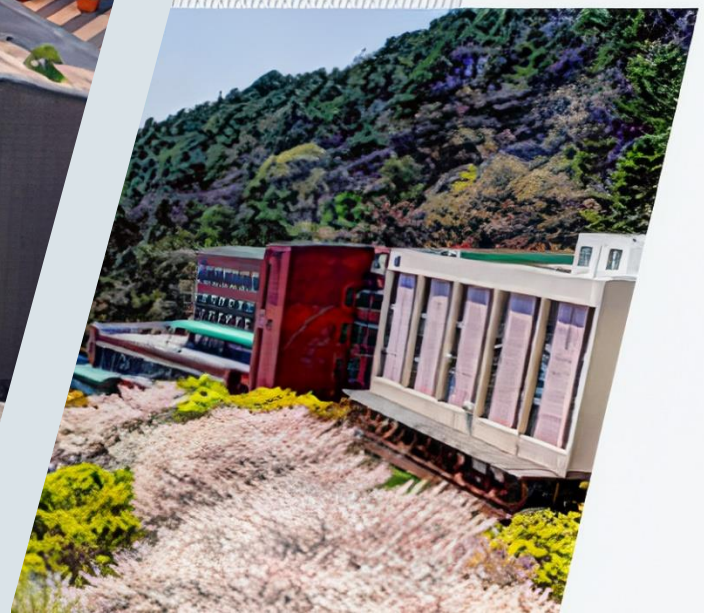
Dong-A Univ. (ISPL)

동아대학교 DONG-A UNIVERSITY

[실습] Convolutional Neural Network

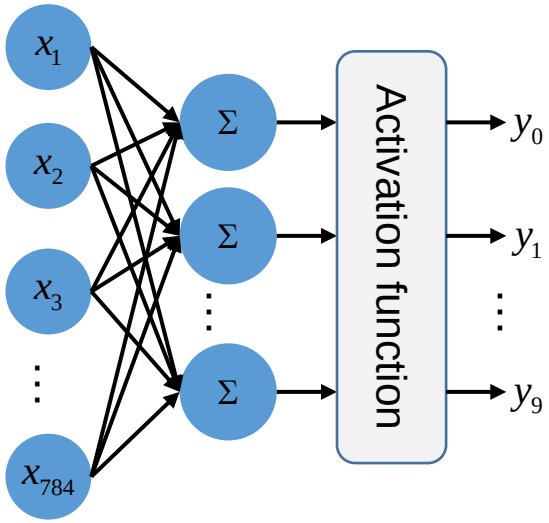
컴퓨터AI공학부 컴퓨터공학과

2025년 1학기 머신러닝



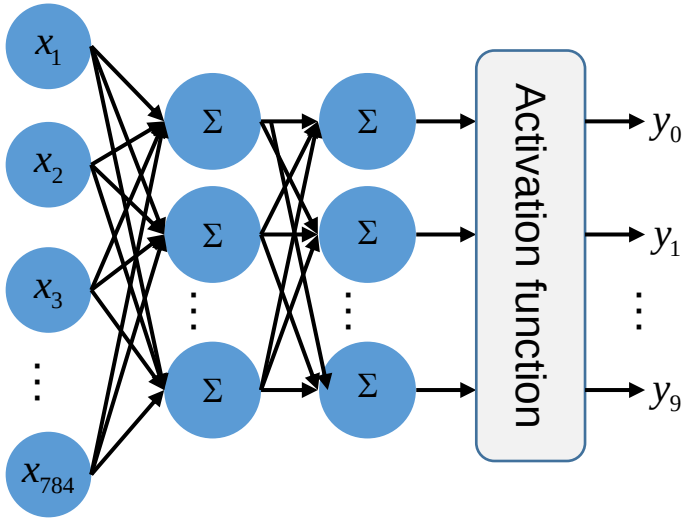
[Remind] Classification: MNIST

Single Layer Perceptron 실습



- ✓ Layer1 (Input: 784, Out: 10)
- ✓ Activation function: Softmax
- ✓ Loss function: Cross Entropy

Multi Layer Perceptron 실습



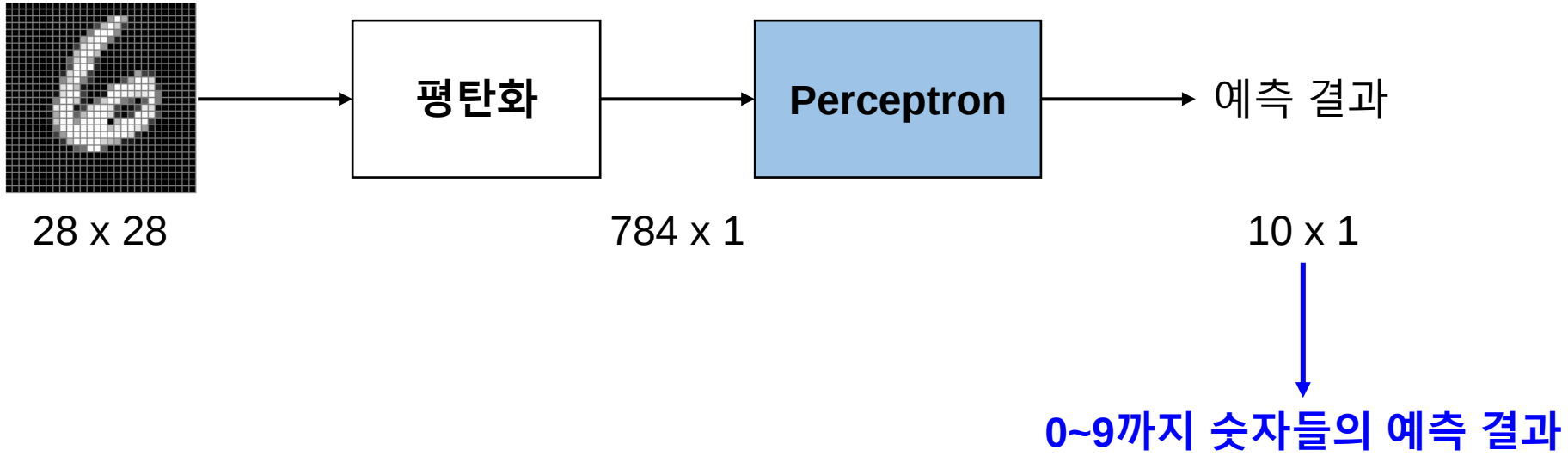
- ✓ Layer1 (Input: 784, Out: 100)
- ✓ Layer2 (Input: 100, Out: 10)
- ✓ Activation function: Softmax, Sigmoid
- ✓ Loss function: Cross Entropy

[Remind] Classification: MNIST

- **MNIST 데이터베이스 (Modified National Institute of Standards and Technology)**
  - 손으로 쓴 숫자들로 이루어진 대형 데이터베이스
  - Train dataset 60,000개, Test dataset 10,000개로 구성됨

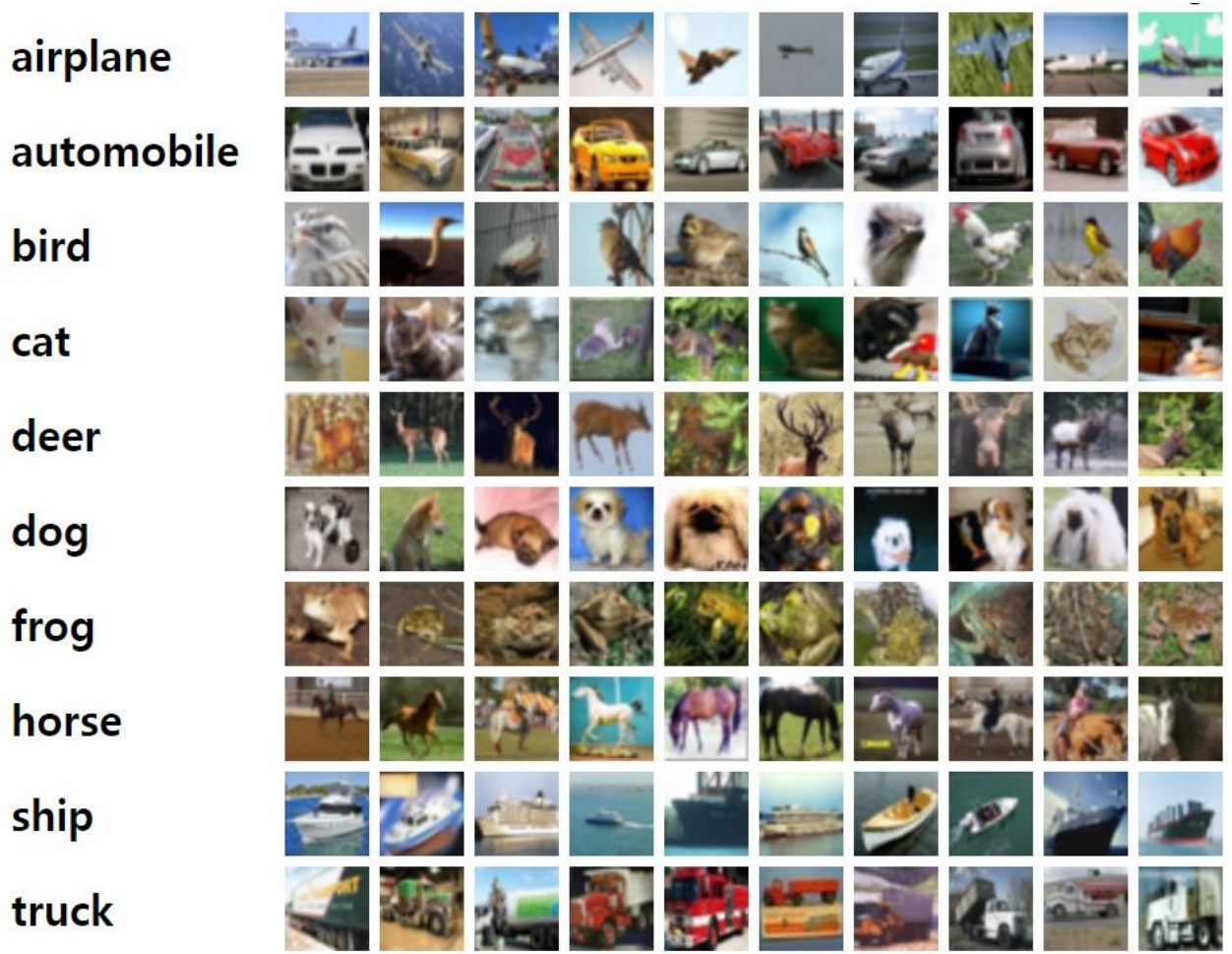
[Remind] Classification: MNIST

- **MNIST 데이터셋의 perceptron 입력 방법**
  - Perceptron의 각 노드는 한번에 1개의 값을 입력 받을 수 있음
  - 따라서 2D 형태 이미지의 전처리가 필요함 → 평탄화



[Remind] Classification: CIFAR 10

- **CIFAR10 dataset 형상**
  - 32x32x3 (RGB) 이미지, 10개의 클래스
  - Train: 50,000개, Test: 10,000개



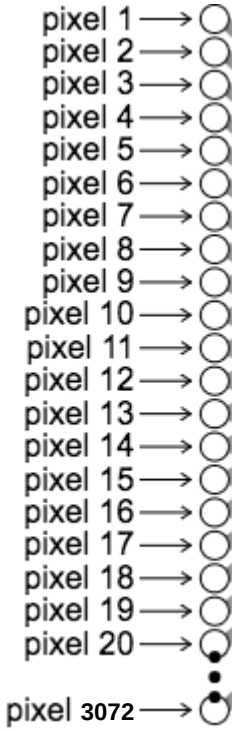
[Remind] Classification: CIFAR 10

- **CIFAR10 dataset 형상**
  - 32x32x3 (RGB) 이미지, 10개의 클래스
  - Train: 50,000개, Test: 10,000개

CIFAR10 input image

(32x32x3)

평탄화



(32x 32x 3) x 1

Perceptron

예측 결과

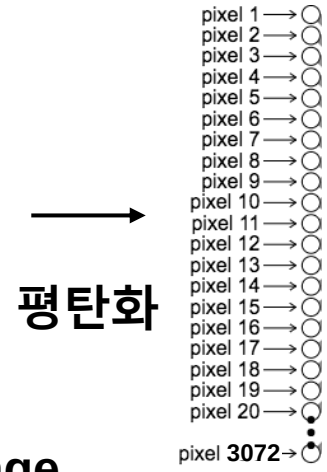
10 x 1

[Remind] Classification: CIFAR 10

- 실습
  - Layer: 5개
  - Node: 100개
  - Activation function: ReLU

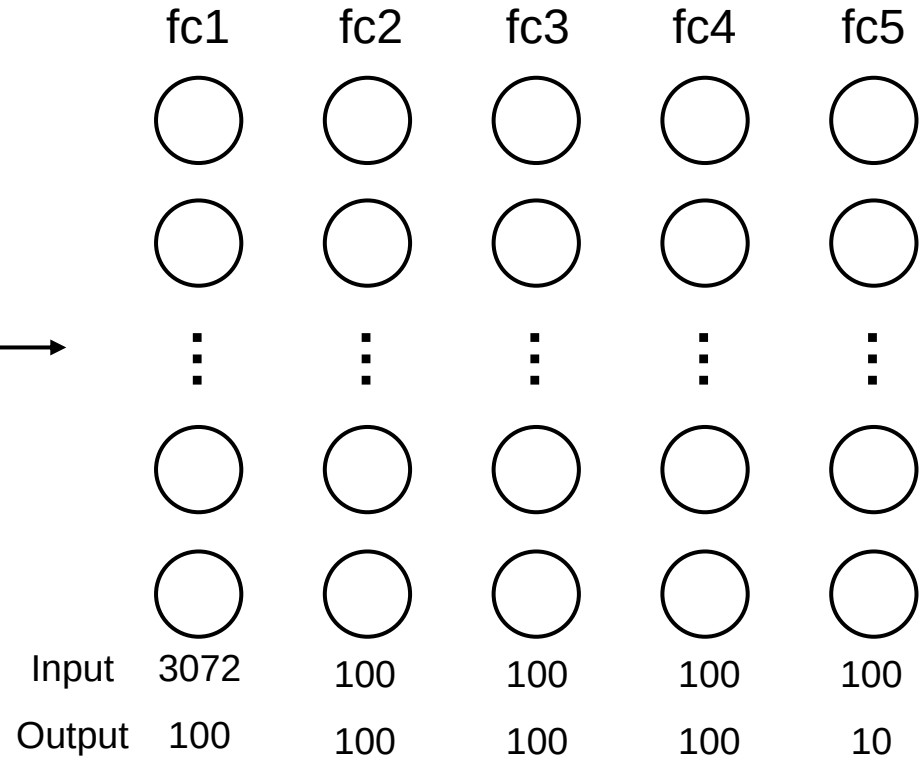
CIFAR10 input image

(32x32x3)



평탄화

(32x 32x 3) x 1



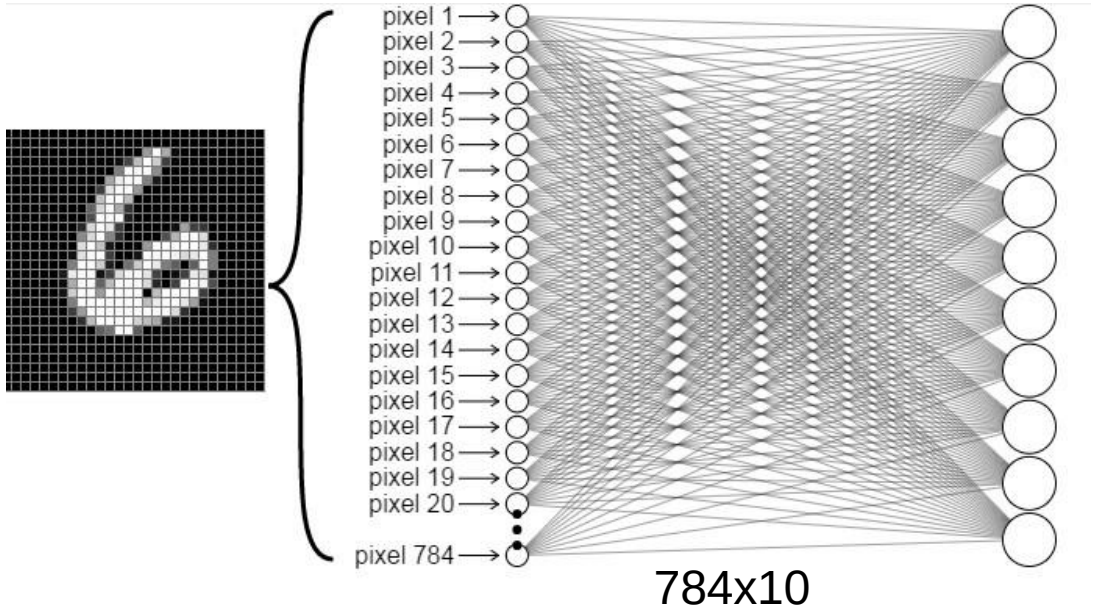
10 x 1

예측 결과 **48%**

Accuracy: 0.488200

[Remind] Convolutional Neural Networks

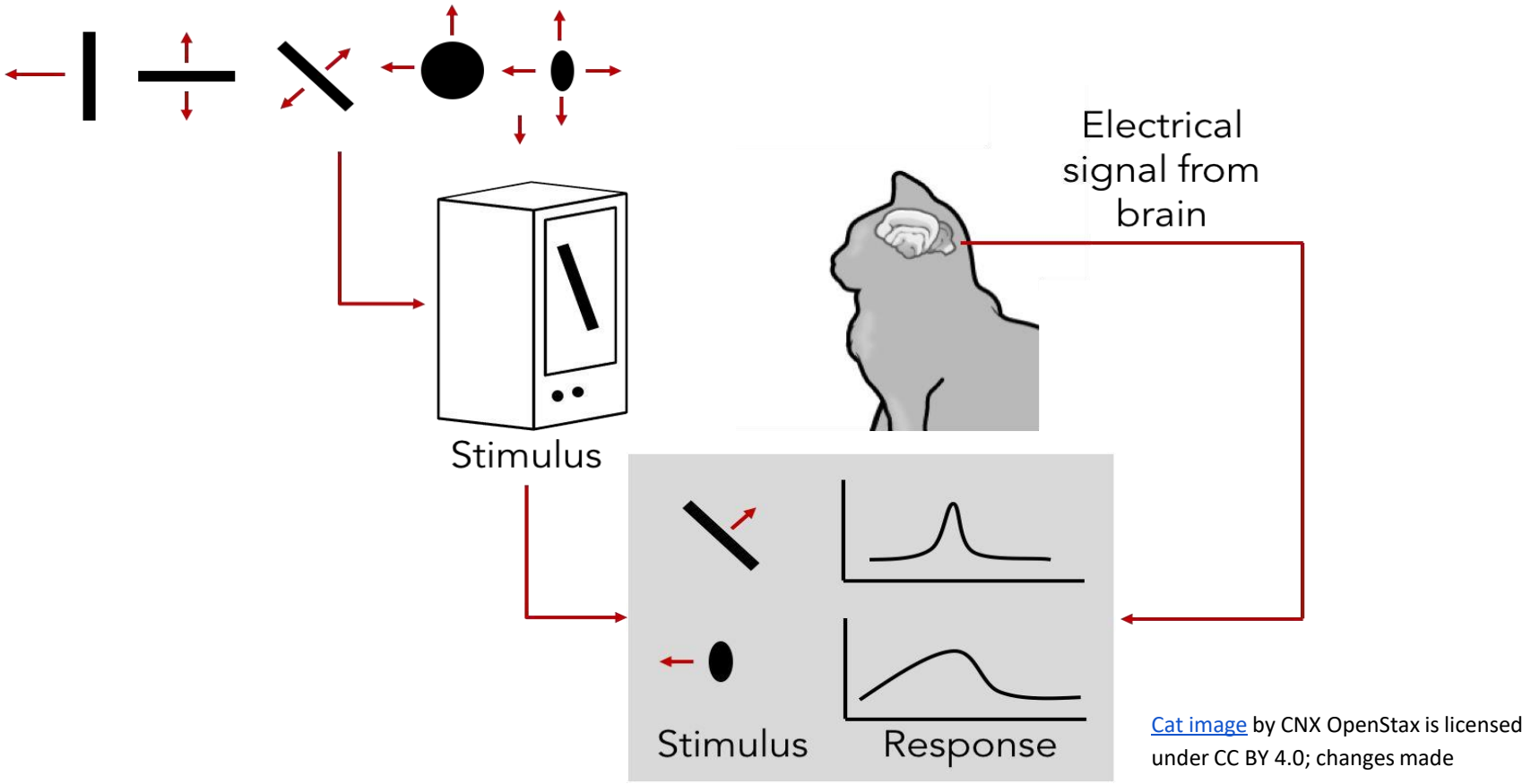
- **Perceptron 모델의 문제점**
  - 평탄화된 이미지를 입력으로 받아 **공간적 (형상) 정보가 사라짐**



Fully connected layer
(#weights: 7,840)

[Remind] Convolutional Neural Networks

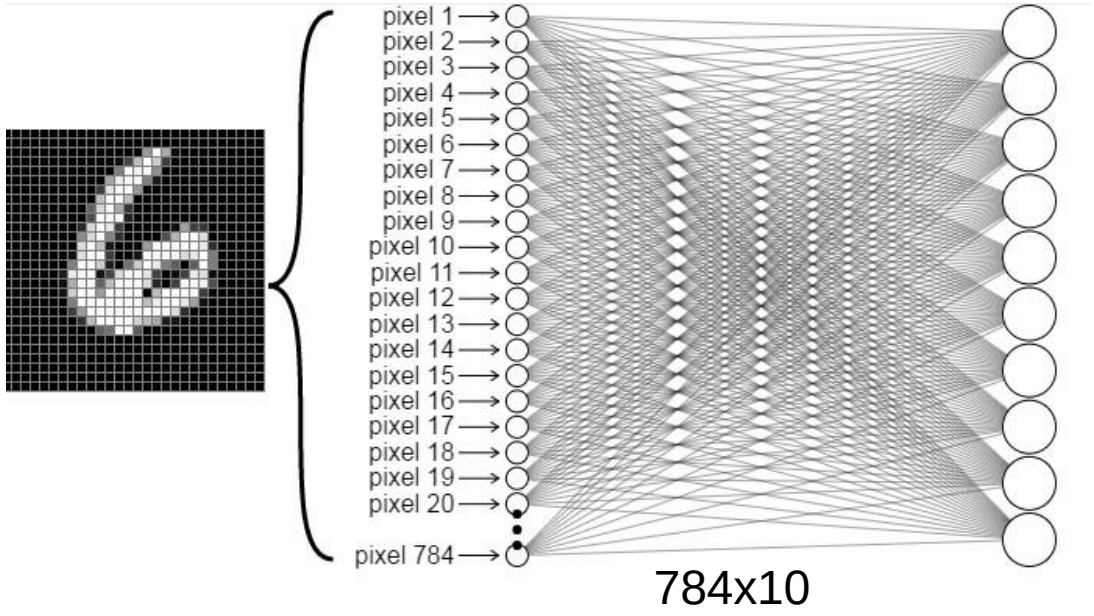
- 이미지의 특성에 따라 뉴런이 다르게 활성화 됨



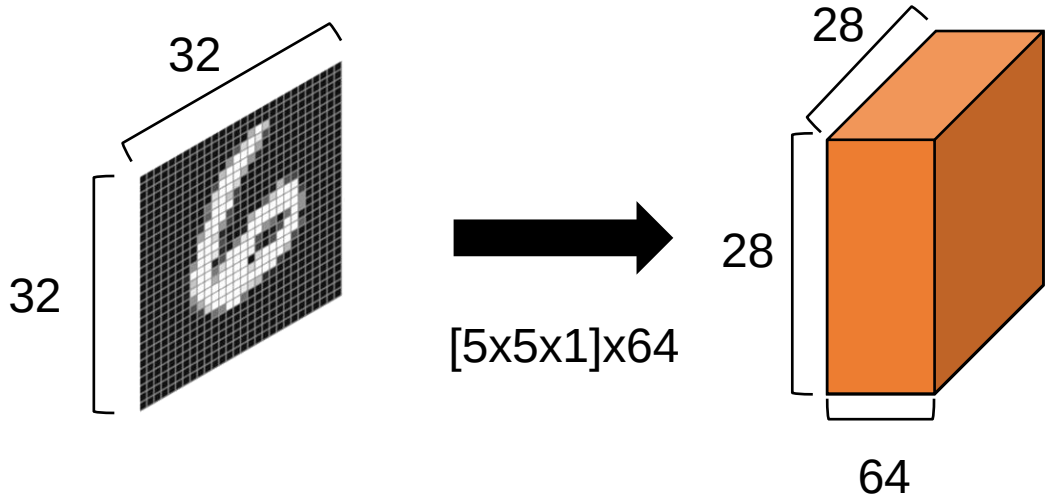
Cat image by CNX OpenStax is licensed under CC BY 4.0; changes made

[Remind] Convolutional Neural Networks

- **Perceptron 모델의 문제점**
  - 평탄화된 이미지를 입력으로 받아 공간적 (형상) 정보가 사라짐
  - CNN에 비해 필요한 weight parameter의 개수가 많음



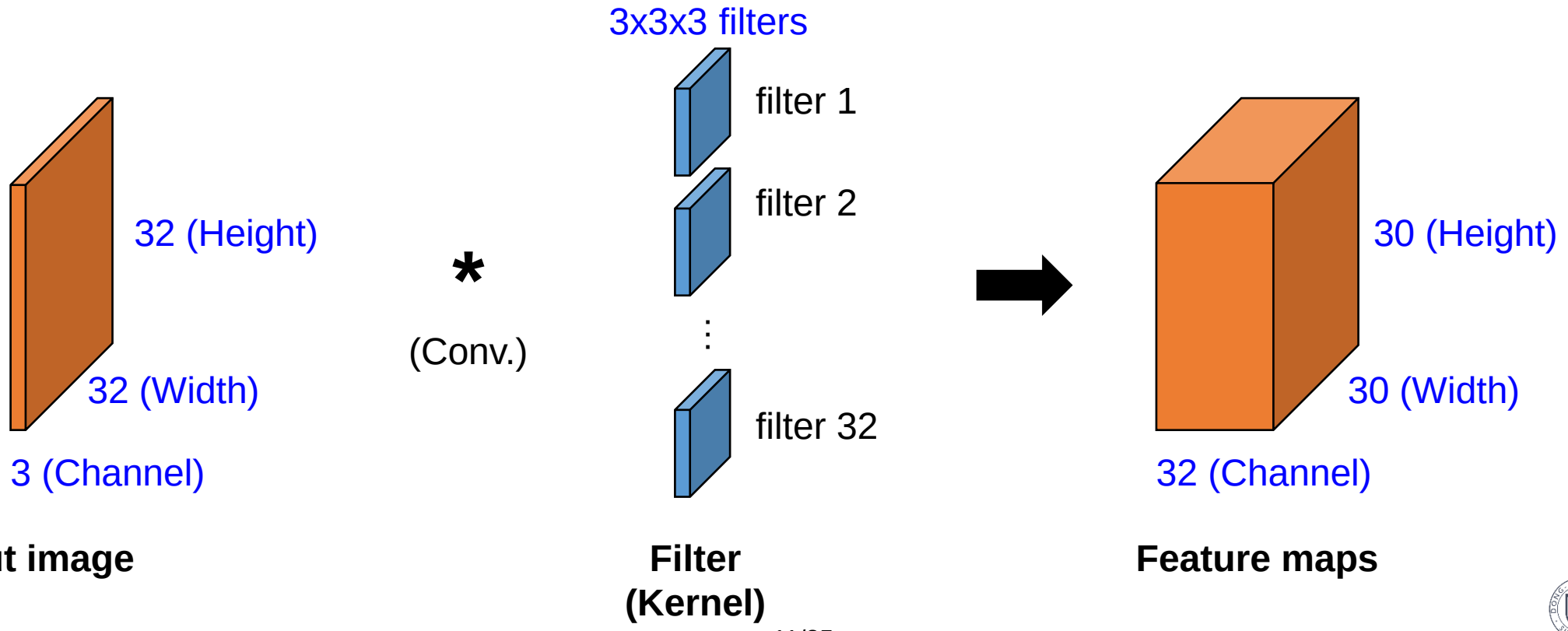
Fully connected layer
(#weights: 7,840)



5x5 Convolution layer
(#weights: 1,600)

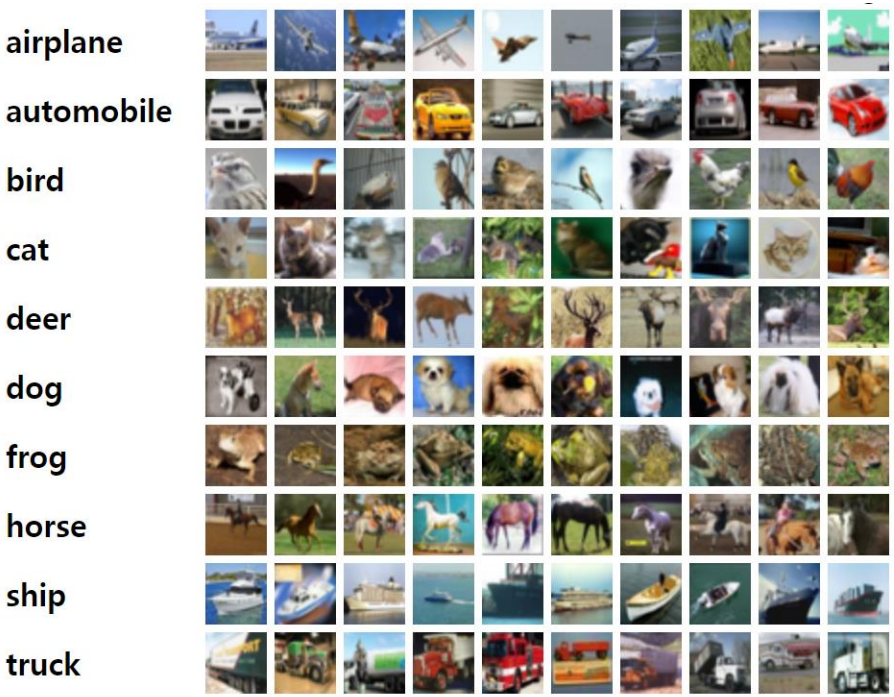
[Remind] Convolutional Neural Networks

- **CNN 모델 개요**
  - 합성곱 연산 (Convolution)을 수행해 이미지의 특징 (Feature) 추출
  - 이미지를 입력으로 받는 분야에서 높은 성능을 보임 (Ex. Image classification, Super-resolution, Denoising)

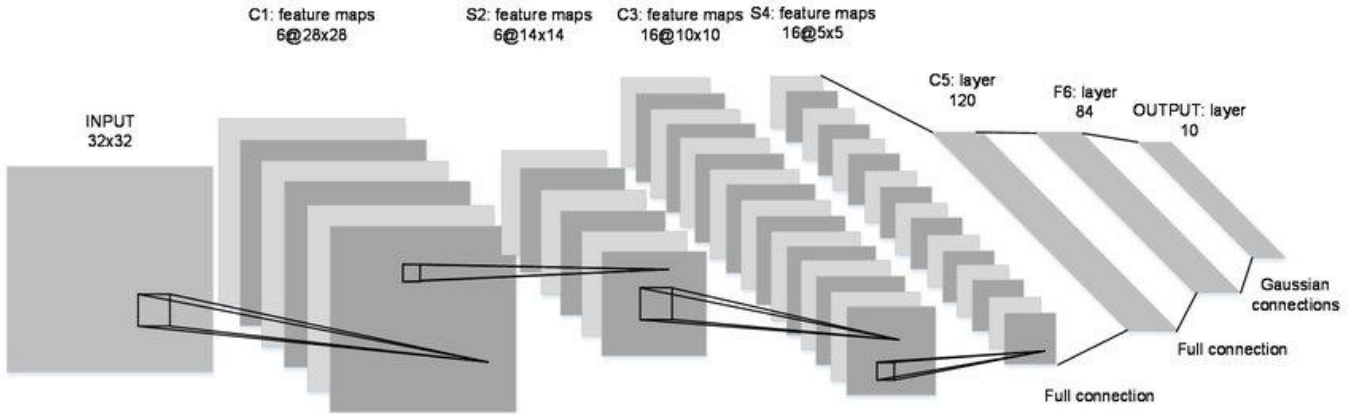


[실습] Convolutional Neural Networks

- **CIFAR10 dataset 형상**
  - 32x32x3 (RGB) 이미지, 10개의 클래스
  - Train: 50,000개, Test: 10,000개



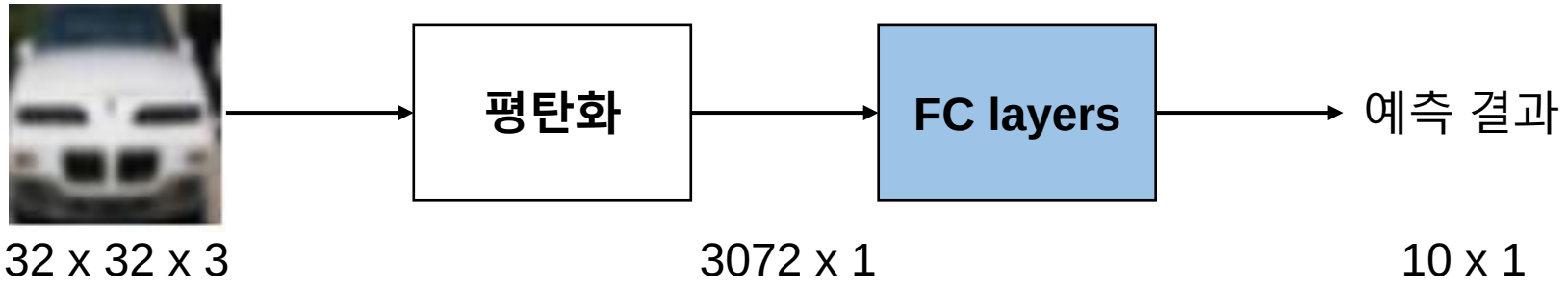
CIFAR-10 dataset 예시

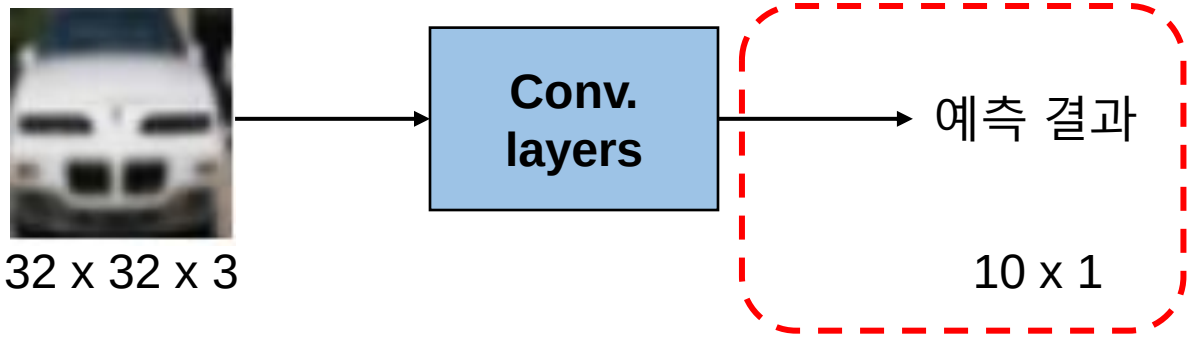


LeNet-5 신경망 구조

[실습] Convolutional Neural Networks: CIFAR10 classification

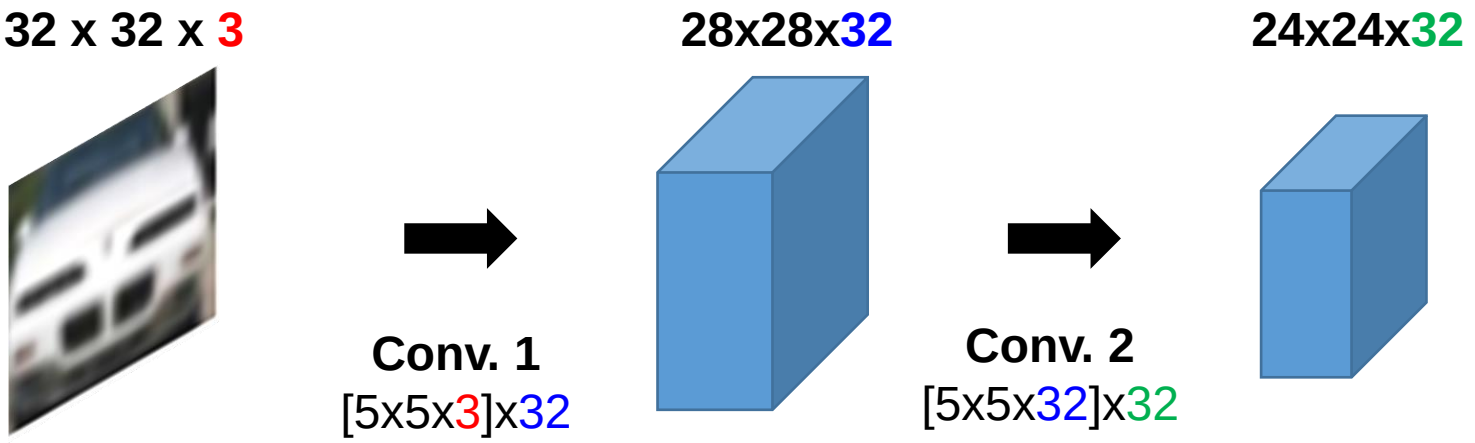
- 입출력 구조 확인





[실습] Convolutional Neural Networks: CIFAR10 classification

- 입출력 구조 확인

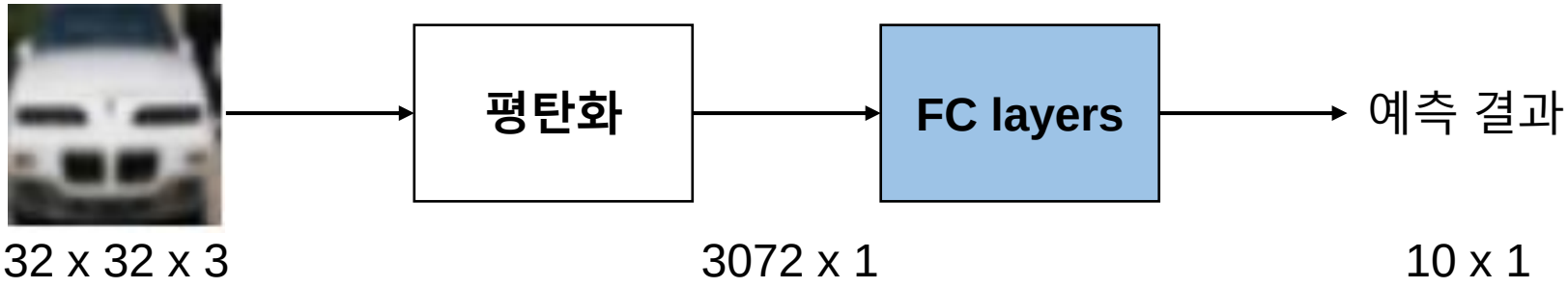


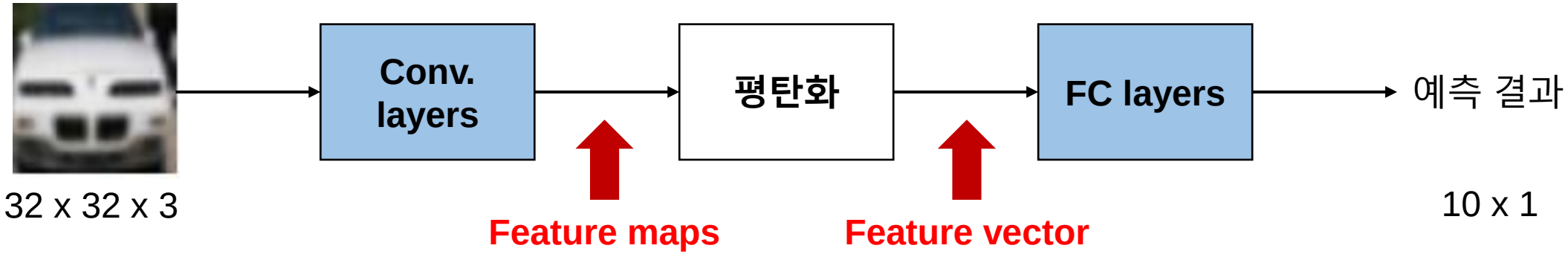
출력의 형태가 cube 이므로
예측 결과를 확인 할 수 없음

CNN model 예시

[실습] Convolutional Neural Networks: CIFAR10 classification

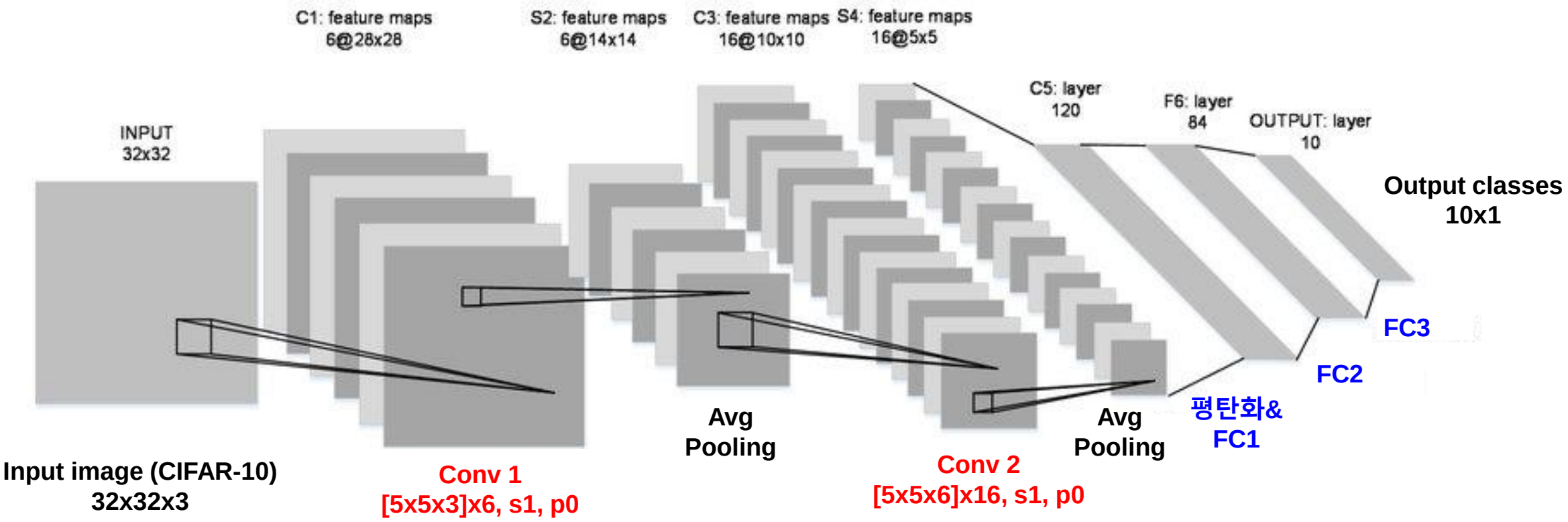
- 입출력 구조 확인





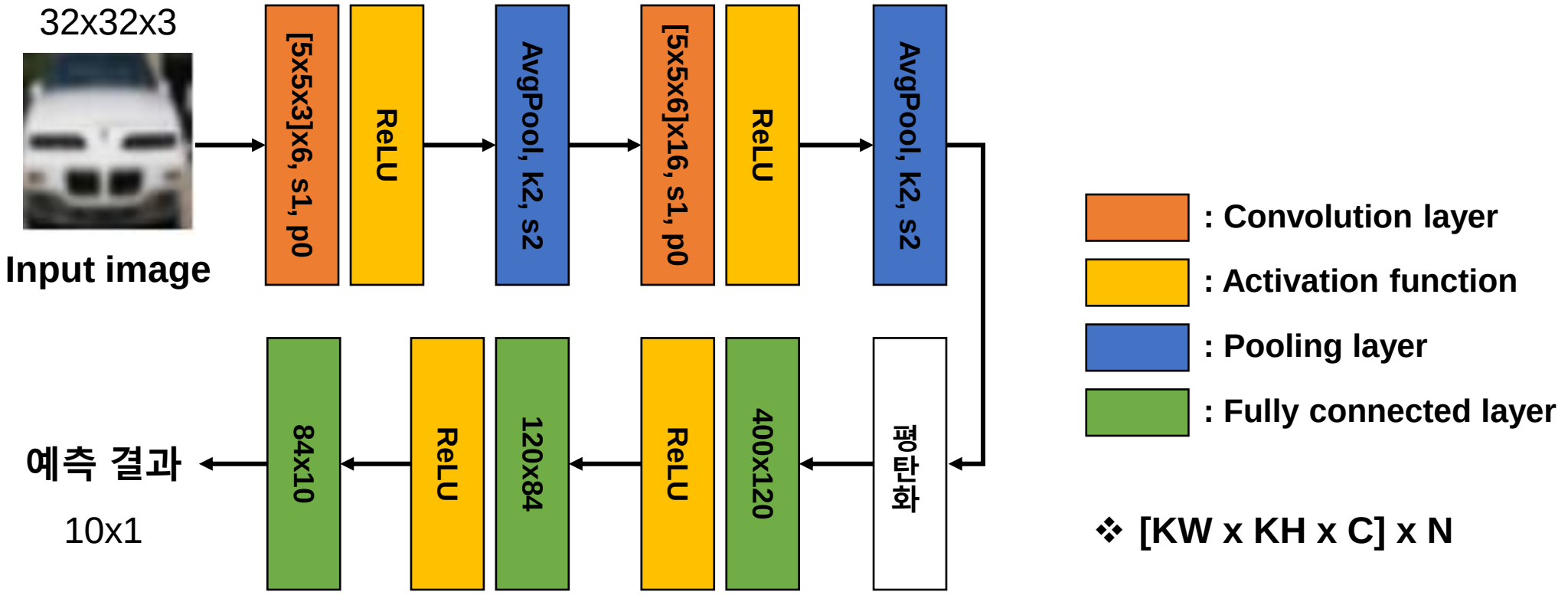
[실습] Convolutional Neural Networks: CIFAR10 classification

- **LeNet-5 신경망 구조**
  - 2개의 convolutional layer, 3개의 fully connected layer로 구성



[실습] Convolutional Neural Networks: CIFAR10 classification

- **LeNet-5 모델 구조 작성 참고사항**
  - Filter size: 5x5, Stride: 1, Padding: 0



LeNet-5 구조

[실습] Convolutional Neural Networks: CIFAR10 classification

- **LeNet-5 모델 구조 작성 참고사항**
  - 참고자료: https://pytorch.org/docs/stable/nn.html

CONV2D

CLASS `torch.nn.Conv2d(in_channels, out_channels, kernel_size, stride=1, padding=0, dilation=1, groups=1, bias=True, padding_mode='zeros', device=None, dtype=None)` [SOURCE]

RELU

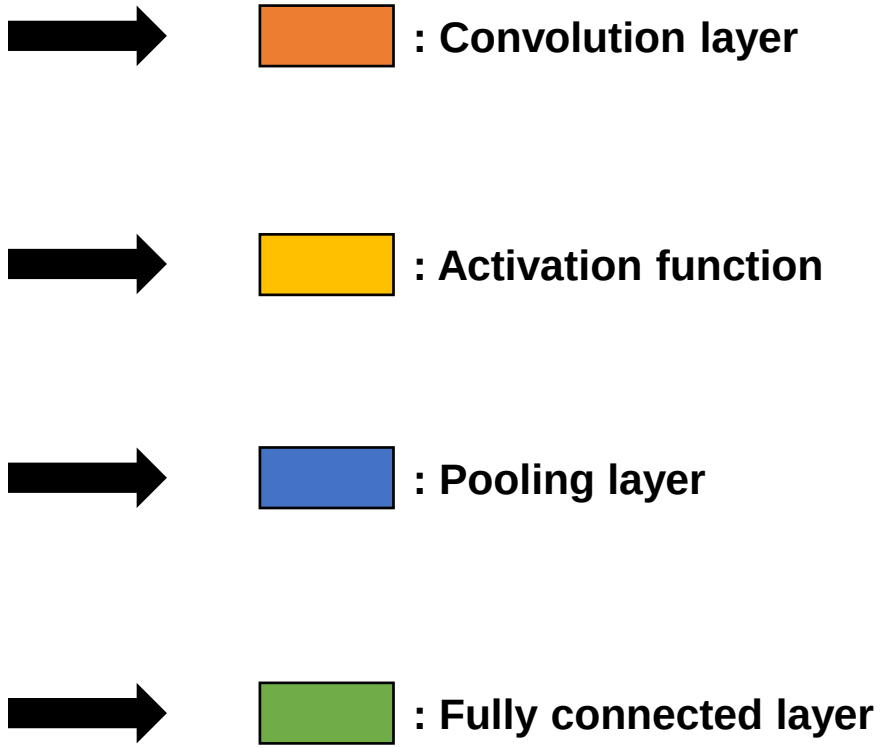
CLASS `torch.nn.ReLU(inplace=False)` [SOURCE]

MAXPOOL2D

CLASS `torch.nn.MaxPool2d(kernel_size, stride=None, padding=0, dilation=1, return_i ceil_mode=False)` [SOURCE]

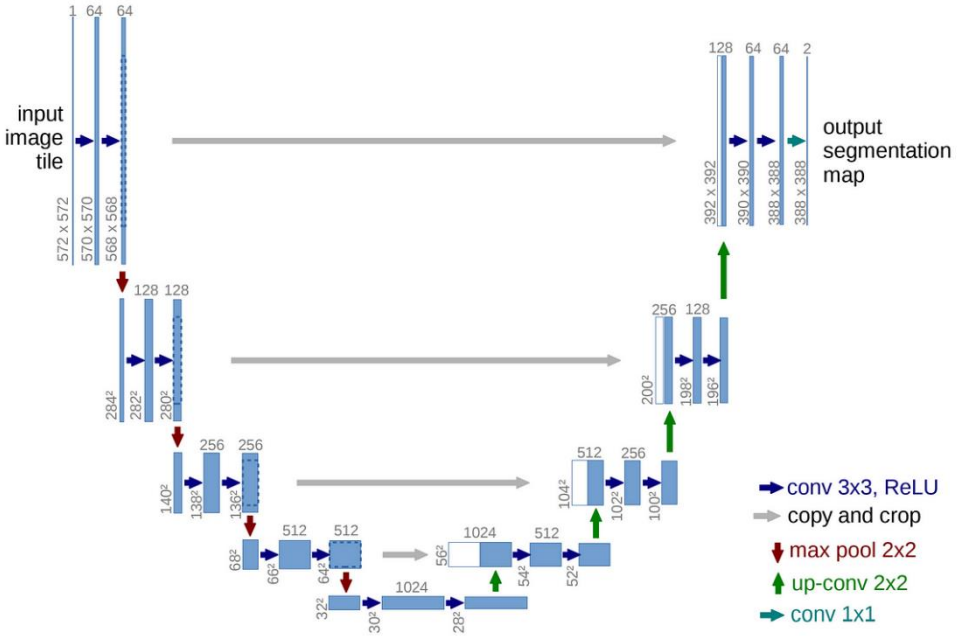
LINEAR

CLASS `torch.nn.Linear(in_features, out_features, bias=True, device=None, dtype=None)` [SOURCE]

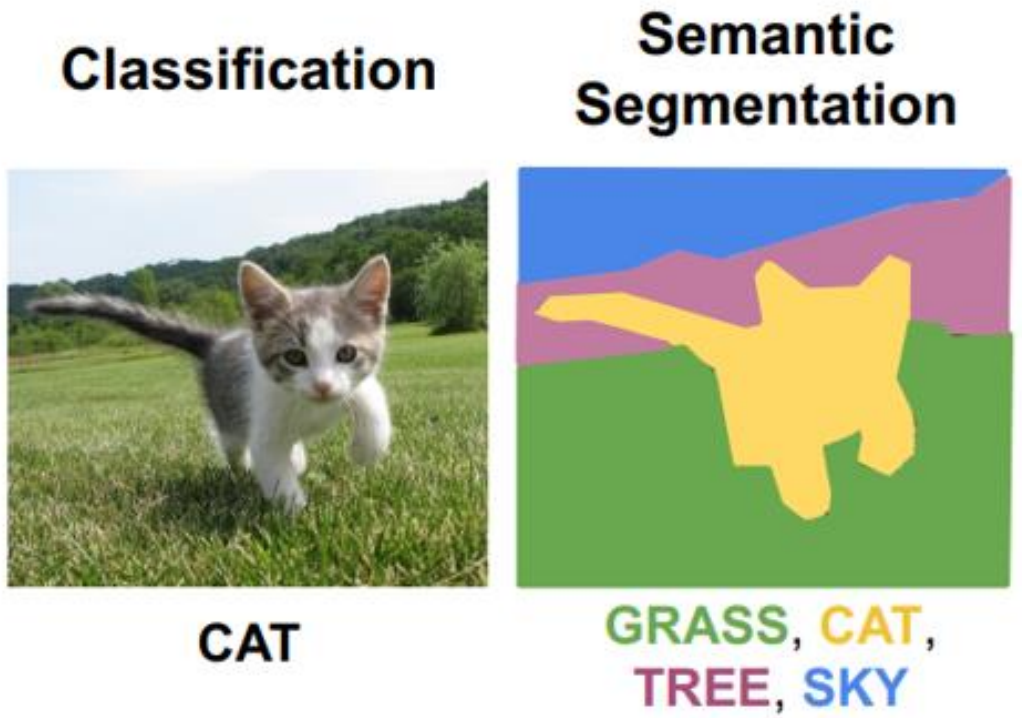


CIFAR-10 분류 실습: U-Net을 이용한 분류

- **U-Net**
  - 2015년 Semantic Segmentation 기법으로 등장한 모델



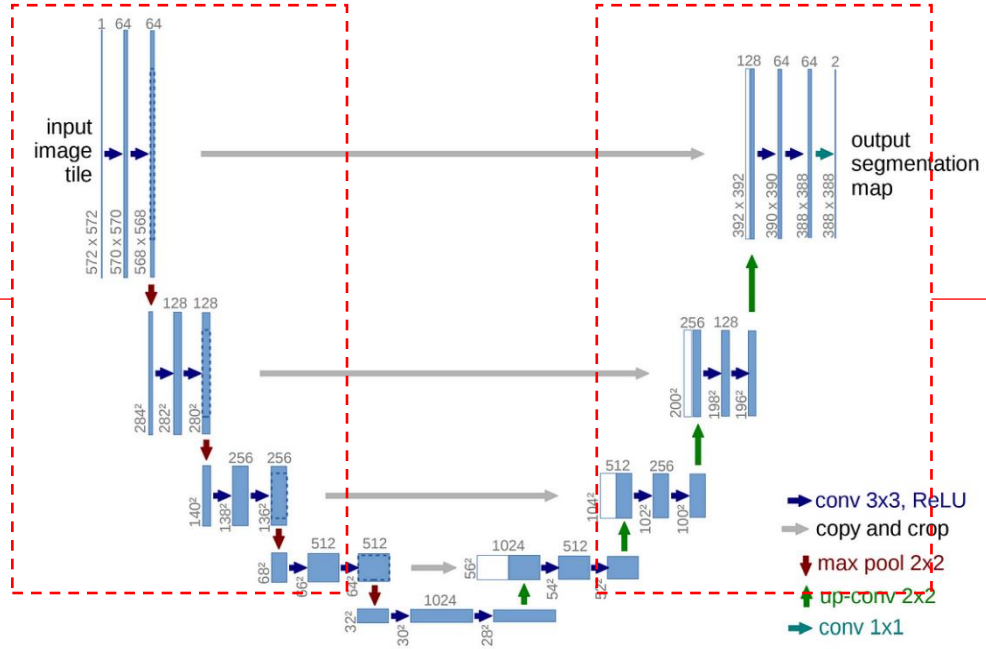
U-net network architecture



CIFAR-10 분류 실습: U-Net을 이용한 분류

- **U-Net**
  - 2015년 Semantic Segmentation 기법으로 등장한 모델

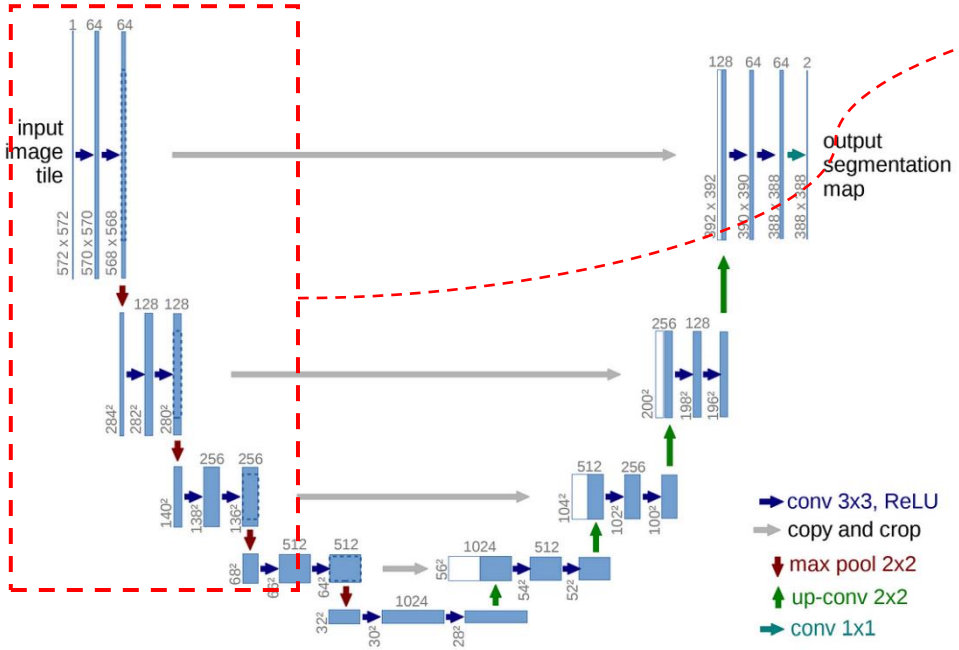
입력 이미지로부터 특징 학습

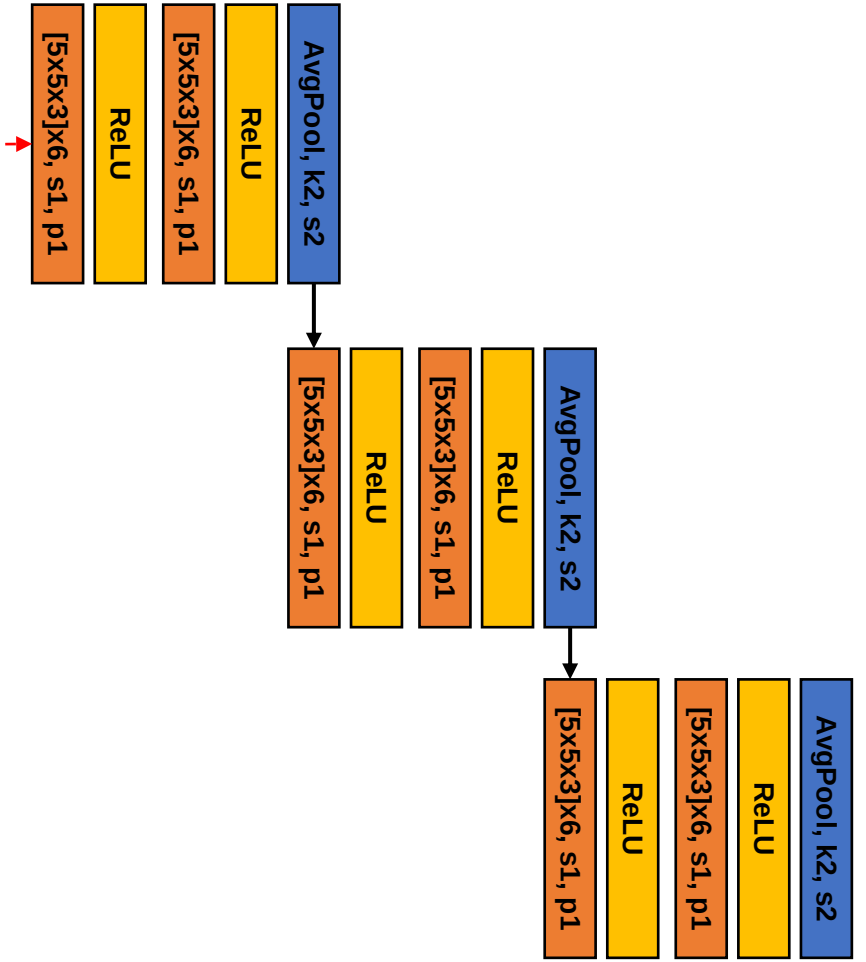


학습된 Feature map으로부터 Segmentation map 생성

CIFAR-10 분류 실습: U-Net을 이용한 분류

- **U-Net**
  - 2015년 Semantic Segmentation 기법으로 등장한 모델

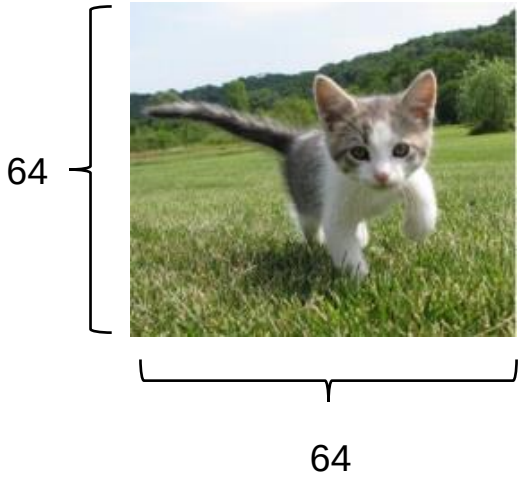


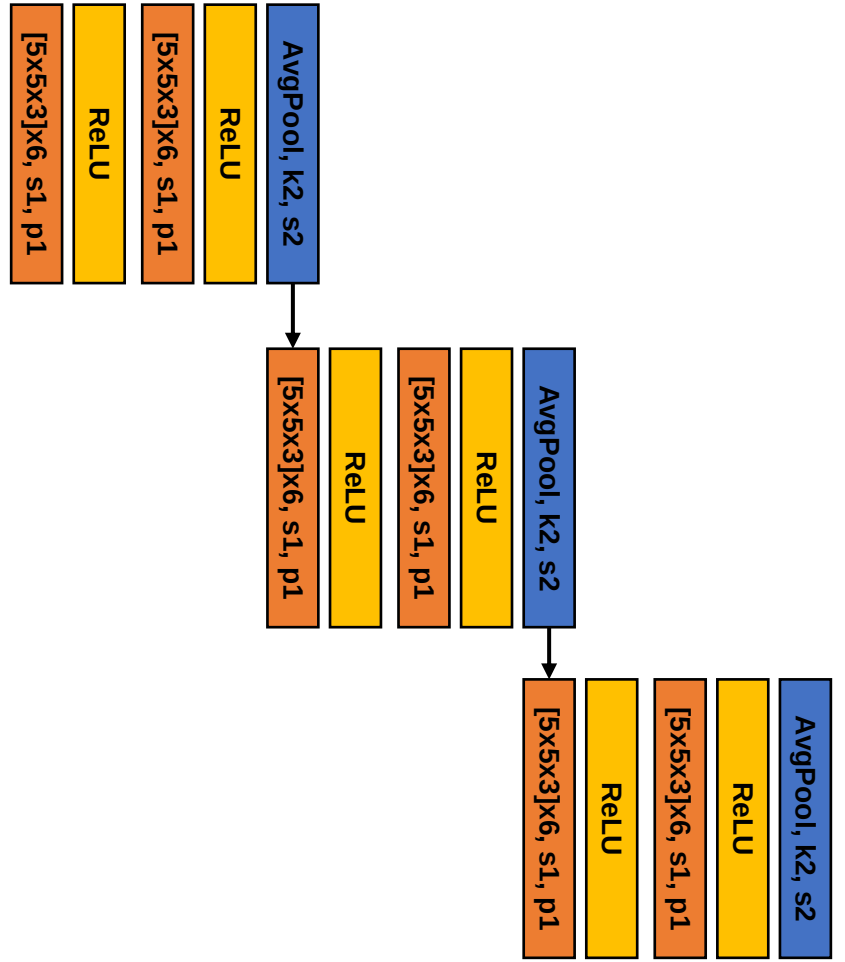


U-net network architecture

CIFAR-10 분류 실습: U-Net을 이용한 분류

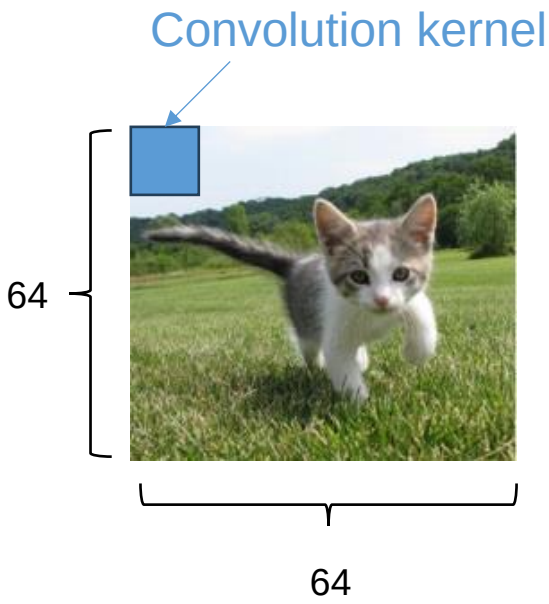
- **U-Net**
  - 2015년 Semantic Segmentation 기법으로 등장한 모델

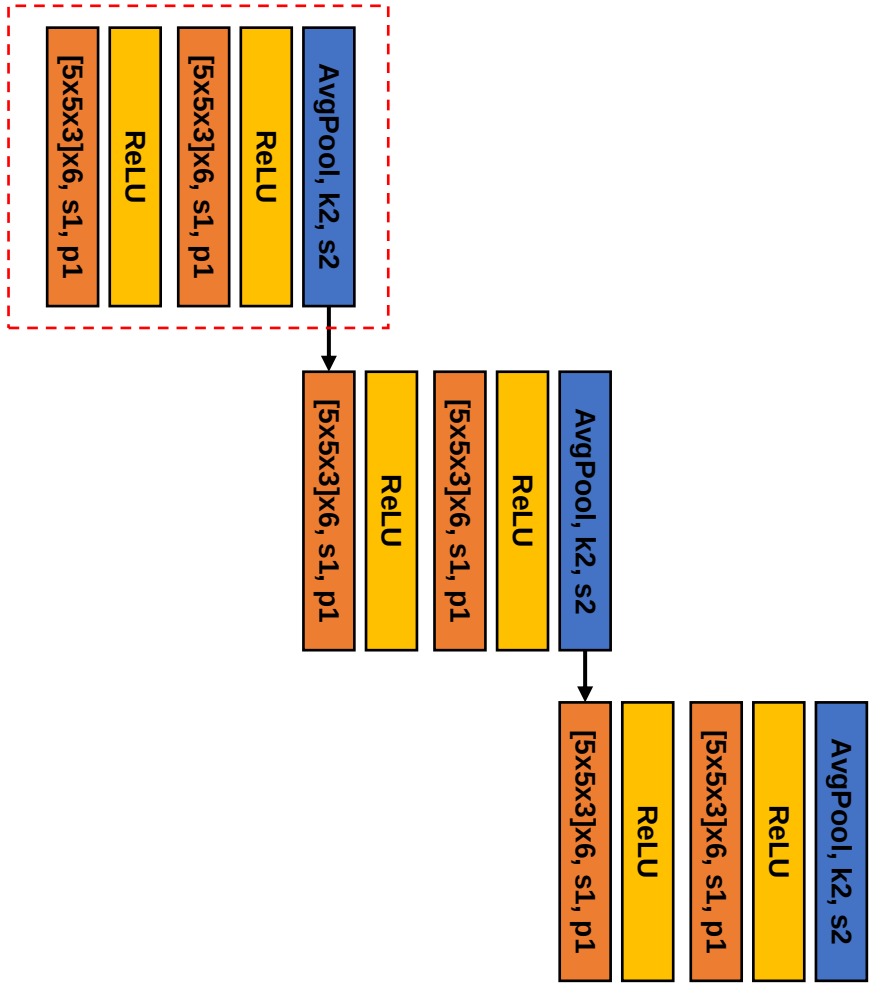




CIFAR-10 분류 실습: U-Net을 이용한 분류

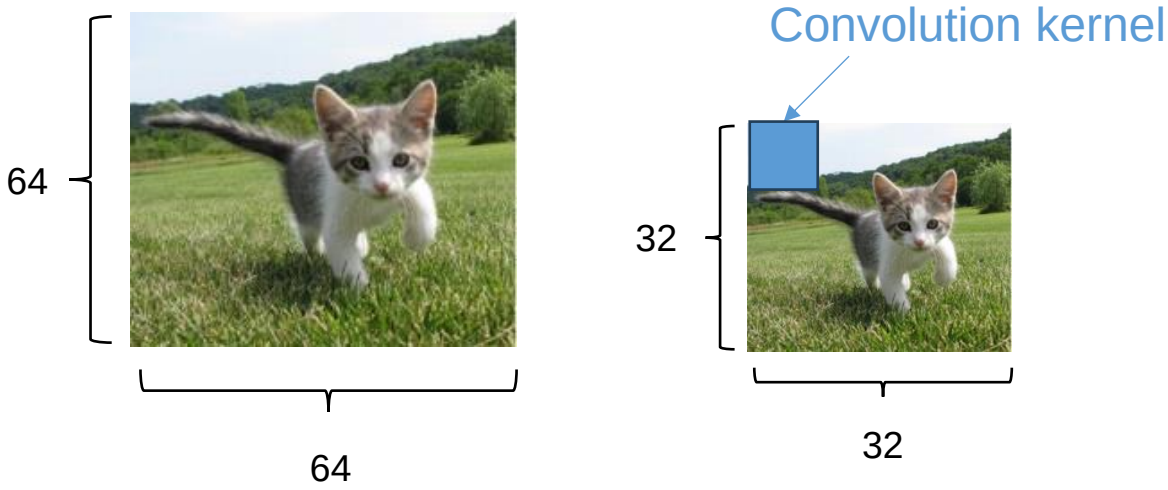
- **U-Net**
  - 2015년 Semantic Segmentation 기법으로 등장한 모델

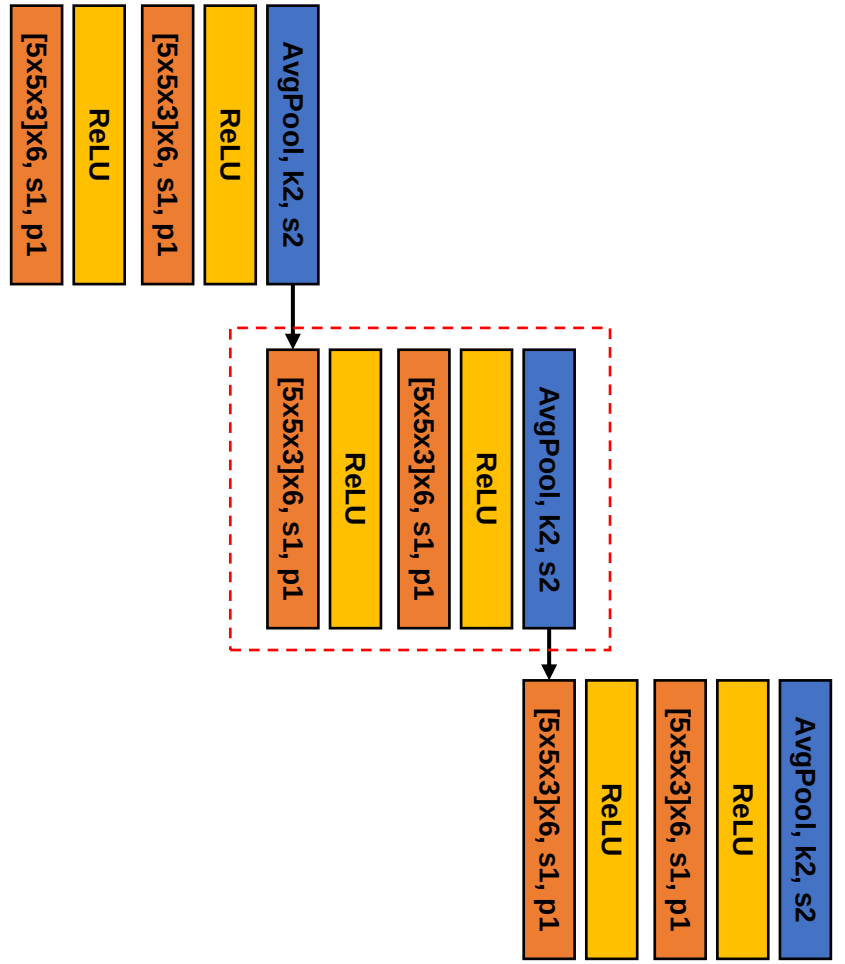




CIFAR-10 분류 실습: U-Net을 이용한 분류

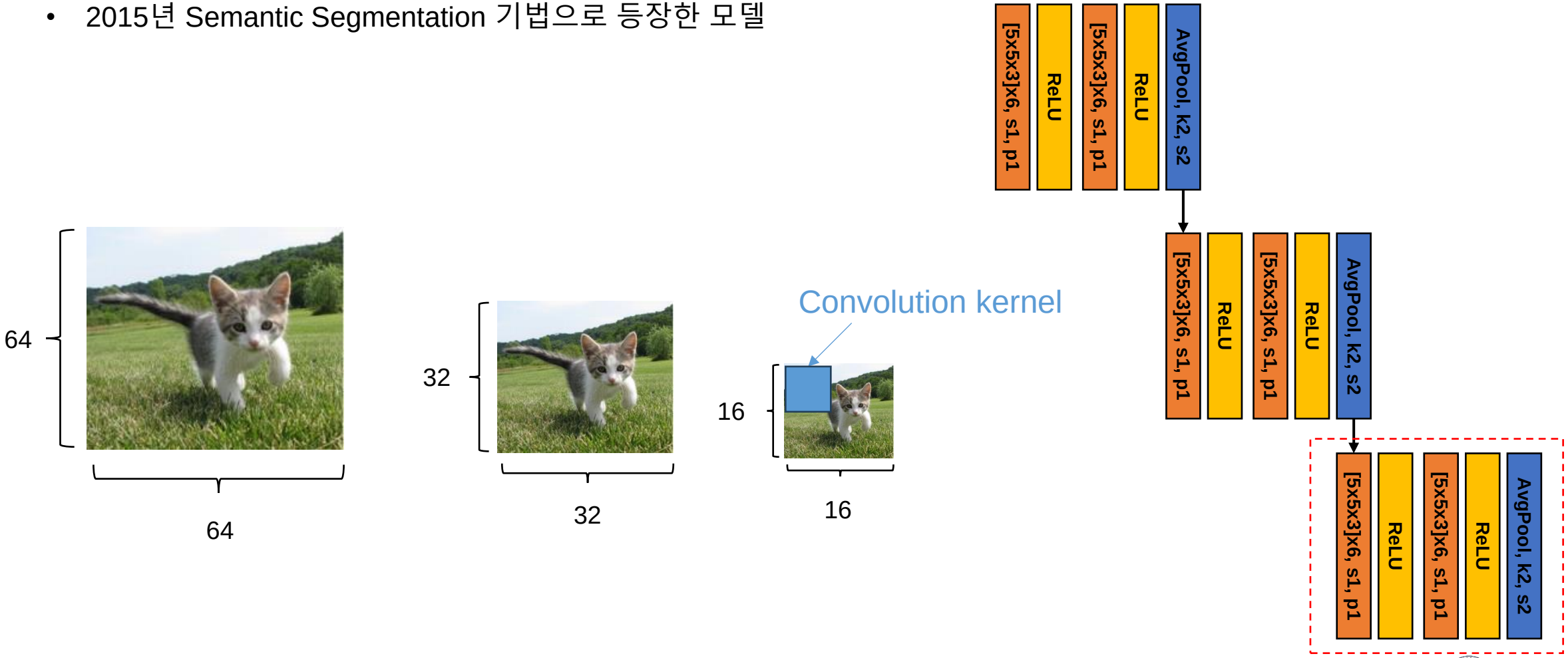
- **U-Net**
  - 2015년 Semantic Segmentation 기법으로 등장한 모델





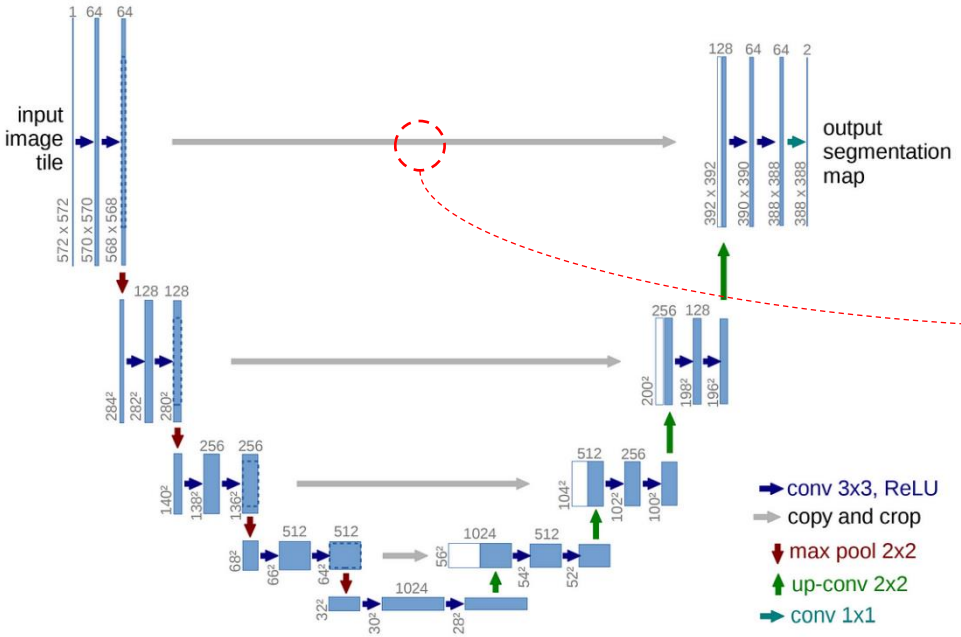
CIFAR-10 분류 실습: U-Net을 이용한 분류

- **U-Net**
  - 2015년 Semantic Segmentation 기법으로 등장한 모델



CIFAR-10 분류 실습: U-Net을 이용한 분류

- **U-Net**
  - 2015년 Semantic Segmentation 기법으로 등장한 모델

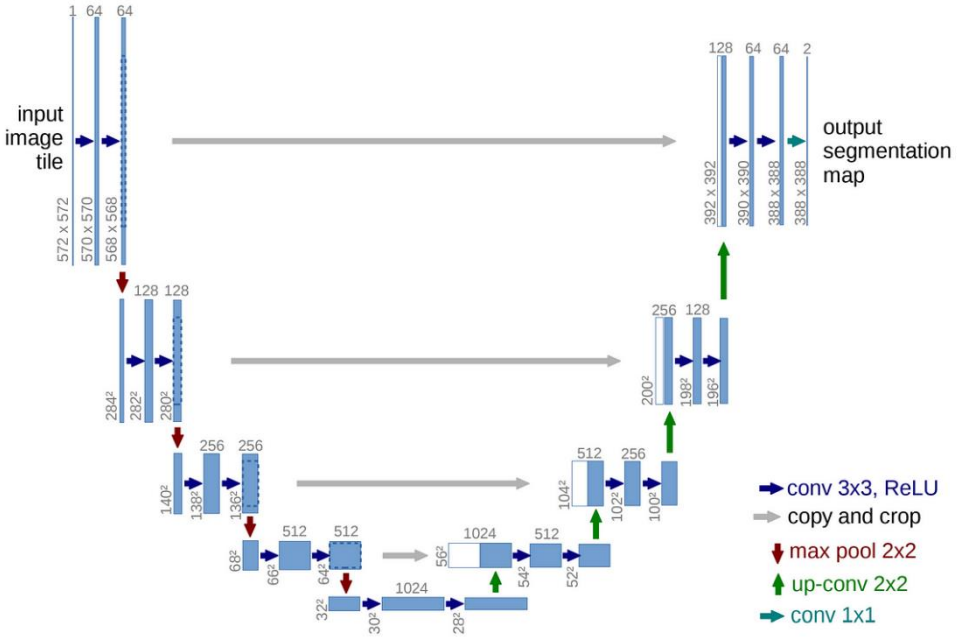


U-net network architecture

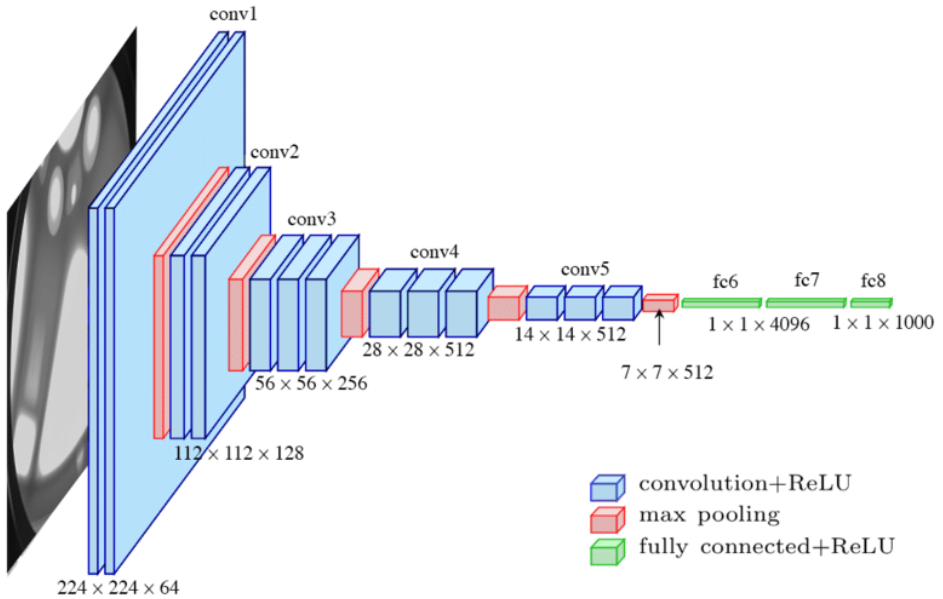
Skip connection을 통해 학습된 feature map을 전달하여 정교한 Segmentation map 생성 가능

CIFAR-10 분류 실습: U-Net을 이용한 분류

- **U-Net**
  - 2015년 Semantic Segmentation 기법으로 등장한 모델
  - Auto encoder 구조를 이용한 입력 이미지 정보 압축 가능



U-net network architecture

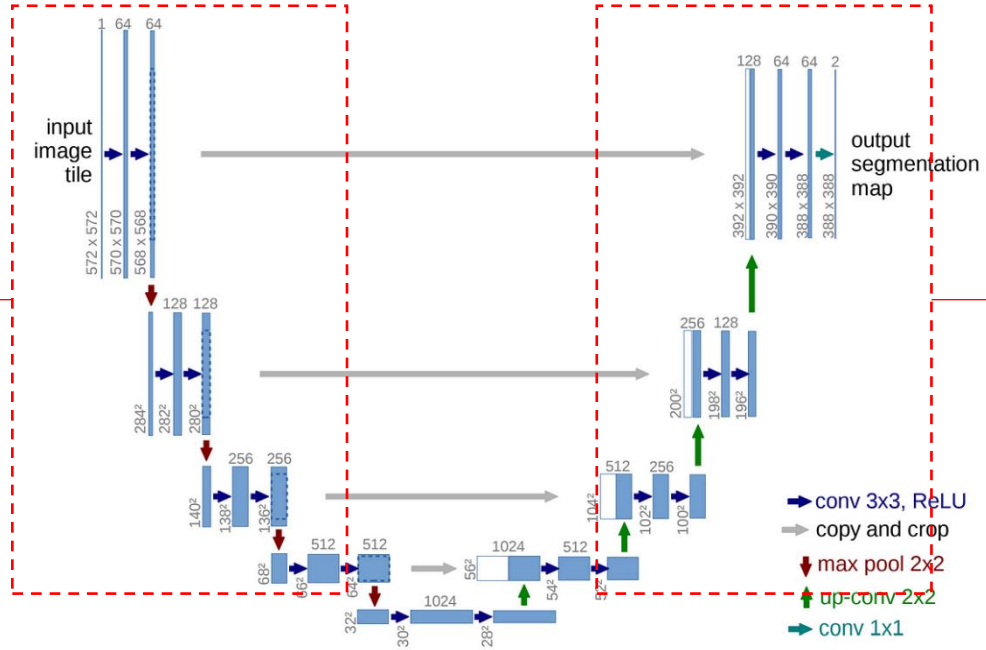


LeNet5 network architecture

CIFAR-10 분류 실습: U-Net을 이용한 분류

- **U-Net**
  - 2015년 Semantic Segmentation 기법으로 등장한 모델
  - Auto encoder 구조를 이용한 입력 이미지 정보 압축 가능

입력 이미지로부터 특징 학습

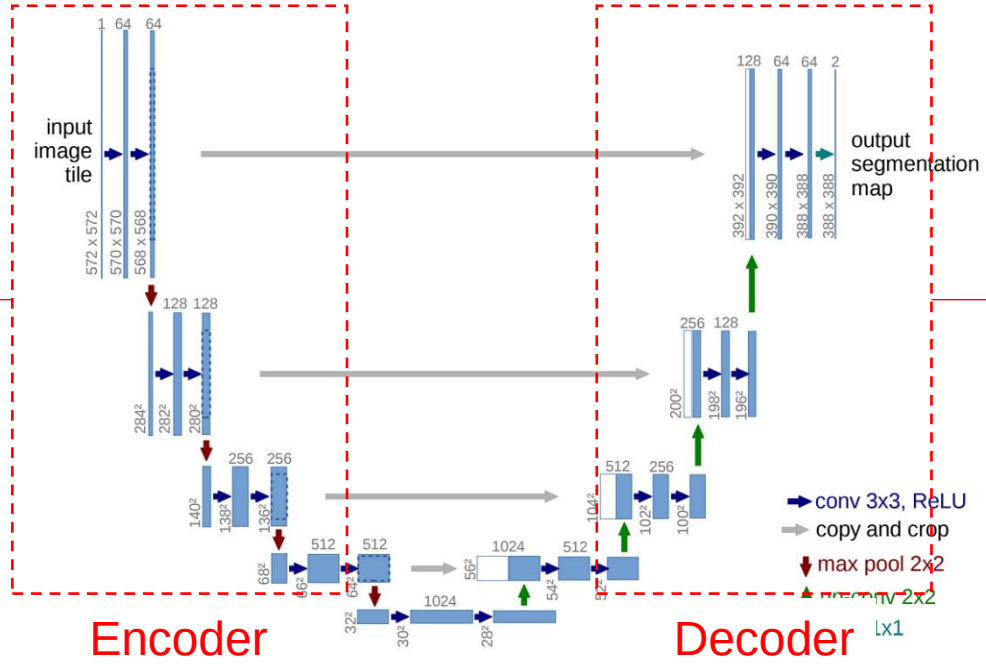


학습된 Feature map으로부터 Segmentation map 생성

CIFAR-10 분류 실습: U-Net을 이용한 분류

- **U-Net**
  - 2015년 Semantic Segmentation 기법으로 등장한 모델
  - Auto encoder 구조를 이용한 입력 이미지 정보 압축 가능

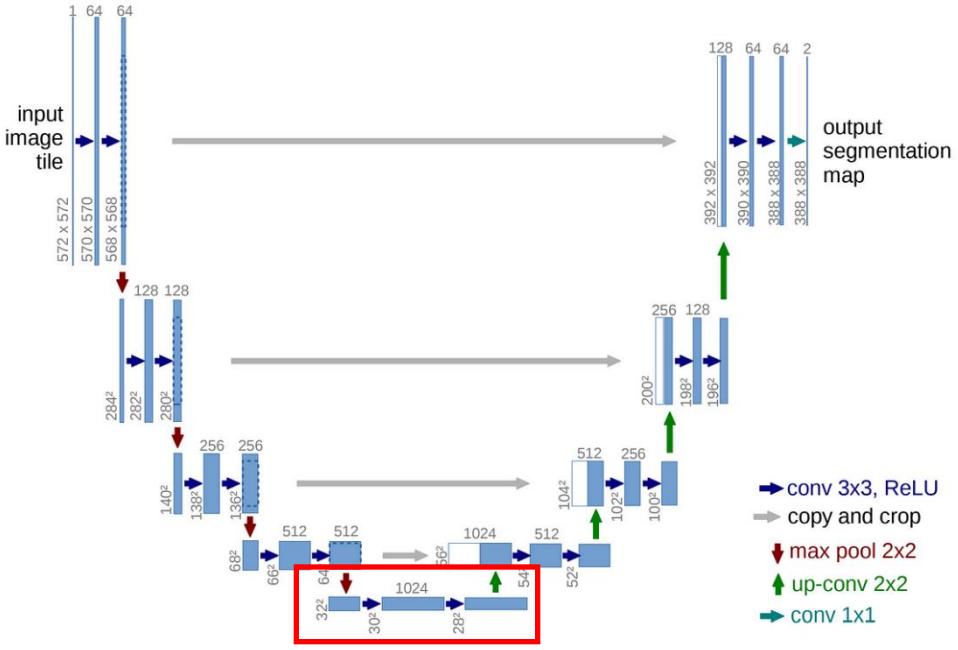
입력 이미지로부터 압축된 vector 생성



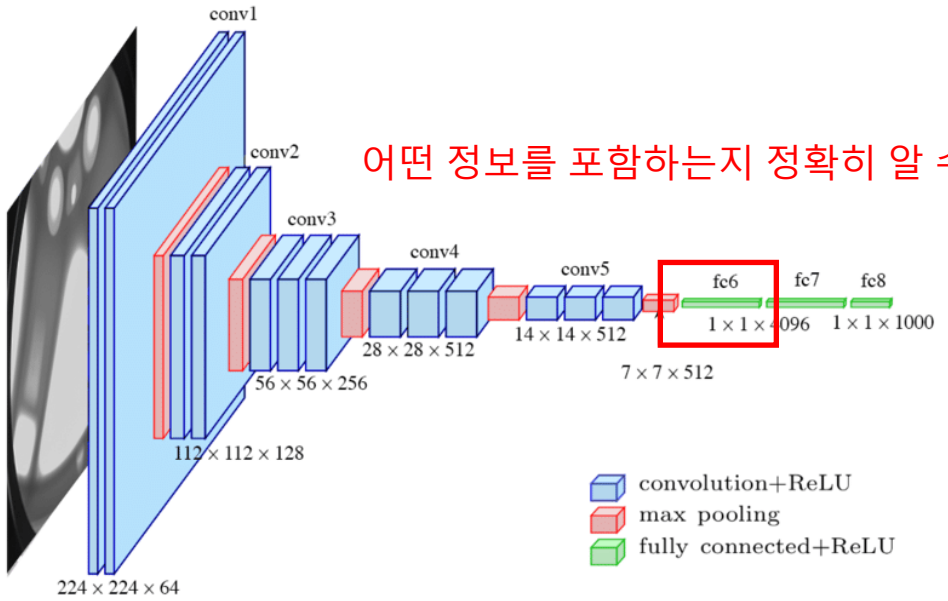
압축된 vector로부터 입력 이미지 생성

CIFAR-10 분류 실습: U-Net을 이용한 분류

- **U-Net**
  - 2015년 Semantic Segmentation 기법으로 등장한 모델
  - Auto encoder 구조를 이용한 입력 이미지 정보 압축 가능



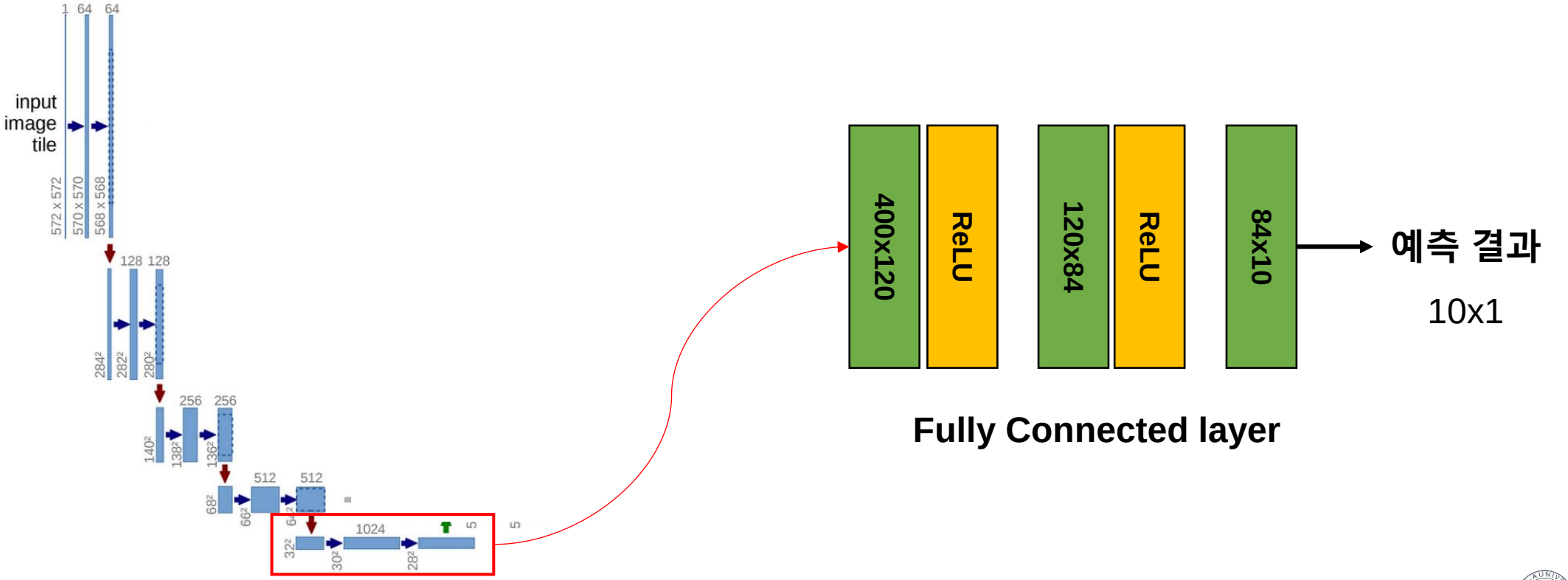
입력 이미지의 정보를 담고 있는 압축된 vector



어떤 정보를 포함하는지 정확히 알 수 없음

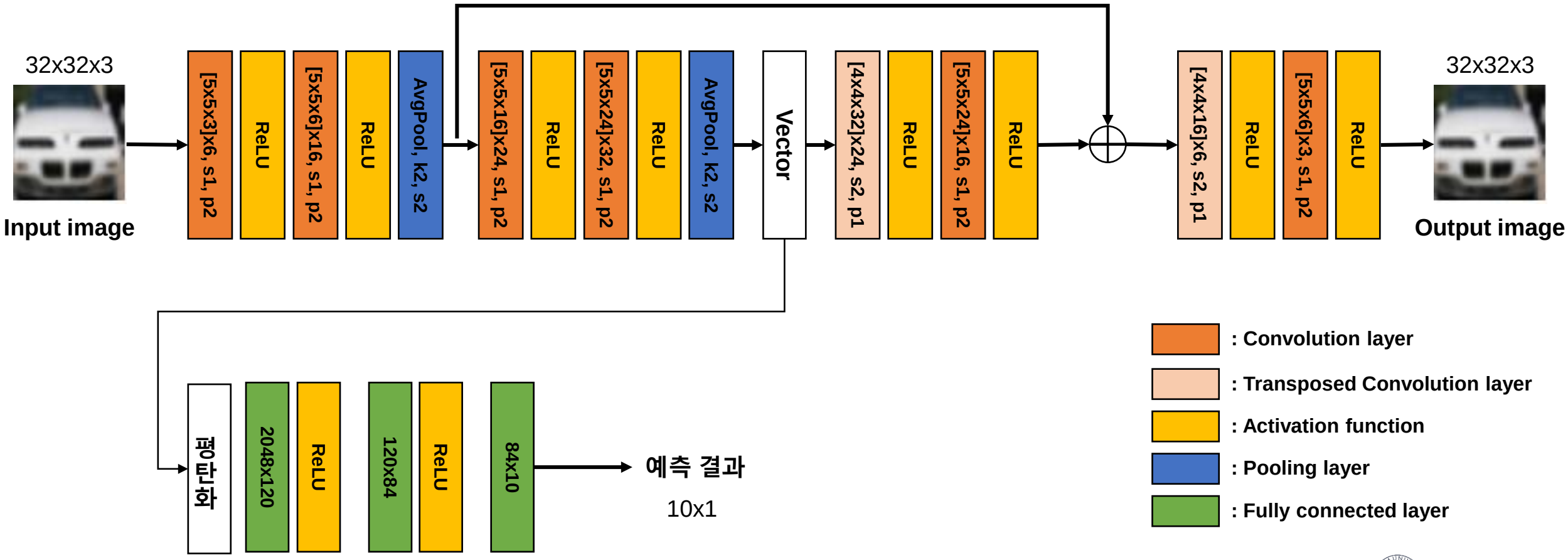
CIFAR-10 분류 실습: U-Net을 이용한 분류

- **U-Net**
  - 2015년 Semantic Segmentation 기법으로 등장한 모델
  - Auto encoder 구조를 이용한 입력 이미지 정보 압축 가능



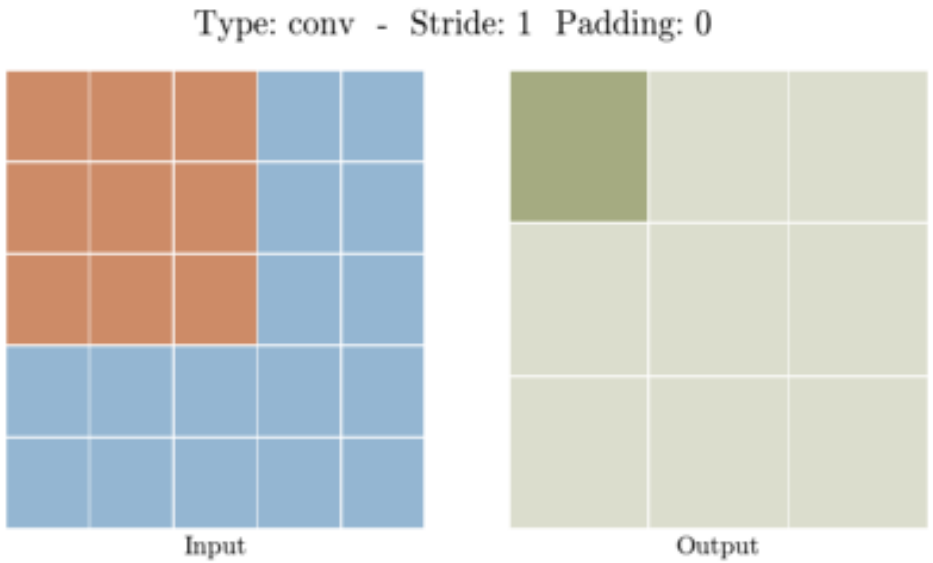
[실습] U-Net: CIFAR10 classification

- **U-Net 모델 구조 작성 참고사항**
  - Filter size: 5x5, Stride: 1, Padding: 2

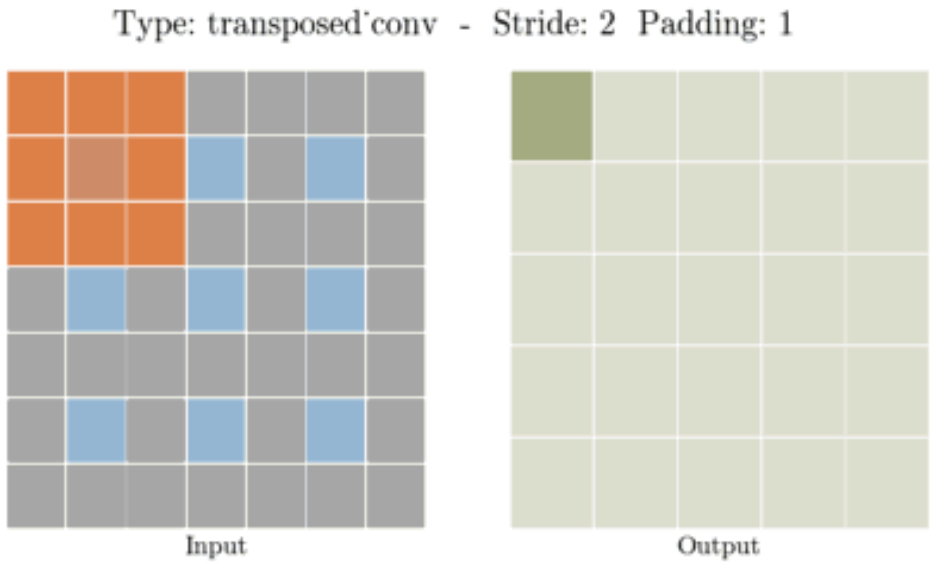


[실습] U-Net: CIFAR10 classification

- **U-Net 모델 구조 작성 참고사항**
  - Transposed Convolution: Up-sampling을 하기위한 convolution layer
  - Feature map 크기를 N배 Up-sampling하기 **위해서는 Kernel size: 2N, Stride: N, Padding = 0.5N으로 설정**



Convolution



Transposed Convolution

[실습] U-Net: CIFAR10 classification

- **U-Net 모델 구조 작성 참고사항**
  - Transposed Convolution: Up-sampling을 하기위한 convolution layer
  - Feature map 크기를 N배 Up-sampling하기 **위해서는 Kernel size: 2N, Stride: N, Padding = 0.5N으로 설정**

ConvTranspose2d

CLASS torch.nn.ConvTranspose2d(*in_channels*, *out_channels*, *kernel_size*, *stride=1*, *padding=0*, *output_padding=0*, *groups=1*, *bias=True*, *dilation=1*, *padding_mode='zeros'*, *device=None*, *dtype=None*) [SOURCE]

: Transposed Convolution layer

Questions & Answers

Dongsan Jun (dsjun@dau.ac.kr)
Image Signal Processing Laboratory (www.donga-ispl.kr)
Dept. of Computer Engineering
Dong-A University, Busan, Rep. of Korea