

CNN Introduction

2021.10.15

Contents

- **Confusion Matrix**
 - **MNIST Dataset**
 - **Activation Function**
 - **FC Layer**
 - **Batch Normalization**
 - **Convolution Layer**
 - **Pooling Layer**
-

Confusion Matrix (1)

- 성능 측정을 위해 예측값과 실제값을 비교한 표

		예측값	
		Positive	Negative
실제값	Positive	TP (True Positive)	FN (False Negative)
	Negative	FP (False Positive)	TN (True Negative)

Confusion Matrix (2)

- 정확도(accuracy)

- N개의 데이터 샘플 중 예측에 성공한 샘플의 비율 ($\frac{TP+TN}{TP+FN+FP+TN}$)

		예측값	
		Positive	Negative
실제값	Positive	TP (True Positive)	FN (False Negative)
	Negative	FP (False Positive)	TN (True Negative)

Confusion Matrix (3)

• 정밀도(precision)

- 모델이 Positive로 예측한 것 중 실제값 또한 Positive인 비율 ($\frac{TP}{TP+FP}$)

예측값

	Positive	Negative
Positive	TP (True Positive)	FN (False Negative)
Negative	FP (False Positive)	TN (True Negative)

실제값

Confusion Matrix (4)

• 재현도(recall)

- 실제값이 Positive인 것 중 모델이 Positive로 예측한 비율 ($\frac{TP}{TP+FN}$)

예측값

	Positive	Negative
실제값	<div>Positive</div> <div>TP (True Positive)</div>	<div>FN (False Negative)</div>
	<div>Negative</div> <div>FP (False Positive)</div>	<div>TN (True Negative)</div>

Confusion Matrix (5)

• F1 Score

- 정밀도와 재현도의 조화 평균 (역수의 산술평균의 역수, $\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$)

예측값

	Positive	Negative
Positive	TP (True Positive)	FN (False Negative)
Negative	FP (False Positive)	TN (True Negative)

실제값

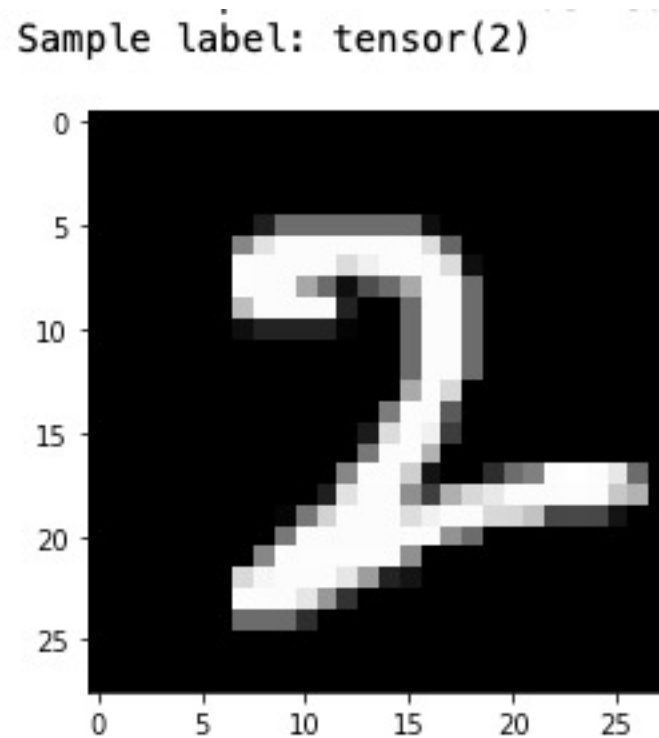
MNIST Dataset (1)

- 0~9 사이 숫자들의 handwritten database
- 이미지 분류 문제에 사용



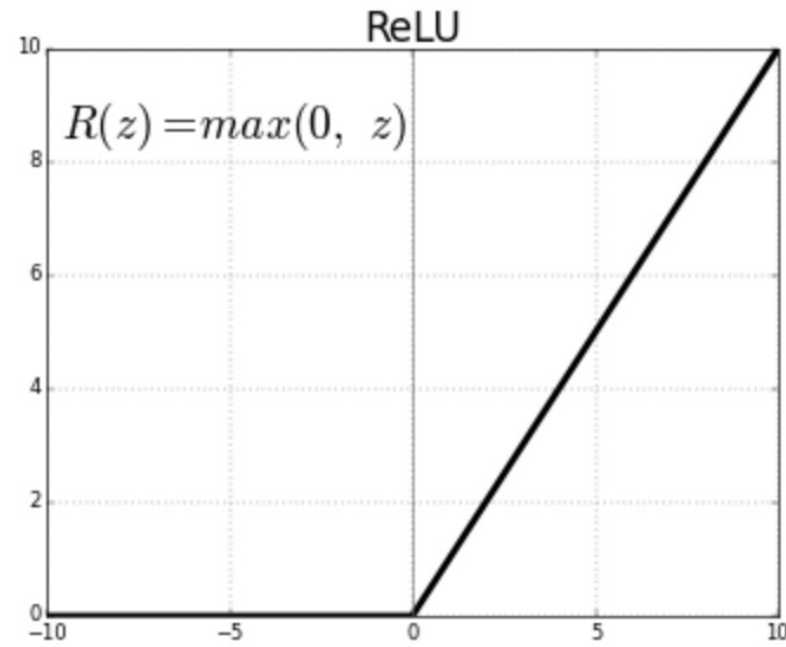
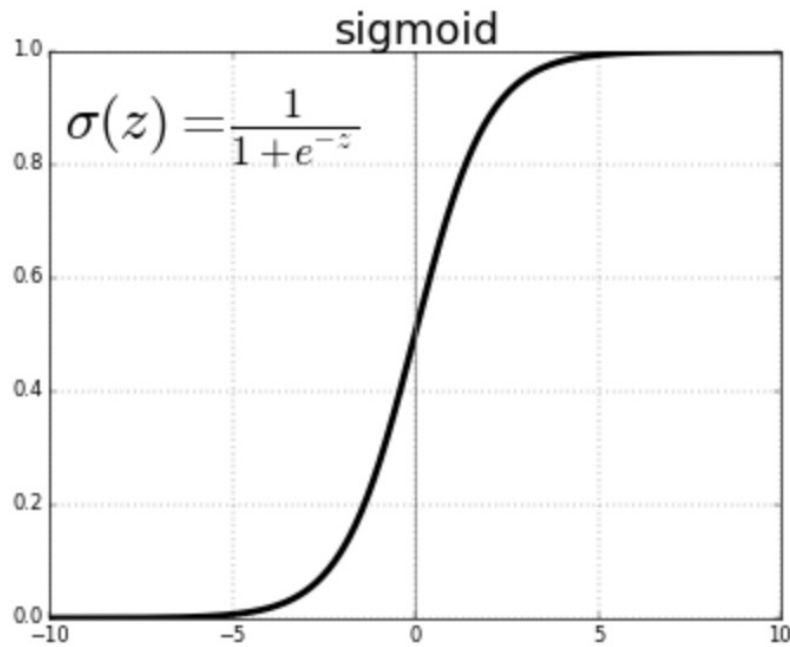
MNIST Dataset (2)

- 28x28 크기의 **흑백** 그림 (28 x 28 x 1)



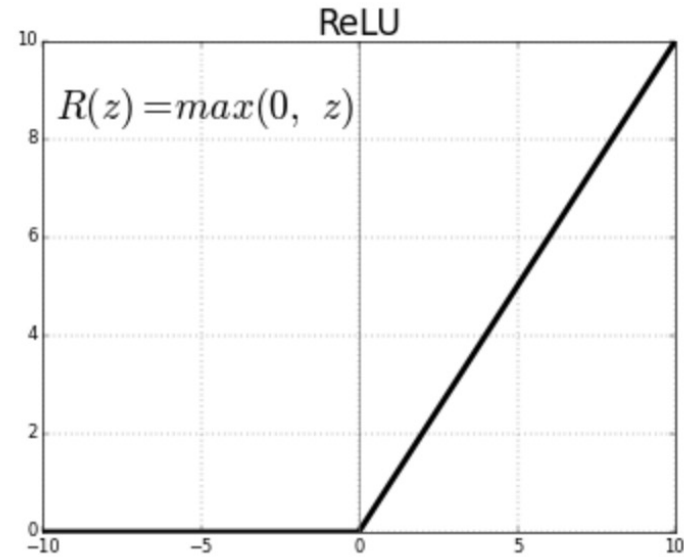
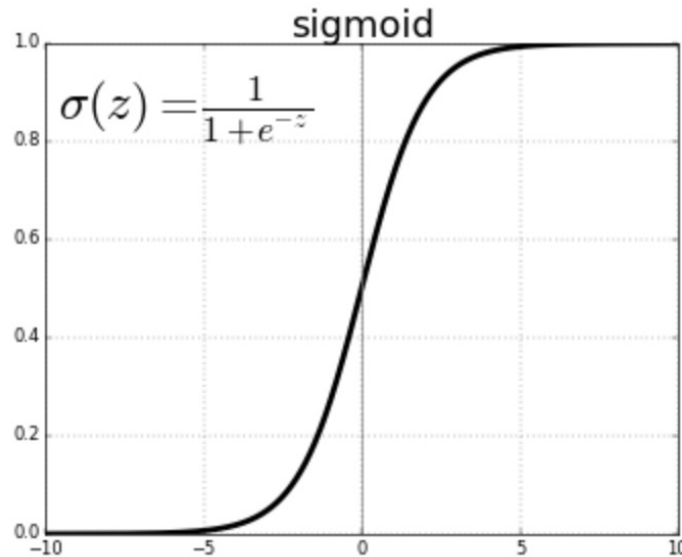
Activation Function (1)

- 선형 모델을 무한정 쌓으면, 그저 하나의 선형 모델이 될 뿐임
- 따라서, 딥러닝 모델에서는 선형 함수 사이에 **활성화 함수**를 넣어 줌
- **비선형적인 활성화 함수**를 넣어줌으로써, 층을 깊이 쌓는 의미를 갖게 됨



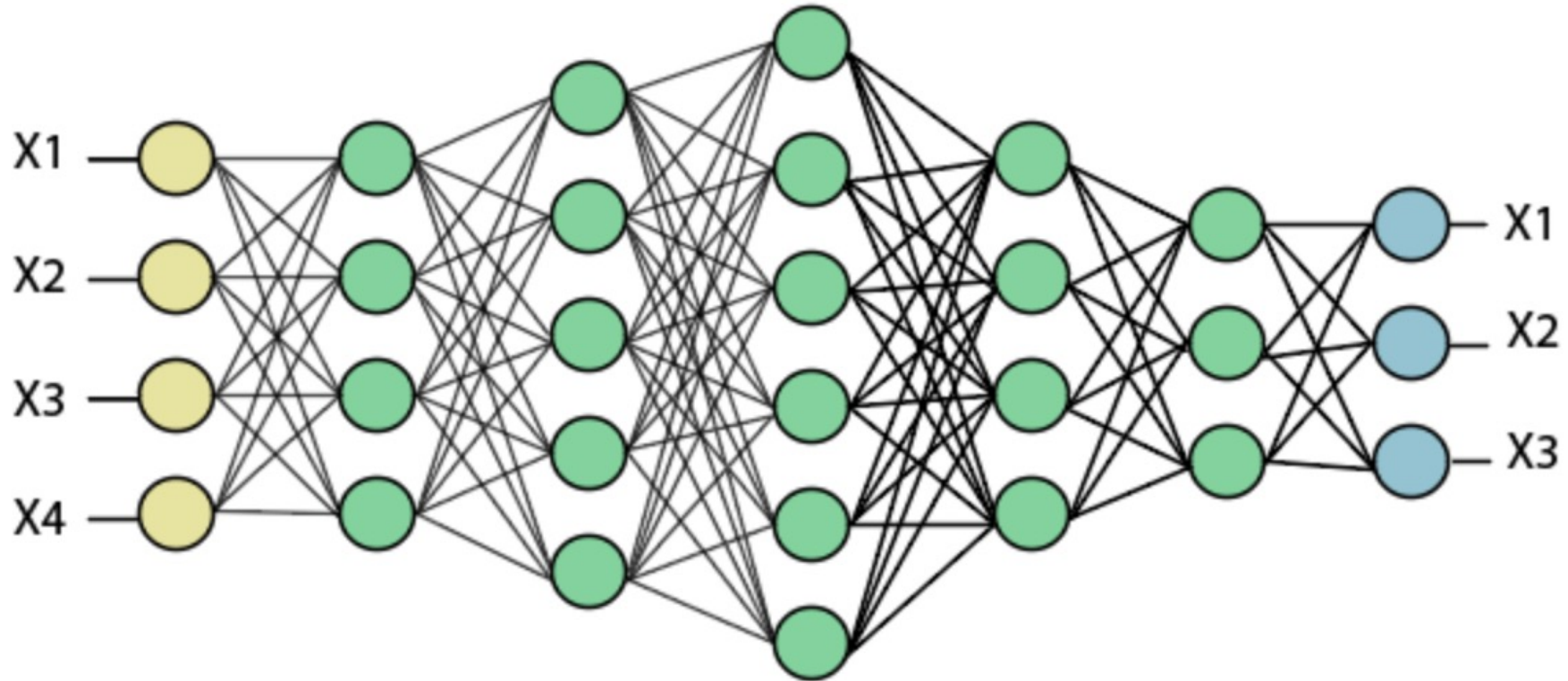
Activation Function (2)

- 딥러닝에서는 **쌓인 층들의 미분값을 모두 곱하는 역전파** 방식으로 학습됨
- 이 때, **Sigmoid 함수**는 양끝에서 0 근처의 미분값을 가져, **vanishing gradient** 문제가 발생할 수 있음
- 양수값에 한해서는 1의 미분값을 뱃는 **ReLU**의 성능을 살펴볼 계획



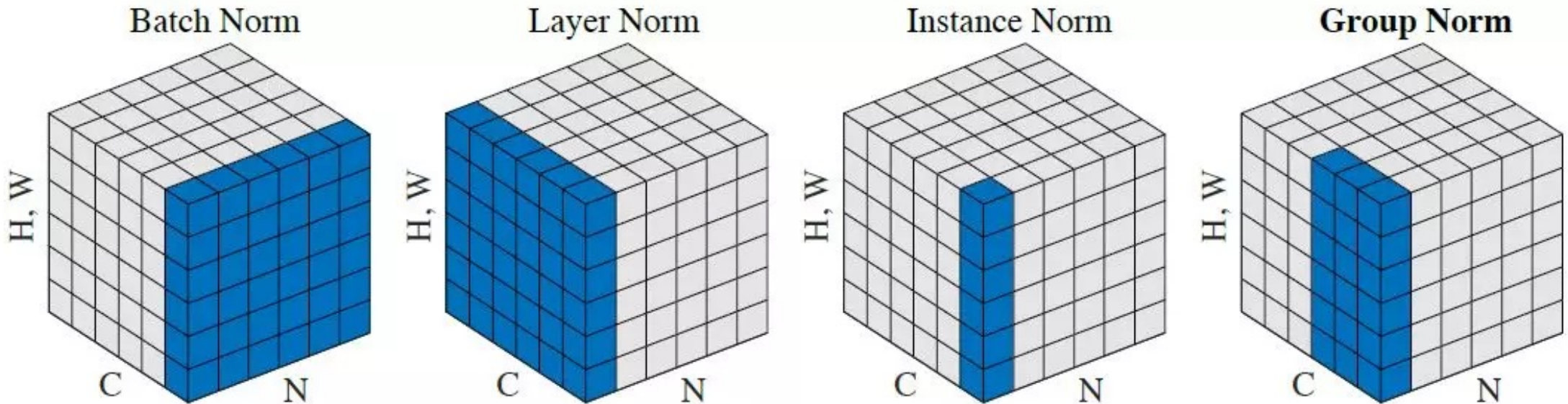
FC Layer

- 모든 Input 노드에 파라미터를 연결하여 Output을 연산하는 layer



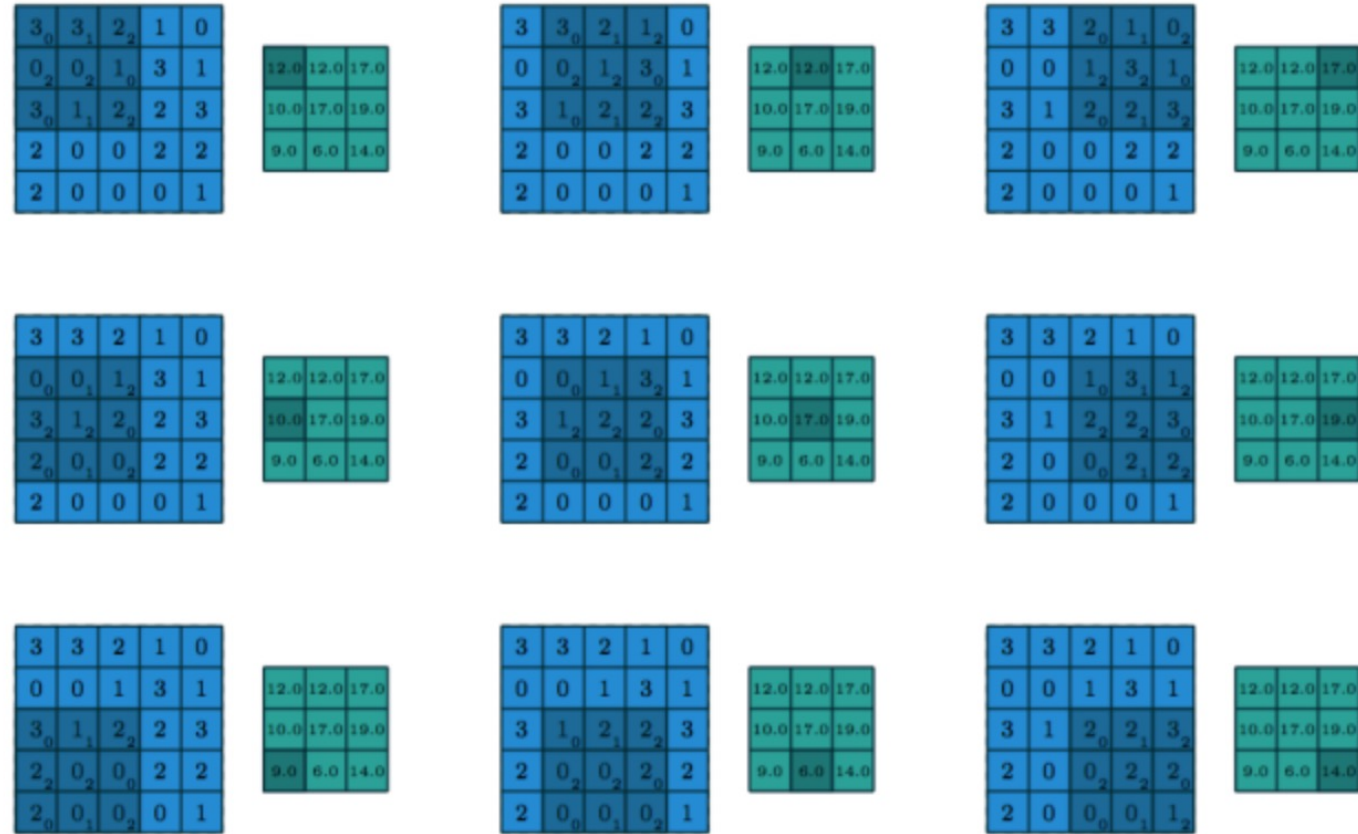
Batch Normalization Layer

- 신경망의 중간 산출값을 **배치별 평균과 분산으로 정규화**하여, 학습의 안정성을 도움.
- **Covariate shift** 문제를 해결한다고 알려져 있음



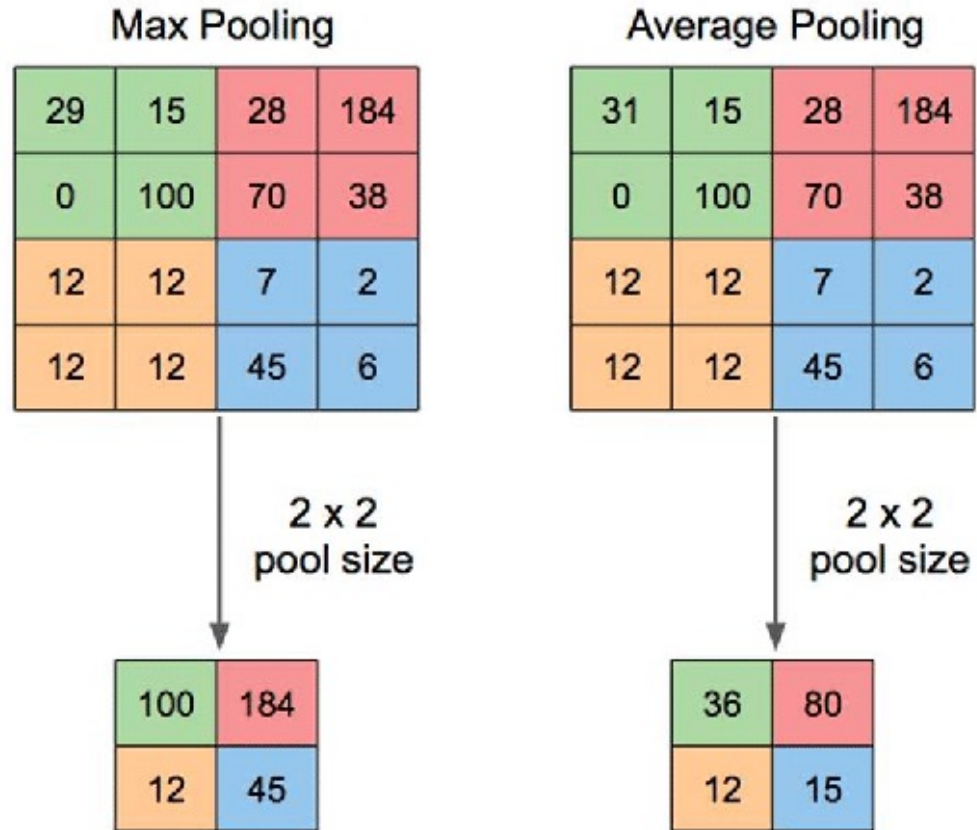
Convolution Layer

- Input을 지정된 간격으로 순회하며 연산하는 층



Pooling Layer

- Input의 차원을 축소하되, 특정한 정보(**Average or Max**)를 유지함.



Thank you

