Toy Project 2주차 중간보고서

Simpson 캐릭터 10가지를 분류하는 cnn 신경망을 colab을 이용한 tensorflow로 만들어보았다.

1. 메인 파일(이름 : project)에 test파일과 valid, test 파일을 만들고 각각 안에 10가지의 simpson 캐릭터 이름 파일을 만든다. 그 각각 파일에 모두 이미지를 넣었는데 캐릭터 당 test set은 각각 50개, valid set과 test set은 각각 10개의 이미지를 이용하였다.

2. project 파일을 압축시켜서 google 드라이브에 올린다.

3. 아래의 명령어를 입력하여 project 파일의 압축을 푼다.

```
!unzip /content/drive/"My Drive"/project.zip -d /content/drive/"My Drive"/
```

4. 필요한 라이브러리를 입력한다.

```
from keras.preprocessing.image import ImageDataGenerator
from keras.utils import np_utils
from keras.models import Sequential
from keras.preprocessing.image import array_to_img, img_to_array, load_img
from keras.layers import *
from keras.callbacks import ModelCheckpoint, EarlyStopping
import os
import pandas as pd
import numpy as np
from sklearn.metrics import accuracy_score
```

5. test, valid, test 파일의 경로를 path로 설정한다.

```
path = r"/content/drive/My Drive/"
```

6. train data 이미지를 증식시킨다.

```
trainDataGen = ImageDataGenerator(rescale=1./255,
                                  rotation_range = 10,
                                  width_shift_range=0.1,
                                  height_shift_range=0.1,
                                  shear_range=0.1,
                                  zoom_range=0.1,
                                  horizontal_flip=False,
                                  vertical_flip=False,
                                  fill_mode='nearest'
                                  )

trainGenSet = trainDataGen.flow_from_directory(
    path + 'train',
    batch_size=50,
    target_size=(28,28),
    class_mode='categorical'
)
```

7. test data 이미지를 정규화한다.

```
testDataGen = ImageDataGenerator(rescale=1./255)

testGenSet = testDataGen.flow_from_directory(
    path + 'test',
    target_size=(28,28),
    batch_size=10,
    class_mode='categorical'
)
```

8. validation data로 사용하기 위해 train data에 조금 더 넓은 범위로 증식시킨다.

```
valDataGen = ImageDataGenerator(rescale=1./255,
                                rotation_range = 15,
                                width_shift_range=0.2,
                                height_shift_range=0.2,
                                shear_range=0.2,
                                zoom_range=0.1,
                                horizontal_flip=False,
                                vertical_flip=False,
                                fill_mode='nearest')

valGenSet = valDataGen.flow_from_directory(
    path + 'valid',
    target_size=(28,28),
    batch_size=10,
    class_mode='categorical'
)
```

9. model을 생성한다.

```
model = Sequential()
model.add(Conv2D(64, kernel_size=(3,3), padding='same', input_shape=(28,28,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))

model.add(Conv2D(128, kernel_size=(3,3),padding='same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))

model.add(Conv2D(256, kernel_size=(3,3), padding='same', activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))

model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(18, activation='softmax'))
```

10. model.summary() 결과

```
Found 500 images belonging to 10 classes.
Found 100 images belonging to 10 classes.
Found 100 images belonging to 10 classes.
Model: "sequential_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_21 (Conv2D)           (None, 28, 28, 64)        1792
_________________________________________________________________
max_pooling2d_21 (MaxPooling (None, 14, 14, 64)        0
_________________________________________________________________
dropout_28 (Dropout)         (None, 14, 14, 64)        0
_________________________________________________________________
conv2d_22 (Conv2D)           (None, 14, 14, 128)       73856
_________________________________________________________________
max_pooling2d_22 (MaxPooling (None, 7, 7, 128)         0
_________________________________________________________________
dropout_29 (Dropout)         (None, 7, 7, 128)         0
_________________________________________________________________
conv2d_23 (Conv2D)           (None, 7, 7, 256)         295168
_________________________________________________________________
max_pooling2d_23 (MaxPooling (None, 3, 3, 256)         0
_________________________________________________________________
dropout_30 (Dropout)         (None, 3, 3, 256)         0
_________________________________________________________________
flatten_7 (Flatten)          (None, 2304)              0
_________________________________________________________________
dense_14 (Dense)             (None, 256)               590080
_________________________________________________________________
dropout_31 (Dropout)         (None, 256)               0
_________________________________________________________________
dense_15 (Dense)             (None, 18)                4626
=================================================================
Total params: 965,522
Trainable params: 965,522
Non-trainable params: 0
_________________________________________________________________
```

11. model 학습하고 결과를 print한다.

```
model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])

model.fit_generator(
    trainGenSet,
    steps_per_epoch=40,
    epochs=400,
    validation_data=valGenSet,
    validation_steps=20
)

scores = model.evaluate_generator(testGenSet)
print(scores)
```

12. model을 학습시키는 과정에서 error가 발생하였다.

```
model.fit_generator(
    trainGenSet,
    steps_per_epoch=40,
    epochs=400,
    validation_data=valGenSet,
    validation_steps=20
)
```

```
Found 500 images belonging to 10 classes.
Found 100 images belonging to 10 classes.
Found 100 images belonging to 10 classes.
Epoch 1/400
---------------------------------------------------------------------------
InvalidArgumentError                      Traceback (most recent call last)
<ipython-input-21-2d8d71f0b264> in <module>()
     88     epochs=400,
     89     validation_data=valGenSet,
---> 90     validation_steps=20
     91 )
     92

                                    10 frames
/usr/local/lib/python3.6/dist-packages/tensorflow/python/eager/execute.py in quick_execute(op_name, num_outputs, inputs, attrs, ctx, name)
     58     ctx.ensure_initialized()
     59     tensors = pywrap_tfe.TFE_Py_Execute(ctx._handle, device_name, op_name,
---> 60                                         inputs, attrs, num_outputs)
     61   except core._NotOkStatusException as e:
     62     if name is not None:

InvalidArgumentError:  logits and labels must be broadcastable: logits_size=[50,18] labels_size=[50,10]
        [[node categorical_crossentropy/softmax_cross_entropy_with_logits (defined at <ipython-input-21-2d8d71f0b264>:90) ]] [Op:__inference_train_function_13903]

Function call stack:
train_function
```

다음주까지 이 error를 고치면 완성될 것으로 생각된다.