Section 11. Convolutional Neural Networks (CNN)

I. ConvNet의 Conv 레이어 만들기

파이토치 첫걸음 Chapter 5 내용이랑 같음.

II. ConvNet Max pooling과 Full Network

파이토치 첫걸음 Chapter 5 내용이랑 같음.

III. ConvNet의 활용예

1. LeNet-5
2. AlexNet
3. GoogLeNet : Inception module
4. ResNet

IV. TensorFlow CNN

Convolution → subsampling → Convolution → subsampling → ... → fully connected

1. Image 만들기

3x3 Color 한가지 이미지 (1) 2, 3, 1

이미지 개수

```
image = np.array([[[[1], [2], [3]],
                  [[4], [5], [6]],
                  [[7], [8], [9]]]],
                 dtype=np.float32)
```

1	2	3
4	5	6
7	8	9

Color ↓ ← filter수

Filter : 2, 2, 1, 1

```
[[[[1.]], [[1.]]],
 [[[1.]], [[1.]]]]
```

shape = (2, 2, 1, 1)

padding='SAME'
⇒ 필요한 만큼 0으로 채움.

2. pooling

pool=tf.nn.max_pool(image, ksize=[1,2,2,1],
strides=[1,1,1,1], padding='SAME')

V. Class, tf.layers, Ensemble

1. tf.layers

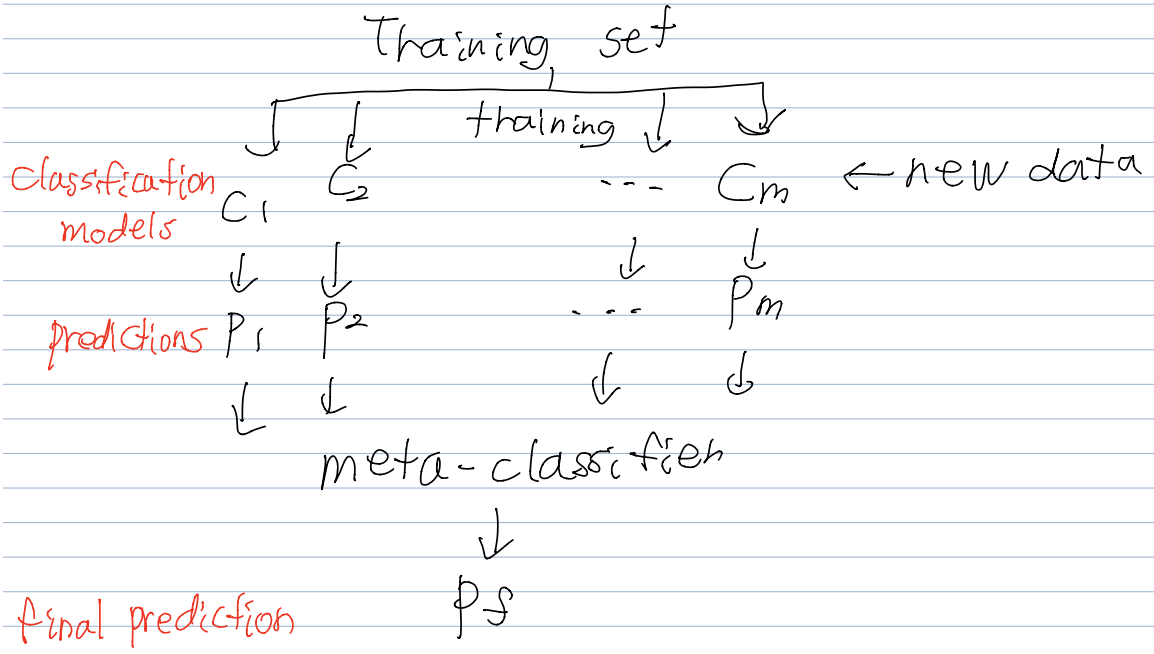
average_pooling1d
average_pooling2d
average_pooling3d
batch_normalization
conv1d
conv2d
conv2d_transpose
conv3d
dense
dropout
max_pooling1d
max_pooling2d
max_pooling3d
separable_conv2d

Ex) conv1 = tf.layers.conv2d(inputs=X_img,
filter=32,
kernel_size=[3,3],
padding="SAME",
activation=tf.nn.relu)

```
pool1 = tf.layers.max_pooling2d(inputs=conv1,
                                pool_size=[2, 2],
                                padding="SAME",
                                stride=2)
dropout1 = tf.layers.dropout(inputs=pool1, rate=0.7,
                             training=self.training)

dense4 = tf.layers.dense(inputs=flat, units=625,
                         activation=tf.nn.relu)
```

2. Ensemble



① 모델 만들기

```
class Model:
    def __init__(self, sess, name):
        self.sess = sess
        self.name = name
        self.build_net()
```

```
def _build_net(self):
    with tf.variable_scope(self.name):

models = []
num_models = 7
for m in range(num_models):
    models.append(Model(sess, "model" + str(m)))
```

② Ensemble prediction

```
test_size = len(mnist.test.labels)
predictions = np.zeros(test_size * 10).reshape
                (test_size, 10)

for m_idx, m in enumerate(models):
    p = m.predict(mnist.test.images)
    predictions += p
```