

# Image Classification – assignment1 보고서

컴퓨터소프트웨어학부

2016025996

조성진

딥러닝 모델을 학습하는데, 항상 통용되는 hyperparameter값이 없고 때로는 training data에 의해 overfitting되어 test의 f1 score, accuracy가 떨어질 수 있으므로 여러번의 테스트를 진행하였다. 최종적으로 선택한 모델의 코드를 보면, 우선 hyperparameter의 값들을 다음과같이 선택하였다.

```
learning_rate = 0.001
training_epochs = 50
batch_size = 128
regularization_strength = 0.01
keep_prob:0.6
```

그리고 epoch를 50으로 늘렸지만, 학습 도중에 성능의 개선이 보이지 않는 경우에 중단시키기 위해 EarlyStopping클래스를 사용했다.

```
class EarlyStopping():
    def __init__(self, patience=0, verbose=0):
        self._loss = float('inf')
        self._step = 0
        self.patience = patience
        self.verbose = verbose
    def validate(self, loss):
        if self._loss < loss:
            self._step += 1
            if self._step > self.patience:
                if self.verbose:
                    print("Early stopped!")
                return True
        else:
            self._loss = loss
            self._step = 0
        return False
```

해당 코드에서 5번 이상 성능이 개선되지 못할 경우 학습을 중단하도록 만들었

다.

먼저 메인함수 부분부터 설명을하면

```
np.random.seed(51)
tf.set_random_seed(51)
```

실행할때마다 다른 값이 나와서 seed값을 고정시켰는데 여전히 실행할때마다 값이 조금씩 다른것 같다.

```
keep_prob = tf.placeholder_with_default(1.0, shape=())
```

Dropout rate를 설정하는 부분에 있어서 최종 결과에 dropout을 사용하지 않으려고 하는데 코드의 수정할 수 없는 부분이 있기 때문에 해당 함수를 찾아서 사용하였다.

```
regularization_loss = regularization_strength * tf.reduce_sum([tf.nn.l2_loss(v) for v in tf.compat.v1.trainable_variables() if len(v.shape) > 1])
cost = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(labels=y, logits=logits)) + regularization_loss
```

해당 코드는 수업시간에 배운 overfitting을 방지하기 위해 regularization을 적용하였다.

```
for epoch in range(training_epochs):
    print("Epoch", epoch+1)
    start = 0
    shuffled_idx = np.arange(0, len(x_train))
    np.random.shuffle(shuffled_idx)
    losses = 0

    for i in range(total_batch):
        batch = batch_data(shuffled_idx, batch_size, x_train, y_train_one_hot.eval(), i*batch_size)
        loss = sess.run([train_step, cost], feed_dict={x: batch[0], y: batch[1], keep_prob:0.6})
        losses += loss
    print("losses:", losses)

    if early_stopping.validate(losses):
        break
```

Dropout에서 60%의 노드들만 학습에 참여(FCNN층에만 적용)하게 만들었으며 early stop될수 있도록 코드를 추가하였다. Loss를 확인하기 위해 출력을 epoch마다 하도록 만들었다.

나머지 부분은 base line 코드와 동일하다.

CNN의 층을 쌓는건 굉장히 많은 층을 쌓았는데, 처음에는 스스로 여러 실험을 해보다가 f1 score가 생각보다 너무 낮게 나와 인터넷에 나와있는 여러 모델들의 층을 줄이기도하고 인자도 조금씩 바꿔보며 실험을 추가적으로 해봤다.

```

W1 = tf.get_variable(name="W1", shape=[3, 3, 3, 64], initializer=tf.contrib.layers.xavier_initializer())
b1 = tf.get_variable(name="b1", shape=[64], initializer=tf.contrib.layers.xavier_initializer())
c1 = tf.nn.conv2d(x_image, W1, strides=[1, 1, 1, 1], padding='SAME')
l1 = tf.nn.relu(tf.nn.bias_add(c1, b1))

W2 = tf.get_variable(name="W2", shape=[3, 3, 64, 64], initializer=tf.contrib.layers.xavier_initializer())
b2 = tf.get_variable(name="b2", shape=[64], initializer=tf.contrib.layers.xavier_initializer())
c2 = tf.nn.conv2d(l1, W2, strides=[1, 1, 1, 1], padding='SAME')
l2 = tf.nn.relu(tf.nn.bias_add(c2, b2))
l2_pool = tf.nn.max_pool(l2, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1], padding='SAME')

W3 = tf.get_variable(name="W3", shape=[3, 3, 64, 128], initializer=tf.contrib.layers.xavier_initializer())
b3 = tf.get_variable(name="b3", shape=[128], initializer=tf.contrib.layers.xavier_initializer())
c3 = tf.nn.conv2d(l2_pool, W3, strides=[1, 1, 1, 1], padding='SAME')
l3 = tf.nn.relu(tf.nn.bias_add(c3, b3))

W4 = tf.get_variable(name="W4", shape=[3, 3, 128, 128], initializer=tf.contrib.layers.xavier_initializer())
b4 = tf.get_variable(name="b4", shape=[128], initializer=tf.contrib.layers.xavier_initializer())
c4 = tf.nn.conv2d(l3, W4, strides=[1, 1, 1, 1], padding='SAME')
l4 = tf.nn.relu(tf.nn.bias_add(c4, b4))
l4_pool = tf.nn.max_pool(l4, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1], padding='SAME')

W5 = tf.get_variable(name="W5", shape=[3, 3, 128, 256], initializer=tf.contrib.layers.xavier_initializer())
b5 = tf.get_variable(name="b5", shape=[256], initializer=tf.contrib.layers.xavier_initializer())
c5 = tf.nn.conv2d(l4_pool, W5, strides=[1, 1, 1, 1], padding='SAME')
l5 = tf.nn.relu(tf.nn.bias_add(c5, b5))

W6 = tf.get_variable(name="W6", shape=[3, 3, 256, 256], initializer=tf.contrib.layers.xavier_initializer())
b6 = tf.get_variable(name="b6", shape=[256], initializer=tf.contrib.layers.xavier_initializer())
c6 = tf.nn.conv2d(l5, W6, strides=[1, 1, 1, 1], padding='SAME')
l6 = tf.nn.relu(tf.nn.bias_add(c6, b6))
l6_pool = tf.nn.max_pool(l6, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1], padding='SAME')

W7 = tf.get_variable(name="W7", shape=[3, 3, 256, 512], initializer=tf.contrib.layers.xavier_initializer())
b7 = tf.get_variable(name="b7", shape=[512], initializer=tf.contrib.layers.xavier_initializer())
c7 = tf.nn.conv2d(l6_pool, W7, strides=[1, 1, 1, 1], padding='SAME')
l7 = tf.nn.relu(tf.nn.bias_add(c7, b7))

W8 = tf.get_variable(name="W8", shape=[3, 3, 512, 512], initializer=tf.contrib.layers.xavier_initializer())
b8 = tf.get_variable(name="b8", shape=[512], initializer=tf.contrib.layers.xavier_initializer())
c8 = tf.nn.conv2d(l7, W8, strides=[1, 1, 1, 1], padding='SAME')
l8 = tf.nn.relu(tf.nn.bias_add(c8, b8))

W9 = tf.get_variable(name="W9", shape=[3, 3, 512, 512], initializer=tf.contrib.layers.xavier_initializer())
b9 = tf.get_variable(name="b9", shape=[512], initializer=tf.contrib.layers.xavier_initializer())
c9 = tf.nn.conv2d(l8, W9, strides=[1, 1, 1, 1], padding='SAME')
l9 = tf.nn.relu(tf.nn.bias_add(c9, b9))

l9_pool = tf.nn.max_pool(l9, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1], padding='SAME')
l9_flat = tf.reshape(l9_pool, [-1, 2*2*512])

```

다음과 같이 convolution의 size는 3x3, stride=1, padding=SAME으로 해서 다음 이미지의 사이즈가 줄어들지 않도록 했다. 이 stride=1이 의미있는 값이라고 생각했는데, 후에 이미지로 실험한 결과를 첨부하겠지만 Pooling을 자주 거치거나 stride때문에 이미지의 크기가 작아져, 초기 이미지의 사이즈도 32 x 32 x 3 밖에 되지 않는 데 더욱 줄어들어서 학습이 좀 덜되지 않았나 라고 생각했다. 그래서 convolution layer을 좀더 많이 두고 pooling을 줄여서 최대한 이미지의 사이즈를 줄이지 않고 많이 학습할 수 있게 했다. 이렇게 filter가 64인층 2개 -> Pooling -> filter 128인층 2개 -> Pooling -> filter 256인층 2개 -> Pooling -> filter 512인층 3개 -> pooling을 진행하였다.

```

W_fc1 = tf.get_variable(name="W_fc1", shape=[2*2*512, 2*2*512], initializer=tf.contrib.layers.xavier_initializer())
b_fc1 = tf.get_variable(name="b_fc1", shape=[2*2*512], initializer=tf.contrib.layers.xavier_initializer())
l10 = tf.nn.bias_add(tf.matmul(l9_flat, W_fc1), b_fc1)
l10 = tf.nn.dropout(l10, keep_prob)
W_fc2 = tf.get_variable(name="W_fc2", shape=[2*2*512, 2*2*512], initializer=tf.contrib.layers.xavier_initializer())
b_fc2 = tf.get_variable(name="b_fc2", shape=[2*2*512], initializer=tf.contrib.layers.xavier_initializer())
l11 = tf.nn.bias_add(tf.matmul(l10, W_fc2), b_fc2)
l11 = tf.nn.dropout(l11, keep_prob)
W_fc3 = tf.get_variable(name="W_fc3", shape=[2*2*512, 10], initializer=tf.contrib.layers.xavier_initializer())
b_fc3 = tf.get_variable(name="b_fc3", shape=[10], initializer=tf.contrib.layers.xavier_initializer())
logits = tf.nn.bias_add(tf.matmul(l11, W_fc3), b_fc3)
hypothesis = tf.nn.softmax(logits)

```

그리고 뒷부분에 FCNN을 연결하여 이부분에서도 충분히 학습을 시켰다. 해당 모델의 dev set을 가지고 f1 score을 계산한 결과 0.76정도를 얻을 수 있었다.

```
Epoch 50
losses: 266.31748992204666
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/python/training/saver.py:1266: checkpoint_exists (from tensorflow.python.training.checkpoint_management) is deprecated and
Instructions for updating:
Use standard file APIs to check for files with this prefix.
dev 데이터 f1 score: 0.760370
```

그전의 여러 실험들이 있었는데,

```
학습시작
Epoch 1
2021-05-18 07:24:17.995961: I tensorflow/stream_executor/dso_loader.cc:152] successfully opened C
losses: 3372.0106596946716
Epoch 2
losses: 444.2444018125534
Epoch 3
losses: 382.9851142168045
Epoch 4
losses: 332.40608608722687
Epoch 5
losses: 295.0811690688133
Epoch 6
losses: 254.51493734121323
Epoch 7
losses: 210.182149797678
Epoch 8
losses: 175.90533816814423
Epoch 9
losses: 143.50683400034904
Epoch 10
losses: 127.28719985485077
Epoch 11
losses: 113.24639330804348
Epoch 12
losses: 98.18449555337429
Epoch 13
losses: 86.27534200251102
Epoch 14
losses: 92.314114138484
Epoch 15
losses: 75.87460998445749
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/python/training/saver.py
Instructions for updating:
Use standard file APIs to check for files with this prefix.
dev 데이터 f1 score: 0.505674
losses: 491.41037106312977
Epoch 12
losses: 483.2114931344986
Epoch 13
losses: 495.7262433767319
Epoch 14
losses: 481.1542818546295
Epoch 15
losses: 477.10681569576263
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/python/training/saver.py
Instructions for updating:
Use standard file APIs to check for files with this prefix.
dev 데이터 f1 score: 0.573845

Epoch 22
losses: 468.3417340517044
min loss: 465.99191761016846
this loss: 468.3417340517044
step: 5
Early stopped!
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/python/training/saver.py
Instructions for updating:
Use standard file APIs to check for files with this prefix.
dev 데이터 f1 score: 0.616821

28
idx = shuffled_idx[start_idx:start_idx+batch_size]
data_shuffle = [data[i] for i in idx]
labels_shuffle = [labels[i] for i in idx]
31
32
return np.asarray(data_shuffle), np.asarray(labels_shuffle)
33
34 def build_CNN_classifier(x):
35     x_image = x
36
37     W1 = tf.get_variable(name="W1", shape=[5, 5, 3, 64], ini
38     b1 = tf.get_variable(name="b1", shape=[64], initializer=
39     c1 = tf.nn.conv2d(x_image, W1, strides=[1, 1, 1, 1], paddi
40     l1 = tf.nn.relu(tf.nn.bias_add(c1, b1))
41     l1_pool = tf.nn.max_pool(l1, ksize=[1, 3, 3, 1], strides
42
43     W2 = tf.get_variable(name="W2", shape=[16, 16, 64, 128],
44     b2 = tf.get_variable(name="b2", shape=[128], initializer=
45     c2 = tf.nn.conv2d(l1_pool, W2, strides=[1, 1, 1, 1], paddi
46     l2 = tf.nn.relu(tf.nn.bias_add(c2, b2))
47     l2_pool = tf.nn.max_pool(l2, ksize=[1, 2, 2, 1], strides
48
49     l2_flat = tf.reshape(l2_pool, [-1, 8*8*128])
50
51     W_fc = tf.get_variable(name="W_fc", shape=[8*8*128, 10],
52     b_fc = tf.get_variable(name="b_fc", shape=[10], initiali
53     logits = tf.nn.bias_add(tf.matmul(l2_flat, W_fc), b_fc)
54     hypothesis = tf.nn.softmax(logits)
55
56     return hypothesis, logits
57
58
59 tf.set_random_seed(51)
60
61 f build_CNN_classifier(x):
62     x_image = x
63
64     W1 = tf.get_variable(name="W1", shape=[5, 5, 3, 64], ini
65     b1 = tf.get_variable(name="b1", shape=[64], initializer=
66     c1 = tf.nn.conv2d(x_image, W1, strides=[1, 1, 1, 1], paddi
67     l1 = tf.nn.relu(tf.nn.bias_add(c1, b1))
68     l1_pool = tf.nn.max_pool(l1, ksize=[1, 3, 3, 1], strides=
69     l1_pool = tf.nn.dropout(l1_pool, keep_prob)
70
71     W2 = tf.get_variable(name="W2", shape=[16, 16, 64, 128],
72     b2 = tf.get_variable(name="b2", shape=[128], initializer=
73     c2 = tf.nn.conv2d(l1_pool, W2, strides=[1, 1, 1, 1], paddi
74     l2 = tf.nn.relu(tf.nn.bias_add(c2, b2))
75     l2_pool = tf.nn.max_pool(l2, ksize=[1, 2, 2, 1], strides=
76     l2_pool = tf.nn.dropout(l2_pool, keep_prob)
77
78     l2_flat = tf.reshape(l2_pool, [-1, 8*8*128])
79
80     W_fc = tf.get_variable(name="W_fc", shape=[8*8*128, 10],
81     b_fc = tf.get_variable(name="b_fc", shape=[10], initializ
82     logits = tf.nn.bias_add(tf.matmul(l2_flat, W_fc), b_fc)
83     hypothesis = tf.nn.softmax(logits)
84
85     return hypothesis, logits
86
87 W1 = tf.get_variable(name="W1", shape=[5, 5, 3, 64], ini
88 b1 = tf.get_variable(name="b1", shape=[64], initializer=
89 c1 = tf.nn.conv2d(x_image, W1, strides=[1, 1, 1, 1], paddi
90 l1 = tf.nn.relu(tf.nn.bias_add(c1, b1))
91 l1_pool = tf.nn.max_pool(l1, ksize=[1, 3, 3, 1], strides=
92 l1_pool = tf.nn.dropout(l1_pool, keep_prob)
93
94 W2 = tf.get_variable(name="W2", shape=[16, 16, 64, 128],
95 b2 = tf.get_variable(name="b2", shape=[128], initializer=
96 c2 = tf.nn.conv2d(l1_pool, W2, strides=[1, 1, 1, 1], paddi
97 l2 = tf.nn.relu(tf.nn.bias_add(c2, b2))
98 l2_pool = tf.nn.max_pool(l2, ksize=[1, 2, 2, 1], strides=
99 l2_pool = tf.nn.dropout(l2_pool, keep_prob)
100
101 l2_flat = tf.reshape(l2_pool, [-1, 8*8*128])
102
103 W_fc = tf.get_variable(name="W_fc", shape=[8*8*128, 10],
104 b_fc = tf.get_variable(name="b_fc", shape=[10], initializ
105 logits = tf.nn.bias_add(tf.matmul(l2_flat, W_fc), b_fc)
106 hypothesis = tf.nn.softmax(logits)
107
108 return hypothesis, logits
109
110 W1 = tf.get_variable(name="W1", shape=[5, 5, 3, 64], ini
111 b1 = tf.get_variable(name="b1", shape=[64], initializer=
112 c1 = tf.nn.conv2d(x_image, W1, strides=[1, 1, 1, 1], paddi
113 l1 = tf.nn.relu(tf.nn.bias_add(c1, b1))
114 l1_pool = tf.nn.max_pool(l1, ksize=[1, 3, 3, 1], strides=
115 l1_pool = tf.nn.dropout(l1_pool, keep_prob)
116
117 W2 = tf.get_variable(name="W2", shape=[16, 16, 64, 128],
118 b2 = tf.get_variable(name="b2", shape=[128], initializer=
119 c2 = tf.nn.conv2d(l1_pool, W2, strides=[1, 1, 1, 1], paddi
120 l2 = tf.nn.relu(tf.nn.bias_add(c2, b2))
121 l2_pool = tf.nn.max_pool(l2, ksize=[1, 2, 2, 1], strides=
122 l2_pool = tf.nn.dropout(l2_pool, keep_prob)
123
124 l2_flat = tf.reshape(l2_pool, [-1, 8*8*128])
125
126 W_fc = tf.get_variable(name="W_fc", shape=[8*8*128, 10],
127 b_fc = tf.get_variable(name="b_fc", shape=[10], initiali
128 logits = tf.nn.bias_add(tf.matmul(l2_flat, W_fc), b_fc)
129 hypothesis = tf.nn.softmax(logits)
130
131 return hypothesis, logits
132
```

위의 경우들은 층을 그다지 깊게 가져가지도 않고 convolution layer 1번에 pooling 1번씩 한 경우 f1 score이 그다지 좋게 얻어지지 않았다.

그런데 최종 결과처럼 층을 깊게 가져가기 시작하니 f1 score이 0.7을 넘기 시작했고 마지막에 FCNN을 2층을 추가하니 0.754에서 0.76으로 미세한 성능의 개

선을 얻었다.

256개의 filter를 가지는 층을 추가하지 않았을 경우( 2번의 시행)

```
Epoch 50
losses: 185.72534388303757
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/python/training/saver.py:1266: checkpoint_exists (from tensorflow.python.training.checkpoint_util) is deprecated and will be removed in a future version.
Instructions for updating:
Use standard file APIs to check for files with this prefix.
dev 데이터 f1 score: 0.715190
```

```
Epoch 50
losses: 203.81536448001862
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/python/training/saver.py:1266: checkpoint_exists (from tensorflow.python.training.checkpoint_util) is deprecated and will be removed in a future version.
Instructions for updating:
Use standard file APIs to check for files with this prefix.
dev 데이터 f1 score: 0.738178
```

마지막 512개의 filter를 가지는 층을 추가하지 않았을 경우

```
losses: 229.1637978553772
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/python/training/saver.py:1266: checkpoint_exists (from tensorflow.python.training.checkpoint_util) is deprecated and will be removed in a future version.
Instructions for updating:
Use standard file APIs to check for files with this prefix.
dev 데이터 f1 score: 0.748226
```

FCNN을 softmax의 1개의 층만 사용하는 경우

```
Epoch 50
losses: 283.01179111003876
WARNING:tensorflow:From /usr/local/lib/python3.7/dist-packages/tensorflow/python/training/saver.py:1266: checkpoint_exists (from tensorflow.python.training.checkpoint_util) is deprecated and will be removed in a future version.
Instructions for updating:
Use standard file APIs to check for files with this prefix.
dev 데이터 f1 score: 0.754192
```