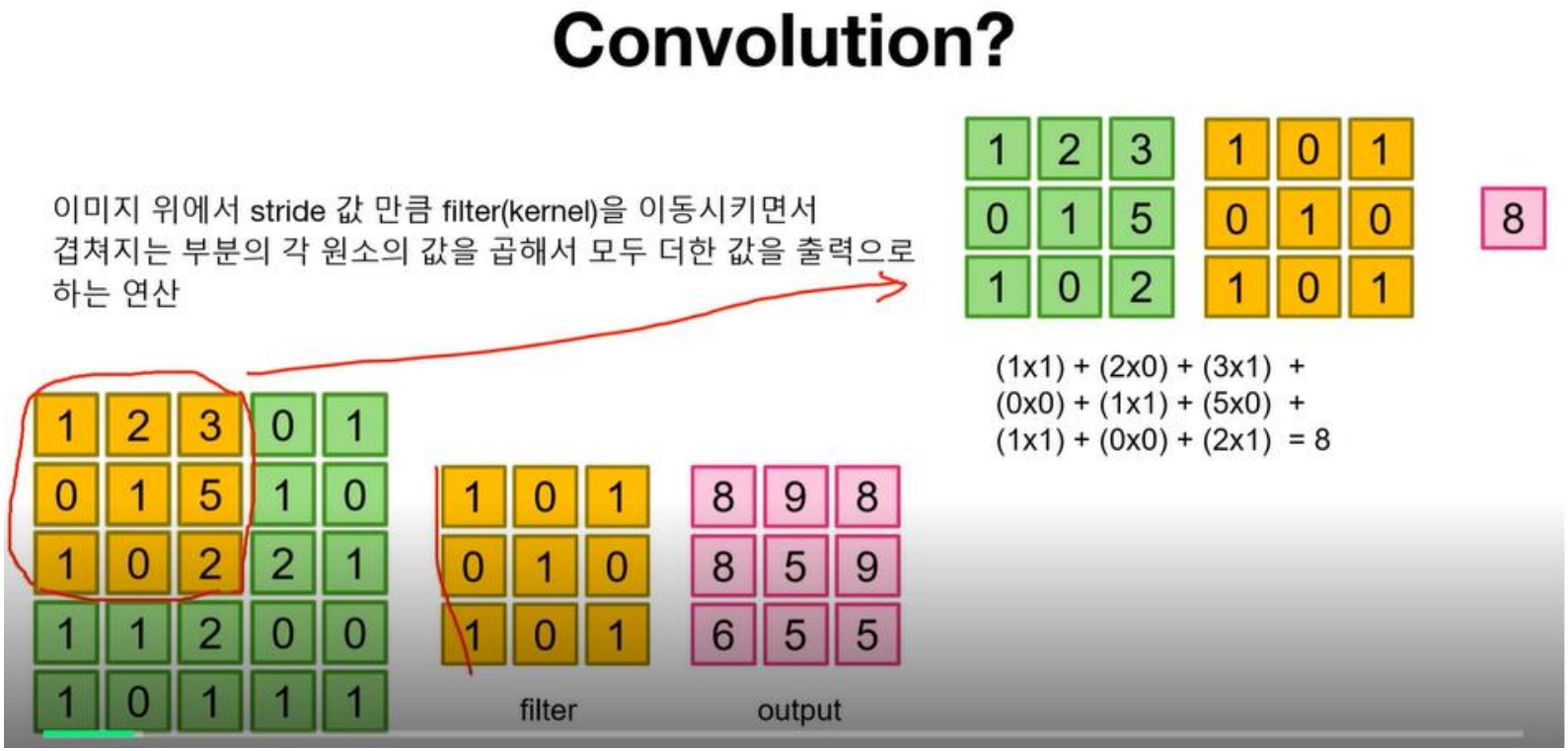
Week01 발표

CNN&RNN

이경선

1Convolution Neural Network

1.Convolution이 어떤 연산인가



이미지 위에서 stride 값 만큼 filter를 이동시키면서
겹쳐지는 부분의 각 원소의 곱을 출력하는 연산

Convolution Neural Network

Convolution의 유래

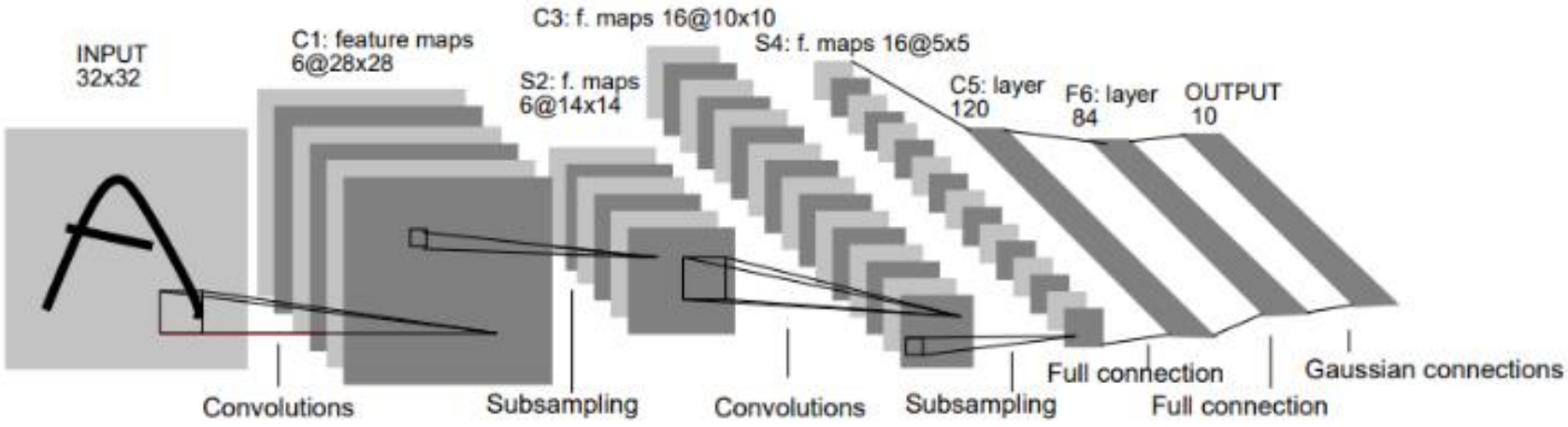
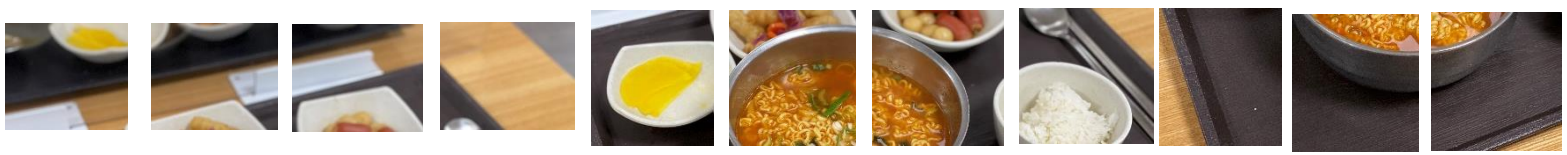


Figure 1: LeNet-5

시각 피질 안의 많은 뉴런들이 작은 local receptive field를 가짐.
일부 범위 안의 시각 자극에만 반응.

Convolution Neural Network

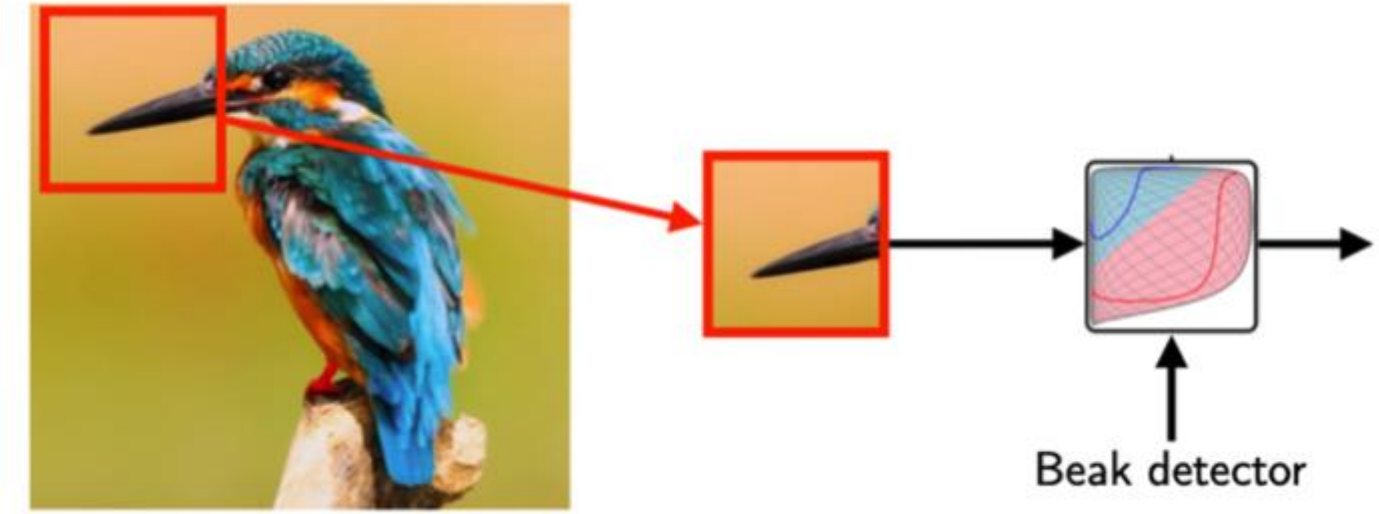
Convolution을 왜 하는가?



기존의 Deep Neural Network에서는 이미지의 위치 정보가 소실됨.
따라서 위치 정보를 유지하면서 연산할 방법을 찾음.

Convolution Neural Network

Convolution을 왜 하는가?



새인지 아닌지 판단하기 위해 부리 부분을 잘라 보는 것이 효율적.
CNN의 뉴런이 패턴(새의 부리)를 파악하기 위해 전체 이미지를 볼 필요가 없음.

Convolution Neural Network

2. Padding

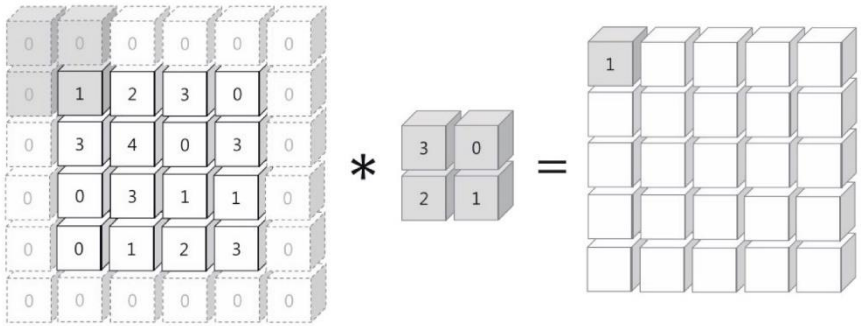


그림 12-55 입력 데이터에 패딩 적용 후 합성곱 연산

입력 데이터 주변을 특정값으로 채우는 것.
출력 데이터의 차원이 줄어드는 현상을 방지.

Convolution Neural Network

2. stride

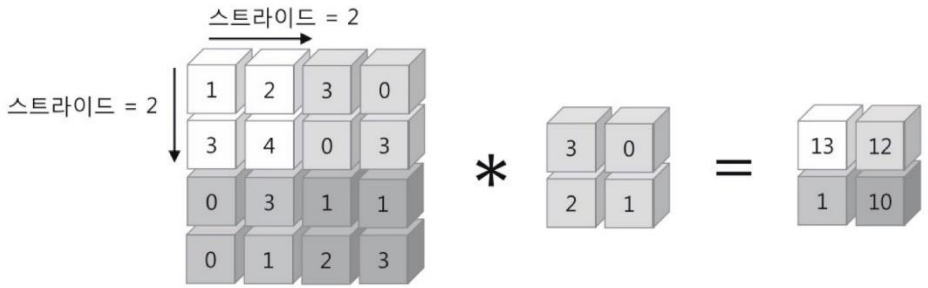
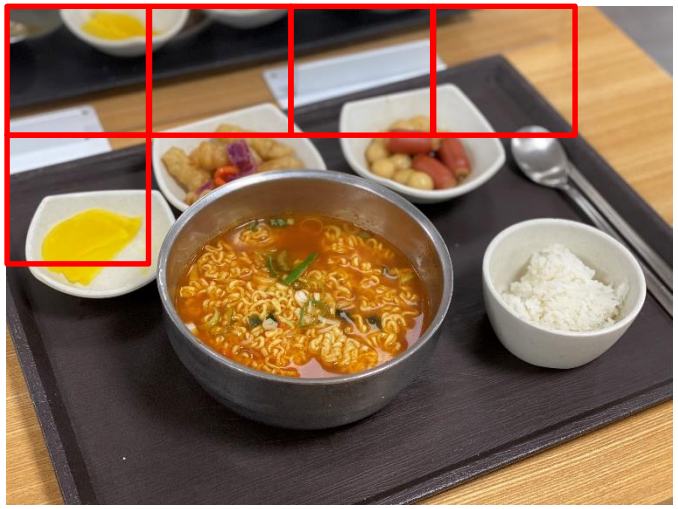


그림 12-57 스트라이드 적용 후 합성곱



한 번 합성곱 연산한 후 다음 계산 영역을 선택할 때 얼마나 이동할지 간격

Convolution Neural Network

Flattening

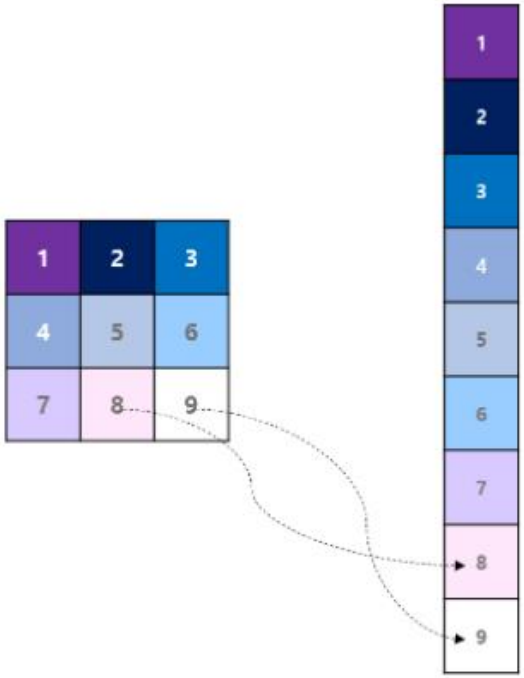
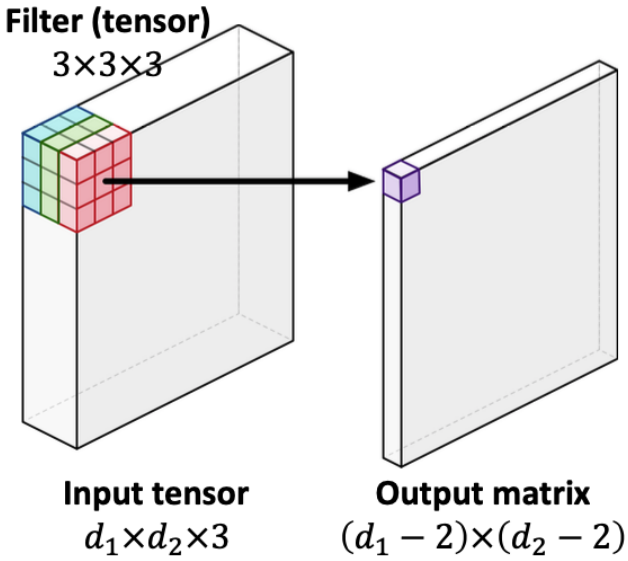


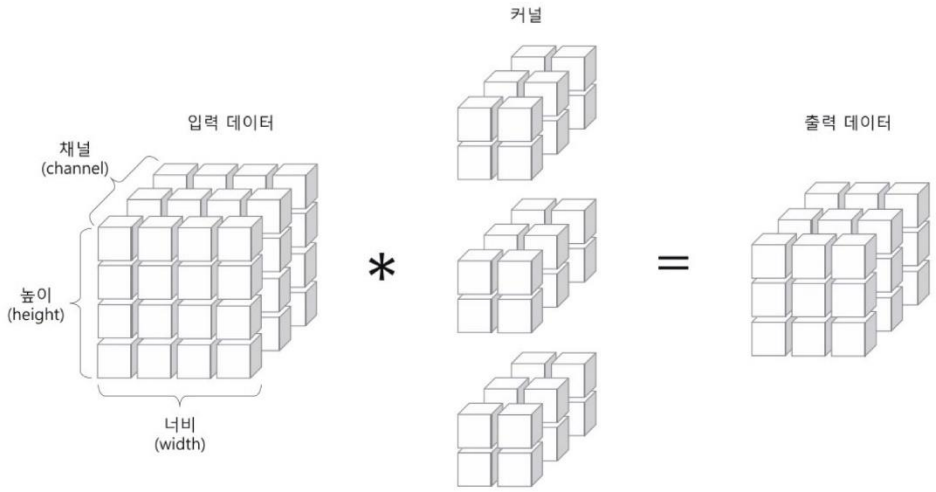
Figure 3: Flattening

추출된 특성을 output layer에 연결하여 어떤 이미지인지 분류하기 위함.
일반 신경망 모델과 동일.

Convolution Neural Network

고차원 데이터 합성곱





차원: 너비x높이x채널

Convolution Neural Network

고차원 데이터 합성곱

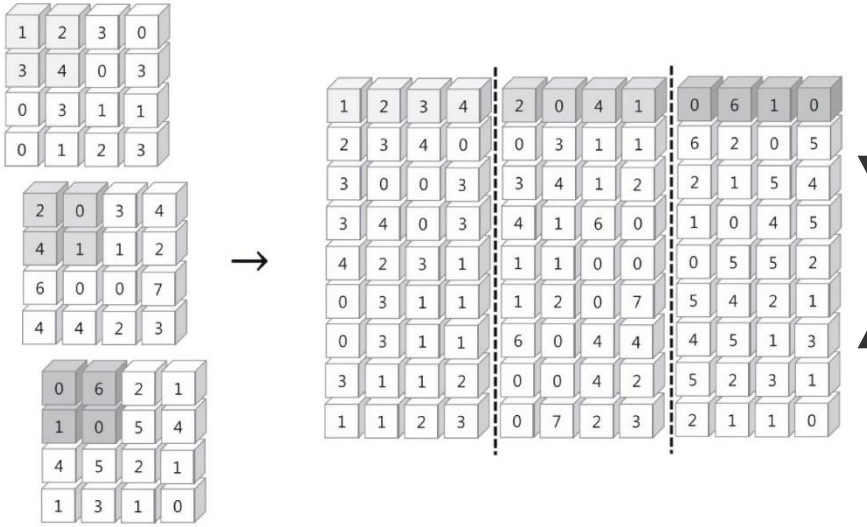


그림 12-64 고차원 데이터를 행렬로 표현

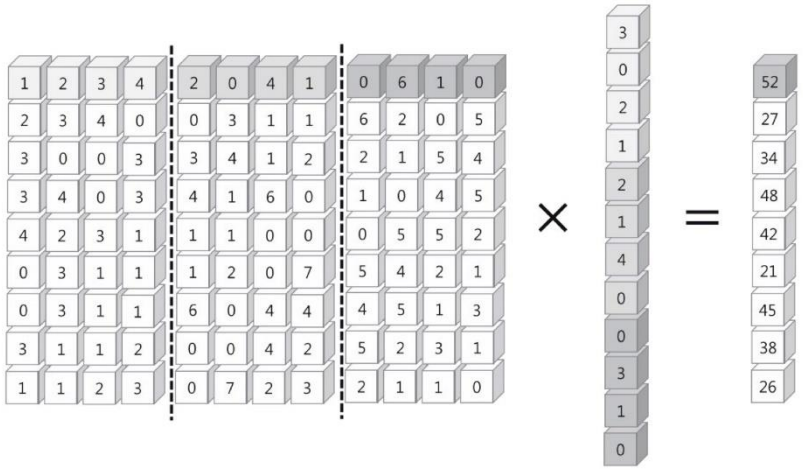


그림 12-66 고차원 입력, 커널 데이터 합성곱

차원: 너비x높이x채널

Convolution Neural Network

Convolution이 어떤 연산인가

Torch.nn.Conv2d
(in_channels, out_channels, kernel_size, stride,padding)

kernel_size: 필터 사이즈
input type: torch.Tensor
input shape: (batch_size, channel, height, width)
output size =(input size - filter size + 2*padding)/stride +1

Convolution Neural Network

MNIST dataset에 CNN적용

딥러닝 학습 단계

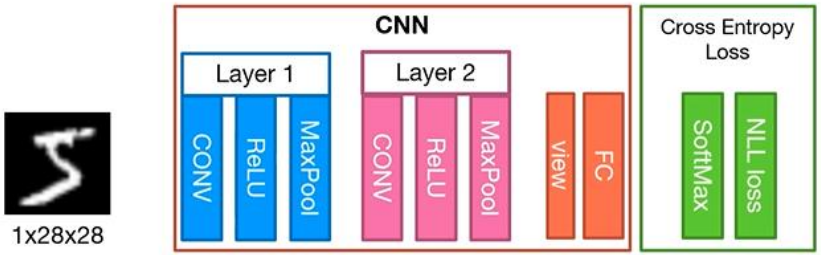
---

1. 라이브러리 가져오기
2. GPU 사용 설정하고 random value를 위한 seed 설정
3. 학습에 사용되는 parameter 설정
4. 데이터셋 가져오고 loader 만들기
5. 학습 모델 만들기 (class CNN(torch.nn.Module))
6. loss function 선택하고 최적화도구 선택
7. 모델 학습 및 loss check
8. 학습된 모델의 성능을 확인한다.

Convolution Neural Network

MNIST dataset에 CNN적용

우리가 만들 CNN 구조 확인!



(Layer 1) Convolution layer = (in_c=1, out_c=32,kernel_size =3, stride=1,padding=1)
(Layer 1) MaxPool layer = (kernel_size=2, stride =2)

(Layer 2) Convolution layer = (in_c=32, out_c=64, kernel_size =3, stride=1,padding=1)
(Layer 2) MaxPool layer = (kernel_size=2, stride =2)

```
Model: "sequential_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_12 (Conv2D)           (None, 24, 24, 32)        832
_________________________________________________________________
max_pooling2d_14 (MaxPooling (None, 12, 12, 32)        0
_________________________________________________________________
dropout_17 (Dropout)         (None, 12, 12, 32)        0
_________________________________________________________________
conv2d_13 (Conv2D)           (None, 10, 10, 32)        9248
_________________________________________________________________
max_pooling2d_15 (MaxPooling (None, 5, 5, 32)          0
_________________________________________________________________
dropout_18 (Dropout)         (None, 5, 5, 32)          0
_________________________________________________________________
flatten_7 (Flatten)          (None, 800)               0
_________________________________________________________________
dense_14 (Dense)             (None, 1024)              820224
_________________________________________________________________
dropout_19 (Dropout)         (None, 1024)              0
_________________________________________________________________
dense_15 (Dense)             (None, 10)                10250
=================================================================
Total params: 840,554
Trainable params: 840,554
Non-trainable params: 0
_________________________________________________________________
```

그림 12-68 합성곱 신경망 구조