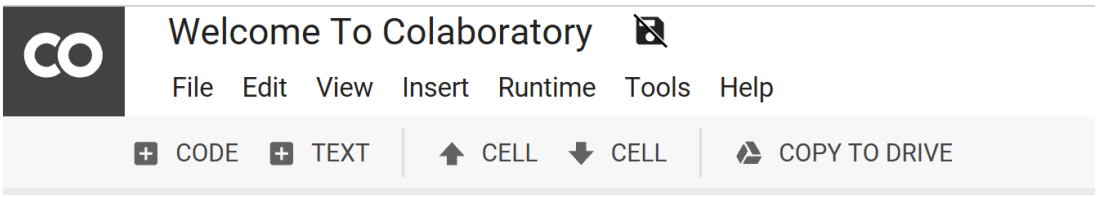
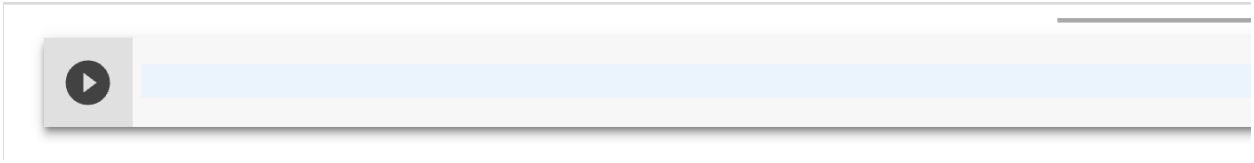
1. Go to https://colab.research.google.com/notebooks/welcome.ipynb
2. Log in with any of your google account.
3. On the left top corner, click File, and open a new python2 notebook.



4. After creating a new notebook, copy the baseline code here and run



The output should be similar to the screenshot below:

```
Train on 60000 samples, validate on 10000 samples
Epoch 1/12
60000/60000 [==============================] - 21s 354us/step - loss: 0.3002 - acc: 0.9049 - val_loss: 0.0902 - val_acc: 0.9711
Epoch 2/12
60000/60000 [==============================] - 20s 339us/step - loss: 0.0773 - acc: 0.9757 - val_loss: 0.0531 - val_acc: 0.9837
Epoch 3/12
60000/60000 [==============================] - 20s 341us/step - loss: 0.0538 - acc: 0.9829 - val_loss: 0.0519 - val_acc: 0.9834
Epoch 4/12
60000/60000 [==============================] - 21s 343us/step - loss: 0.0430 - acc: 0.9862 - val_loss: 0.0438 - val_acc: 0.9862
Epoch 5/12
60000/60000 [==============================] - 20s 341us/step - loss: 0.0347 - acc: 0.9891 - val_loss: 0.0440 - val_acc: 0.9850
Epoch 6/12
60000/60000 [==============================] - 20s 342us/step - loss: 0.0285 - acc: 0.9911 - val_loss: 0.0373 - val_acc: 0.9882
Epoch 7/12
60000/60000 [==============================] - 20s 341us/step - loss: 0.0240 - acc: 0.9922 - val_loss: 0.0345 - val_acc: 0.9894
Epoch 8/12
60000/60000 [==============================] - 20s 340us/step - loss: 0.0205 - acc: 0.9933 - val_loss: 0.0392 - val_acc: 0.9883
Epoch 9/12
60000/60000 [==============================] - 20s 341us/step - loss: 0.0182 - acc: 0.9942 - val_loss: 0.0328 - val_acc: 0.9896
Epoch 10/12
60000/60000 [==============================] - 21s 343us/step - loss: 0.0150 - acc: 0.9952 - val_loss: 0.0427 - val_acc: 0.9879
```

After verifying the baseline, you can change

1. Batchsize,

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

batch_size = 128
num_classes = 10
epochs = 12

# input image dimensions
img_rows, img_cols = 28, 28
```

2. Number of feature maps and kernel size

```
model = Sequential()
model.add(Conv2D(6, kernel_size=(3, 3),
                 activation='relu',
                 input_shape=input_shape))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(16, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(120, activation='relu'))
model.add(Dense(84, activation='relu'))

model.add(Dense(num_classes, activation='softmax'))
```

3. Learning rate

```
model.add(Dense(num_classes, activation='softmax'))

model.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.Adadelta(lr=1.0, rho=0.95, epsilon=None, decay=0.0),
              metrics=['accuracy'])

model.fit(x_train, y_train,
```

4. different optimizer. For details about the optimizer, you can check it here. https://keras.io/optimizers/

```
model.add(Dense(num_classes, activation='softmax'))

model.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.Adadelta(lr=1.0, rho=0.95, epsilon=None, decay=0.0),
              metrics=['accuracy'])

model.fit(x_train, y_train,
```