딥러닝 발전에 힘입어 가장 많이 발전한 기술

CNN을 이용한 영상분석으로 인해서 컴퓨터 비전 분야가 각광받게 되었다.

Fully connected layer로만 이루어져 있는 경우의 한계점

# DNN의 한계

DNN의 입력 데이터는 1D데이터

각각의 위치가 하나의 Attribute(특성) 각각은 특성을 나타내는 값들이 채워져 있다.

* 입력데이터로 영상을 넣고자 하는 경우, DNN의 특성에 따른 한계

1. 2D 데이터를 1D로 평탄화 하여 DNN 적용할 수 있다.

ex) MNIST예제.

28\*28의 2D데이터를 -> 784 1D로 평탄화

최종적으로 10개의 클래스를 결정짓는 출력노드를 만든다.

1. DNN모델은 위치에 상관없이 동일한 수준의 중요도를 갖는다.

하지만 실제 2D의 경우 위치정보가 중요하다. ex) 6과 9는 같지 않음

1. 영상데이터가 위치정보를 인코딩할 수 없다. 6p

조금만 픽셀을 이동해도 새로운 데이터로 이해한다. = 입력데이터가 달라지게 된다.

크기, 회전, 변형과 같은 모든 상황을 일일이 학습시켜주지 않는 이상 좋은 결과를 얻기는 어렵다.

🡪 DNN 구조의 단점 7p

학습시간이 오래걸린다. 크기와 회전, 이동변형 등의 데이터를 모두 학습해야 하므로.

또한 입력 영상이 커지면 연결노드의 개수(weight의 개수)가 기하급수적으로 증가한다.

이렇게 입력영상이 커지고, Layer가 깊어지면 모델의 크기가 커지고, 학습해야하는 파라미터의 수가 많아진다.

# 해결방안 🡪 **CNN** 8p

: 영상, 2D domain, 공간상의 위치 정보가 중요한 영상데이터를 다룰 수 있는 방안.

사진데이터, 2D영상데이터를 전처리없이 영상데이터를 그대로 넣는다.

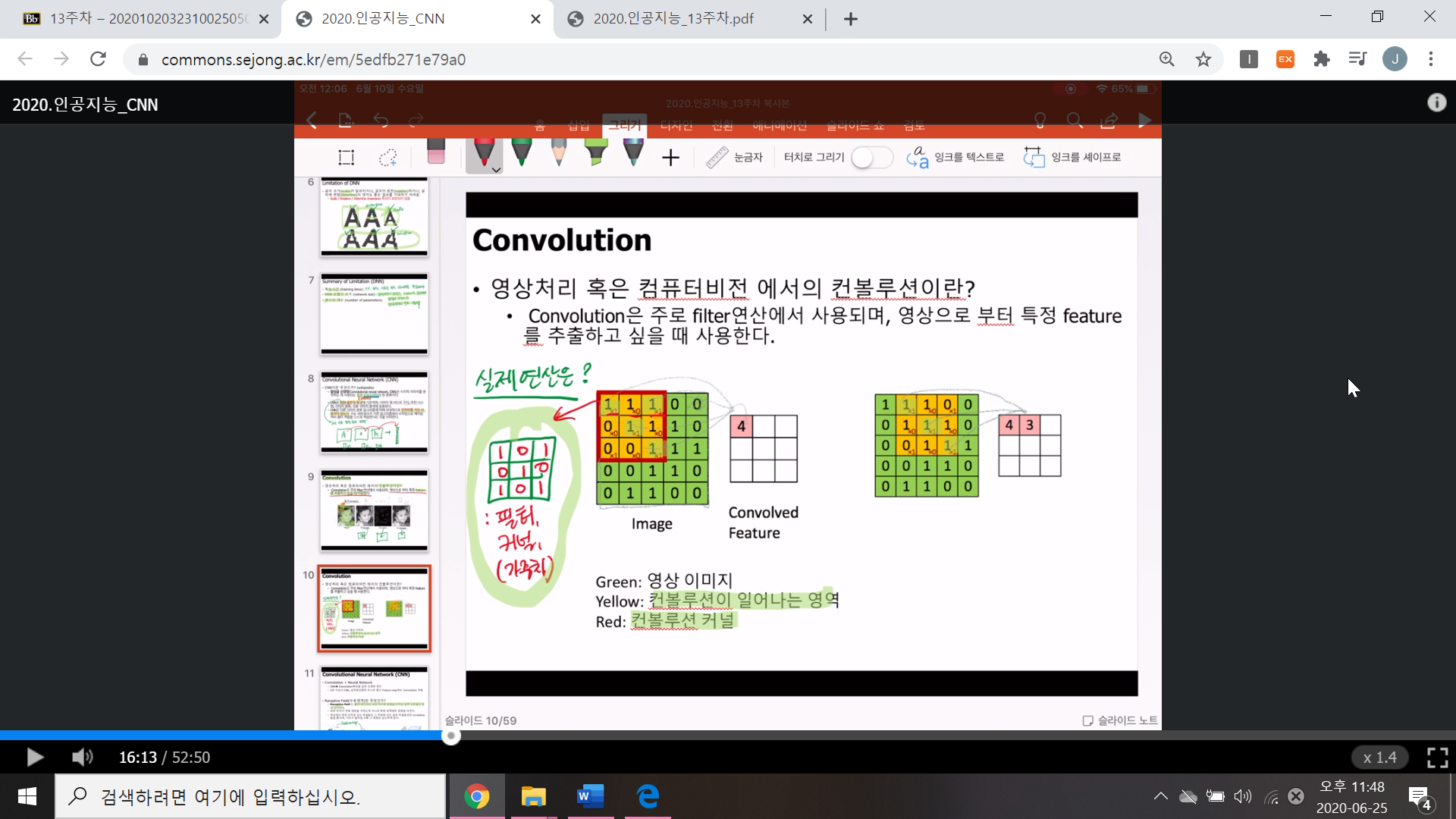
시각적 이미지: 영상, 사진, 비디오 데이터

변환 불변성 특성: 크기, 이동, 회전 등의 변형에 강인하다. 불변하다.

변형되어도 다 같은 하나의 벡터로 기술된다.

## 10p **convolution**

: filter연산에서 사용되며, 영상으로 부터 특정 feature를 추출하고 싶을 때 사용한다.



하나하나의 픽셀 값으로 이루어져 있다.

Yellow내의 빨간색 값들은 필터, 커널의 값이다.

컨볼루션 커널이 3\*3으로 계속 이동하면서 값을 채워 넣는다.

곱했을 경우 실제로 1이 남는 픽셀은 총 4개. (픽셀\*커널값)

곱했을 경우 실제로 1이 남는 픽셀은 총 3개

🡪 원본 영상위에 커널(필터)를 스캐닝해서 값을 각각의 영상에 채워 넣는 과정.

## 11p **CNN**

: Convolution을 여러 번 중첩시키는 것 (Convolution + NN)

: 필터연산을 입력영상 뿐만 아니라 중간 feature map에도 convolution을 적용.

지속적으로 값을 추출해낸다.

\* **Receptive Field**(수용영역) 이란?

연산을 통해 나온 파란색 값을 결정짓기 위해 필요한 영역 32\*32\*3

커널은 5\*5\*3. 값을 결정짓기 위해 사용된 입력영상의 영역은 5\*5\*3이다.

이와 같은 Convolution layer의 receptive field의 크기는 5\*5이다.

Layer가 여러 개가 쌓일수록 주변에 있는 일부 픽셀들만 영향을 미친다.

12p CNN의 장점

DNN은 2D데이터 영상데이터를 1D로 만들었다.

하지만 CNN은 평탄화 하지 않고 그대로 사용할 수 있어, 공간적 정보를 그대로 유지할 수 있다.

CNN은 공간적 정보를 유지할 수 있도록 Input은 2D, 중간 정보는 depth를 포함한 3D.

12p **DNN과 CNN 비교**

- 필터의 크기가 매우 차이가 많이 난다.

이미지가 32\*32\*3일 때,

DNN은 10개의 클래스가 출력을 위해 3072를 10으로 만드는 가중치W(3072\*10)가 필요하다.

CNN은 이미지의 Receptive field를 커버할 수 있는 5\*5\*3필터를 임의로 정의해서 스캐닝하여 사용한다. 한 번에 커버하는 것이 아니라 부분부분을 연산을 통해 판단하는 디자인 설계 기법이다.

🡪 장점: 적은 수의 가중치(파라미터)를 갖는다.

Convolution 커널은 실제 사람이 부분의 합으로 판단하는 인간의 뇌를 모사한 방식이다.

14p CNN모델의 예시

DNN: (NN+ RELU+ Dropout) + (NN+ RELU+ Dropout) + ()…

모듈을 쌓아서 MNIST 분류문제를 해결

CNN: (**CONV**+ RELU+ **CONV**+ RELU+ POOL) + (CONV+ RELU+ CONV+ RELU+ POOL) + ()…

Convolution layer를 이용해서 분류문제를 풀 수 있다

15p 입력영상의 커널이 어떻게 동작하는지

32\*32\*3에서

- 3이 의미하는 것은? 컬러정보 R+G+B 데이터의 합이다.

컬러영상은 R, G, B 3채널로 이루어져 있다. (흑백영상만 1채널)

5\*5\*3 convolution을 설계

영상에 5\*5\*3 필터를 씌운 후 연산을 하면 하나의 값이 나온다. Wx+b

모든 영역을 커버할 때까지 커널을 움직여서 연산을 한다.

커널을 끝까지 움직여서 모든 영역을 커버하고 나면 28\*28 activation maps을 얻을 수 있다.

5\*5\*3 필터를 6개를 사용 🡪 28\*28\*6 (activation maps의 feature)

5\*5\*3 필터로 모든 영역을 커버하는데, 이 과정을 6번반복한다.

25p

커널을 영상위에서 돌릴 때의 가정: Stride==1

Stride: 커널을 움직일 때, 센터를 한 칸씩 움직이는 것.

ex) 25p~29p

7\*7 input이 stride=1을 통해 5\*5 output이 나온다.

31p

입력데이터가 N\*N, 커널의 크기가 F\*F, Stride가 주어졌을 때

Output size = (N-F) / stride+ 1

예시를 보면, 영상사이즈가 계속 줄어든다..

32p **Padding** 영상사이즈가 줄어들지 않고 유지하는 방법

: 원래 영상사이즈 주변 값을 0으로 채워서 영상사이즈를 키우기 (=zero padding)

- Pad 1은 한 칸 padding, pad 2는 두 칸 padding

- 커널 사이즈에 따른 pad를 구하는 방법: F/2에서 올림하기

33p 레이어의 파라미터 개수 구하기

Input은 32\*32\*3 커널은 5\*5 10개.

(5\*5\*3 + 1)\*10개 = 760개의 파라미터

5\*5\*3=weight, 1=bias

- Stride와 pad는 파라미터가 아니다.

35p W, H, K, S, P에 따라 출력이 어떻게 결정되어지는지

36p~38p Convolution layer

: 입력영상에 필터를 정의하고 연산하는 레이어

32\*32\*3이미지에 필터 5\*5\*3을 입히면

크기가 줄어든 feature map(=activation map) 하나가 나온다.

커널의 개수만큼의 feature map이 생긴다.

커널의 개수=Depth(6개)

39p weight variable 개수 구하기

필터크기\*필터개수

(5\*5\*3)\*6 (5\*5\*6)\*10

Input의 커널 5\*5\*3에서 depth 3은 컬러 입력채널을 의미. (필터개수가 아닌)

이후 layer에서는 depth는 필터의 개수

40p

Activation map: 입력영상에 convolution layer를 통과시킨 후의 결과물

물리적 의미: Low level feature는 edge성분들을, High level feature는 low level feature을 기술할 수 있도록 되어있다.

## 41p **Pooling layer**

: feature map을 resizing

32\*32 -> 16\*16으로 sampling

42p pooling layer의 연산방법

Max pool은 filter에서 가장 큰 값을 추출하는 것.

원본사이즈를 sampling, resize하는 효과가 있다.

44p max pooling

- computational complexity를 감소시킨다.

- 여러가지 pooling기법이 있다.

Average pooling: filter의 평균값으로 채우는 것.

Structure를 보존하되 이미지가 스무스해진다.

= 중간중간 튀는 값들을 필터링한다.

- Pooling은 depth를 줄이지 않고, height와 width만 줄인다.

32\*32\*3 🡪 16\*16\*3 depth는 유지된다.

45p

FC layer = NN

Pooling layer가 [1\*1024] 라면,

5개의 클래스로 출력하기 위해 w=1024\*5 **FC layer**