교육부 Ministry of Education

NRF 한국연구재단 National Research Foundation of Korea

인공지능시스템

Convolutional Neural Network

Convergence and Open Sharing System COSS 인공지능

전남대학교 CHONNAM NATIONAL UNIVERSITY

성균관대학교 SUNG KYUN KWAN UNIVERSITY

KNU 경북대학교 KYUNGPOOK NATIONAL UNIVERSITY

전주대학교 JEONJU UNIVERSITY

서울시립대학교 UNIVERSITY OF SEOUL

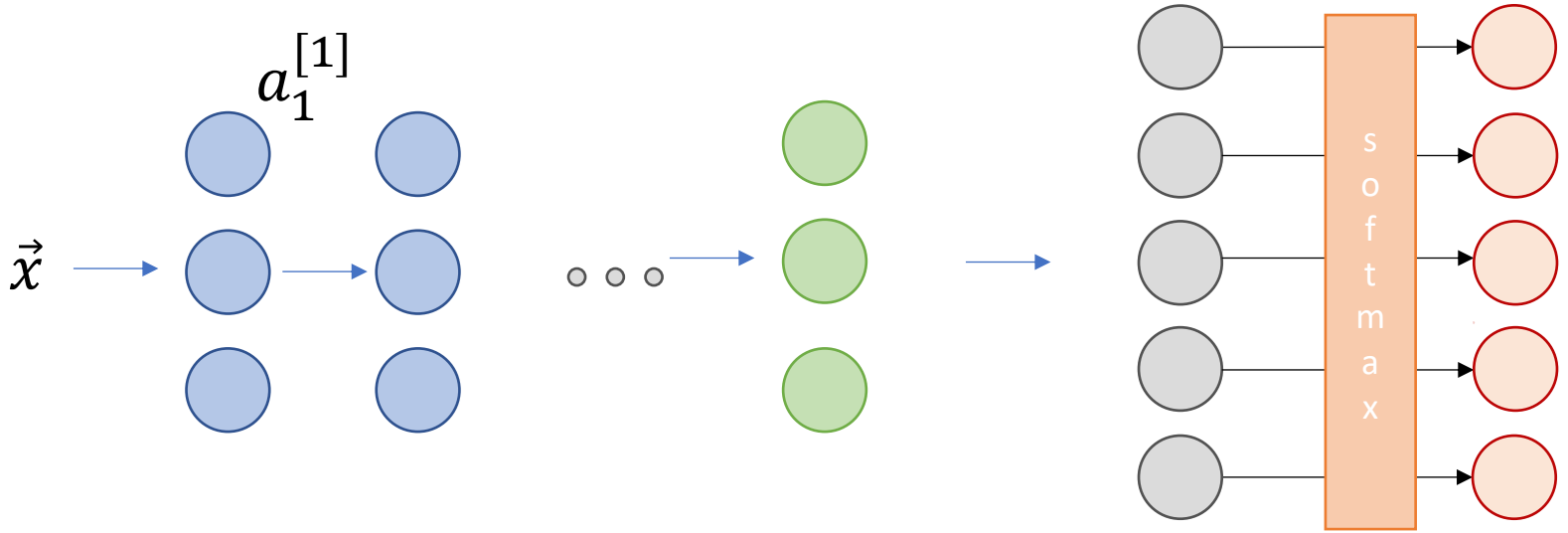
서울과학기술대학교 SEOUL NATIONAL UNIVERSITY OF SCIENCE & TECHNOLOGY

영진전문대학교 YEUNGJIN UNIVERSITY

Disadvantages of using MLP

Disadvantages of using MLP for image classification

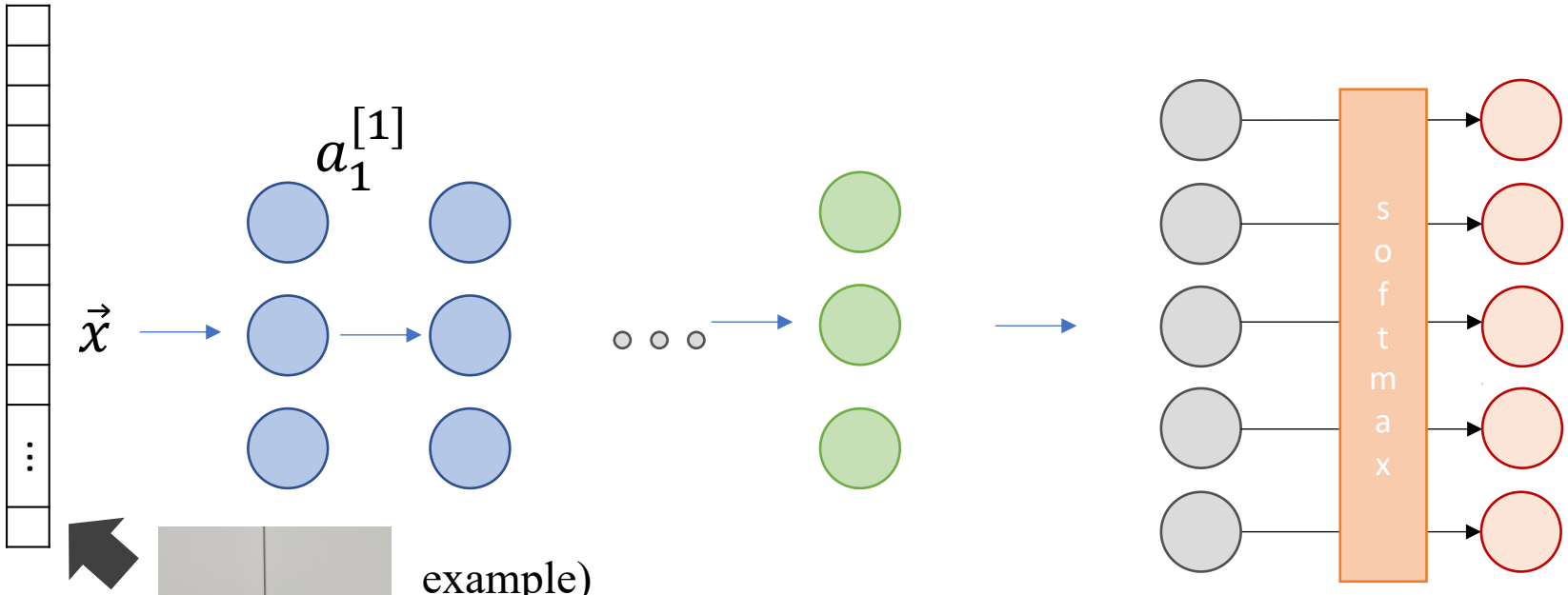
- 많은 연산을 요구함
- 이미지 안에 가까운 픽셀간의 관계를 다루지 못함
- 이미지 내에 물체(Object)가 존재하는 위치에 예민하게 됨

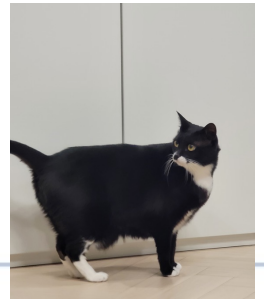


Disadvantages of using MLP

Disadvantages of using MLP for image classification

- 많은 연산을 요구함
- 이미지 안에 가까운 픽셀간의 관계를 다루지 못함
- 이미지 내에 물체(Object)가 존재하는 위치에 예민하게 됨





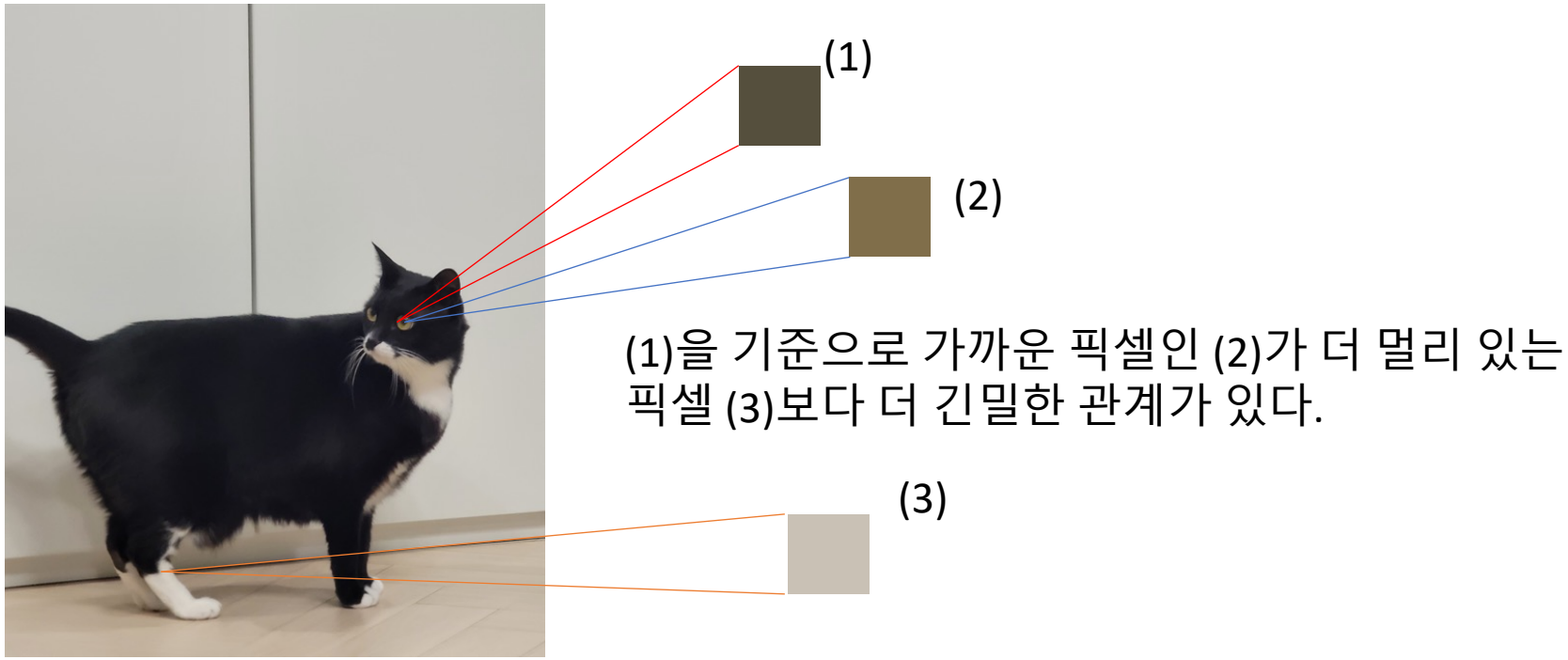
example)

- Image size(FHD) = 1920 x 1080 x 3
- weights for $a_1^{[1]}$= 1920 x 1080 x 3 = 6,220,800 (6M)

Disadvantages of using MLP

Disadvantages of using MLP for image classification

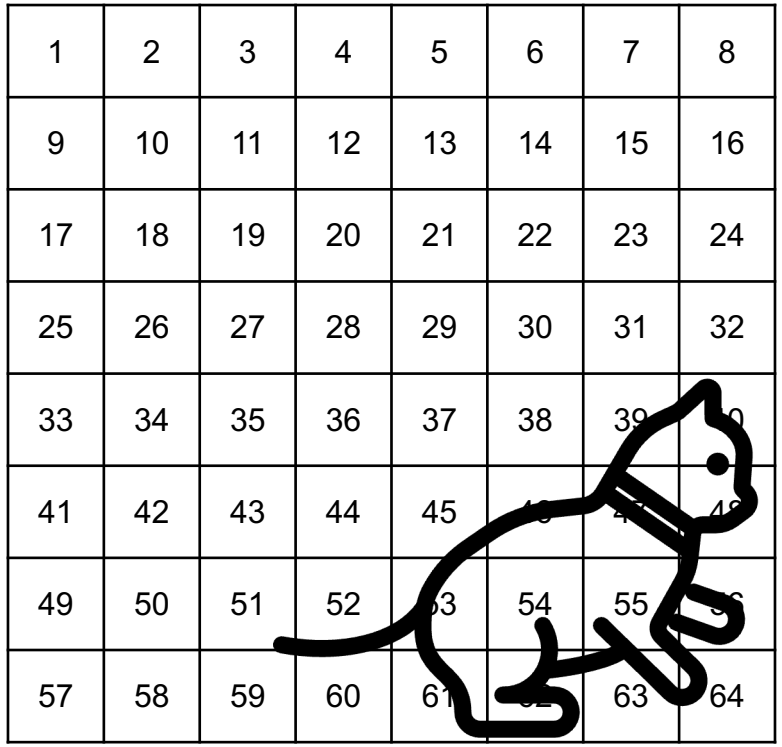
- 많은 연산을 요구함
- 이미지 안에 가까운 픽셀간의 관계를 다루지 못함
- 이미지 내에 물체(Object)가 존재하는 위치에 예민하게 됨

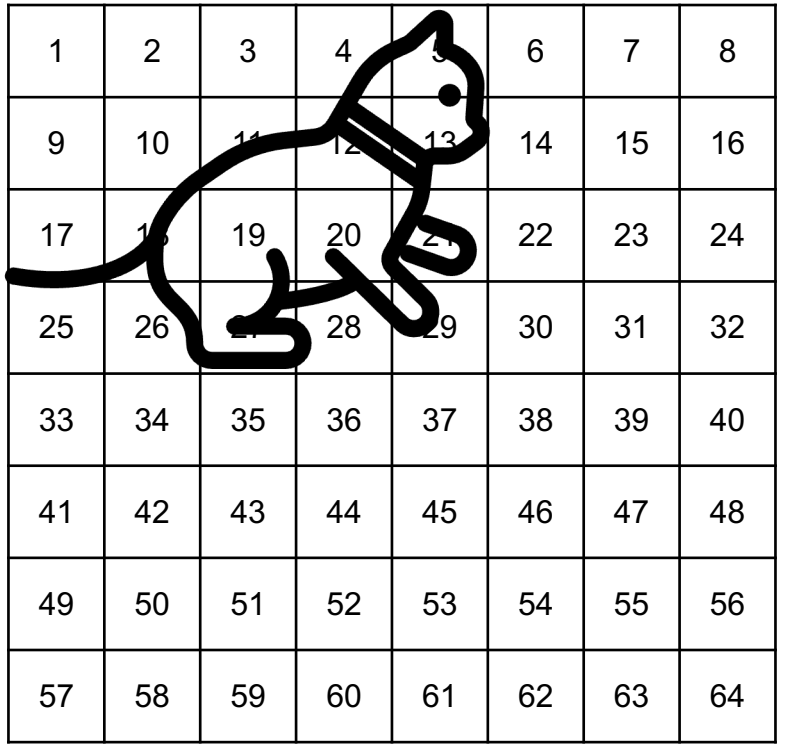


Disadvantages of using MLP

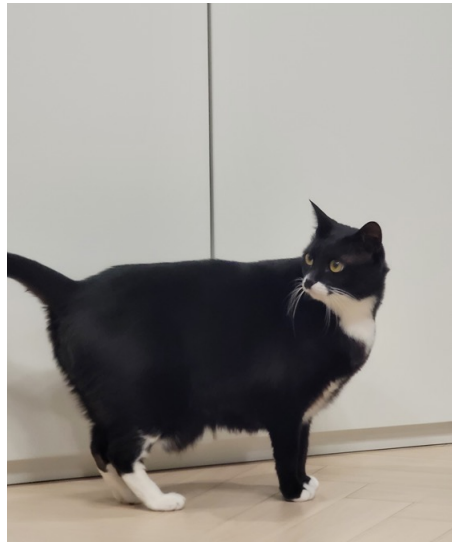
Disadvantages of using MLP for image classification

- 많은 연산을 요구함
- 이미지 안에 가까운 픽셀간의 관계를 다루지 못함
- 이미지 내에 물체(Object)가 존재하는 위치에 예민하게 됨

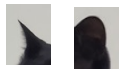




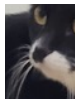
How does human recognize images?



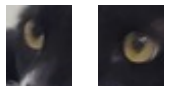
고양이의 귀? = 예/아니오

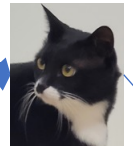


고양이의 코? = 예/아니오



고양이의 눈? = 예/아니오

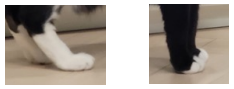




고양이의 머리? = 예/아니오

고양이? = 예/아니오

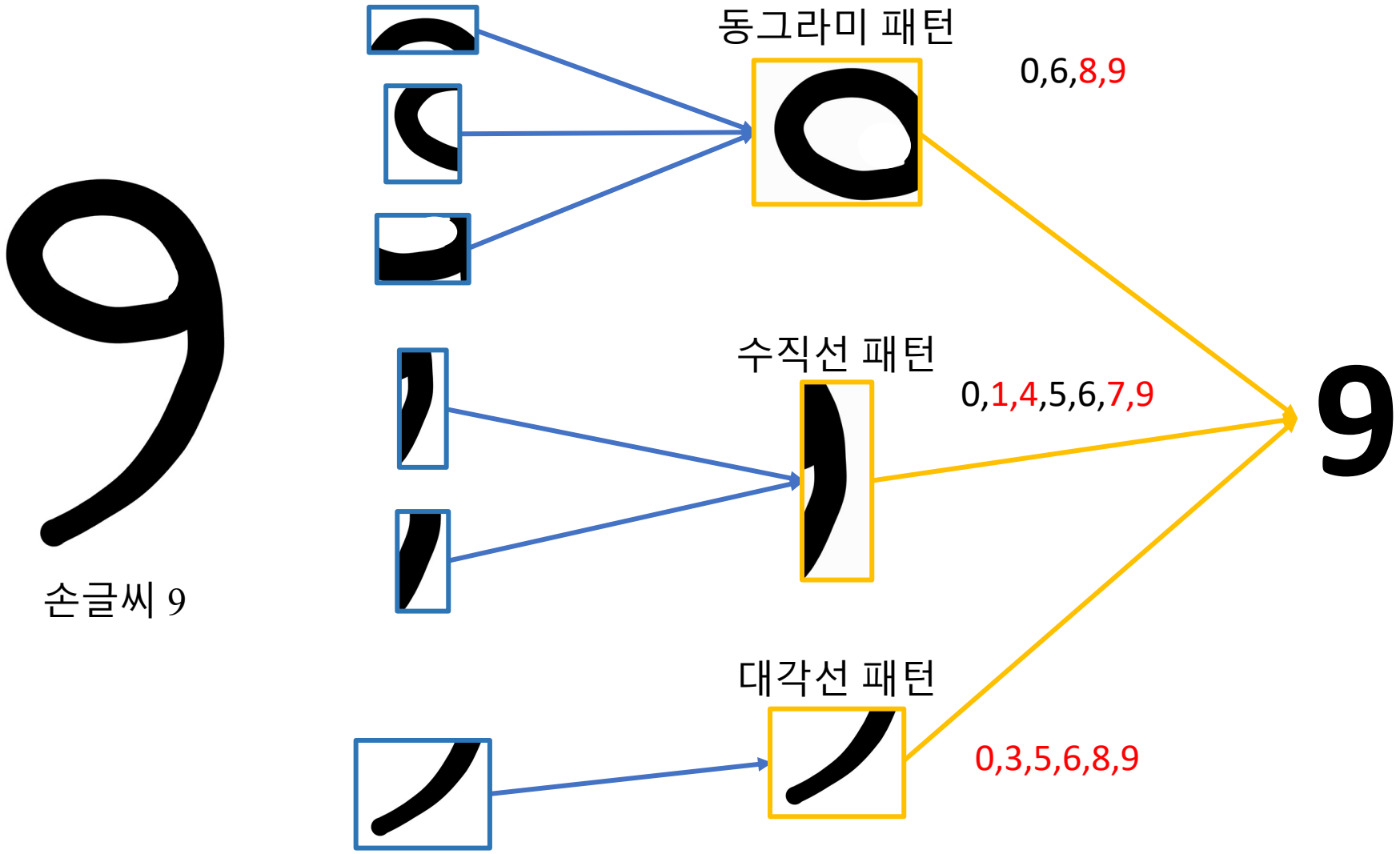
고양이 다리? = 예/아니오



고양이 꼬리? = 예/아니오

고양이의 몸? = 예/아니오

Recognize the handwritten digit 9



Recognize the handwritten digit 9

손글씨 9

-1	1	1	1	-1
-1	1	-1	1	-1
-1	1	1	1	-1
-1	-1	-1	1	-1
-1	-1	-1	1	-1
-1	-1	1	-1	-1
-1	1	-1	-1	-1

-1	1	1	1	-1
-1	1	-1	1	-1
-1	1	1	1	-1
-1	-1	-1	1	-1
-1	-1	-1	1	-1
-1	-1	1	-1	-1
-1	1	-1	-1	-1

동그라미 패턴
filter

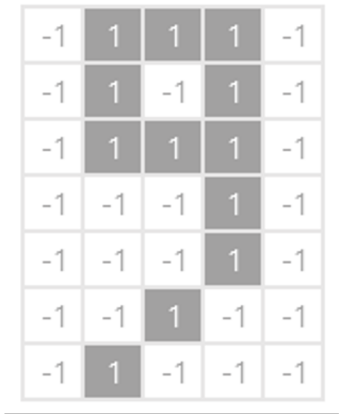
-1	1	1	1	-1
-1	1	-1	1	-1
-1	1	1	1	-1
-1	-1	-1	1	-1
-1	-1	-1	1	-1
-1	-1	1	-1	-1
-1	1	-1	-1	-1

수직선 패턴
filter

-1	1	1	1	-1
-1	1	-1	1	-1
-1	1	1	1	-1
-1	-1	-1	1	-1
-1	-1	-1	1	-1
-1	-1	1	-1	-1
-1	1	-1	-1	-1

대각선 패턴
filter

Convolution Operation



-1	1	1	1	-1
-1	1	-1	1	-1
-1	1	1	1	-1
-1	-1	-1	1	-1
-1	-1	-1	1	-1
-1	-1	1	-1	-1
-1	1	-1	-1	-1

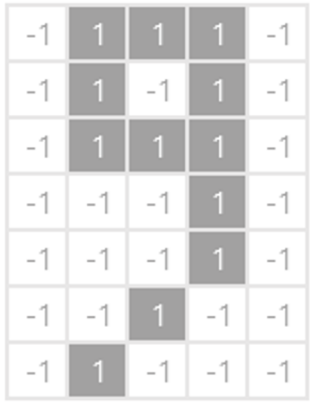
* 1/9

1	1	1
1	-1	1
1	1	1

=

-0.11		

Feature Map



-1	1	1	1	-1
-1	1	-1	1	-1
-1	1	1	1	-1
-1	-1	-1	1	-1
-1	-1	-1	1	-1
-1	-1	1	-1	-1
-1	1	-1	-1	-1

* 1/9

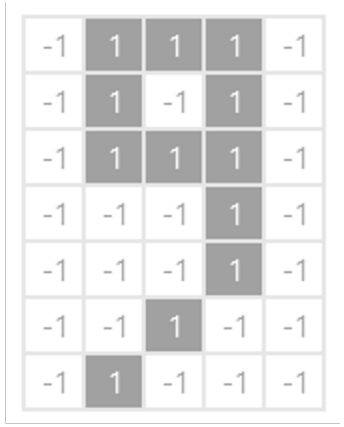
1	1	1
1	-1	1
1	1	1

=

-0.11	1	

Feature Map

Convolution Operation



-1	1	1	1	-1
-1	1	-1	1	-1
-1	1	1	1	-1
-1	-1	-1	1	-1
-1	-1	-1	1	-1
-1	-1	1	-1	-1
-1	1	-1	-1	-1

* 1/9

1	1	1
1	-1	1
1	1	1

=

-0.11	1	-0.11
-0.55	0.11	-0.33
-0.33	0.33	-0.33
-0.55	-0.11	-0.55
-0.33	-0.55	-0.33

Feature Map

Convolution Operation

*

1	1	1
1	-1	1
1	1	1

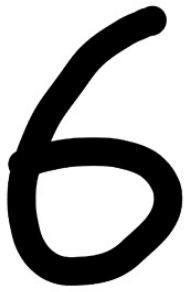
동그라미 패턴

filter

=

	1	

Feature Map



*

1	1	1
1	-1	1
1	1	1

동그라미 패턴

filter

=

	1	

Feature Map

*

1	1	1
1	-1	1
1	1	1

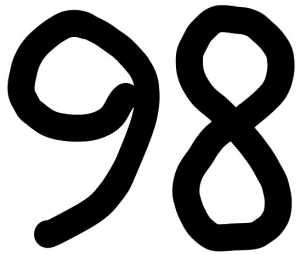
동그라미 패턴

filter

=

	1	
	1	

Feature Map



*

1	1	1
1	-1	1
1	1	1

동그라미 패턴

filter

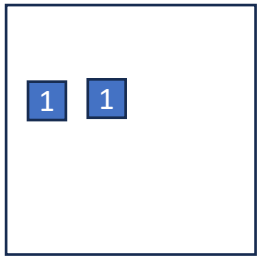
=

	1		1	
			1	

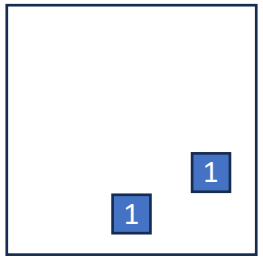
Feature Map

Recognize the cat

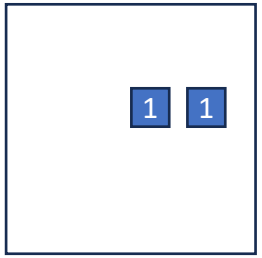
* 눈 탐지기 filter =



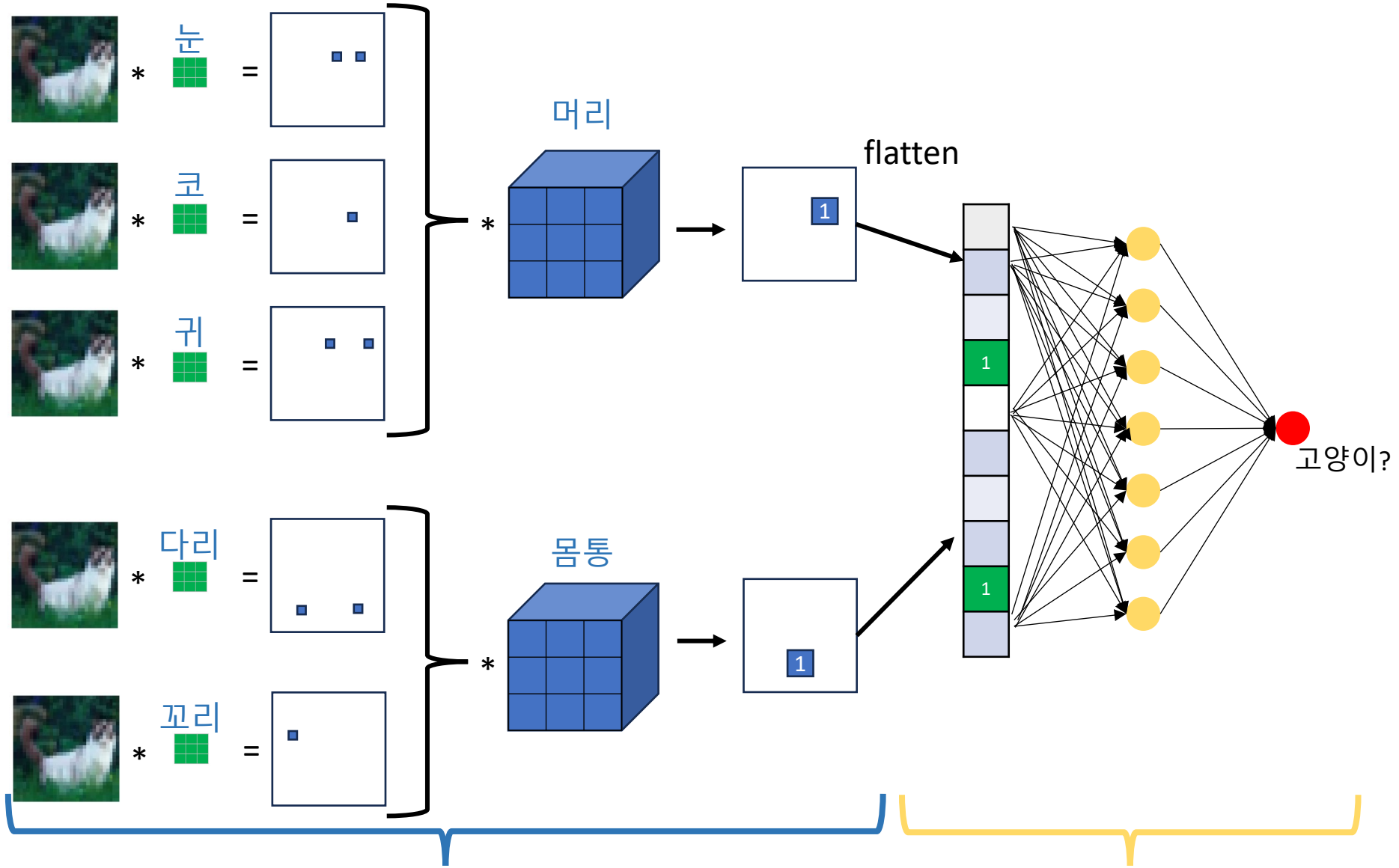
* 눈 탐지기 filter =



* 눈 탐지기 filter =



Recognize the cat



ReLU Function

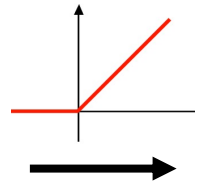
-1	1	1	1	-1
-1	1	-1	1	-1
-1	1	1	1	-1
-1	-1	-1	1	-1
-1	-1	-1	1	-1
-1	-1	1	-1	-1
-1	1	-1	-1	-1

*

-1	1	-1
1	-1	-1
-1	-1	-1

=

-0.11	1	-0.11
-0.55	0.11	-0.33
-0.33	0.33	-0.33
-0.55	-0.11	-0.55
-0.33	-0.55	-0.33

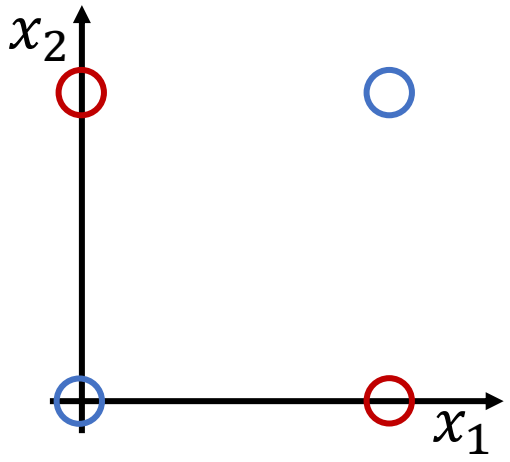


ReLU

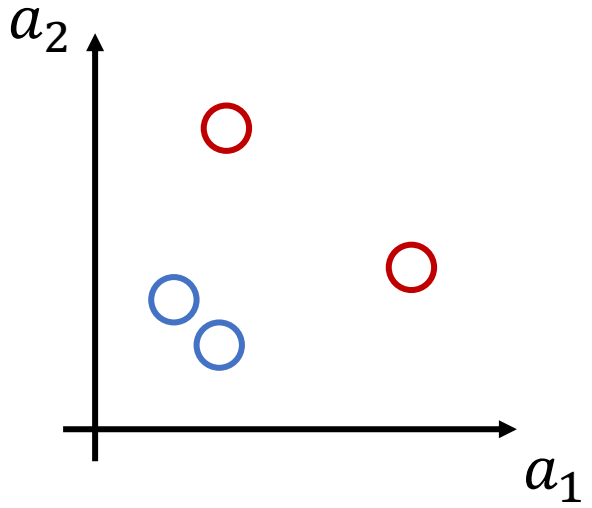
0	1	0
0	0.11	0
0	0.33	0
0	0	0
0	0	0

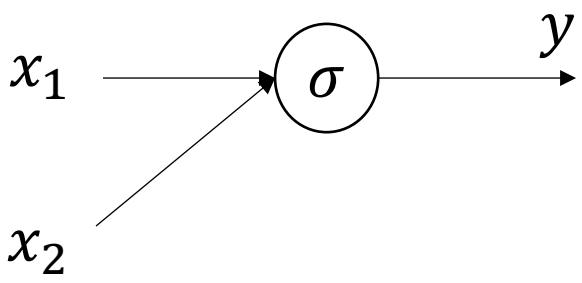
- $ReLU(x) = Max(0, x)$
- ReLU는 모델을 non-linear 하게 만듦

XOR Problem

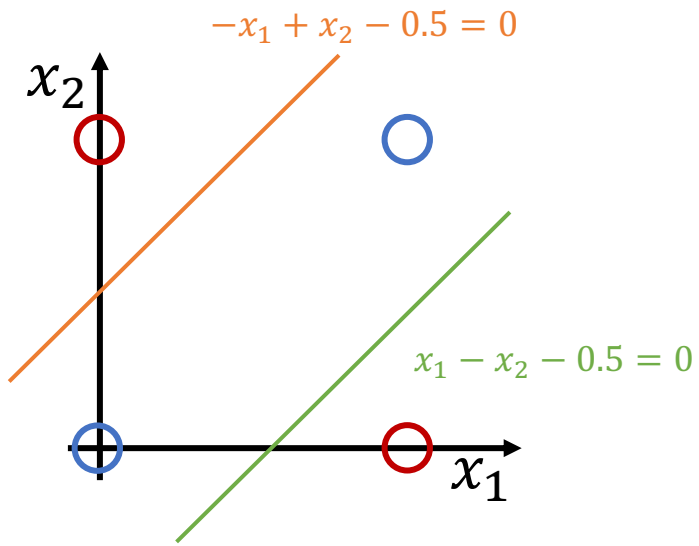


x_1	x_2	y
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**0**

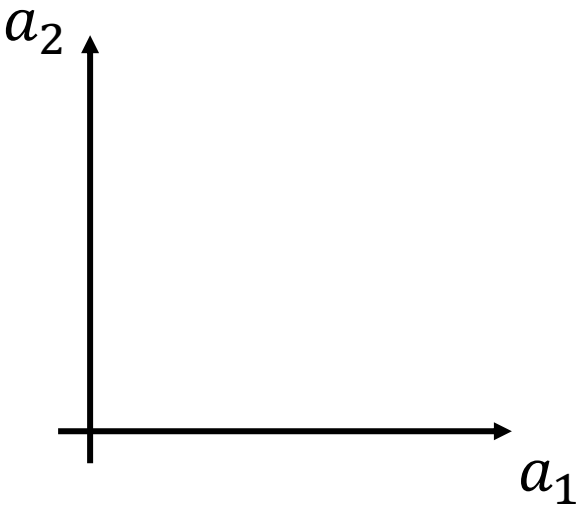


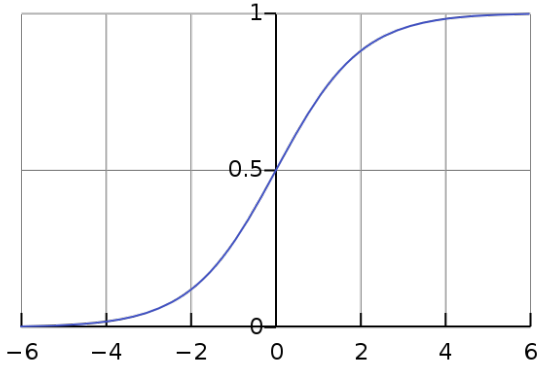


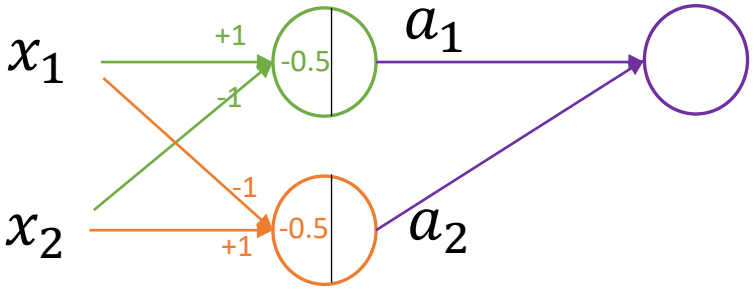
XOR Problem with sigmoid



x_1	x_2	y
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**0**

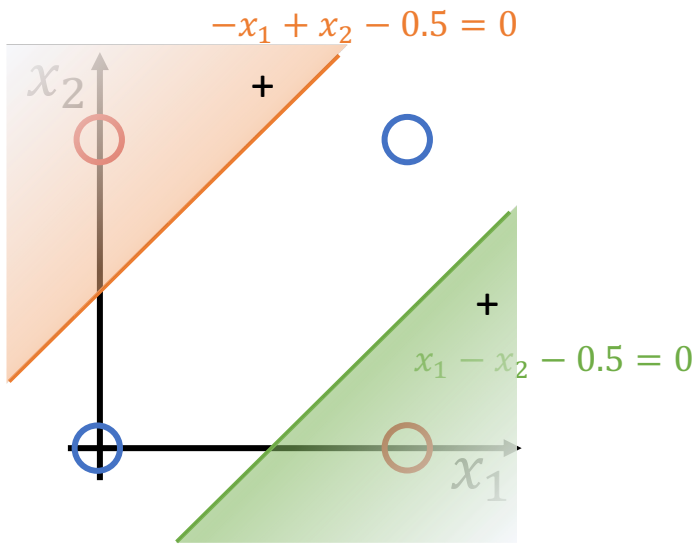




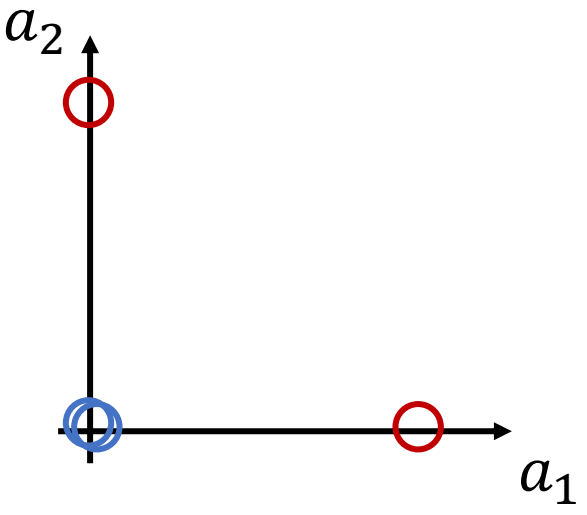


x_1	x_2	a_1	a_2	y
0	0			**0**
0	1			**1**
1	0			**1**
1	1			**0**

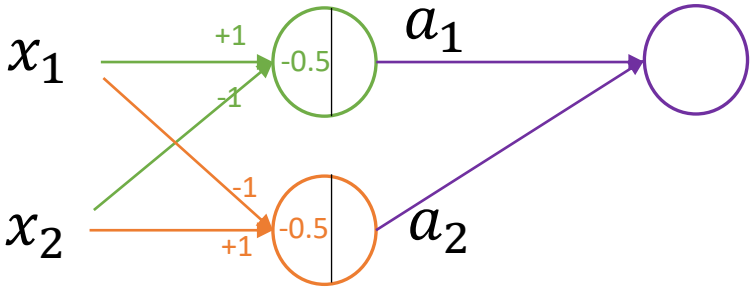
XOR Problem with sigmoid



x_1	x_2	y
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**0**

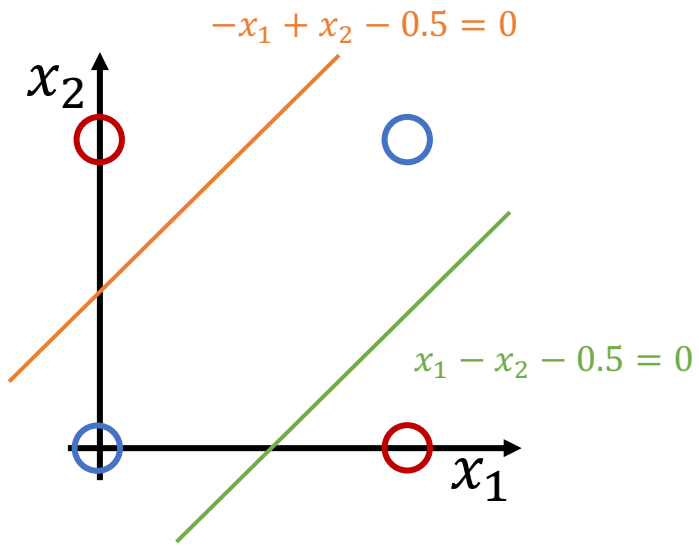


x_1	x_2	a_1	a_2	y
0	0	0	0	**0**
0	1	0	1	**1**
1	0	1	0	**1**
1	1	0	0	**0**

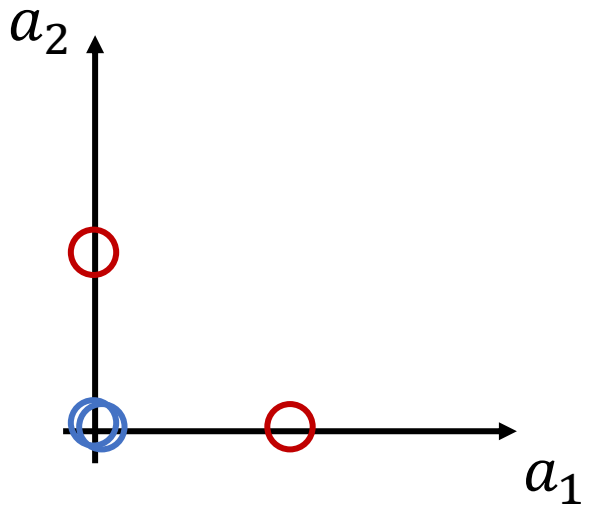


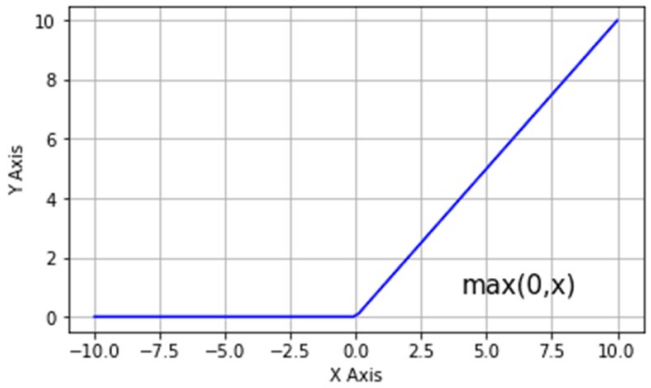
x_1	x_2	a_1	a_2	y
0	0	0	0	**0**
0	1	0	1	**1**
1	0	1	0	**1**
1	1	0	0	**0**

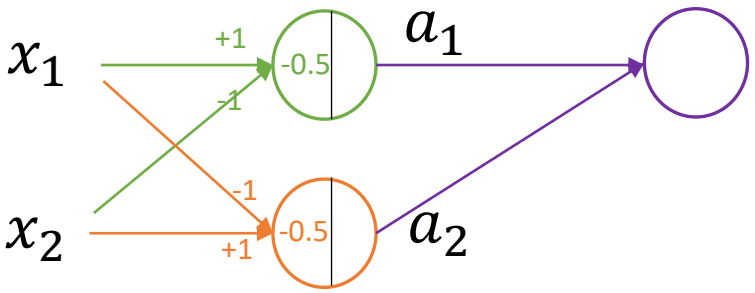
XOR Problem with ReLU



x_1	x_2	y
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**0**



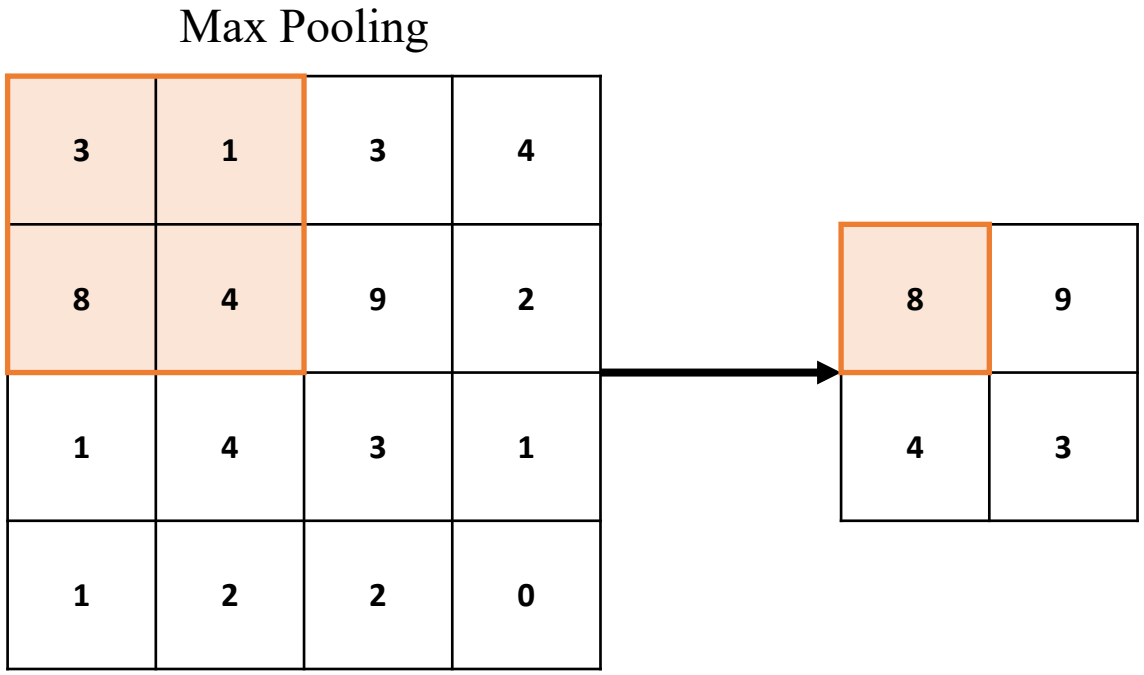




x_1	x_2	a_1	a_2	y
0	0	0	0	**0**
0	1	0	0.5	**1**
1	0	0.5	0	**1**
1	1	0	0	**0**

Pooling Layer

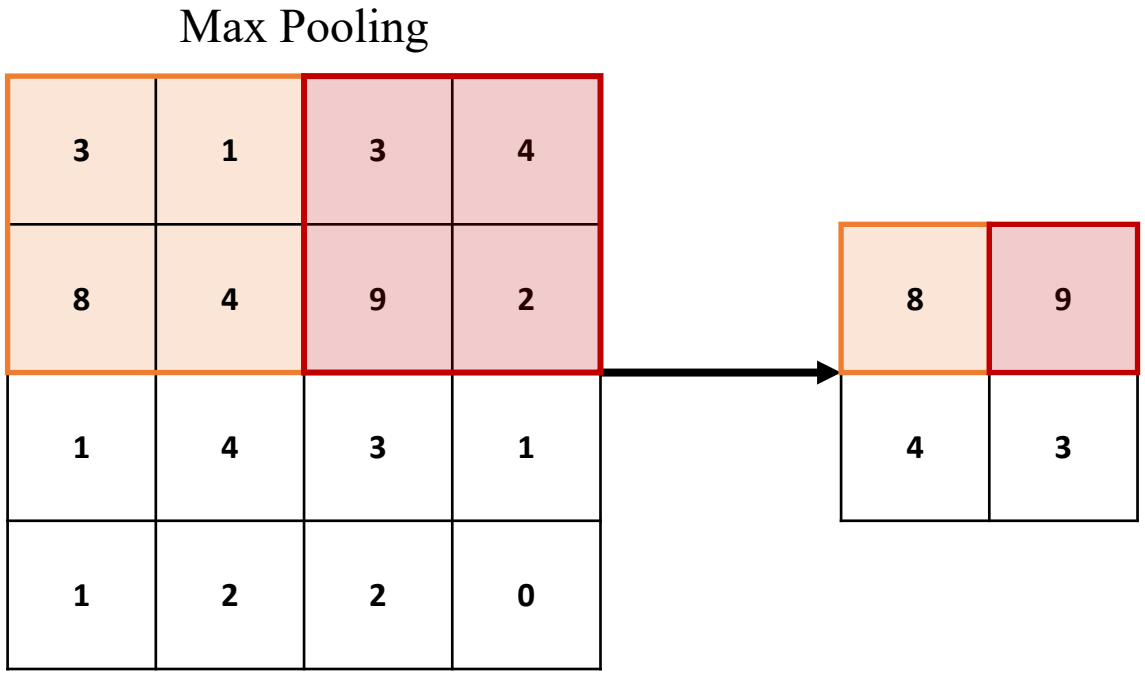
2 x 2 filter with stride = 2



- Pooling Layer 는 공간적 크기와 계산량을 줄이는 데 사용
- 학습해야 하는 파라미터가 없고, 다음 layer의 계산량을 줄이는데 기여
- 입력의 변형 또는 왜곡에 대해서 민감하지 않음

Pooling Layer

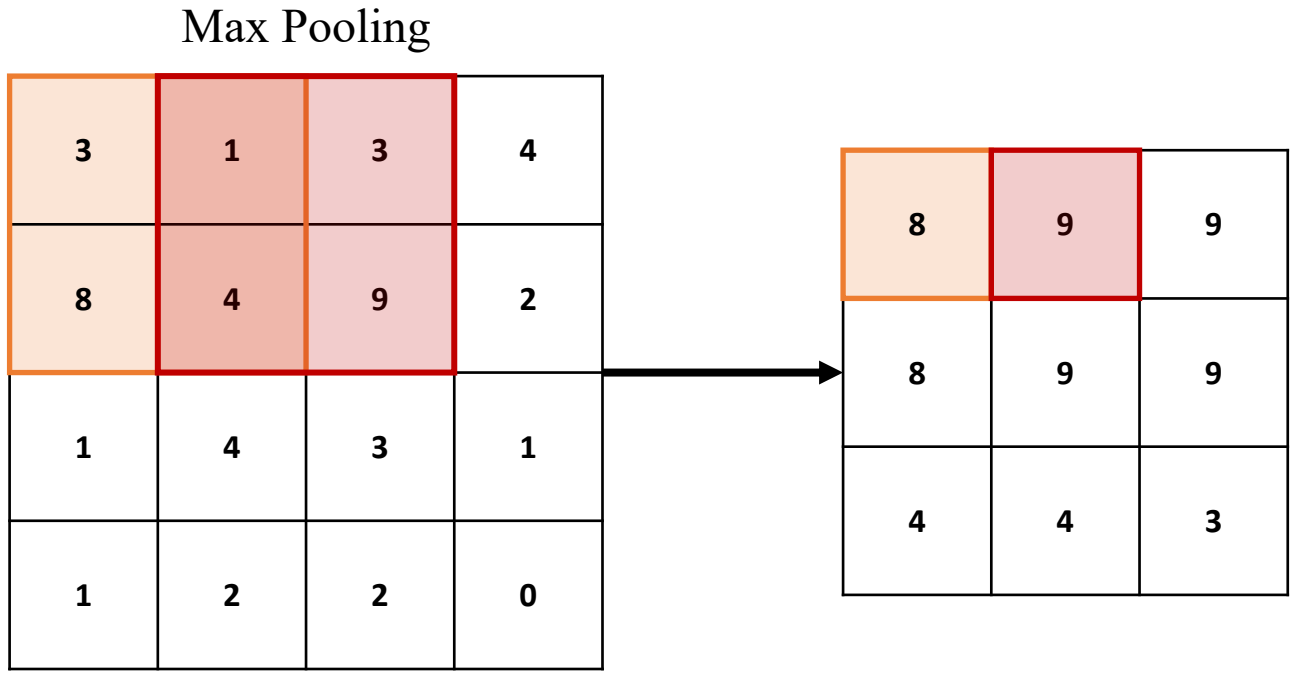
2 x 2 filter with stride = 2



- Pooling Layer 는 공간적 크기와 계산량을 줄이는 데 사용
- 학습해야 하는 파라미터가 없고, 다음 layer의 계산량을 줄이는데 기여
- 입력의 변형 또는 왜곡에 대해서 민감하지 않음

Pooling Layer

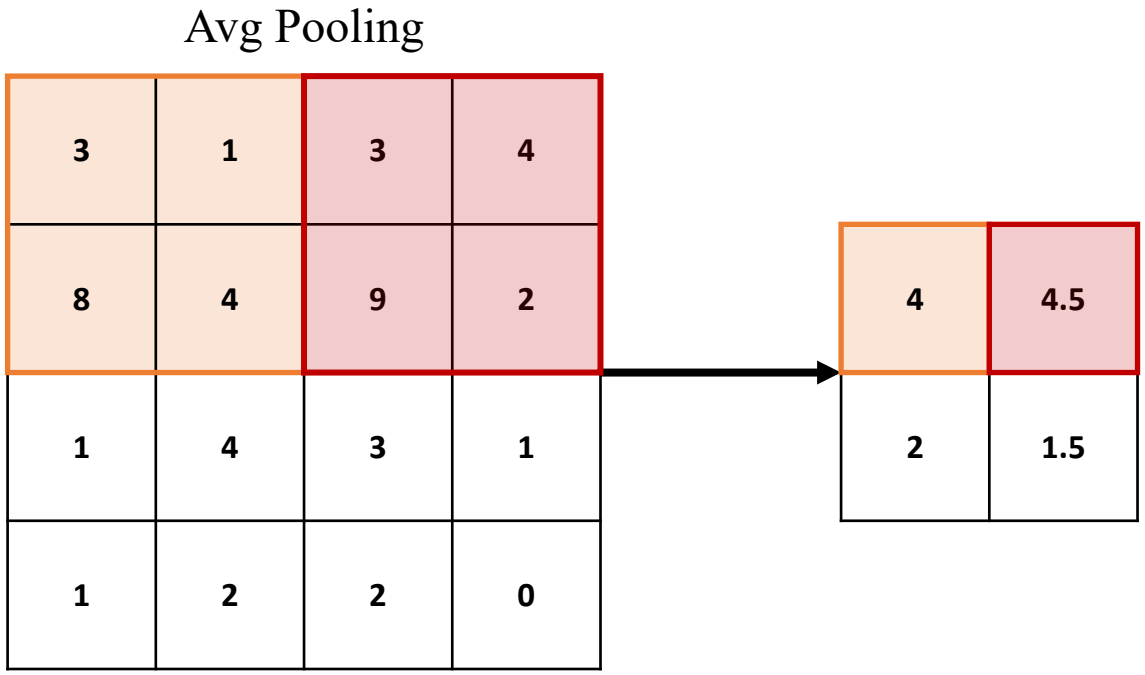
2 x 2 filter with stride = 1



- Pooling Layer 는 공간적 크기와 계산량을 줄이는 데 사용
- 학습해야 하는 파라미터가 없고, 다음 layer의 계산량을 줄이는데 기여
- 입력의 변형 또는 왜곡에 대해서 민감하지 않음

Pooling Layer

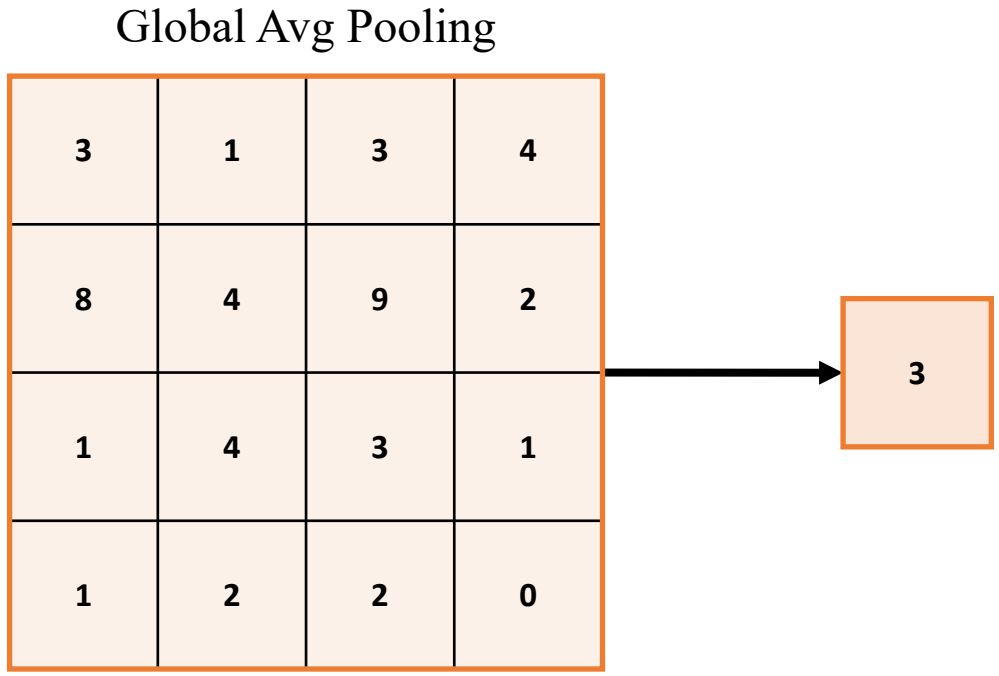
2 x 2 filter with stride = 2



- Pooling Layer 는 공간적 크기와 계산량을 줄이는 데 사용
- 학습해야 하는 파라미터가 없고, 다음 layer의 계산량을 줄이는데 기여
- 입력의 변형 또는 왜곡에 대해서 민감하지 않음

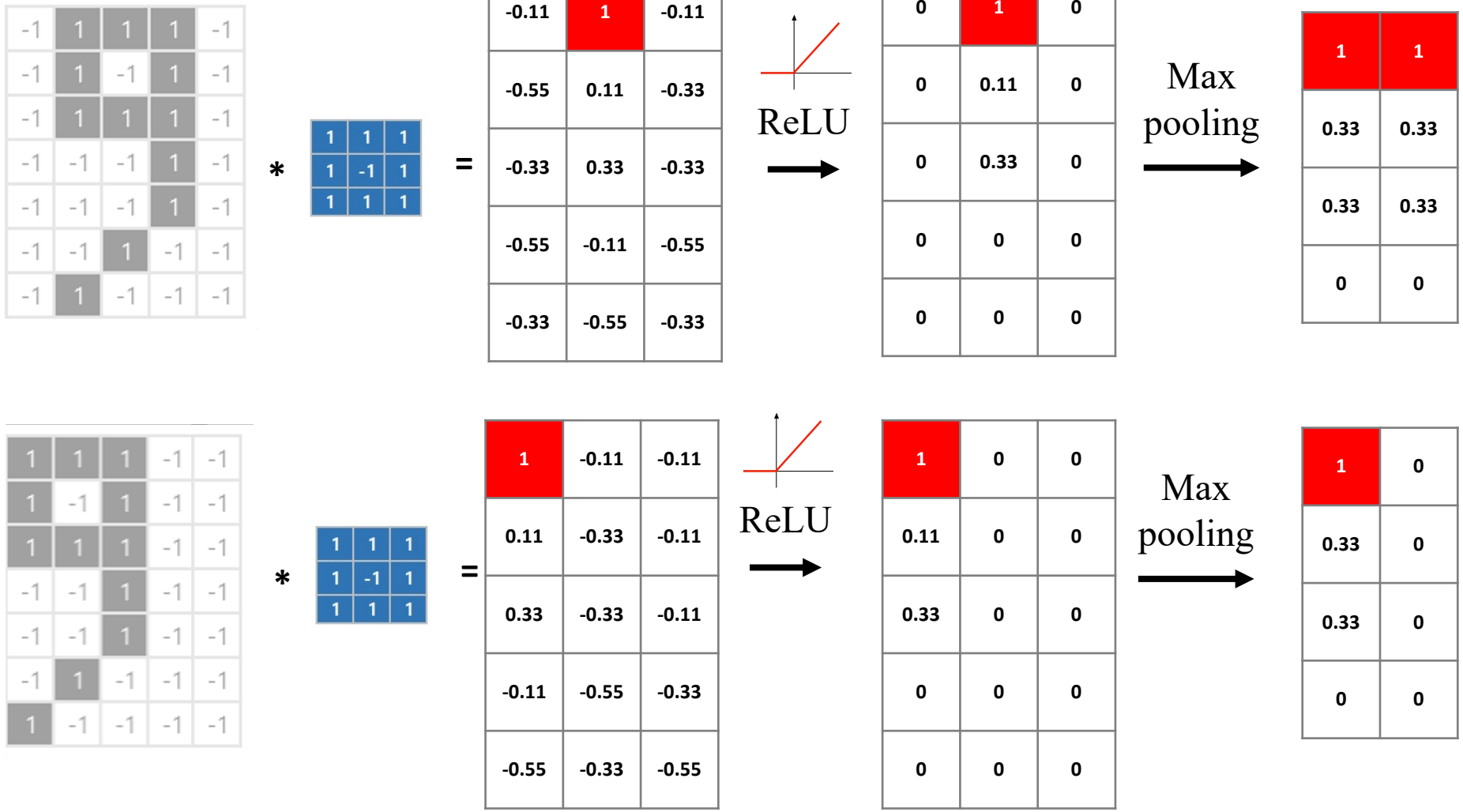
Pooling Layer

No filter and stride size

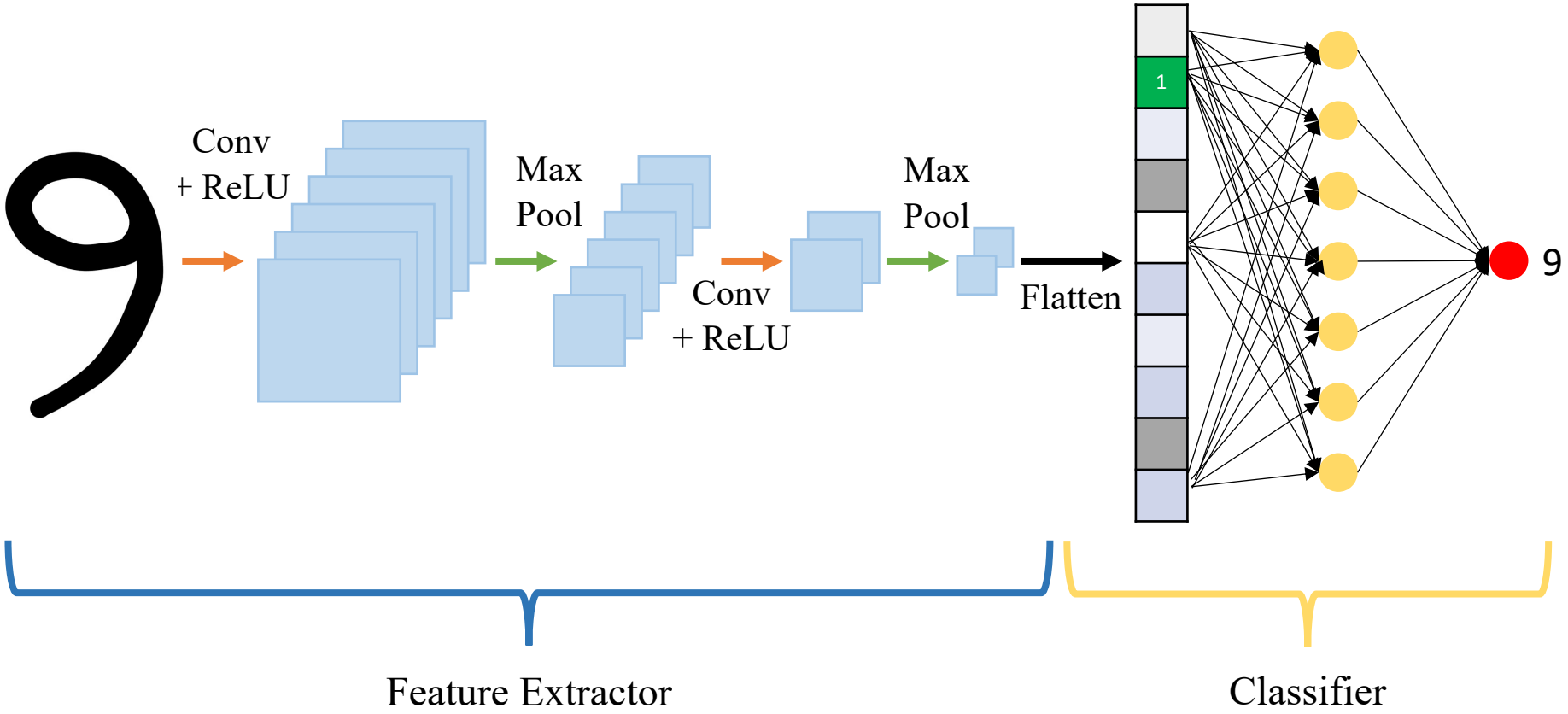


- 공간정보를 하나의 값으로 표현한다.
- Feature extractor의 마지막 출력에 해당하는 feature에 Global Avg Pooling을 적용하여 3-d tensor를 vector로 변형하는데 주로 사용된다.

Pooling Layer



Classification



교육부 Ministry of Education

NRF 한국연구재단 National Research Foundation of Korea

thank you

본 과제(결과물)는 교육부와 한국연구재단의 재원으로 지원을 받아 수행된
디지털신기술인재양성 혁신공유대학사업의 연구결과입니다.

Convergence and Open Sharing System COSS 인공지능 · 전남대학교 CHONNAM NATIONAL UNIVERSITY · 성균관대학교 SUNGKYUNKWAN UNIVERSITY · KNU 경북대학교 KYUNGPOOK NATIONAL UNIVERSITY · 전주대학교 JEONJU UNIVERSITY · 서울시립대학교 UNIVERSITY OF SEOUL · 서울과학기술대학교 SEOUL NATIONAL UNIVERSITY OF SCIENCE & TECHNOLOGY · 영진전문대학교 YEUNGJIN UNIVERSITY