



Week4 Presentation

**Conv Layer, Convnet Max pooling
Cat-Dog Classifier**

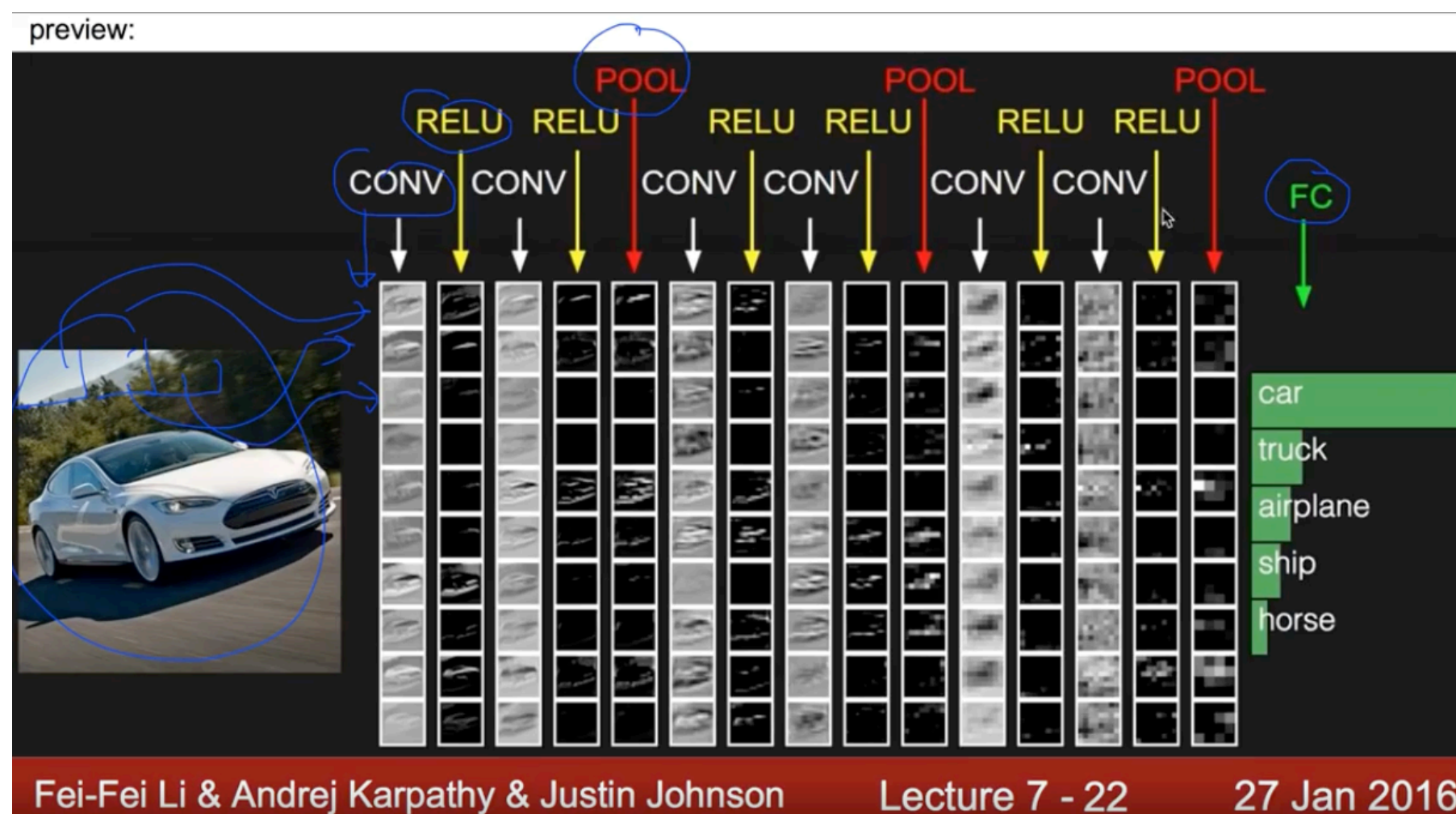
GDSC Hanyang ML/DL : Basic, Jaeseung Lee

Index

- CNN
 - Basic concepts
 - Filter, Stride, Padding
 - Max pooling
 - How to work
- Skills
 - Model Ensemble
 - Data Augmentation

Basic concepts

- CNN (Convolutional Neural Network)
 - 이미지/영상 데이터를 여러 개의 뉴런을 통해 학습하는 방법
 - Convolution 전처리를 통해 이미지/영상 데이터를 효과적으로 학습
 - Filter 기법과 Max pooling을 통해 특징들을 잘 학습 가능



CNN

Basic concepts

- Filter
 - 한 번에 이미지를 얼마 만큼 볼 지 정의함.
 - Weight 가 여기에 정의됨. -> filter 행렬의 값을 조정해 학습
- Stride
 - Filter를 한 번에 얼마 만큼 이동할 것인지 정의함. 값이 커질 수록 정보 건너뛰게 됨.
 - N : size, F : filter size -> $\text{Output size} = (N - F) / \text{stride} + 1$
- Padding
 - 이미지 배열 테두리에 0값을 가지는 배열을 두르는 방법.
 - Padding을 이용하는 이유는 입력 이미지와 출력 이미지의 사이즈를 같게 하기 위함.

In practice: Common to zero pad the border

0	0	0	0	0	0			
0								
0								
0								
0								

e.g. input 7x7

3x3 filter, applied with **stride 1**

pad with 1 pixel border => what is the output?

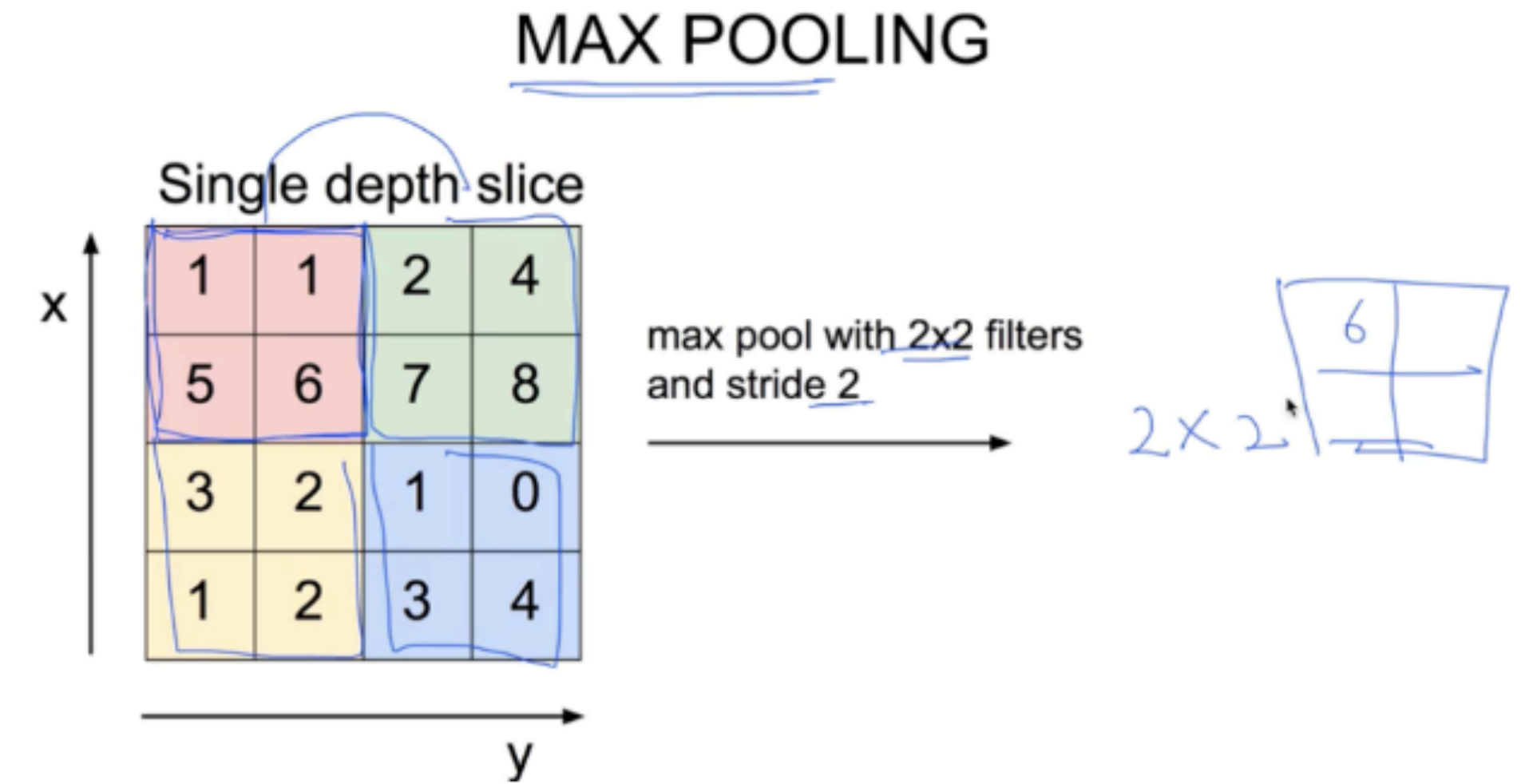
(recall:)

$$(N - F) / \text{stride} + 1$$

CNN

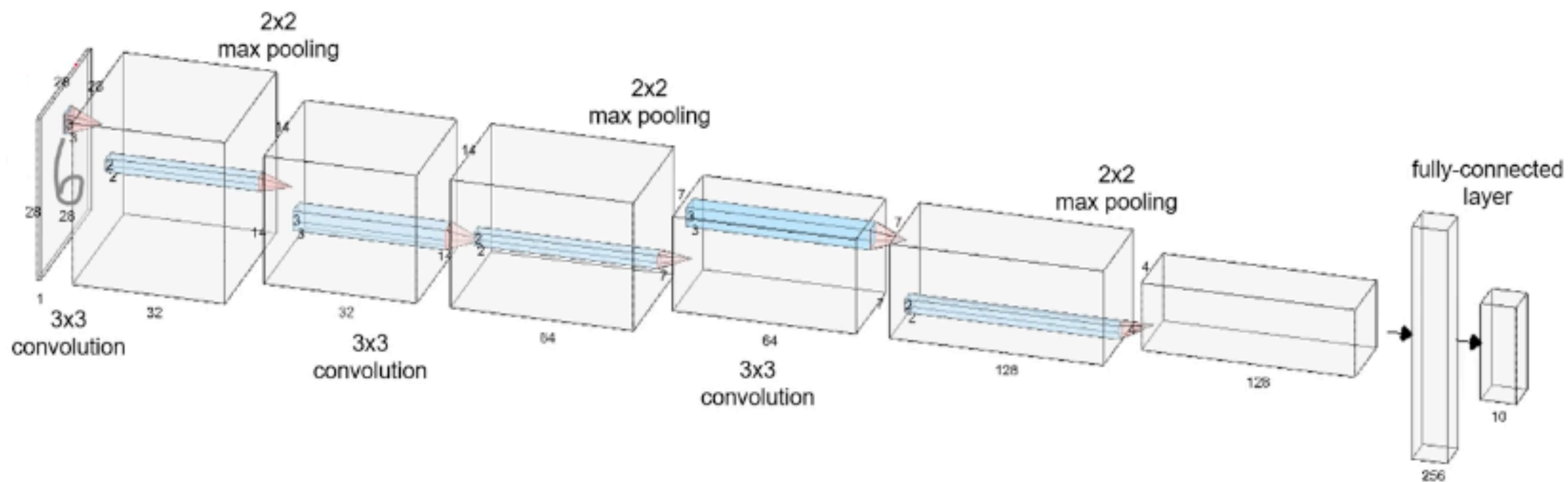
Max pooling

- Pooling
 - Sampling : Convolution Layer를 Resize 하여 한 개의 Layer 생성
 - Layer를 계속 Sampling 하여 쌓인 것을 다음 단계로 넘김
- Max pooling
 - Sampling 할 때 filter 내에 있는 가장 큰 값을 추출
 - Average pooling도 있지만, Max pooling이 특징을 잘 잡아내 주로 사용



CNN

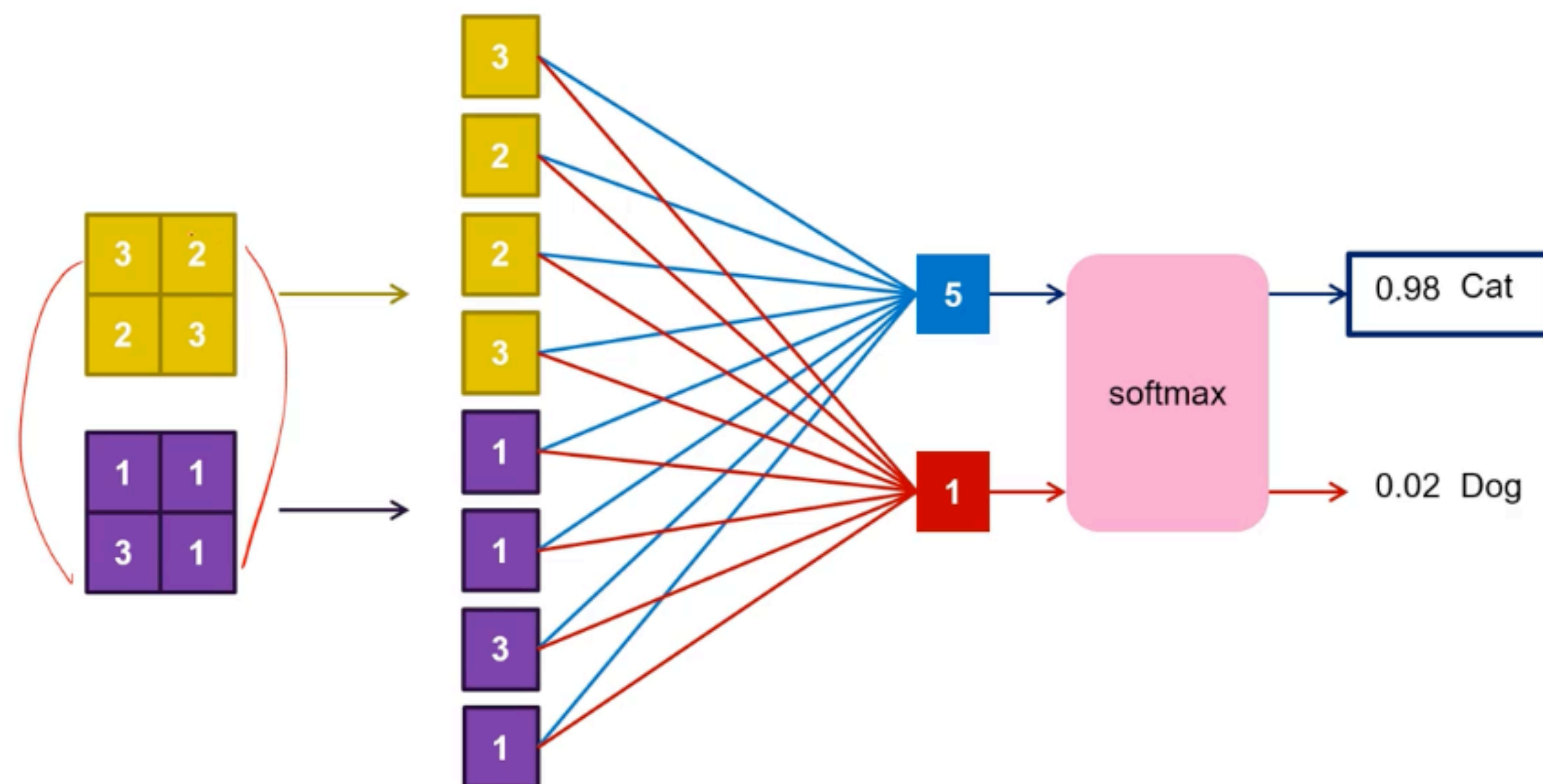
How to work



CNN

How to work

Fully Connected(Dense) Layer

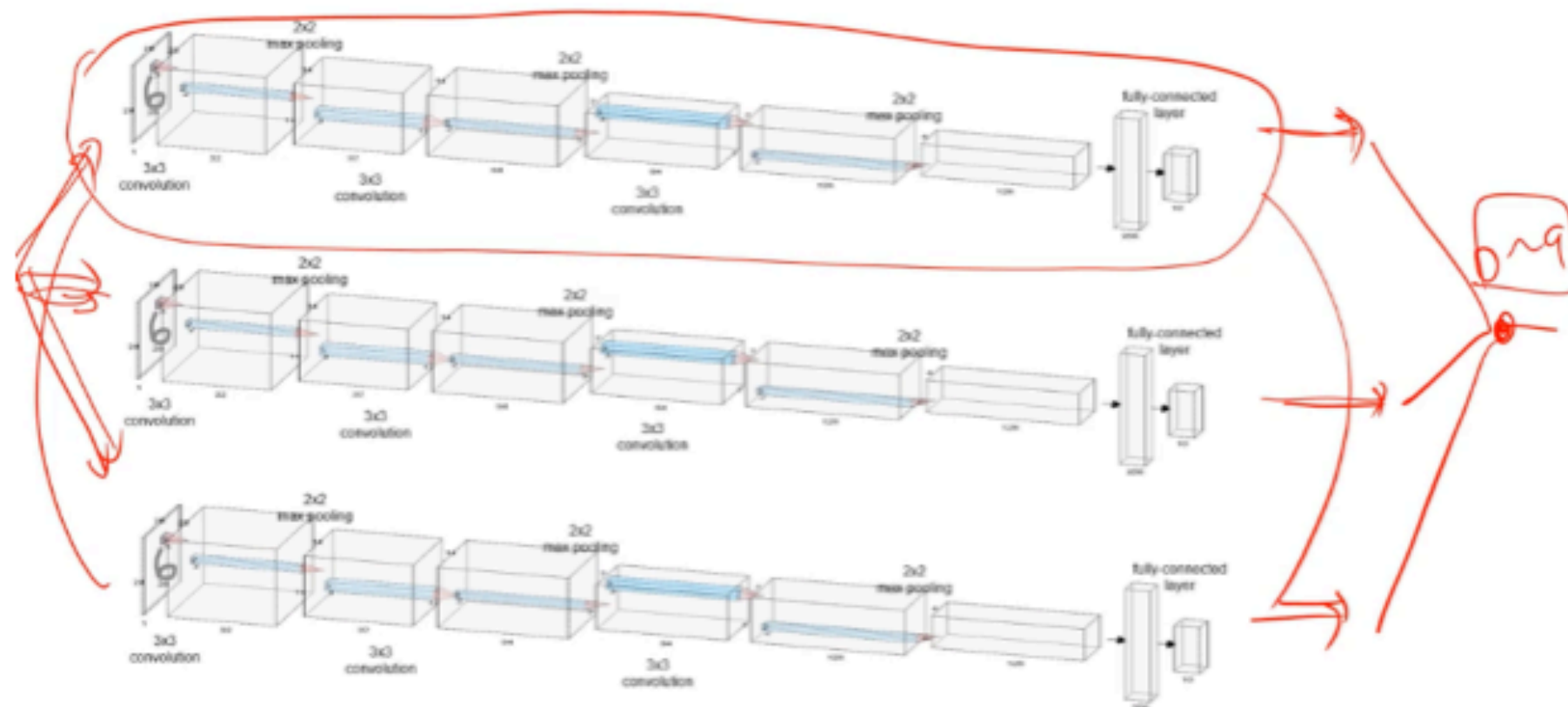


Skills

CNN Model Ensemble

- Model Ensemble
 - 모델 여러개를 사용하여 더 좋은 성능을 이끌어내는 방법
 - 모델에 모두 같은 데이터들을 넣어 주고, 결과의 합을 종합적으로 판단

CNN Model Ensemble with MNIST Data



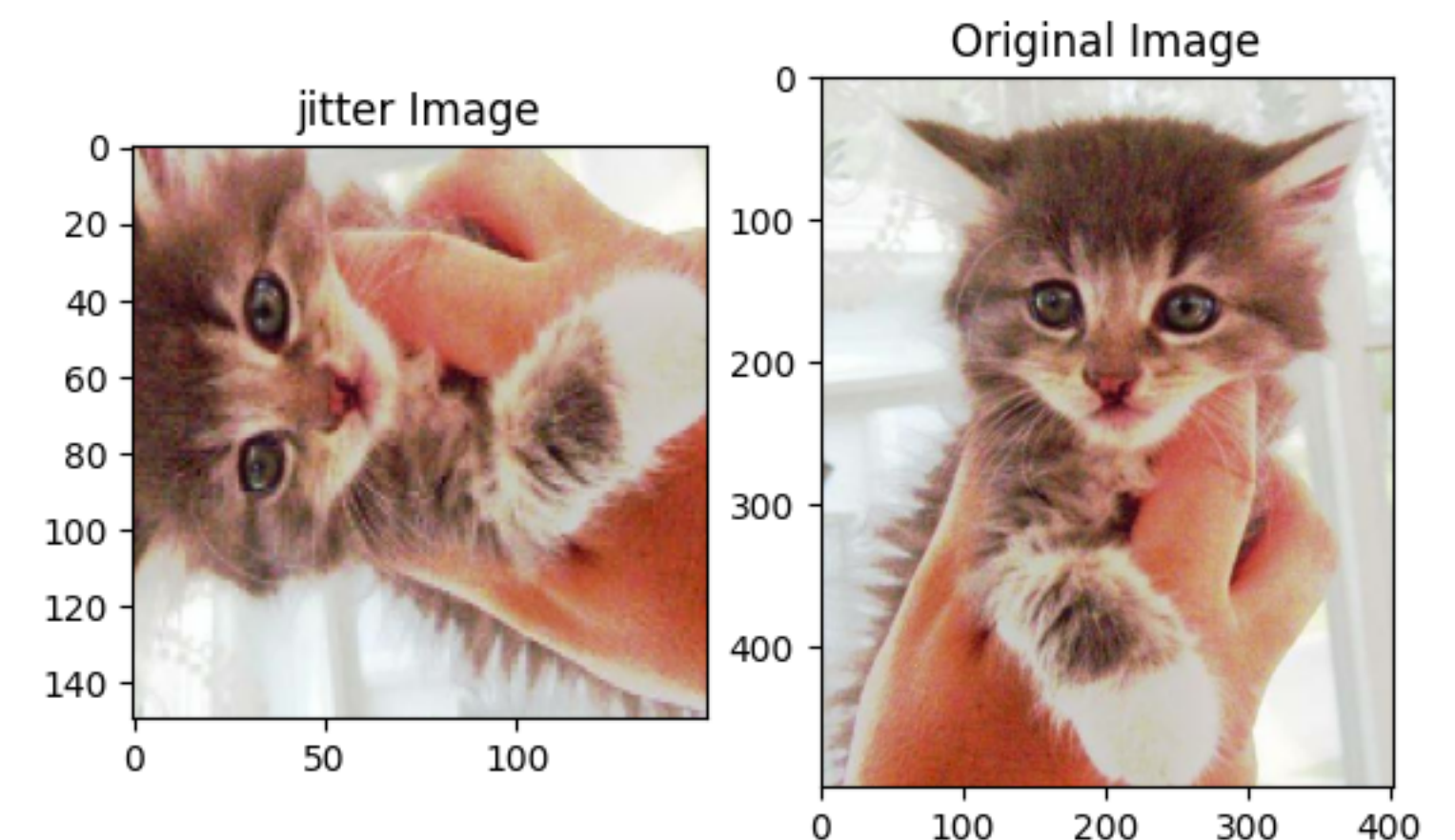
```
models = []  
num_models = 5  
for m in range(num_models):  
    models.append(MNISTModel())
```


Skills

Data Augmentation

- Data Augmentation
 - 사용 가능한 데이터가 적은 경우에 사용, 데이터를 변형하여 데이터 갯수 늘리는 방법
 - Crop, Rotate, Resize 등 여러 기법을 통해 데이터를 늘림

```
def random_jitter(input_image):  
    # resize, random_crop, random_rotation, random_flip 함수들을 이용하여 augmentation을 합니다.  
  
    ## 코드 시작 ##  
    # resizing to 176 x 176 x 3  
    input_image = resize(input_image, 176, 176)  
    # randomly cropping to 150 x 150 x 3  
    input_image = random_crop(input_image)  
    # randomly rotation  
    input_image = random_rotation(input_image)  
    # randomly mirroring  
    input_image = tf.image.random_flip_left_right(input_image)  
    ## 코드 종료 ##  
  
    return input_image
```





Thank you for listening

**Conv Layer, Convnet Max pooling
Cat-Dog Classifier**

GDSC Hanyang ML/DL : Basic, Jaeseung Lee