

21 - 1 딥러닝 및 응용 과제 1

# 이미지 분류

---

조교 이봉석, 최원혁  
tjr4090@hanyang.ac.kr  
gandet09@hanyang.ac.kr

# 과제 개요

- 수업시간에 배운 다양한 방법들을 이용해서 이미지 분류 task 성능 향상.
- 세부사항
  - 제공된 base line 코드를 수정해서 **성능 향상**이 목표.
  - **10종류**의 이미지를 각각에 맞는 class로 **분류**하는 것이 목표.
  - class의 종류는 airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck
  - 주어지는 데이터 (트레이닝 데이터 48,000개, 테스트 데이터 12,000개)
    - x\_train.npy => 트레이닝 데이터, 이미지
    - y\_train.npy => 트레이닝 데이터, 각 이미지에 대한 라벨
    - x\_test.npy => 성능 측정을 위한 test 데이터, 이미지
  - 트레이닝 데이터로 학습 후 x\_test.npy를 분류한 **result를 이용해서 측정된 성능으로 점수 산출.**

# 성능 평가 metrics

- **F1 score**

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

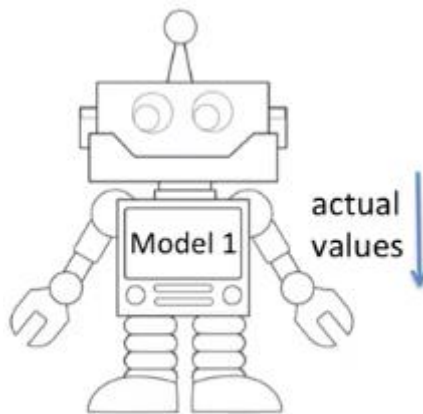
Precision : 모델이 true라고 분류한 것 중에서 실제 True인 것의 비율

Recall : 실제 True인 것 중에서 모델이 True라고 분류한 것의 비율

# 성능 평가 metrics

- **F1 score**

Precision : 모델이 true라고 분류한 것 중에서 실제 True인 것의 비율



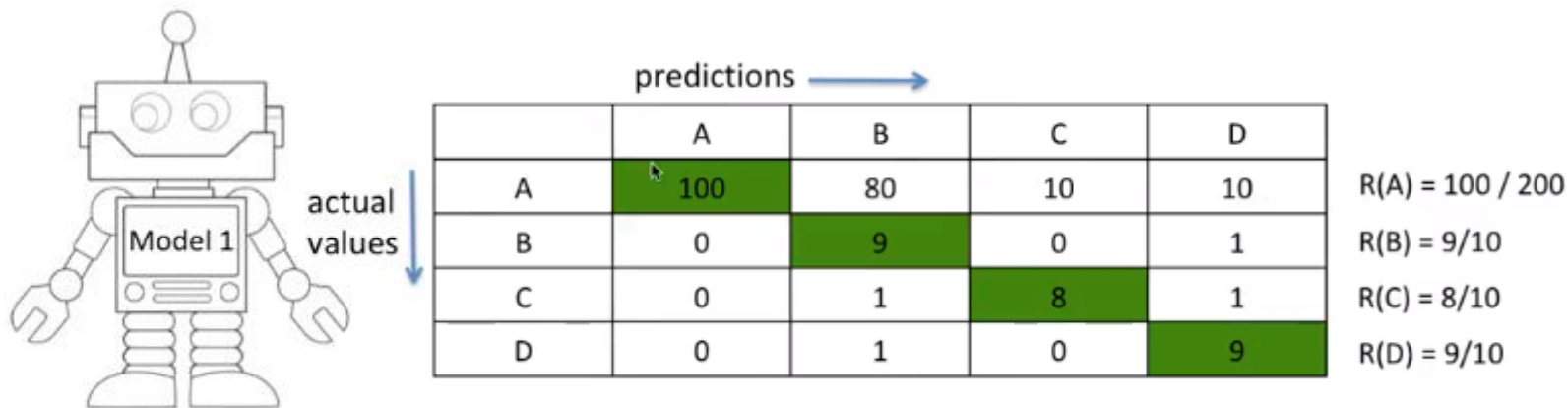
	predictions →			
	A	B	C	D
A	100	80	10	10
B	0	9	0	1
C	0	1	8	1
D	0	1	0	9

Precision =       $P(A) = 1$        $P(B) = 9/91$     $P(C) = 8/18$     $P(D) = 9/21$

# 성능 평가 metrics

- **F1 score**

Recall : 실제 True인 것 중에서 모델이 True라고 분류한 것의 비율



average를 하는 방식에 따라 다양한 F1 score가 존재

# 제공된 코드

- `image_show.py` : 주어진 데이터 이미지를 보기 위한 코드

```
1  # 각 숫자 라벨별 class
2
3  # 0: airplane
4  # 1: automobile
5  # 2: bird
6  # 3: cat
7  # 4: deer
8  # 5: dog
9  # 6: frog
10 # 7: horse
11 # 8: ship
12 # 9: truck
13
14 import matplotlib.pyplot as plt
15 import numpy as np
16
17 x_train = np.load("data/x_train.npy")
18 y_train = np.load("data/y_train.npy")
19
20 plt.imshow(x_train[0], interpolation="bicubic")
21 plt.show()
22 print(y_train[0])
23
```

## 제공된 코드

- eval.py : x\_test 파일을 분류하여 나온 결과를 성능평가하기 위한 코드

```
1  import sys
2  from sklearn.metrics import f1_score
3  import numpy as np
4
5  result_path = sys.argv[1] # result.npy
6  GT = sys.argv[2] # "data/y_test.npy"
7
8  result = np.load(result_path)
9  y_prediction = np.argmax(result, 1)
10 y_test = np.load(GT)
11 y_true = np.squeeze(y_test ,axis=1)
12
13 print(f1_score(y_true, y_prediction, average="weighted"))
```

# 제공된 코드

- model4student.py : baseline 코드

```
1  import tensorflow as tf
2  import numpy as np
3  from sklearn.metrics import f1_score
4
5  learning_rate = 0.001
6  training_epochs = 10
7  batch_size = 128
8
9
10 def batch_data(shuffled_idx, batch_size, data, labels, start_idx):
11     idx = shuffled_idx[start_idx:start_idx+batch_size]
12     data_shuffle = [data[i] for i in idx]
13     labels_shuffle = [labels[i] for i in idx]
14
15     return np.asarray(data_shuffle), np.asarray(labels_shuffle)
16
17 def build_CNN_classifier(x):
18     x_image = x
19
20     W1 = tf.get_variable(name="W1", shape=[5, 5, 3, 64], initializer=tf.contrib.layers.xavier_initializer())
21     b1 = tf.get_variable(name="b1", shape=[64], initializer=tf.contrib.layers.xavier_initializer())
22     c1 = tf.nn.conv2d(x_image, W1, strides=[1, 1, 1, 1], padding='SAME')
23     l1 = tf.nn.relu(tf.nn.bias_add(c1, b1))
24     l1_pool = tf.nn.max_pool(l1, ksize=[1, 3, 3, 1], strides=[1, 2, 2, 1], padding='SAME')
25
26     l1_flat = tf.reshape(l1_pool, [-1, 16*16*64])
27
28     W_fc = tf.get_variable(name="W_fc", shape=[16*16*64, 10], initializer=tf.contrib.layers.xavier_initializer())
29     b_fc = tf.get_variable(name="b_fc", shape=[10], initializer=tf.contrib.layers.xavier_initializer())
30     logits = tf.nn.bias_add(tf.matmul(l1_flat, W_fc), b_fc)
31     hypothesis = tf.nn.softmax(logits)
32
33     return hypothesis, logits
34
35 ckpt_path = "output/"
36
37 x = tf.placeholder(tf.float32, shape=[None, 32, 32, 3])
38 y = tf.placeholder(tf.float32, shape=[None, 10])
39
40 x_train = np.load("data/x_train.npy")
41 y_train = np.load("data/y_train.npy")
42
43 dev_num = len(x_train) // 4
44
45 x_dev = x_train[dev_num:]
46 y_dev = y_train[dev_num:]
47
```



# 제공된 코드

- model4student.py : baseline 코드

```
47
48 x_train = x_train[dev_num:]
49 y_train = y_train[dev_num:]
50
51 y_train_one_hot = tf.squeeze(tf.one_hot(y_train, 10),axis=1)
52 y_dev_one_hot = tf.squeeze(tf.one_hot(y_dev, 10),axis=1)
53
54 y_pred, logits = build_CNN_classifier(x)
55
56 cost = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(labels=y, logits=logits))
57 train_step = tf.train.AdamOptimizer(learning_rate).minimize(cost)
58
59 total_batch = int(len(x_train)/batch_size) if len(x_train)%batch_size == 0 else int(len(x_train)/batch_size) + 1
60
61 with tf.Session() as sess:
62     sess.run(tf.global_variables_initializer())
63     print("학습시작")
64
65     for epoch in range(training_epochs):
66         print("Epoch", epoch+1)
67         start = 0
68         shuffled_idx = np.arange(0, len(x_train))
69         np.random.shuffle(shuffled_idx)
70
71         for i in range(total_batch):
72             batch = batch_data(shuffled_idx, batch_size, x_train, y_train_one_hot.eval(), i*batch_size)
73             sess.run(train_step, feed_dict={x: batch[0], y: batch[1]})
74
75     saver = tf.train.Saver()
76     saver.save(sess, ckpt_path)
77     saver.restore(sess, ckpt_path)
78
79     y_prediction = np.argmax(y_pred.eval(feed_dict={x: x_dev}), 1)
80     y_true = np.argmax(y_dev_one_hot.eval(), 1)
81     dev_f1 = f1_score(y_true, y_prediction, average="weighted")
82     print("dev 데이터 f1 score: %f" % dev_f1)
83
84     # 밑에는 건드리지 마세요
85     x_test = np.load("data/x_test.npy")
86     test_logits = y_pred.eval(feed_dict={x: x_test})
87     np.save("result", test_logits)
88
```

# 점수 산출

- 코드 (70%)
  - 파일 이름
    - model4student.py (파이썬 파일 **한 개 로만 작동** 가능 하도록)
  - Test
    - 작동 여부를 평가하여 점수 산출
- 보고서 (30%)
  - 코드 설명
    - 모델(코드)에 대한 **설명** 명시. (15%)
  - 실험결과
    - 제출한 result.npy을 통해 모델의 **성능** 비교 (15%)

# 과제 조건

- **환경**

- 프로그래밍 언어 : **Python 3.7, tensorflow 1.13.1버전**
- OS : **Windows**
- 설치가 필요한 라이브러리 : sklearn, matplotlib(colab에서는 설치 필요 無)  
설치 명령어 : `pip install sklearn`  
`pip install matplotlib`
- 보고서 : **PDF**

- **제출 사항**

- 파이썬 파일 : **model4student.py**
- 결과 보고서 : **본인학번\_assignment\_1.pdf**
- 결과파일 : **result.npy, output** 폴더
- 기타 기본 제공 파일 : **eval.py, image\_show.py, data** 폴더

# 주의 사항

- **파일명** 반드시 준수.
- 파일은 **GitLab**에 올려주세요.
- 제출 기한 : **2021.05.21 (23:59)**
- 추가 제출 기한 **없음**.
- 점수 비중 : **코드 70% 보고서 30%**

# 주의 사항

- 제출하는 파일들을 경로대로 GitLab에 제출!
- 경로: (GitLab init 경로) – (image classification) – [파일]



Thank you!

---