

Deep Learning III

김연지

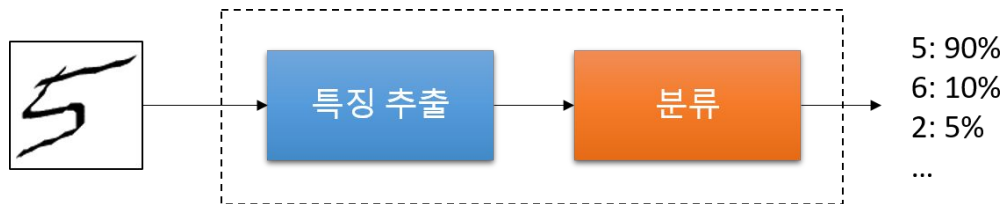
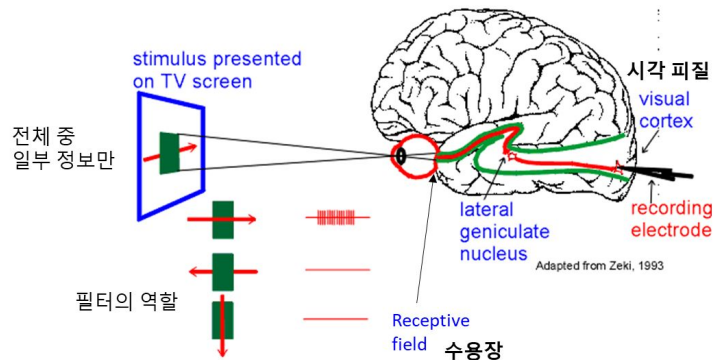
kimyeonji3@gmail.com

CNN

(Convolution Neural Network,
합성곱신경망)

CNN(Convolution Neural Network, 합성곱신경망)

- CNN은 전통적인 ANN(MLP) 앞단에 여러 계층의 컨볼루셔널 계층을 붙인 모양
- 컨볼루셔널 레이어를 통해서 입력 받은 이미지에 대한 **특징(Feature)**을 추출하고, 이를 기반으로 기존의 전부 연결된 뉴럴 네트워크로 **분류**를 하는 원리



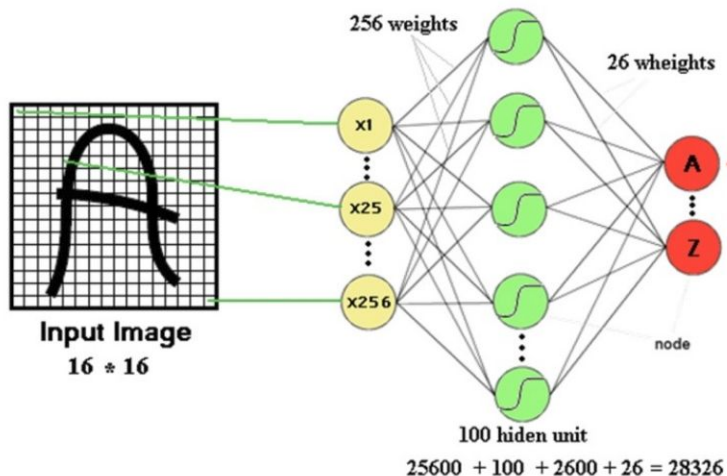
- RF(Receptive Field, 수용 영역): 전체 자극 중 일부 자극에만 반응하는 영역을 의미
- 수용 영역이 전체 중 일부 자극에만 반응하게 만들어진 이유: 실제 이미지에서 특정 위치의 픽셀들은 주변에 있는 픽셀과 관련있고 멀어질수록 그 영향력이 낮아지기 때문에 수용 영역처럼 전체 중 일부 자극만 받아들여 처리하는게 사물 인식 시 더 효과적

CNN(Convolution Neural Network, 합성곱신경망)

- 합성곱층(Convolution layer)와 풀링층(Pooling layer)으로 구성됨

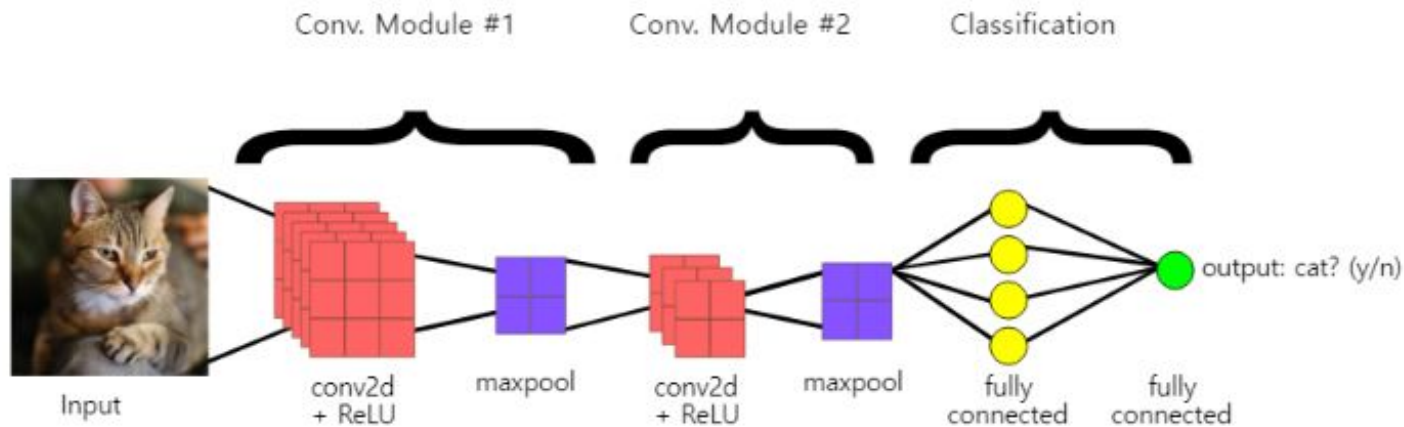
이미지를 멀티퍼셉트론으로 분류하려면?

- 1차원 텐서인 벡터로 변환해야 함



- 완전연결 계층의 문제점은 바로 '**데이터의 형상이 무시**'된다는 것
- 이미지 데이터의 경우 3차원(세로, 가로, 채널)의 **공간적 구조(spatial structure)**를 가지며, 공간적으로 가까운 픽셀은 값이 비슷하거나, RGB의 각 채널은 서로 밀접하게 관련되어 있거나, 거리가 먼 픽셀끼리는 관련이 없는 등, 3차원 공간에서 정보들이 내포되어있음
- 하지만, 완전연결 계층에서 1차원의 데이터로 펼쳐게 되면 이러한 정보들이 사라지게 된다.

CNN(Convolution Neural Network, 합성곱신경망)



<https://developers.google.com/machine-learning/practica/image-classification/convolutional-neural-networks?hl=ko>

CNN(Convolution Neural Network, 합성곱신경망)

이미지 인식, 음성 인식에서 자주 사용됨

- 합성곱 계층(Convolution Layer), 풀링 계층(Pooling Layer) 존재
 - 풀링 계층은 생략 가능
- 완전 연결 계층(Fully-Connected Layer): 인접하는 계층의 모든 뉴런과 결합되어 있는 계층
 - 출력과 가까운 층에는 'Fully_Connected-ReLU' 계층 사용 가능
 - 마지막 출력 계층에서는 'Affine-Softmax' 조합 사용
 - 완전연결계층(FC Layer)에서는 데이터의 형상(shape)이 무시됨.
 - input이 3차원 이라고 하더라도 완전연결계층에 입력할 때는 1차원 데이터로 평탄화해줘야 함.
 - 형상(shape)을 유지하기 위해 합성곱 계층 사용

CNN 네트워크 구조

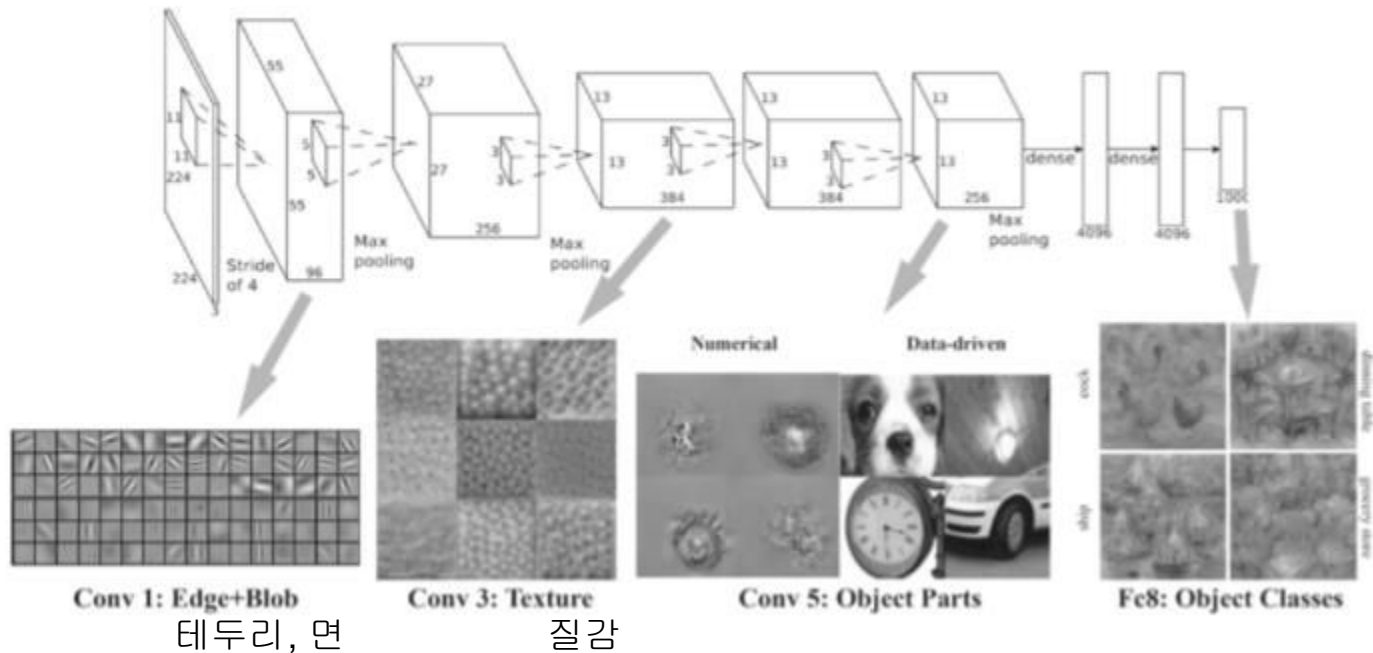


특징맵이란? 합성곱 계층의 입출력 데이터를 의미

입력 데이터 : 입력 특징 맵

출력 데이터 : 출력 특징 맵

CNN(Convolution Neural Network, 합성곱신경망)



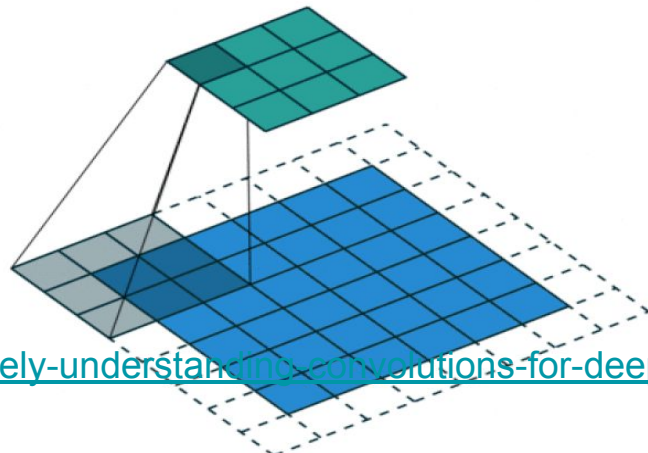
- 딥러닝이 데이터로부터 학습하는 특징
- 층을 거치면서 점진적으로 더 복잡한 표현이 만들어짐
 - 점진적인 중간 표현이 공동으로 학습

합성곱

- 원본 이미지의 각 픽셀을 포함한 주변 픽셀과 필터의 모든 픽셀은 각각 곱연산 수행
- 그 결과를 모두 합해서 새로운 이미지에 넣어주기 때문에 합성곱이라고 함
- 컨볼루션 신경망은 입력된 이미지에서 다시 한번 특징을 추출하기 위해 마스크 (필터, 윈도우 또는 커널이라고도 함)를 도입하는 기법

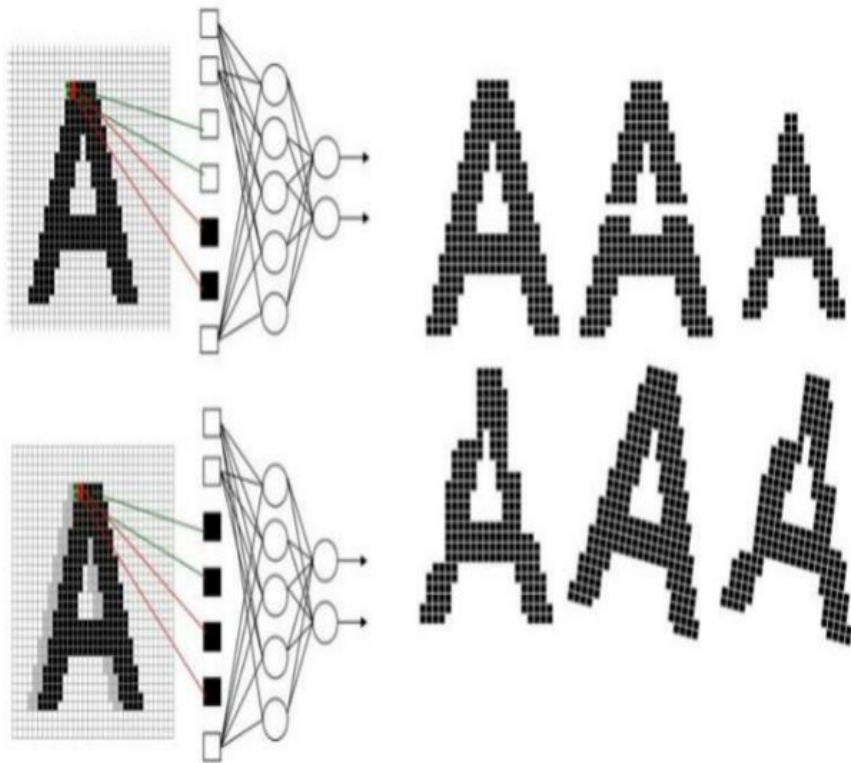
3 ₀	3 ₁	2 ₂	1	0
0 ₂	0 ₂	1 ₀	3	1
3 ₀	1 ₁	2 ₂	2	3
2	0	0	2	2
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0



<https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1>

CNN 네트워크 구조



각 픽셀을 본래 픽셀과 그 주변 픽셀의 조합으로 대체하는 동작

연산에 쓰이는 작은 행렬을 **filter or kernel** 이라 함

CNN(Convolution Neural Network, 합성곱신경망)

필터의 효과 : 수직선, 수평선, 검출, 흐림(blur), 날카로운(sharpen) 등 다양한 이미지 효과 적용 가능

- 딥러닝 기반의 컨볼루션 연산은 특징 검출하는 필터를 수작업으로 설계하는 것이 아님. 네트워크가 특징을 추출하는 필터를 자동으로 생성해 줌
- 학습을 지속적으로 하면서 네트워크로 구성하는 각 뉴런들은 입력한 데이터에 대해 특정 패턴을 잘 추출할 수 있도록 적용

<https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1>

CNN(Convolution Neural Network, 합성곱신경망)

마스크를 한 칸씩 옮겨 모두 적용하면?

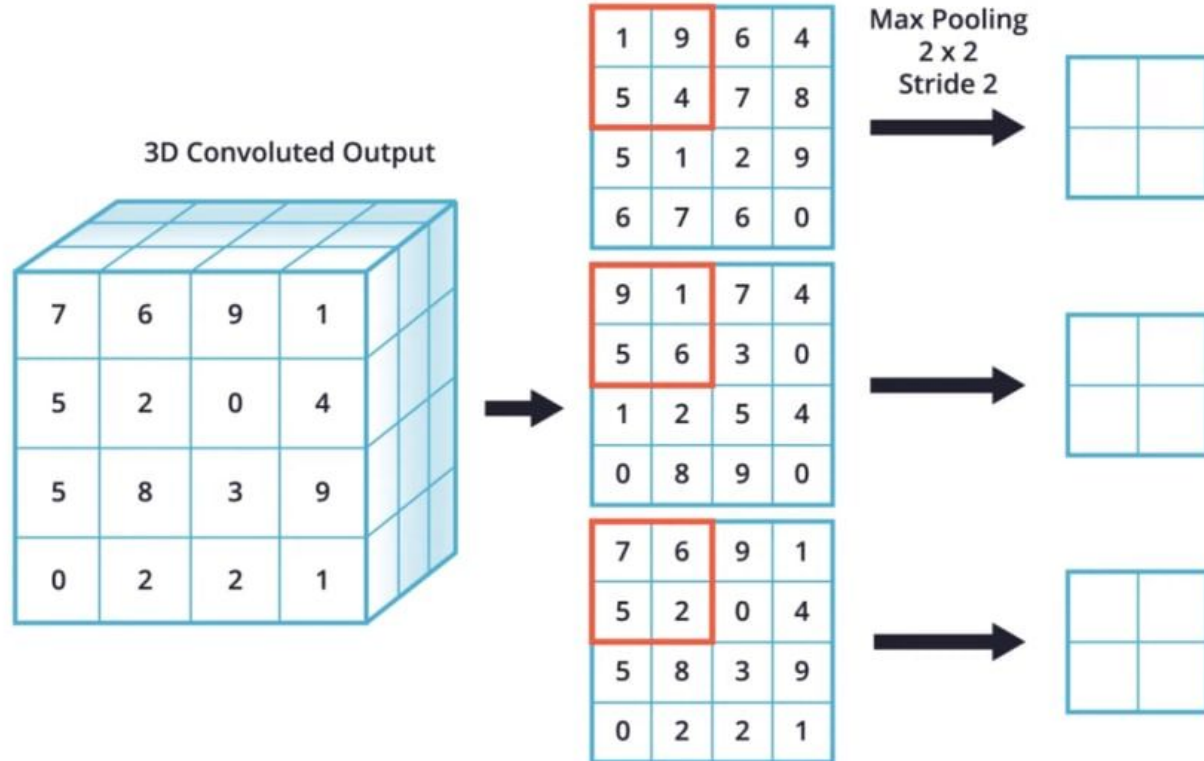
<table><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td></tr></table>	1	0	1	0	0	1	1	0	0	0	1	1	0	0	1	0	<table><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td></tr></table>	1	0	1	0	0	1	1	0	0	0	1	1	0	0	1	0	<table><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td></tr></table>	1	0	1	0	0	1	1	0	0	0	1	1	0	0	1	0
1	0	1	0																																															
0	1	1	0																																															
0	0	1	1																																															
0	0	1	0																																															
1	0	1	0																																															
0	1	1	0																																															
0	0	1	1																																															
0	0	1	0																																															
1	0	1	0																																															
0	1	1	0																																															
0	0	1	1																																															
0	0	1	0																																															
<table><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td></tr></table>	1	0	1	0	0	1	1	0	0	0	1	1	0	0	1	0	<table><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td></tr></table>	1	0	1	0	0	1	1	0	0	0	1	1	0	0	1	0	<table><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td></tr></table>	1	0	1	0	0	1	1	0	0	0	1	1	0	0	1	0
1	0	1	0																																															
0	1	1	0																																															
0	0	1	1																																															
0	0	1	0																																															
1	0	1	0																																															
0	1	1	0																																															
0	0	1	1																																															
0	0	1	0																																															
1	0	1	0																																															
0	1	1	0																																															
0	0	1	1																																															
0	0	1	0																																															
<table><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td></tr></table>	1	0	1	0	0	1	1	0	0	0	1	1	0	0	1	0	<table><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td></tr></table>	1	0	1	0	0	1	1	0	0	0	1	1	0	0	1	0	<table><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td></tr></table>	1	0	1	0	0	1	1	0	0	0	1	1	0	0	1	0
1	0	1	0																																															
0	1	1	0																																															
0	0	1	1																																															
0	0	1	0																																															
1	0	1	0																																															
0	1	1	0																																															
0	0	1	1																																															
0	0	1	0																																															
1	0	1	0																																															
0	1	1	0																																															
0	0	1	1																																															
0	0	1	0																																															

2	1	1
0	2	2
0	1	1

새롭게 만들어진 층을 컨볼루션(합성곱)이라고 함
컨볼루션을 만들면 입력 데이터로부터 더욱 정교한
특징 추출 가능

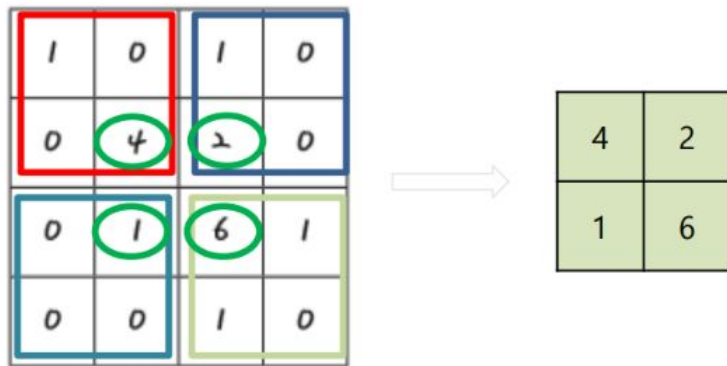
마스크를 여러 개 만들 경우 여러개의 컨볼루션이
구성됨

Max Pooling



Max Pooling

- 맥스 풀링
- 컨볼루션 층을 통해 이미지 특징을 도출 한 경우 그 결과가 여전히 크고 복잡하면 이를 다시 한번 축소 해야 함
- 그 과정을 풀링(pooling) 또는 서브 샘플링(sub sampling)이라고 함
- 풀링 기법 중 가장 많이 사용되는 방법이 맥스 풀링(max pooling)
- 정해진 구역 안에서 가장 큰 값만 다음 층으로 넘기고 나머지는 버림
- 이 과정을 거쳐 불필요한 정보를 간추림
- 맥스 풀링은 `MaxPooling2D()` 함수를 사용



딥러닝시 고려 사항

과적합(overfitting)

- 모델이 학습 데이터셋 안에서는 일정 수준 이상의 예측 정확도를 보이지만, 새로운 데이터에 적용하면 잘 맞지 않는 것을 의미
- 학습한 데이터에만 종속적으로 맞춰져 있어 그 외 데이터는 잘 맞지 않는 상황을 의미
- **Overfitting에 대한 해결책 : Dropout(망 부분 생략) 적용한 분류 DNN**
- 드롭아웃(Dropout)
 - 과적합을 해결하는 가장 효과가 좋은 방법
 - 단순한 기법을 통해 신경망의 일반화 성능을 향상시킬 수 있는데, 학습시킬 때 무작위로 뉴런을 '드롭아웃 (제외)'하기임
 - 드롭 아웃을 적용하면 실질적으로 많은 모델을 생성, 학습하고 그 안에서 예측을 실행하기 때문에 성능이 향상됨

딥러닝시 고려 사항

- 드롭아웃(Dropout)

- 망에 있는 입력 layer나 hidden layer의 일부 뉴런을 생략(drop out)하고 줄어든 신경망을 통해 학습 수행 일정
- mini-batch 구간 동안 생략 된 망에 대한 학습을 끝내면 다 시 무작위로 다른 뉴런들을 생략하면서 반복적으로 학습 수행

