Deep Learning 101

3주차

12기 이두형
12기 임효진

KU-BIG

Curriculum

1주차 : 딥러닝 소개 및 기초 (XOR문제, 퍼셉트론, 활성화 함수 등)

2주차 : Multi-layer Neural Network (Loss Function, Gradient Descending, Backpropagation, MNIST practice, Optimization)

3주차 : CNN 소개 및 기초 (Convolution, Padding, Stride, Pooling등 기초 개념 소개)

4주차 : CNN 실습 (세션 후 조별 과제 부여)

5주차 : RNN, LSTM, GRU

6주차 : seq2seq, 실습 (세션 후 조별 과제 부여)

7주차 : 조별 과제 발표

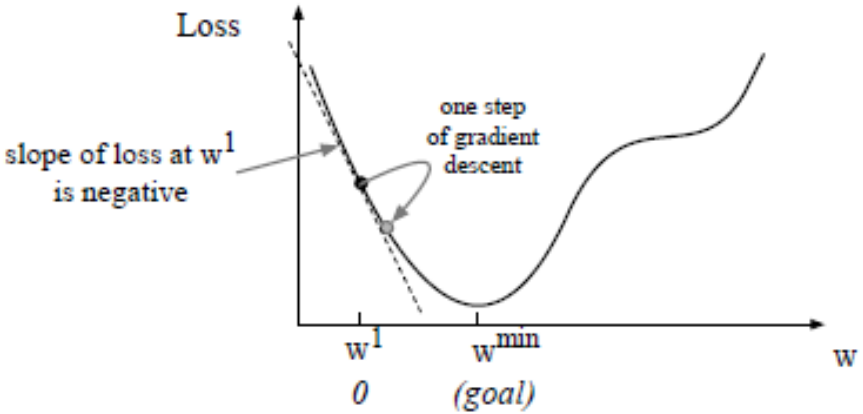
KU-BIG

Review

KU-BIG

Review

- Gradient Descending

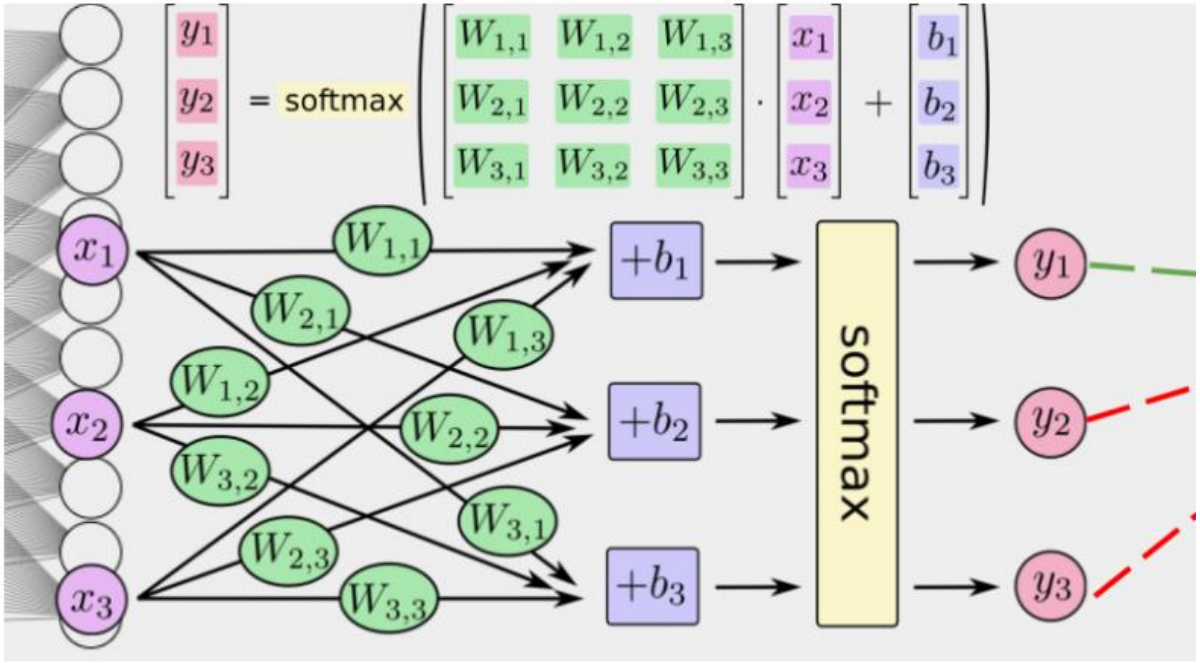


- Batch Gradient Descent
- Stochastic Gradient Descent
- Mini-Batch Gradient Descent

KU-BIG

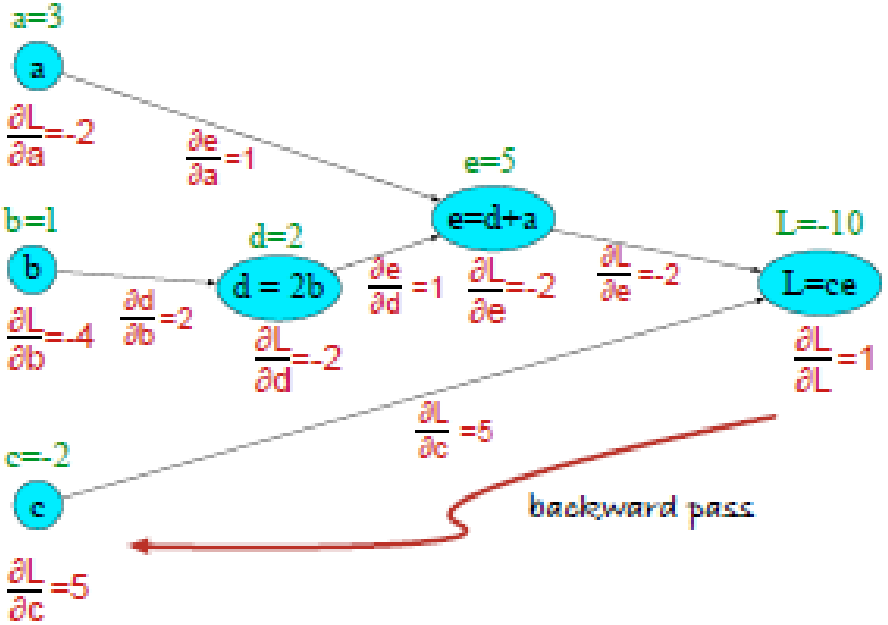
Review

- Multi-Layer Perceptron



Review

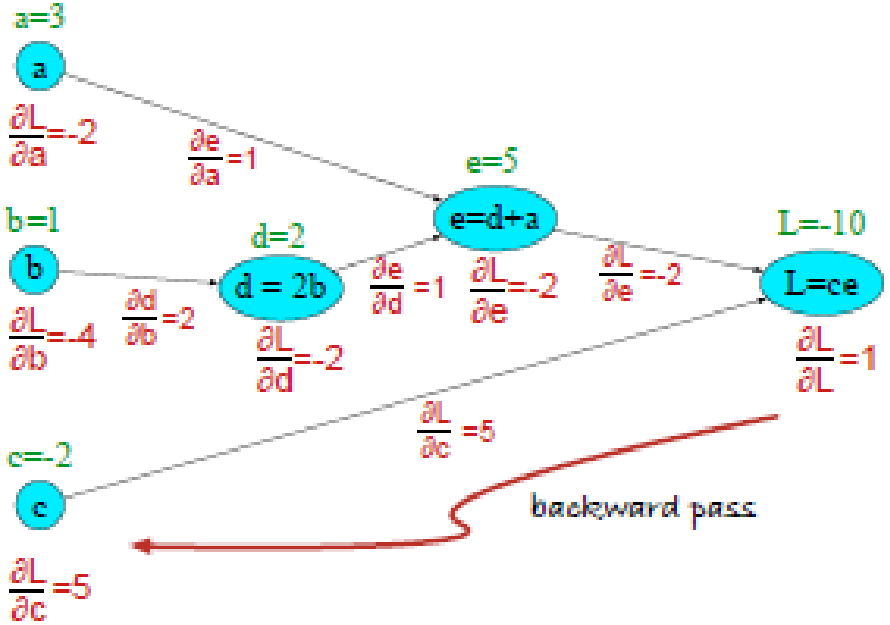
- Back Propagation



KU-BIG

Review

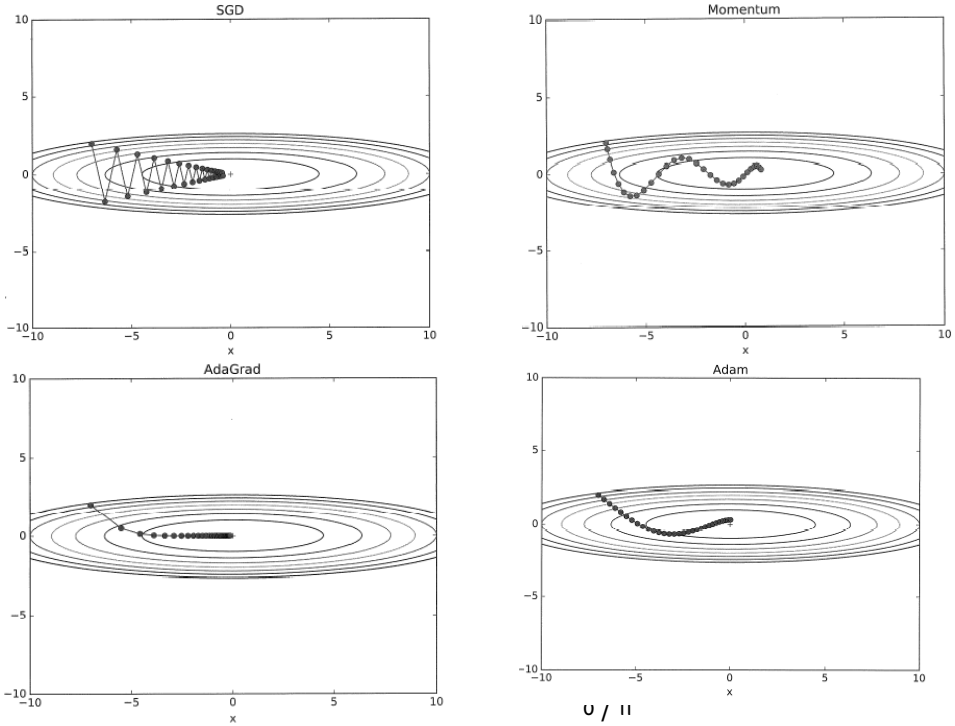
- Back Propagation



KU-BIG

Review

- Optimizer



Review

- Weight Initialization

- Xavier Initialization

$$W \sim Unif\left(-\sqrt{\frac{6}{n_{in}+n_{out}}}, +\sqrt{\frac{6}{n_{in}+n_{out}}}\right)$$

$$W \sim N\left(0, \left(\sqrt{\frac{2}{n_{in}+n_{out}}}\right)^2\right)$$

- He Initialization

$$W \sim Unif\left(-\sqrt{\frac{6}{n_{in}}}, +\sqrt{\frac{6}{n_{in}}}\right)$$

$$W \sim N\left(0, \left(\sqrt{\frac{2}{n_{in}}}\right)^2\right)$$

KU-BIG

Review

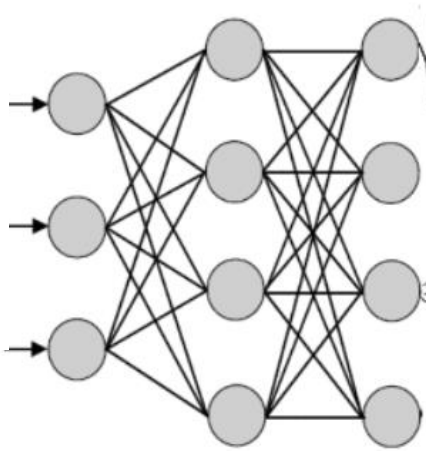
- Batch Normalizatioin & Dropout

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

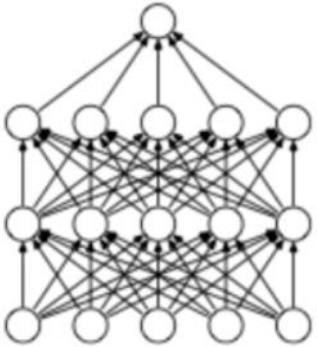
$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

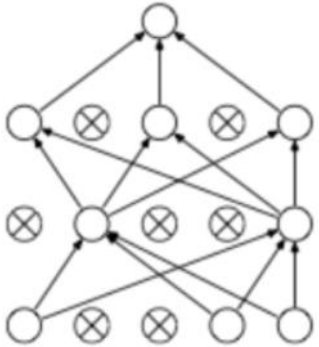
$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$



Dropout: A Simple Way to Prevent Neural Networks from Overfitting [Srivastava et al. 2014]



(a) Standard Neural Net



(b) After applying dropout.

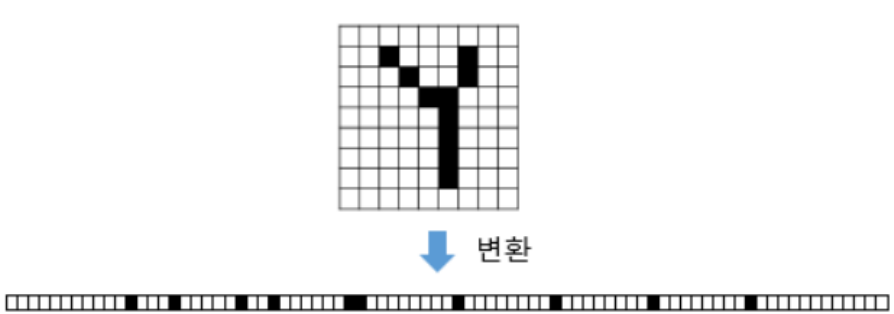
KU-BIG

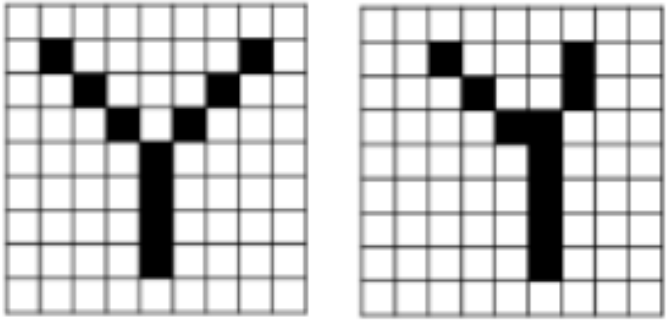
Week3

KU-BIG

Multi-Layer Perceptron

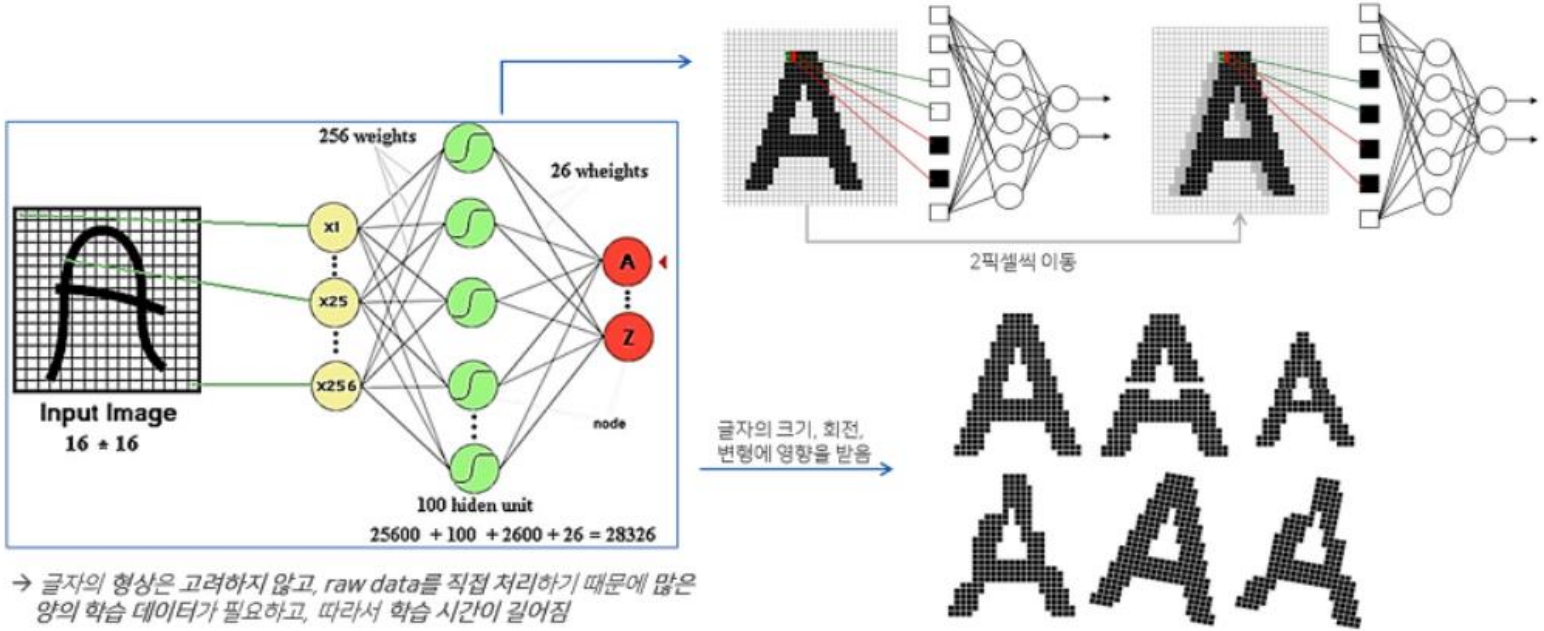
- Problem





Multi-Layer Perceptron

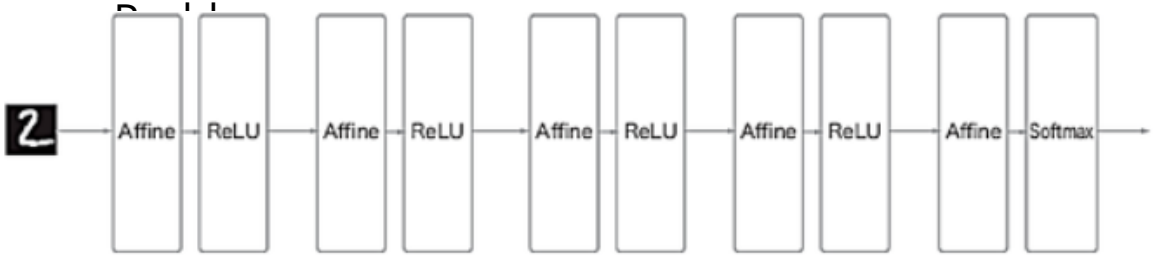
- Problem

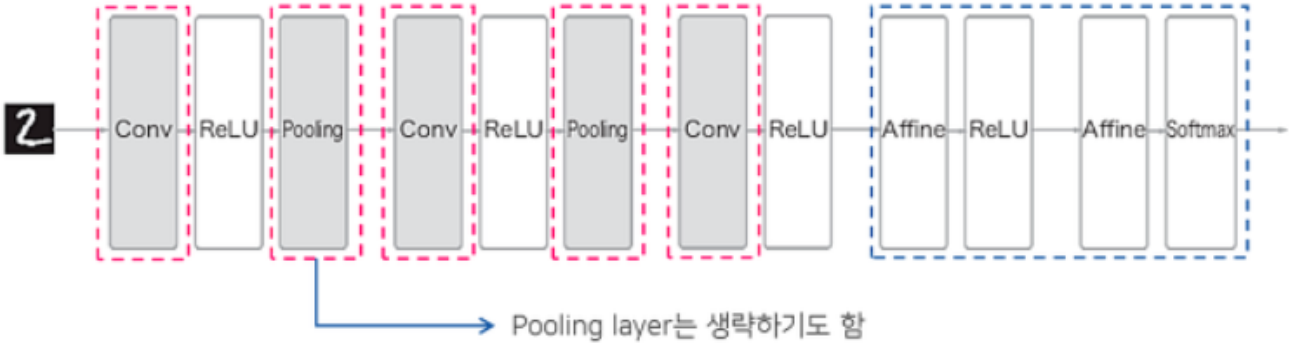


→ *글자의 형상은 고려하지 않고, raw data를 직접 처리하기 때문에 많은 양의 학습 데이터가 필요하고, 따라서 학습 시간이 길어짐*

KU-BIG

CNN (Convolution Neural Network)

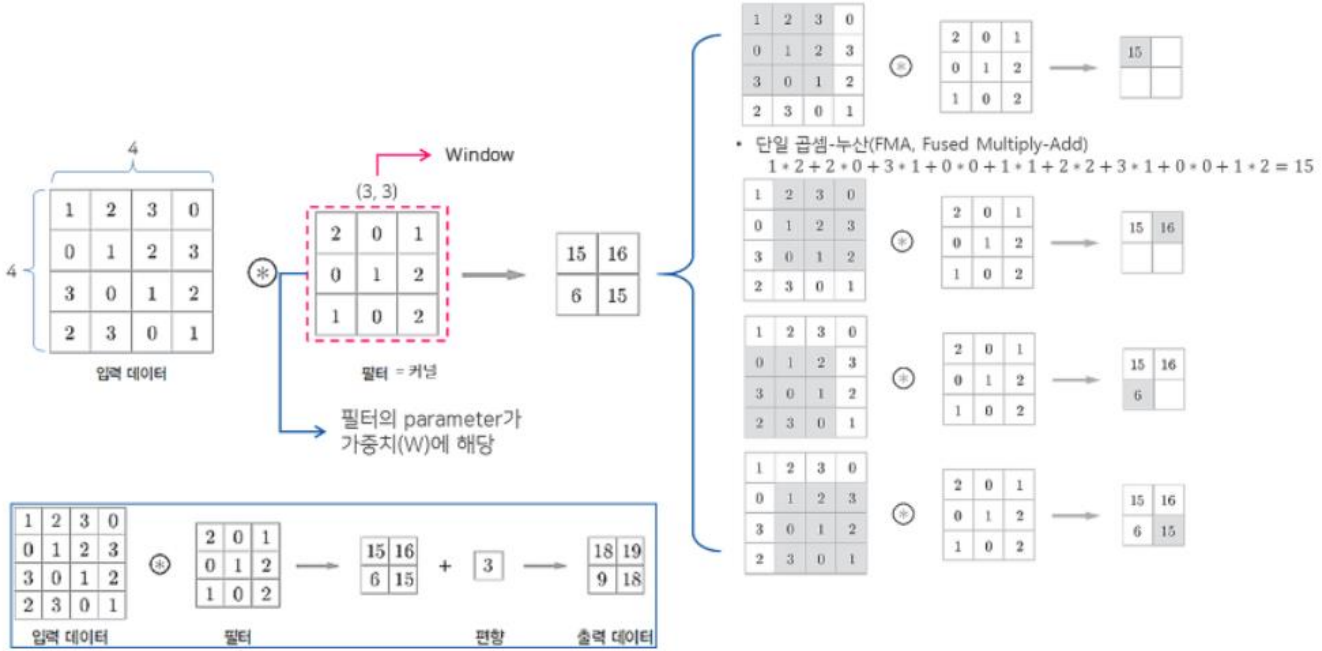




KU-BIG

CNN (Convolution Neural Network)

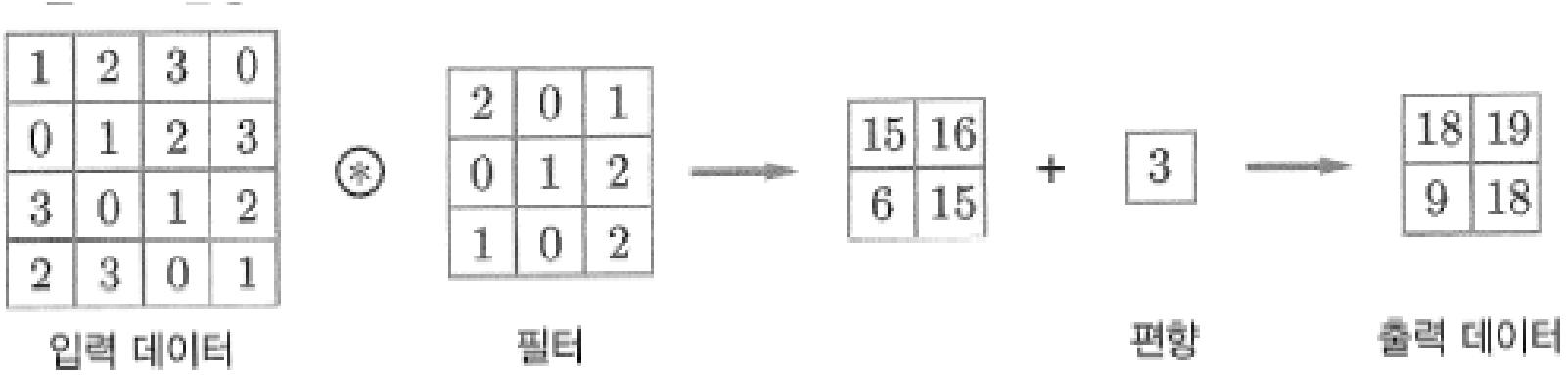
- Convolution



KU-BIG

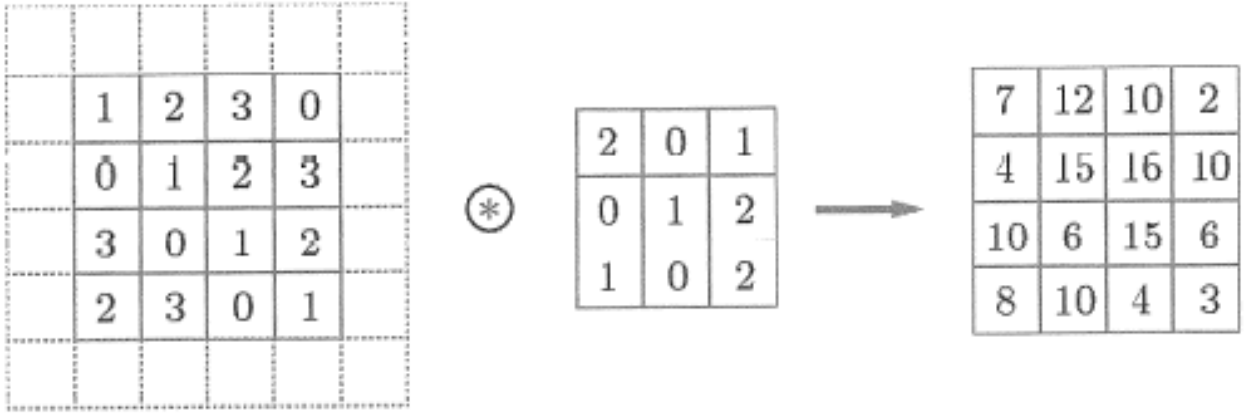
CNN (Convolution Neural Network)

- Padding



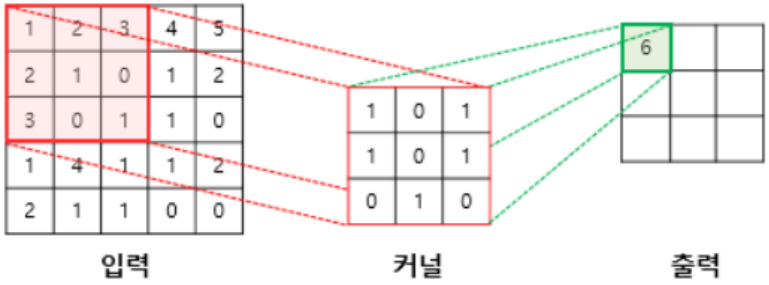
CNN (Convolution Neural Network)

- Padding

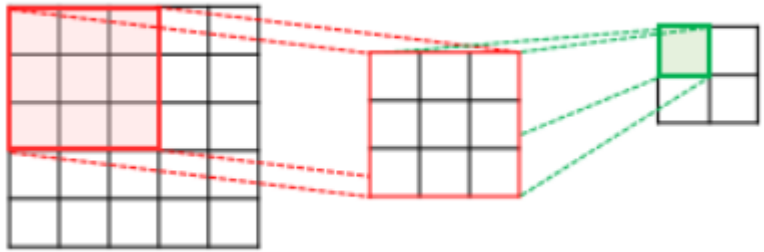


CNN (Convolution Neural Network)

- Stride



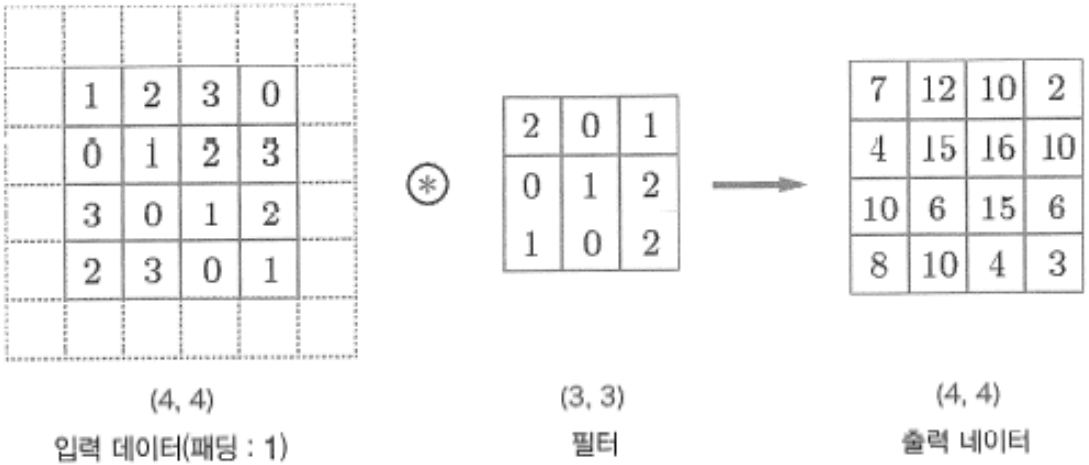
Stride:1



Stride:2

CNN (Convolution Neural Network)

- Feature map size



(4, 4)
입력 데이터(패딩 : 1)

(3, 3)
필터

(4, 4)
출력 데이터

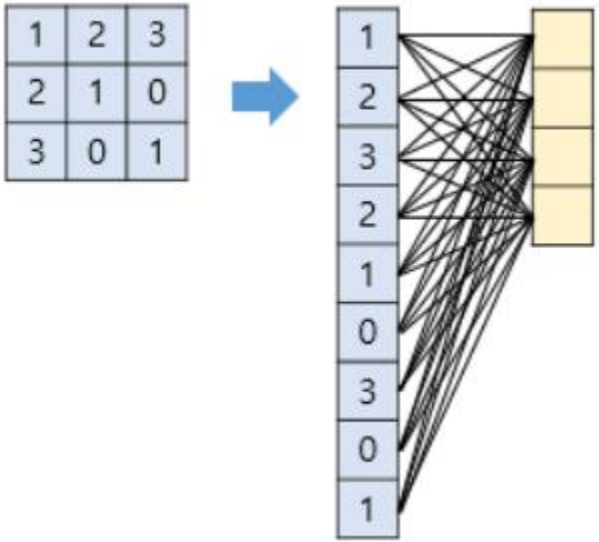
$$OH = \frac{H + 2P - FH}{S} + 1$$

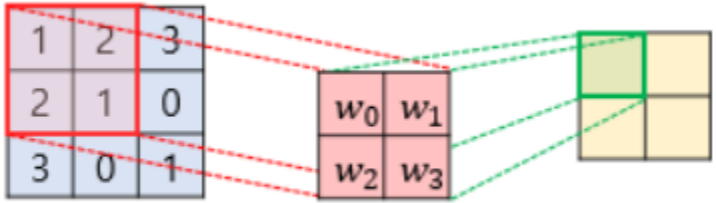
$$OW = \frac{W + 2P - FW}{S} + 1$$

KU-BIG

CNN (Convolution Neural Network)

- Parameter

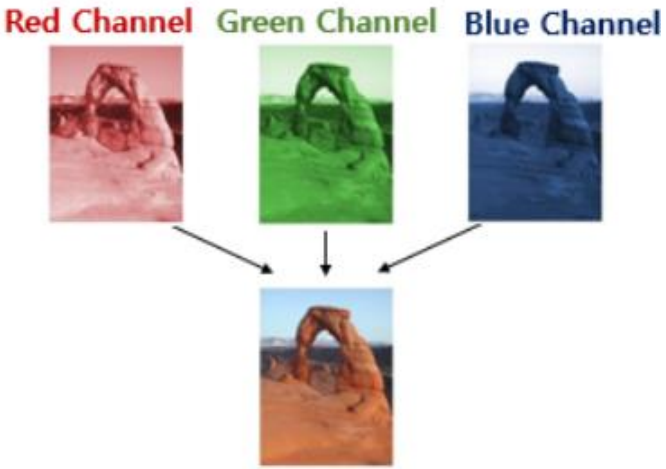


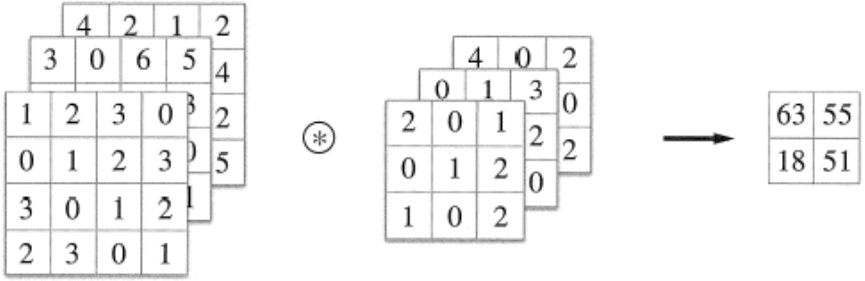


KU-BIG

CNN (Convolution Neural Network)

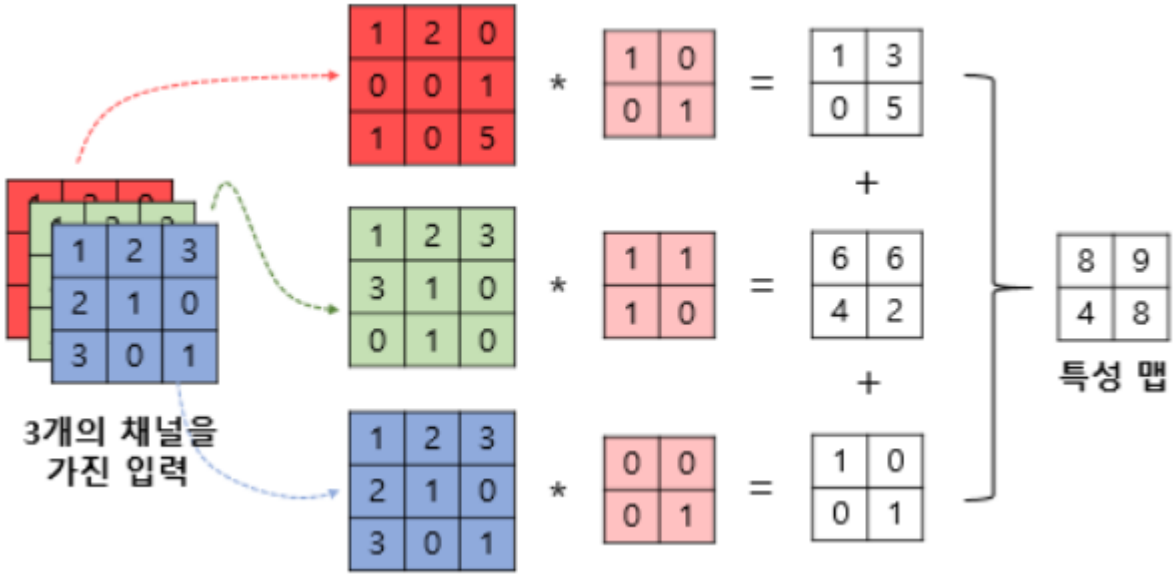
- Channel





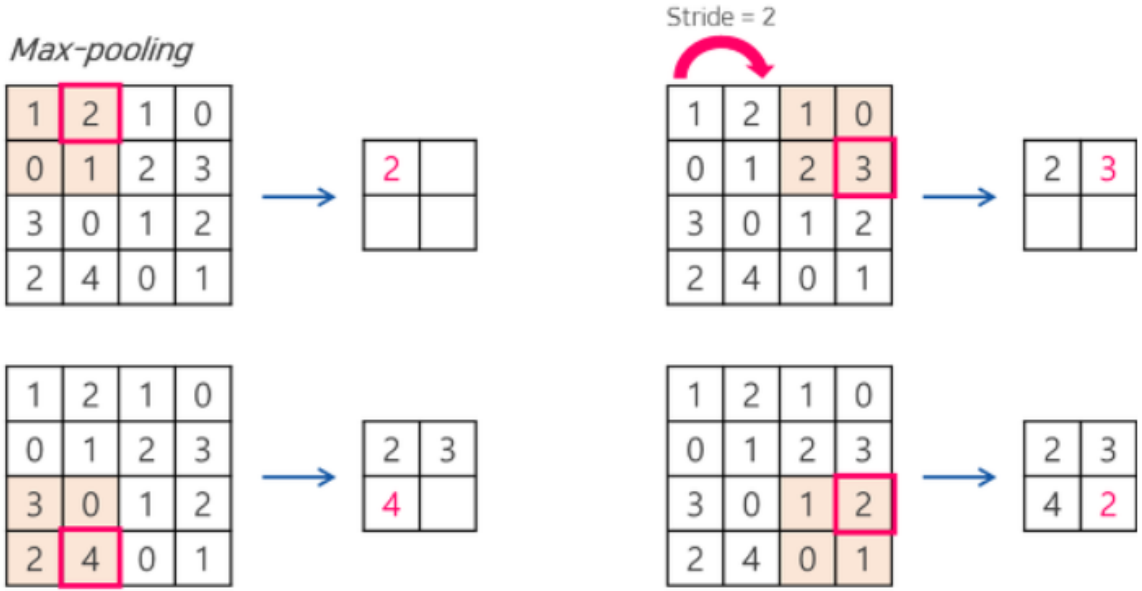
CNN (Convolution Neural Network)

- 3-Dimensional Convolution



CNN (Convolution Neural Network)

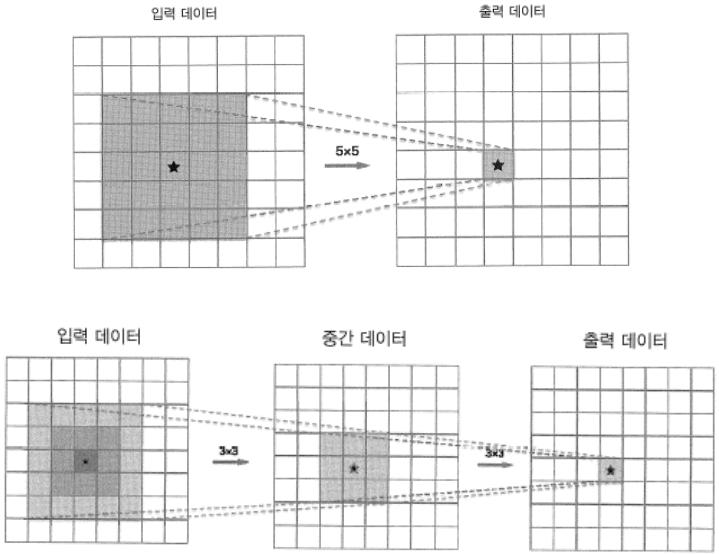
- Pooling

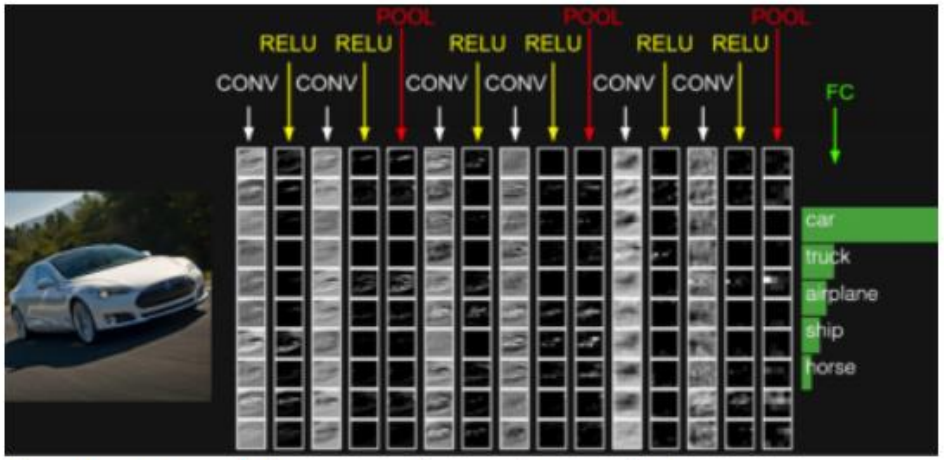


[그림 7] Max-pooling 예제

CNN (Convolution Neural Network)

- Deep Learning



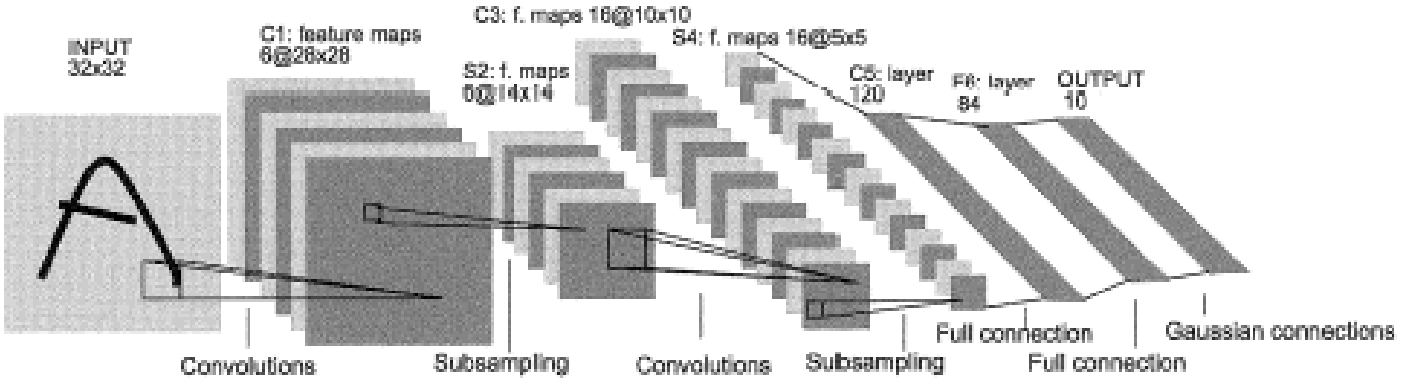


KU-BIG

CNN (Convolution Neural Network)

- LeNet

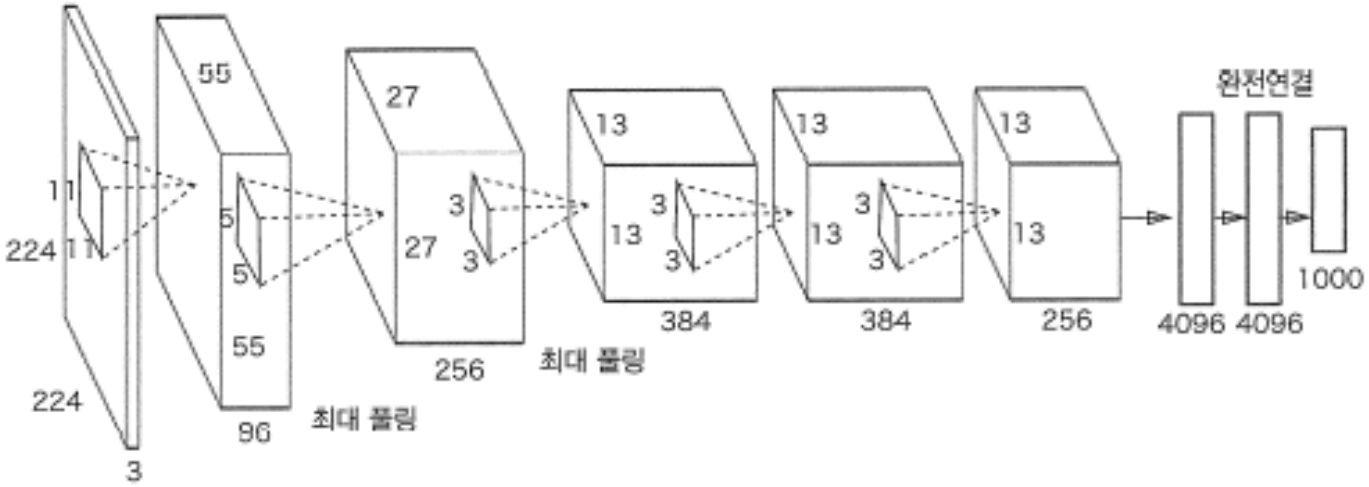
그림 7-27 LeNet의 구성[20]



CNN (Convolution Neural Network)

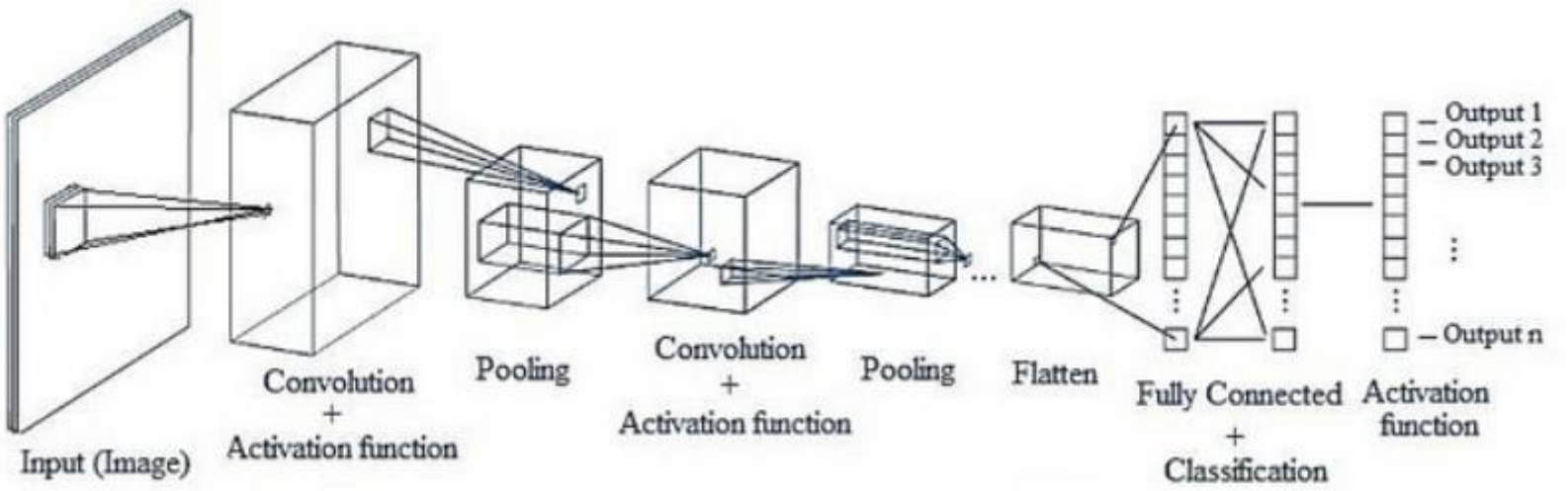
- AlexNet

그림 7-28 AlexNet의 구성[21]



CNN (Convolution Neural Network)

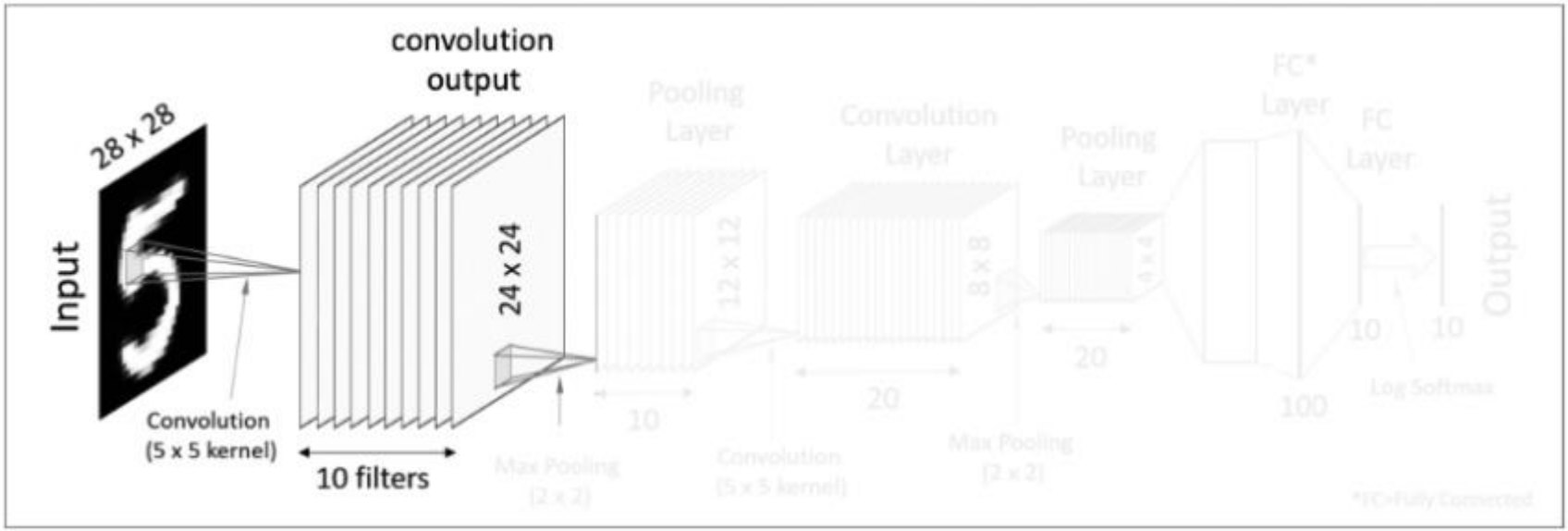
- Alex net Structure



KU-BIG

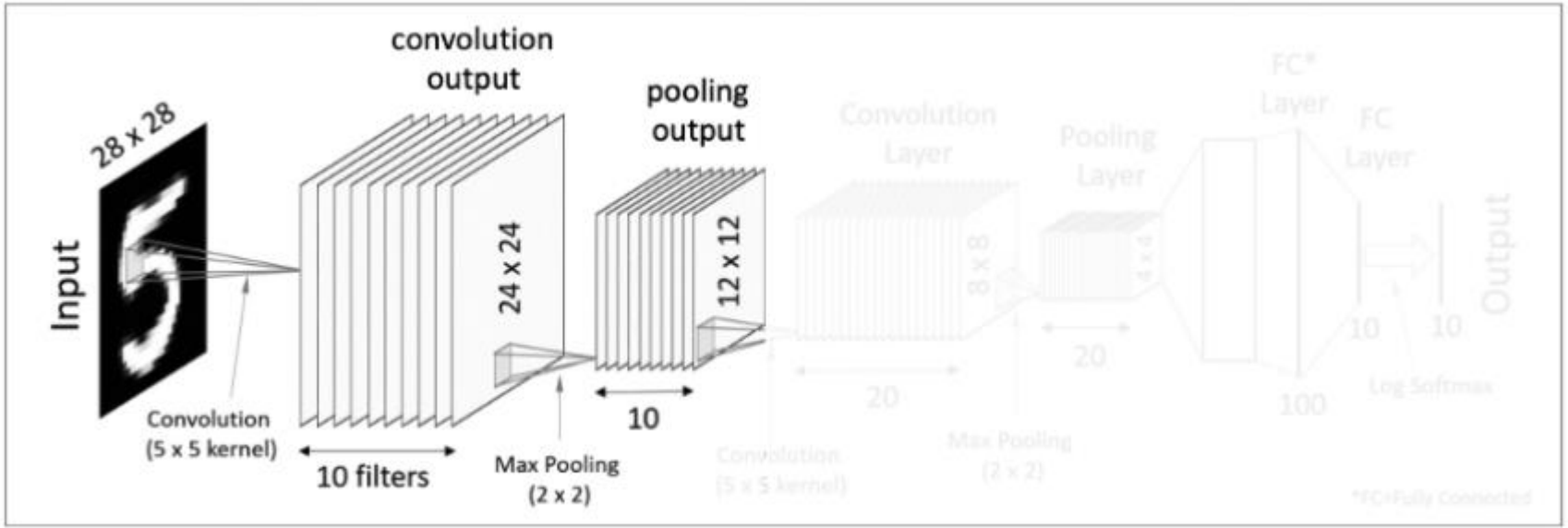
CNN (Convolution Neural Network)

- 1st Convolution Layer



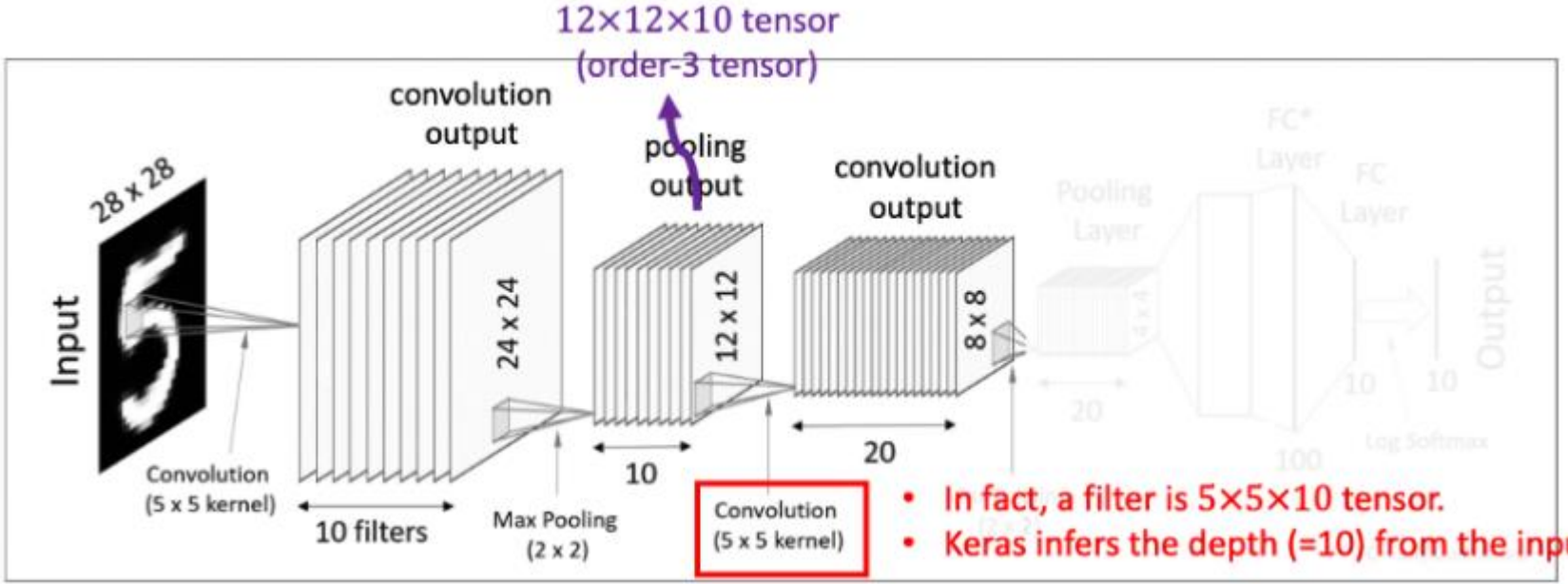
CNN (Convolution Neural Network)

- 1st Pooling Layer



CNN (Convolution Neural Network)

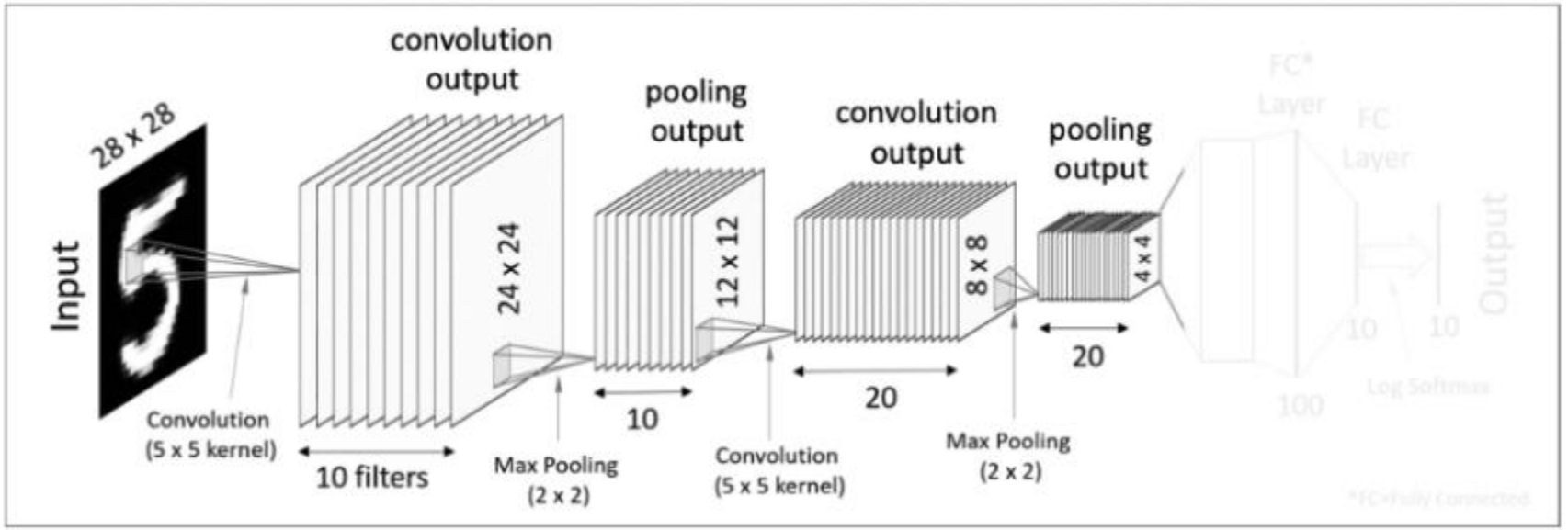
- 2nd Convolution Layer



KU-BIG

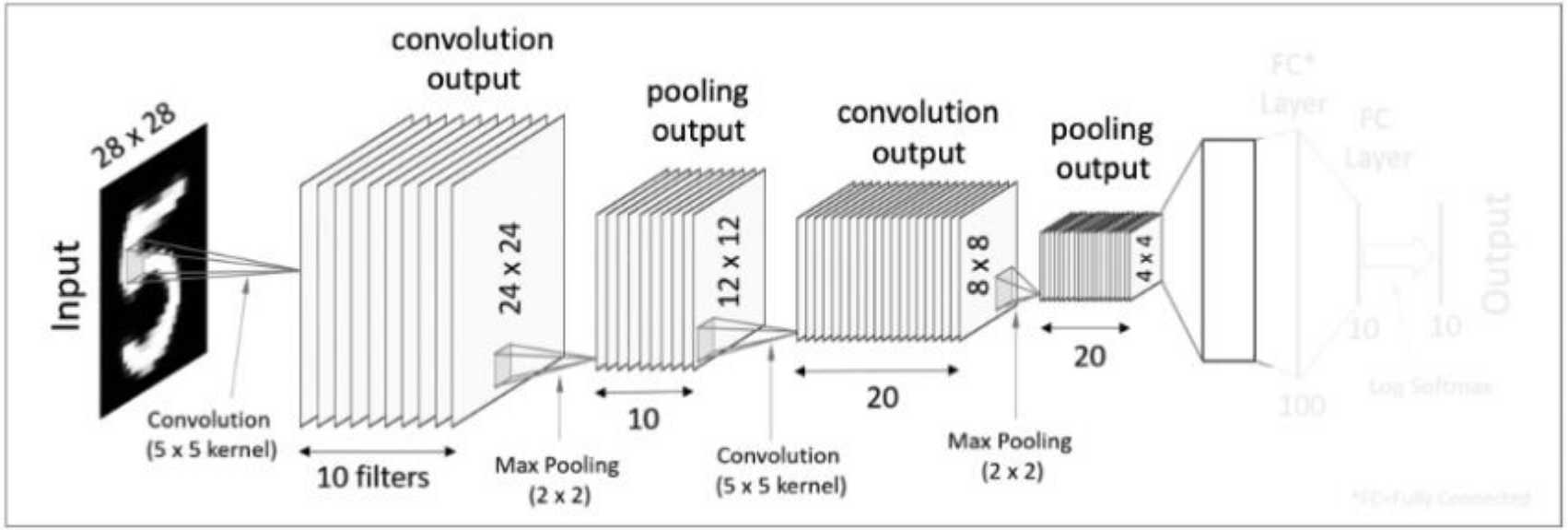
CNN (Convolution Neural Network)

- 2nd Pooling Layer



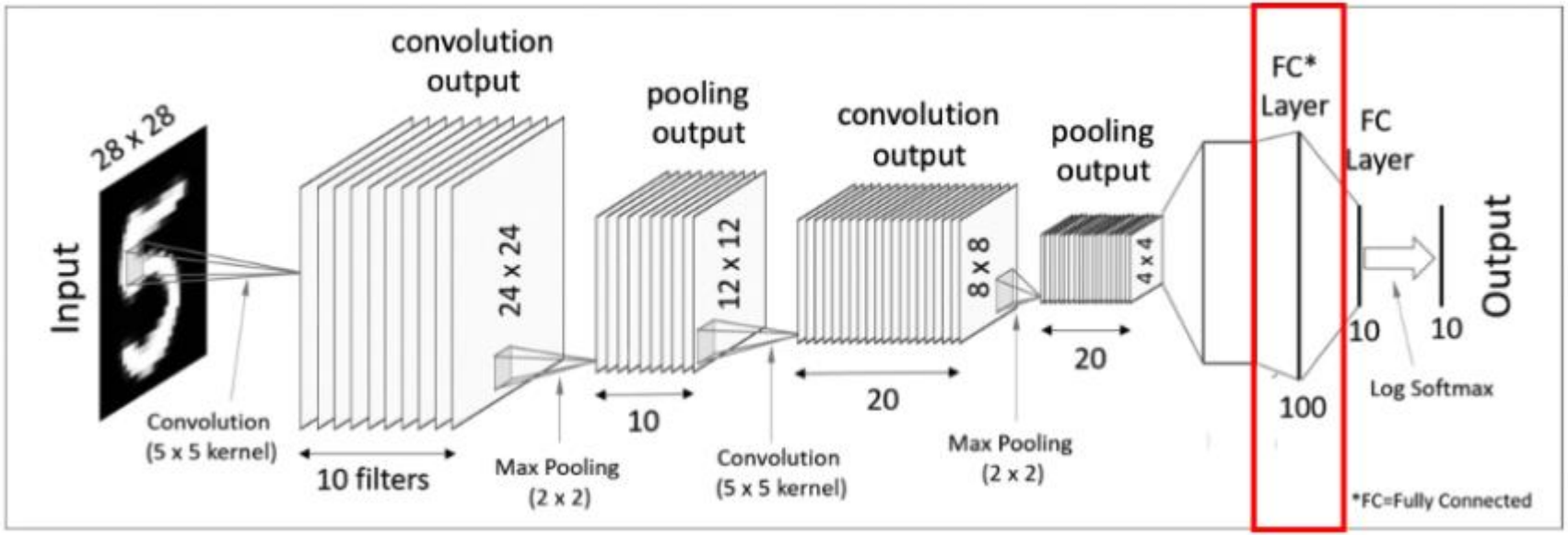
CNN (Convolution Neural Network)

- Flatten



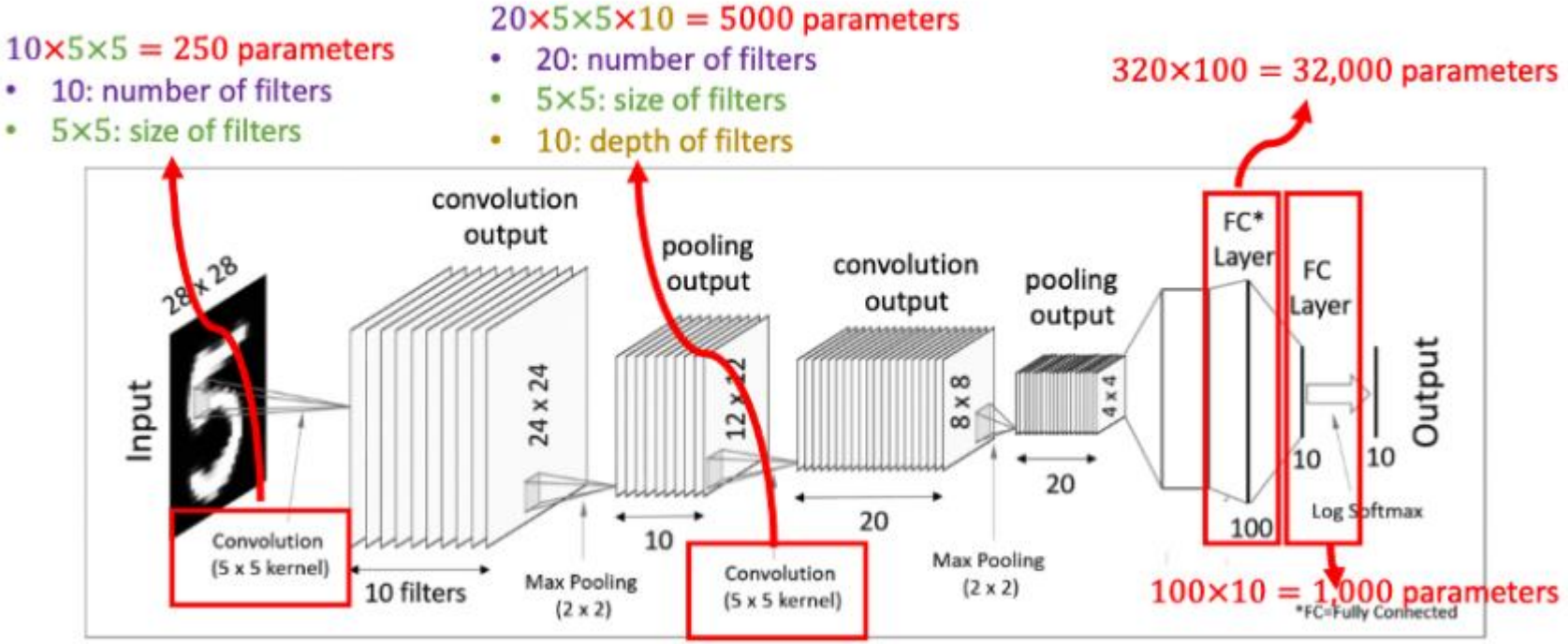
CNN (Convolution Neural Network)

- Fully Connected Layer



KU-BIG

CNN (Convolution Neural Network)



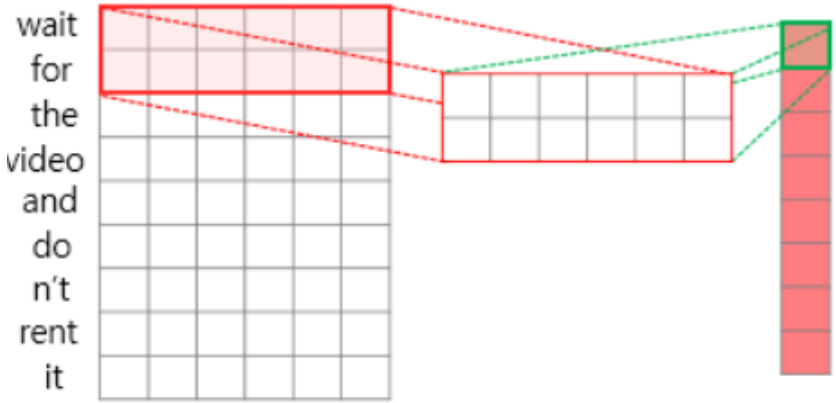
KU-BIG

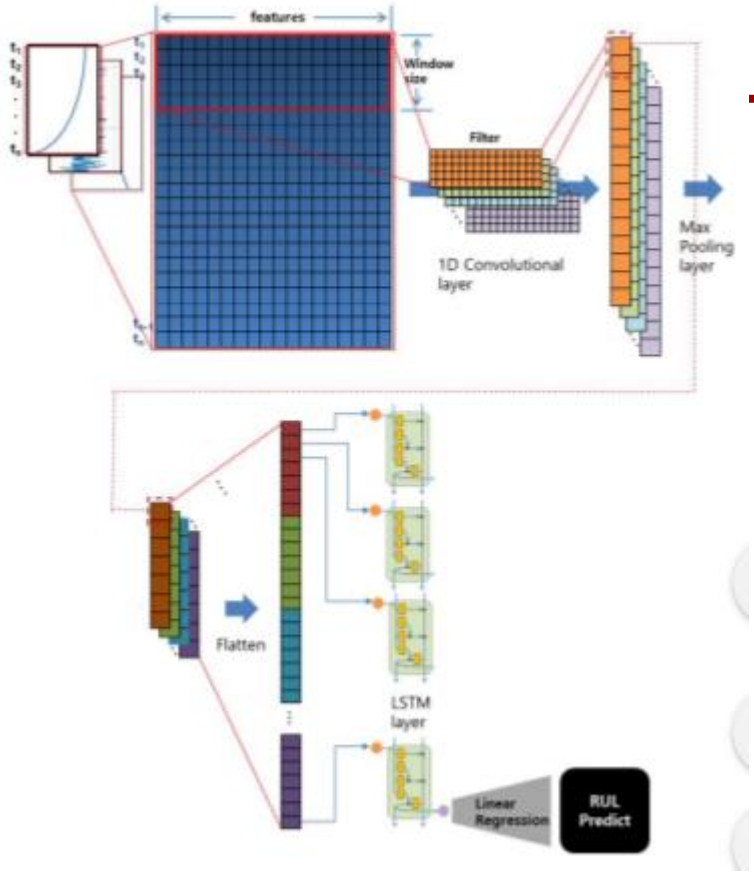
CNN (Convolution Neural Network)

- Hyper-Parameter

- Convolutional layers : 필터 개수, 필터 크기, stride 값, zero-padding
- Pooling Layer : Pooling 방식, Pool 크기, Pool stride 값
- Fully-Connected Layers : 넓이 (width)
- 활성함수 : ReLU, SoftMax, Sigmoid
- Loss Function :
- Optimizer : SGD, AdaGrad, Momentum, Adam + learning rate
- Random Initialization : Gaussian or uniform

1d CNN

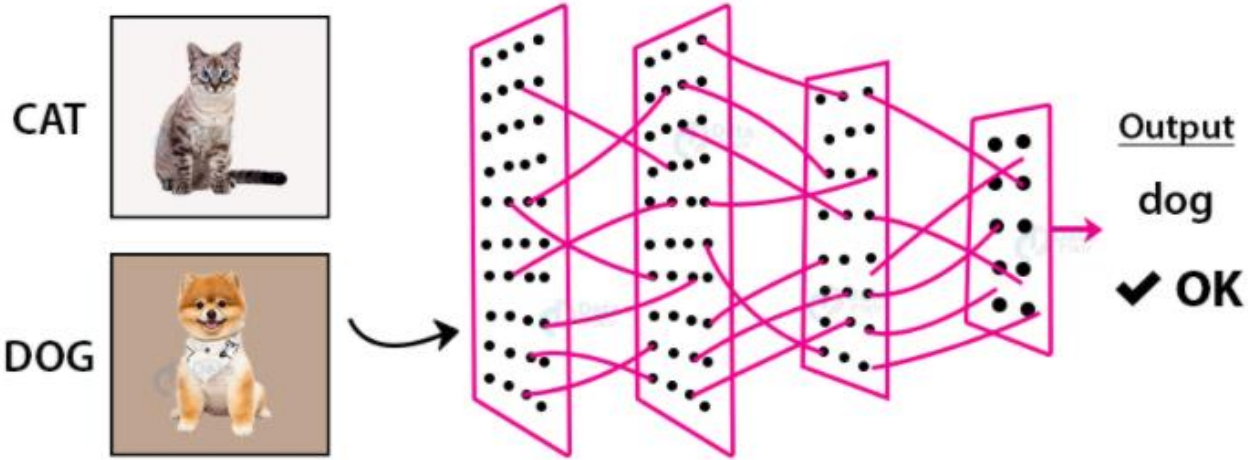

wait
for
the
video
and
do
n't
rent
it


features
Window size
Filter
1D Convolutional layer
Max Pooling layer
Flatten
LSTM layer
Linear Regression
RUL Predict

KU-BIG

Dogs vs cats

- practice



KU-BIG