

```

from __future__ import print_function
import keras
from keras import ...
from keras import ...
from keras import ...
from keras import ...

```

softmax

드라이브에서 찾기  
실습 모드에서 열기

새 Python 3 노트  
새 Python 2 노트

▼ Hype

노트 열기... Ctrl+O  
노트 업로드...  
이름 바꾸기...  
휴지통으로 이동

# input  
img\_rows

드라이브에 사본 저장...  
GitHub Gist로 사본 저장...

# the data  
(x\_train

GitHub에 사본 저장...

if K.image

저장 Ctrl+S

x\_train

버전 저장 및 고정 Ctrl+M S

x\_test

input

업데이트 기록

else:

x\_train

x\_test

input

.ipynb 다운로드

.py 다운로드

x\_train

x\_test =

x\_train

x\_test /

print('x

print(x\_train

print(x\_test

드라이브 미리보기 업데이트

인쇄

Ctrl+P



```

# convert class vectors to binary class matrices
y_train = keras.utils.to_categorical(y_train, num_classes)
y_test = keras.utils.to_categorical(y_test, num_classes)

```

```

model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3),
                  activation='relu',
                  input_shape=input_shape))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))
model.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.Adadelta(),

```

```
metrics=['accuracy'])
```

```
model.summary()
```



```
model.fit(x_train, y_train,  
          batch_size=batch_size,  
          epochs=epochs,  
          verbose=1,  
          validation_data=(x_test, y_test))  
score = model.evaluate(x_test, y_test, verbose=0)  
print('Test loss:', score[0])  
print('Test accuracy:', score[1])
```



