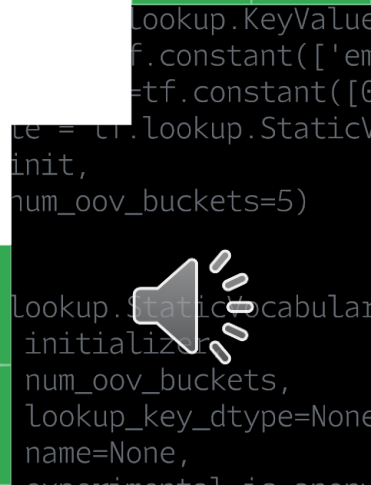
Google Developer Student Clubs

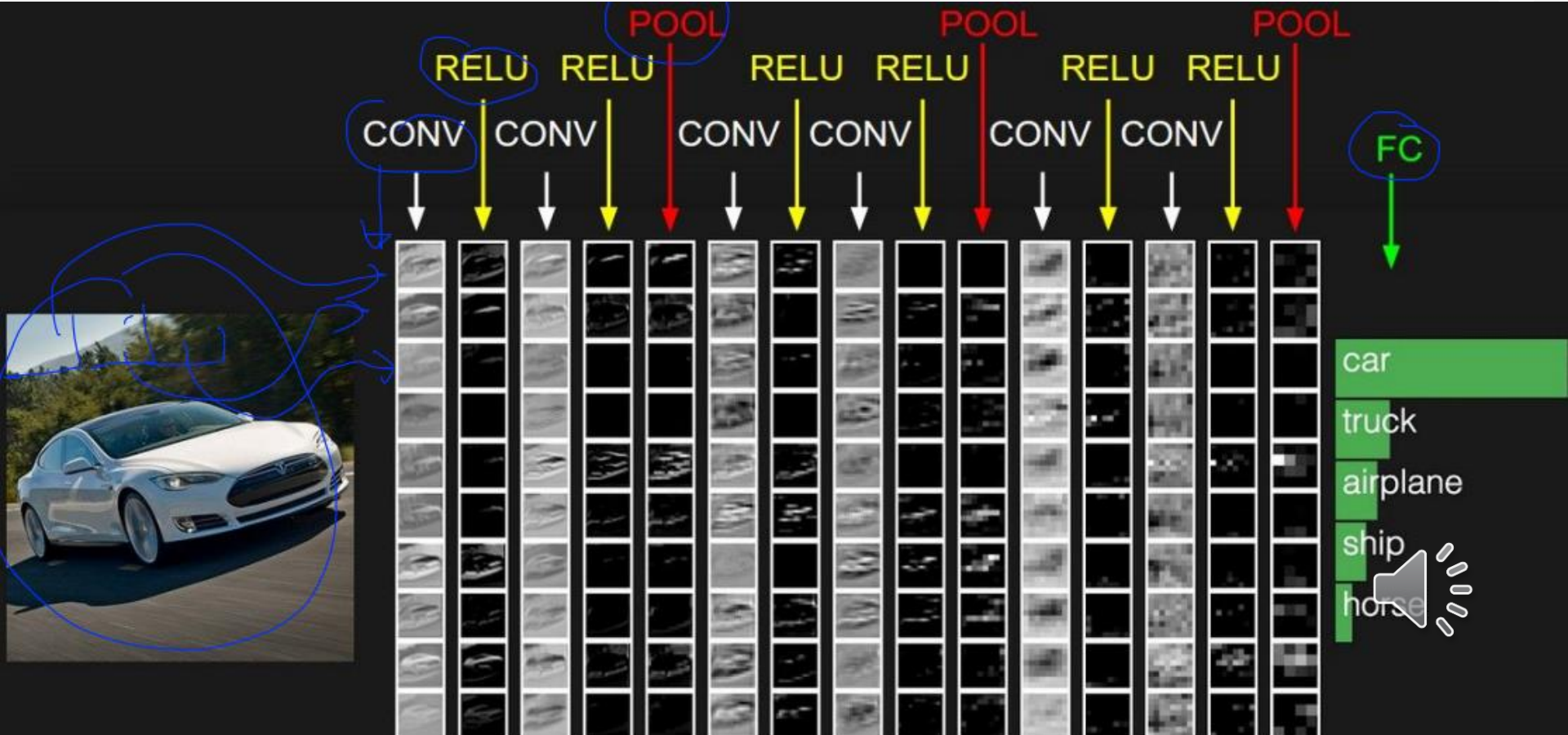
ML/DL Study W04

- Convolutional Neural Network
- Layer : Conv, Pool, Fully connected
- Ensemble
- Data Augmentation



Convolutional Neural Network

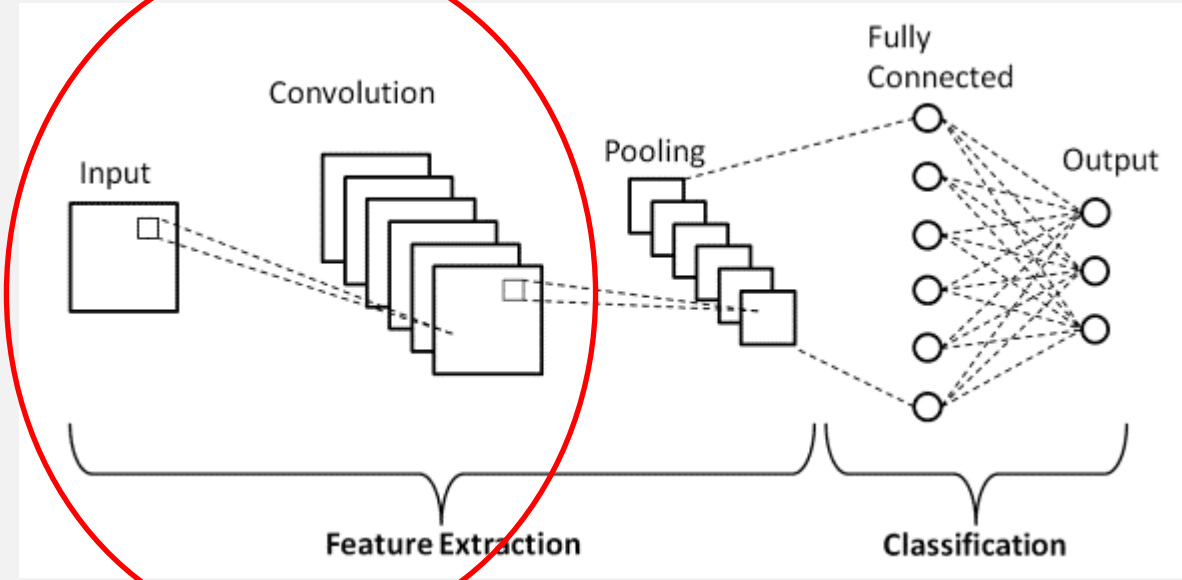
Convolutional Neural Network



Layer : Conv, Pool, Fully connected

Layer : Conv, Pool, Fully connected

Convolution Layer : 이미지 및 다차원 데이터의 특징을 감지하고 추출



- Padding : output data size를 조절할 수 있음, padding = valid or same
- Stride : filter가 input data를 이동하는 간격, (input – filter) // stride + 1로 추측 가능

Padding = same

0	0	0	0	0	0	0	0
0	3	3	4	4	7	0	0
0	9	7	6	5	8	2	0
0	6	5	5	6	9	2	0
0	7	1	3	2	7	8	0
0	0	3	7	1	8	3	0
0	4	0	4	3	2	2	0
0	0	0	0	0	0	0	0

$6 \times 6 \rightarrow 8 \times 8$

$*$

1	0	-1
1	0	-1
1	0	-1

3×3

$=$

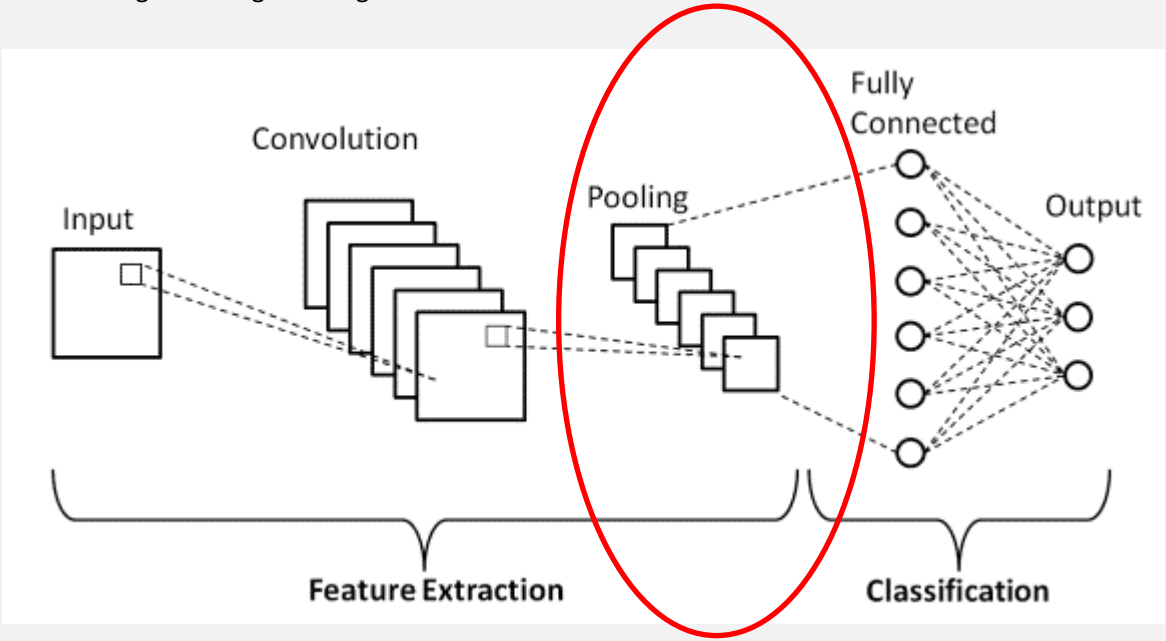
-10	-13	1			
-9	3	0			

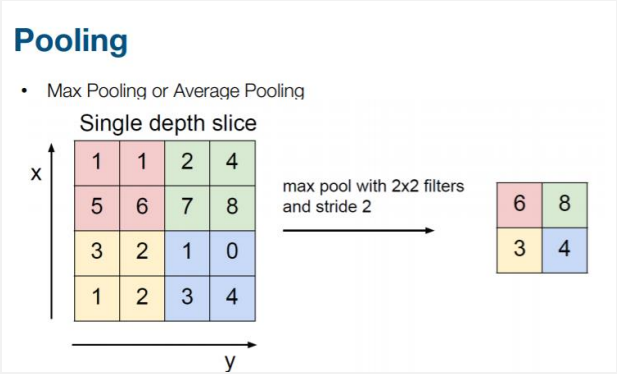
6×6

Layer : Conv, Pool, Fully connected

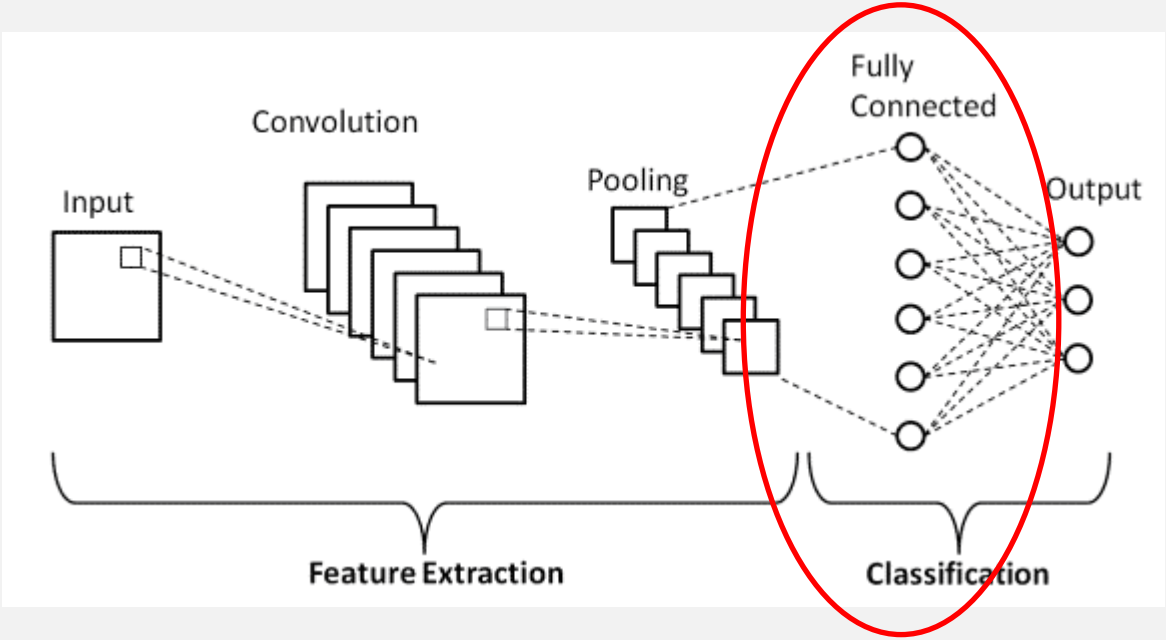
Pooling : 공간 차원(height, width) 줄이고 계산량을 감소시키면서 중요한 정보를 보존

Max Pooling vs Average Pooling





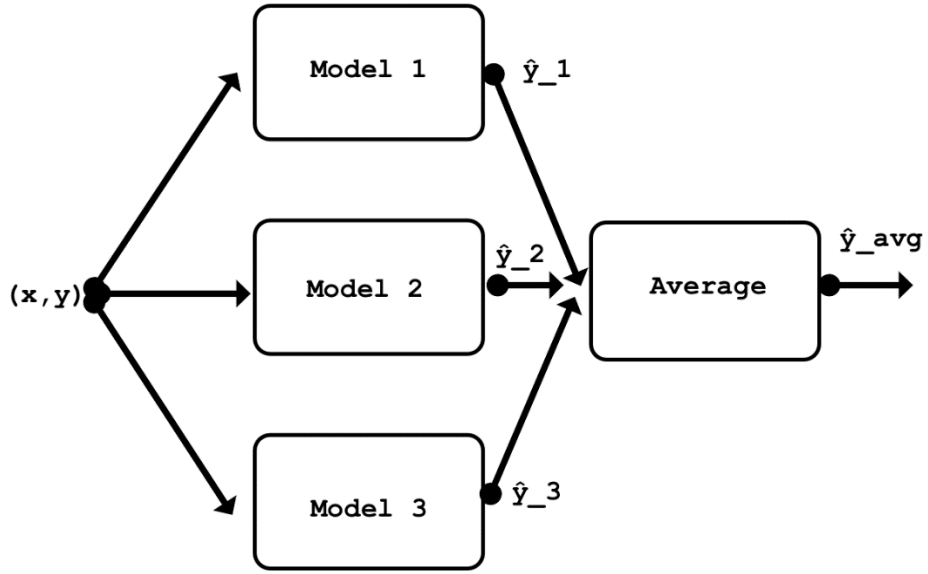
Layer : Conv, Pool, Fully connected



Ensemble

Ensemble

Ensemble : CNN model 여러 개를 조합하여 보다 정확한 model을 만드는 것

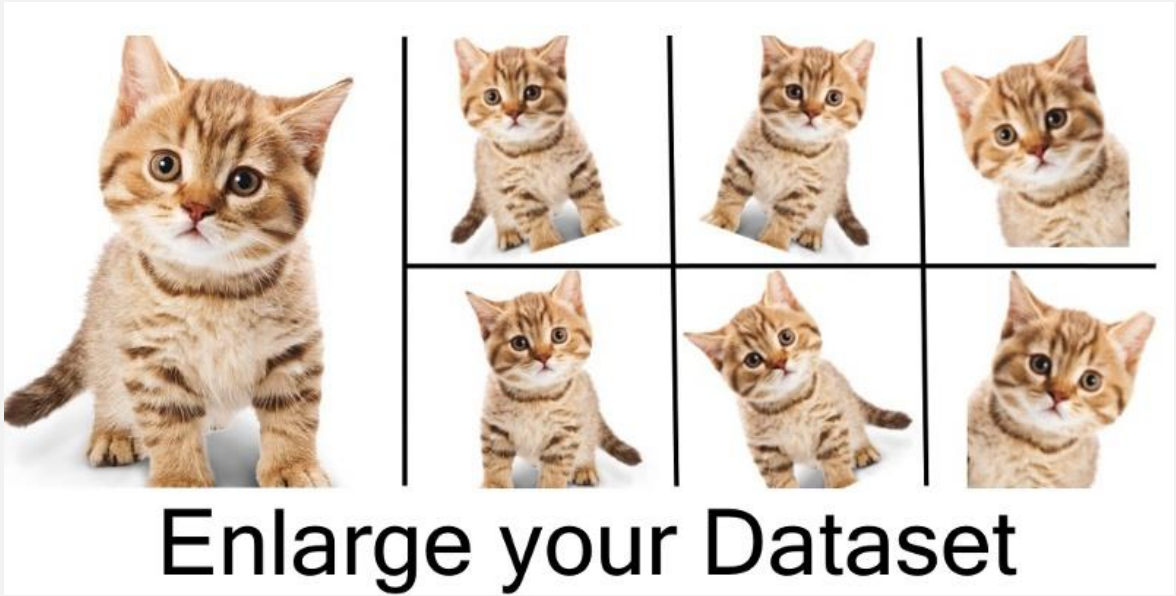



```
models = []
num_models = 3
for m in range(num_models):
    models.append(MNISTModel())
```

Data Augmentation

Data Augmentation

Data Augmentation : origin data set에서 새롭게 training data를 생성해서 model의 성능을 향상시킴



```python
def data_augmentation(images, labels):
    aug_images = []; aug_labels = []

    for x, y in zip(images, labels):
        aug_images.append(x)
        aug_labels.append(y)

        bg_value = np.median(x)
        for _ in range(4):
            angle = np.random.randint(-15, 15, 1)
            rot_img = ndimage.rotate(x, angle, reshape=False, cval=bg_value)

            shift = np.random.randint(-2, 2, 2)
            shift_img = ndimage.shift(rot_img, shift, cval=bg_value)

            aug_images.append(shift_img)
            aug_labels.append(y)
    aug_images = np.array(aug_images)
    aug_labels = np.array(aug_labels)
    return aug_images, aug_labels
```