencoder model
2  # this is our input placeholder
3  input_img = Input(shape=(784,))
4
5  # "encoded" is the encoded representation of the input
6  encoded = Dense(32,
7                  activation='relu')(input_img)
8
9  # "decoded" is the lossy reconstruction of the input
10 decoded = Dense(784,
11                activation='sigmoid')(encoded)
12
13 # this model maps an input to its reconstruction
14 autoencoder = Model(input_img, decoded)
15
16 ## Show model information
17 autoencoder.summary()
18
19 o_RMSprop = optimizers.RMSprop(lr=0.001,
20                                rho=0.9,
21                                epsilon=None,
22                                decay=0.0)
23
24 ## Set train methodology
25 autoencoder.compile(loss='binary_crossentropy',
26                    optimizer=o_RMSprop)
```

① Input dimension (flattened image dimension)

② The number of hidden layer's neuron

③ Activation function in encoder

④ Output dimension (flattened image dimension)

⑤ Activation function in decoder

⑥ Optimizer

⑦ Learning rate

⑧ Loss function (objective function)

# Autoencoder example in Keras

## Autoencoder for MNIST (Hand written images)

```
1  ## Train model with train set.
2  autoencoder.fit(x=x_train, y=x_train,
3                  batch_size=128, epochs=1000,
4                  shuffle=True,
5                  validation_data=(x_test, x_test))
```

- ① x: Input set (flattened image dimension)
- ② y: Output set (flattened image dimension)  
※ This is same with x (input set)
- ③ batch\_size: the number of data in input subset
- ④ epochs: the number of iterations
- ⑤ shuffle:  
shuffling of input set for model's robustness
- ⑥ validation\_data:  
Validation set for hyperparamter or  
early stopping, etc.

# Autoencoder example in Keras

- Techniques for boosting performance of Autoencoder
  - ▣ Change dimension of input, output, and hidden
  - ▣ Change loss function
  - ▣ Change optimizer function (RMSprop, Adagrad, Adam, and etc)
  - ▣ Add more layers in encoder and decoder (**Stacked autoencoder**)
  - ▣ Try different activation function (ReLU, Sigmoid, Tanh, Softmax, etc)
  - ▣ Add noise to input set (**Denoising autoencoder**)
  - ▣ Add regularization term for loss (**Sparse autoencoder**)

# Practice

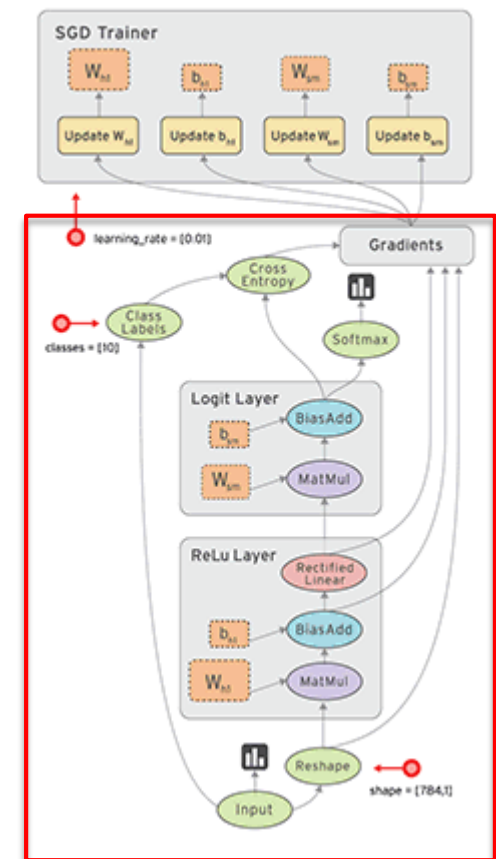
---

- Deep Learning Framework
  - Tensorflow
  - Keras
  - PyTorch
- Google Colaboratory (Colab)
- Running an example using Google Colab



# TensorFlow

- Open source platform for numerical computation by Google
  - ▣ Using data flow graphs
- Data flow graphs (DFG)
  - ▣ Each node in the **graph** represents the instance of a mathematical operation (multiply, add, divide, ...).
  - ▣ Each edge is a multi-dimensional **data** set (tensors) on which the operations are performed.




Data flow graphs

# PyTorch

- Open source platform for numerical computation by Facebook
  - ▣ Tensor computation with strong GPU acceleration.
  - ▣ Deep learning built on a tape-based autodiff system.
  - ▣ Can change DFG at runtime unlike TensorFlow
- Difference between PyTorch vs TensorFlow

For example, to calculate  $1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots$ ,



```
import torch


x = torch.Tensor([0.])
y = torch.Tensor([1.])

for iteration in range(50):
    x = x + y
    y = y / 2

print(x)
```

Construction  
Phase

Execution  
Phase



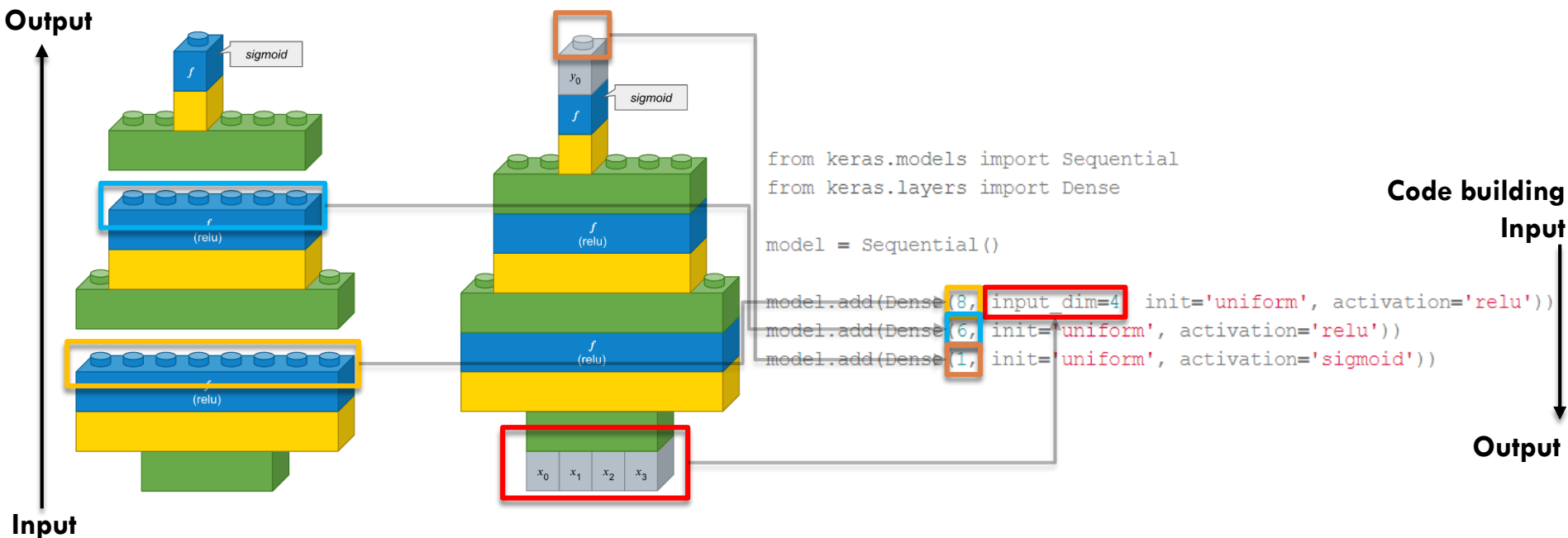
```
import tensorflow as tf

x = tf.Variable(0.)
y = tf.Variable(1.)
add_op = x.assign(x + y)
div_op = y.assign(y / 2)
init = tf.global_variables_initializer()

with tf.Session() as sess:
    init.run()
    for iteration in range(50):
        sess.run(add_op)
        sess.run(div_op)
    print(x.eval())
```

# Keras

- A high-level neural networks API
  - ▣ Running on top of TensorFlow, CNTK, Theano.
  - ▣ Allowing for easy and fast prototyping (through user friendliness, modularity, and extensibility).

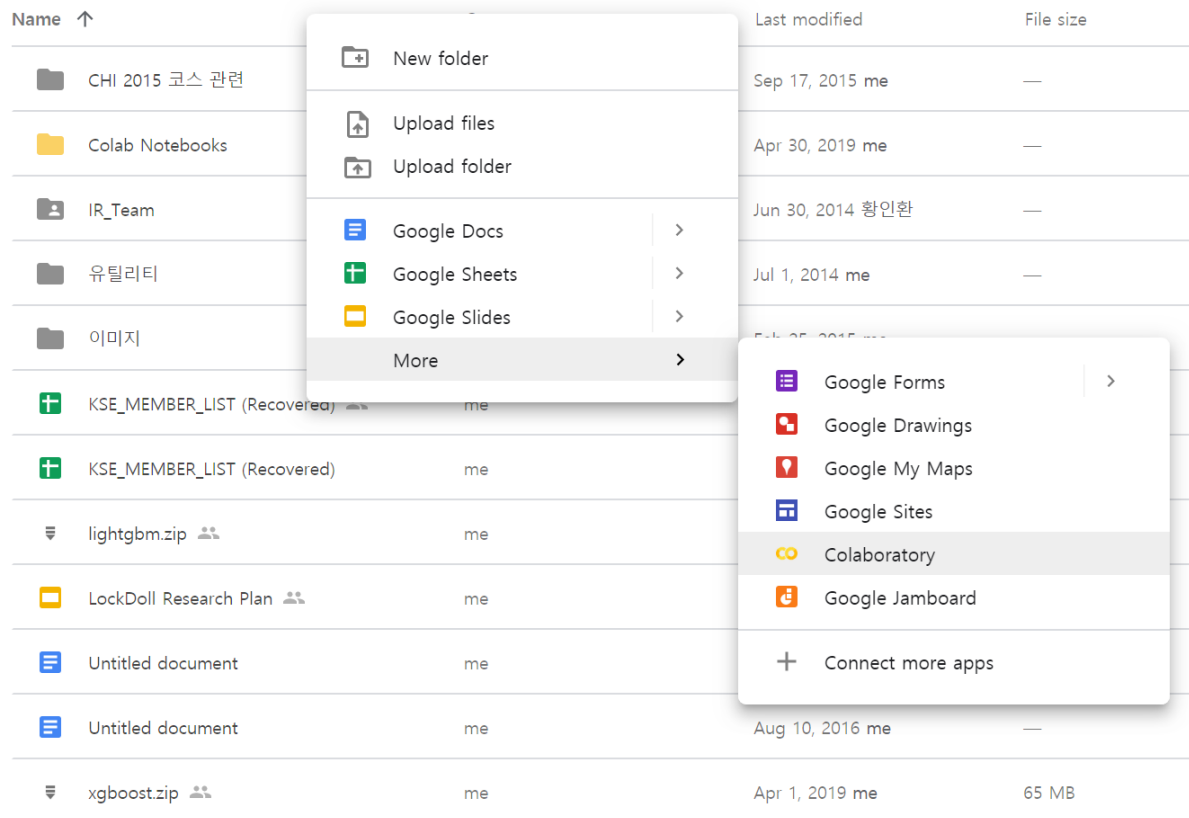


Conceptual model diagram with Lego vs Actual Implementation

# Google Colaboratory (Colab)

- Free service provided by Google for academy  
(<https://colab.research.google.com/notebooks/welcome.ipynb>)
  - ▣ Jupyter Notebook + Google Drive
    - It requires Gmail account and you can connect personal Google drive to save results and models.
    - It is also available Linux command to handle environment.
  - ▣ Computer Specifications
    - OS: Ubuntu 18.04
    - CPU: Xenon 2.3GHz
    - GPU: Tesla K80
    - Memory: 13G
  - ▣ Available for up to 12 hours, and after 12 hours initializing all environment settings and files.

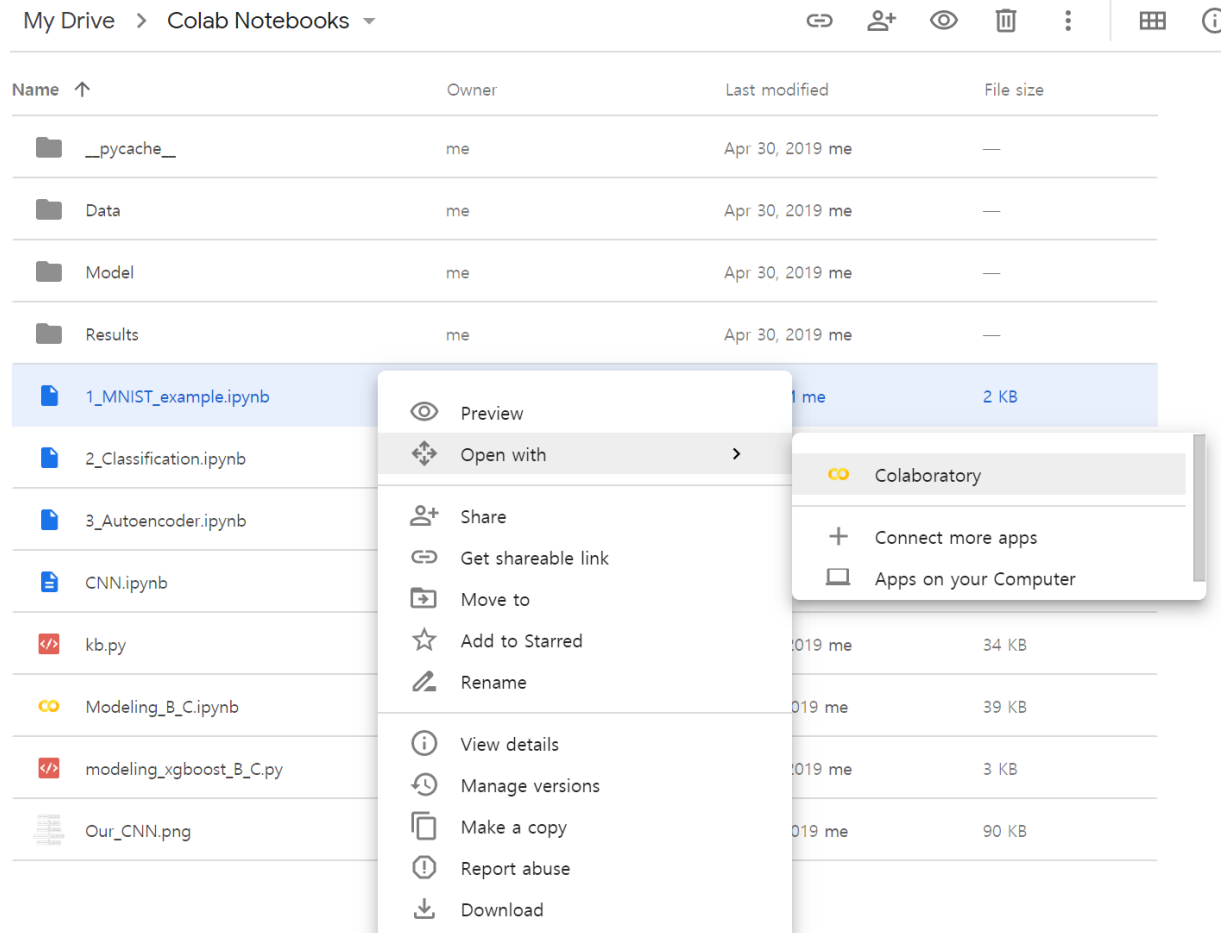
# Example – First starts



Getting Started with Google Colab:

<https://towardsdatascience.com/getting-started-with-google-colab-f2fff97f594c>

# Example – First starts



Or, upload provided ipynb file, and open with Colaboratory

# Example – Sync Google drive

Sync your Google Drive

```
from google.colab import auth
auth.authenticate_user()

from google.colab import drive
drive.mount('/content/gdrive')
```

... Go to the following link in your browser:

① [https://accounts.google.com/o/oauth2/auth?redirect\\_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3Aob&prompt=select\\_account&response\\_type=code&client\\_id=32555940559.apps](https://accounts.google.com/o/oauth2/auth?redirect_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3Aob&prompt=select_account&response_type=code&client_id=32555940559.apps)

Enter verification code:

④



② Google Cloud SDK이(가) 내  
Google 계정에 액세스하려고 합니  
다

fanaticalworld@gmail.com

이렇게 하면 Google Cloud SDK에서 다음 작업을 할 수 있습니다.

Google 드라이브 파일 보기, 수정, 생성, 삭제 ⓘ

● Google 클라우드 플랫폼 서비스 전반에서 데 ⓘ  
이더 주히 미 과리

Google

로그인

이 코드를 복사하여 애플리케이션으로 전환한 다음 붙여넣으세요.

4/TwE16hNPWG7-  
pJS\_hHkLwVWXRlyyoWdqVDn6IRIGAX6UHzyPGI-uow

③

Then, sync your google drive to save results and models

# Example – Sync Google drive

Mount Google drive's root and drive folder

```
[18] cd /content/gdrive/My Drive
```

```
↳ /content/gdrive/My Drive
```

```
[20] ls -ltr
```

```
↳ total 133480
drwx----- 2 root root    4096 Jun 30  2014 IR_Team/
drwx----- 2 root root    4096 Jul   1  2014 유틸리티/
drwx----- 2 root root    4096 Feb 25  2015 이미지/
drwx----- 2 root root    4096 Sep 17  2015 'CHI 2015 코스 관련 '/
-rw----- 1 root root 68912717 Mar 31 19:43 lightgbm.zip
-rw----- 1 root root 67745435 Mar 31 19:43 xgboost.zip
drwx----- 3 root root    4096 Apr 30 10:38 'Colab Notebooks'/
drwx----- 2 root root    4096 May 19 15:10 문서/
```

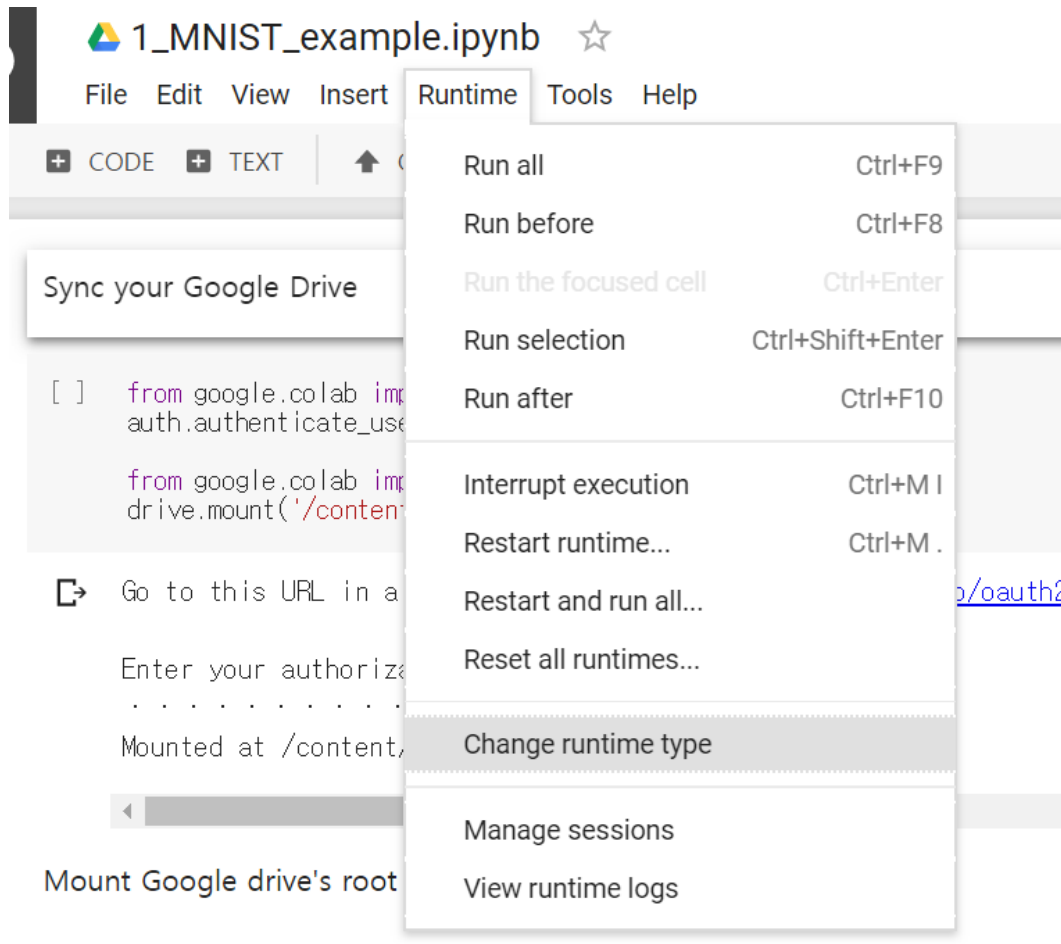
Select your own folder and Go into a folder

```
[21] cd Colab Notebooks
```

```
↳ /content/gdrive/My Drive/Colab Notebooks
```



# Example – GPU Acceleration



The screenshot shows a Google Colab notebook titled "1\_MNIST\_example.ipynb". The "Runtime" menu is open, displaying various options. The option "Change runtime type" is highlighted. The notebook content includes code for mounting Google Drive and a URL input field.

File Edit View Insert Runtime Tools Help

+ CODE + TEXT

Sync your Google Drive

```
[ ] from google.colab import auth
auth.authenticate_user()

from google.colab import drive
drive.mount('/content/drive')
```

Go to this URL in a web browser: [https://colab.research.google.com/oauth2/authorize?...](#)

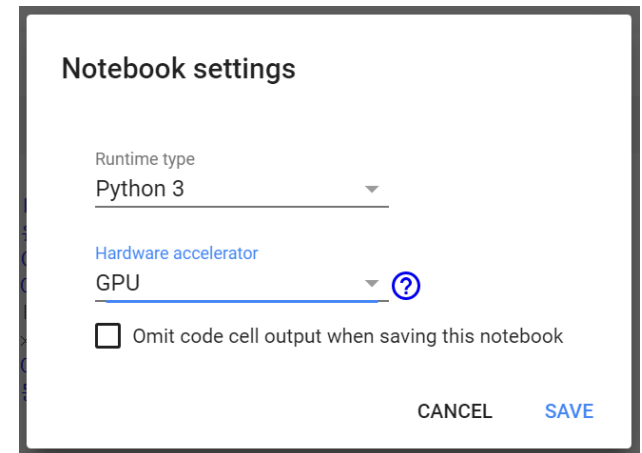
Enter your authorization code: .....

Mounted at /content/drive/

Mount Google drive's root to /content/drive/

Runtime menu options:

- Run all (Ctrl+F9)
- Run before (Ctrl+F8)
- Run the focused cell (Ctrl+Enter)
- Run selection (Ctrl+Shift+Enter)
- Run after (Ctrl+F10)
- Interrupt execution (Ctrl+M I)
- Restart runtime... (Ctrl+M .)
- Restart and run all...
- Reset all runtimes...
- Change runtime type**
- Manage sessions
- View runtime logs



The "Notebook settings" dialog box is shown. The "Runtime type" is set to "Python 3". The "Hardware accelerator" is set to "GPU". There is a checkbox for "Omit code cell output when saving this notebook" which is currently unchecked. The "SAVE" button is highlighted in blue.

Notebook settings

Runtime type  
Python 3

Hardware accelerator  
GPU

☐ Omit code cell output when saving this notebook

CANCEL SAVE