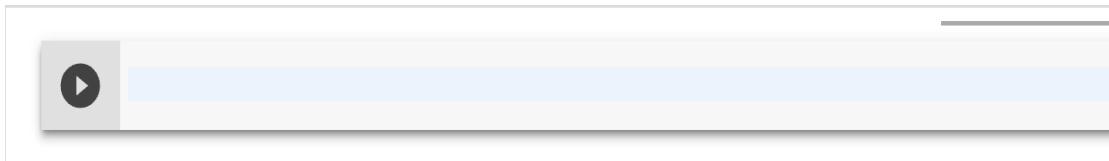


1. Go to <https://colab.research.google.com/notebooks/welcome.ipynb>
2. Log in with any of your google account.
3. On the left top corner, click File, and open a new python2 notebook.



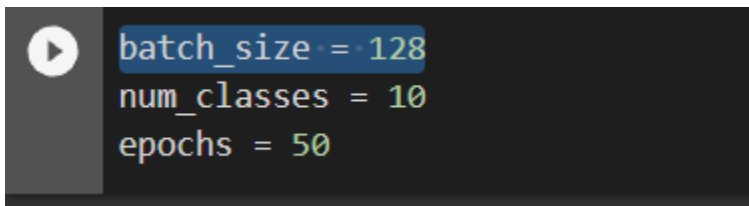
4. After creating a new notebook, copy the baseline code into the following window and click on the “play” button to run the code:



The output should be similar to the screenshot below: (due to the internet connection and server computing capability, you might need to wait for a while to get the results, please be patient.)

```
Epoch 1/50
391/391 [=====] - 16s 37ms/step - loss: 2.2247 - accuracy: 0.3950 - val_loss: 1.6065 - val_accuracy: 0.4440
Epoch 2/50
391/391 [=====] - 14s 37ms/step - loss: 1.2388 - accuracy: 0.5567 - val_loss: 1.2185 - val_accuracy: 0.5761
Epoch 3/50
391/391 [=====] - 14s 36ms/step - loss: 1.0286 - accuracy: 0.6356 - val_loss: 0.9805 - val_accuracy: 0.6499
Epoch 4/50
391/391 [=====] - 14s 36ms/step - loss: 0.8861 - accuracy: 0.6895 - val_loss: 0.7772 - val_accuracy: 0.7307
Epoch 5/50
391/391 [=====] - 14s 36ms/step - loss: 0.7954 - accuracy: 0.7231 - val_loss: 0.9082 - val_accuracy: 0.6868
Epoch 6/50
391/391 [=====] - 14s 36ms/step - loss: 0.7330 - accuracy: 0.7438 - val_loss: 0.7673 - val_accuracy: 0.7410
Epoch 7/50
```

5. After verifying the baseline, you can change different hyperparameters in the code. Some examples are shown below:
 - a. Batch size



- b. Feature map size

```

model = Sequential()
model.add(Conv2D(filters = 32, kernel_size = (3,3), padding='same', input_shape = input_shape))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(Conv2D(filters = 32, kernel_size = (3,3), padding='same'))
model.add(Activation('relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(2,2))
model.add(Dropout(0.2))

```

c. Learning rate

```

# https://keras.io/optimizers/
model.compile(loss=keras.losses.categorical_crossentropy, optimizer=keras.optimizers.Adam(lr=0.01), metrics=['accuracy'])

```

d. Optimizer. For details about different optimizers, you can check <https://keras.io/optimizers/>

```

# https://keras.io/optimizers/
model.compile(loss=keras.losses.categorical_crossentropy, optimizer=keras.optimizers.Adam(lr=0.01), metrics=['accuracy'])

```