



# 실무요건 기반 핀테크 데이터분석가 과정

- 딥러닝 CNN -

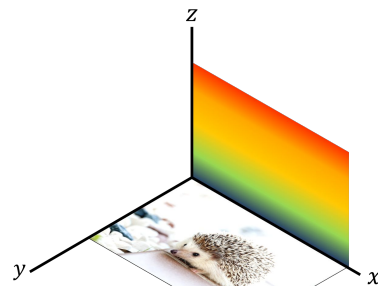
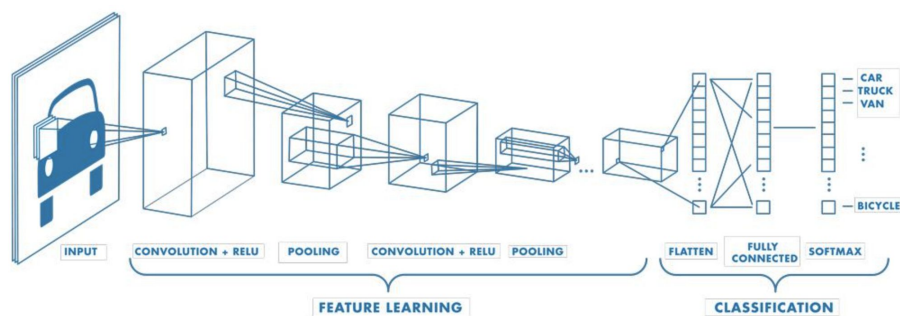
핀인사이트  
데이터분석가 김현진  
jinny@fins.ai

# CNN

# CNN(Convolution Neural Network)

## ✓ CNN(합성곱 신경망)

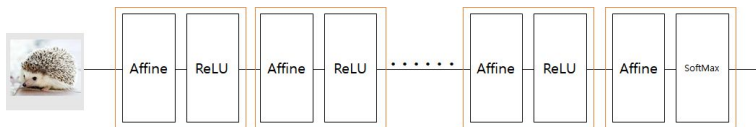
- Lecun이 1989년에 발표한 논문 「Backpropagation applied to handwritten zip code recognition」에서 필기체 이미지를 인식하기 위해 처음으로 등장
- 데이터 고유의 특징을 학습할 수 있는 컨볼루션 계층, ReLU 계층, 풀링 계층이 추가된 형태
- 이미지의 구성 : x 축, y 축, 채널(channel, 색상) 으로 구성된 3차원 데이터로 일반적으로 R(Red), G(Green), B(Blue) 색상 3개 채널을 가짐 (cf. 흑백의 경우 채널 1개)
- 의료영상, 오디오처리, 객체검출, 합성데이터 생성 등 다양한 분야에 사용됨



가치를 높이는 금융 인공지능 실무교육

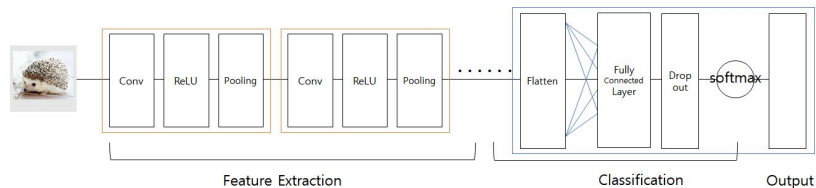
# 기존 인공신경망과 CNN의 구조적 차이

## 완전연결신경망



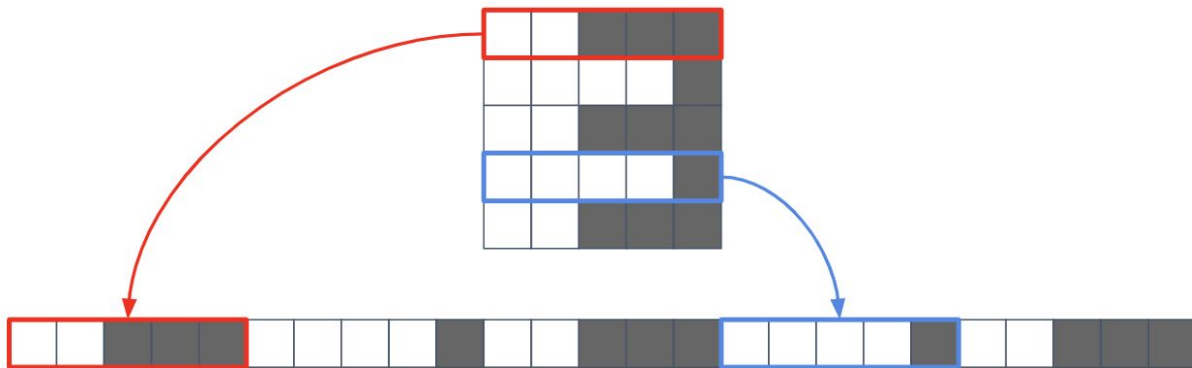
- 인접한 계층이 모든 뉴런과 결합되어 있는 것을 ‘완전결합’이라고 하며 완전연결된 계층을 Affine계층이라고 부름
- 초기 이미지 분류/인식의 경우 완전연결신경망을 사용했는데, 3차원인 이미지를 1차원 배열로 바꾸어 학습함
- 그러다보니 차원이 축소되면서 공간정보에 대한 손실이 발생함

## 합성곱신경망



- 합성곱계층(Conv)과 풀링계층(Pooling)을 추가하여 이미지의 공간정보를 유지하고 특징추출을 통해 완전 연결 신경망 문제를 해결한
- 이미지의 차원을 그대로 유지하고 이미지의 특징을 추출하여 이 특징과 유사한 특징을 가진 다른 이미지를 찾아내는 방식으로 학습
- 특징추출단계와 분류단계로 구성되는데, 분류단계에서만 기존의 완전신경망 구조가 사용됨

# 완전연결신경망의 문제점

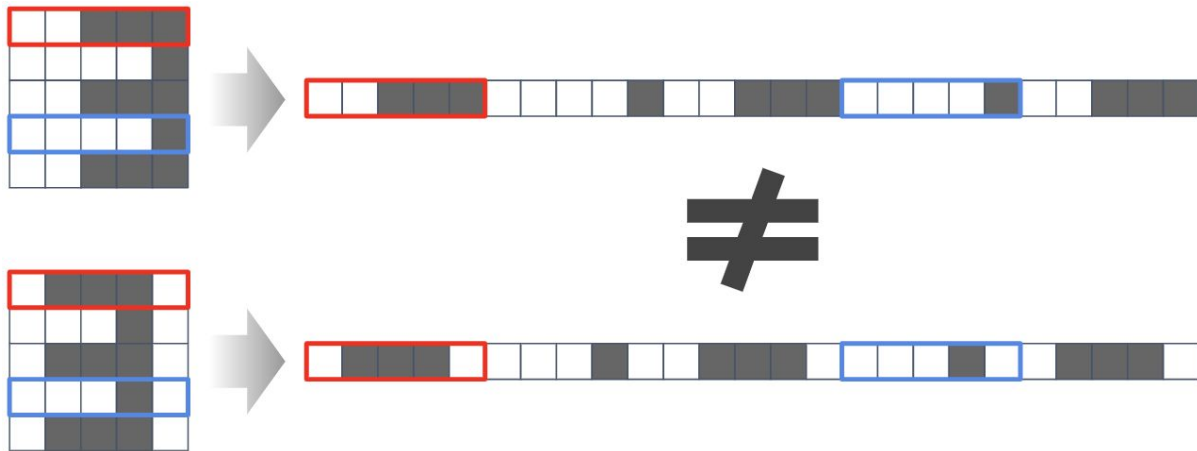


- ✓ 2차원 데이터가 1차원으로 변환하면서 공간 정보가 손실됨  
: 이미지를 2차원으로 표현 했을때 3이라는 것을 알 수 있었지만 1차원으로 변환되었을 때, 어떤 숫자인지 추측하기 어려움. 이는 변환하면서 공간 정보손실이 발생했기 때문임

# 완전연결신경망의 문제점



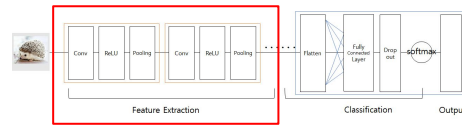
## 완전연결신경망의 문제점



✓ 이미지에 약간의 변화만 있어도 같은 이미지인지 인식하기 어려움

→ 이러한 문제를 해결한 CNN의 합성곱 계층과 풀링 계층은 3차원의 이미지를 그대로 입력받아  
3차원 데이터로 출력하여 다음 계층으로 전달함

# CNN 구조 - Feature Extraction



## ✓ Feature Extraction

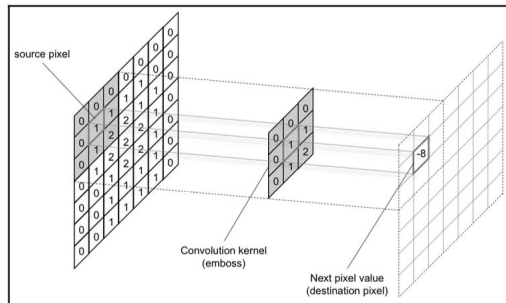
: 특징추출 단계는 합성곱 계층과 풀링계층을 여러겹 쌓는 형태로 구성됨

### ■ 합성곱계층(Convolution Layer)

: 입력 데이터에 필터를 적용하여 데이터의 특징을 추출하고, 활성화함수를 사용해 필터의 값을 비선형값으로 바꿔주는 역할

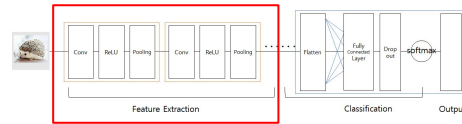
### ■ 합성곱이란?

: 특정 너비와 높이를 갖는 필터를 일정간격으로 이동(Stride)해가며 입력 데이터에 적용하는 과정





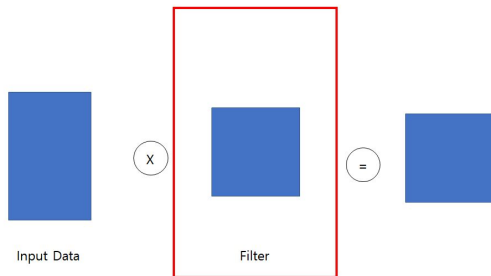
# CNN 구조 - Feature Extraction



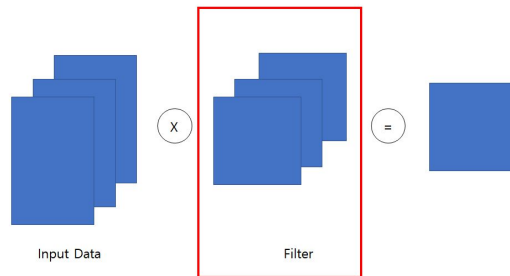
## ■ 필터(Filter) = 커널(Kernel) 란?

: 이미지의 특징을 찾아내기 위해 사용되는 함수

- 추출하려는 이미지의 특징이 입력데이터에 있는지 없는지 확일 할 수 있음
- 입력데이터에 필터가 일정한 간격으로 움직이며 적용됨
- 입력되는 이미지의 차원이 여러차원이라면 필터는 입력되는 데이터의 차원 수 만큼 필요함

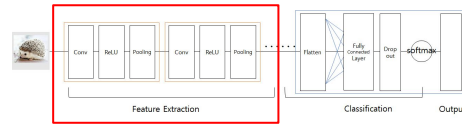


필터에서 - 1차원



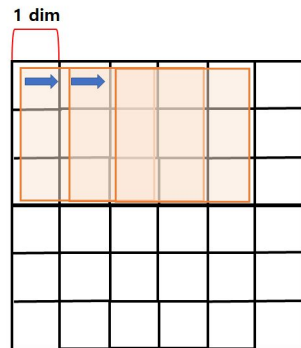
필터에서 - 3차원

# CNN 구조 - Feature Extraction

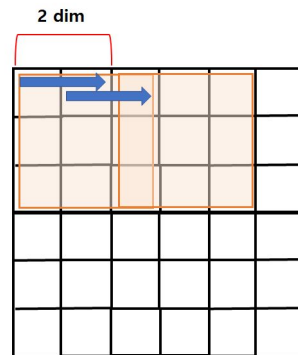


## ■ 스트라이드(Stride)란?

: 합성곱을 계산할 때 필터는 입력값에 대해 일정한 간격에 따라 움직이며 계산되어야 하는데, 이때 일정한 간격으로 움직이는 크기를 스트라이드(Stride)라고 부르며 차원(dimension)이라고 부르기도 함

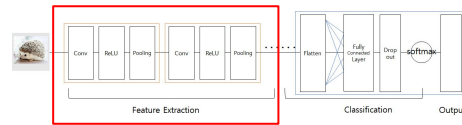


스트라이드예시 - 1dim

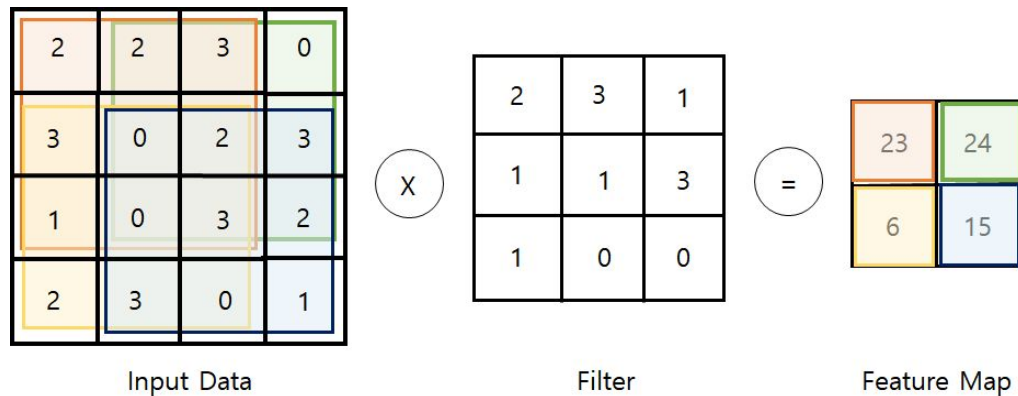


스트라이드예시 - 2dim

# CNN 구조 - Feature Extraction



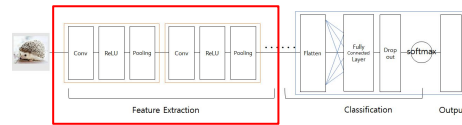
- 합성곱 계산 예시
  - 입력값 : 4 by 4 / 스트라이드 : 1 / 필터 : 3 by 3



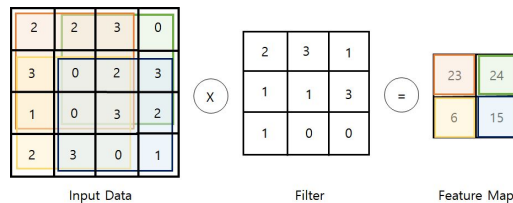
합성곱 예시

또는 convolution feature라고 부름

# CNN 구조 - Feature Extraction



- 합성곱 계산 예시
  - 입력값 : 4 by 4 / 스트라이드 : 1 / 필터 : 3 by 3



2	2	3
3	0	2
1	0	3

×

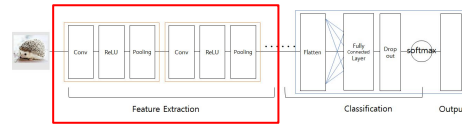
2	3	1
1	1	3
1	0	0

=

$$\begin{aligned}
 &2 \times 2 + 2 \times 3 + 3 \times 1 \\
 &+ 3 \times 1 + 0 \times 1 + 2 \times 3 \\
 &+ 1 \times 1 + 0 \times 0 + 3 \times 0 = 23
 \end{aligned}$$

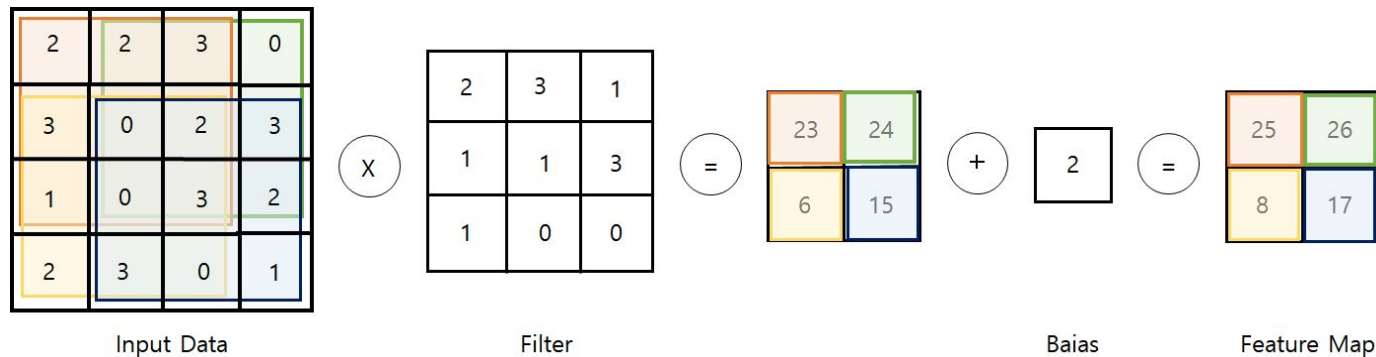
각각의 위치에 대응하는 값을 서로 곱한 후 총합을 구하면 첫번째 값을 구할 수 있고,  
이를 4번 반복하면 Feature Map 전체를 구할 수 있음

# CNN 구조 - Feature Extraction



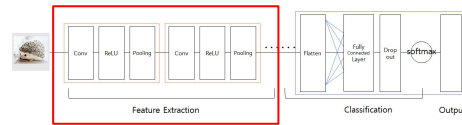
## ■ 합성곱 계산 예시

- 입력값 : 4 by 4 / 스트라이드 : 1 / 필터 : 3 by 3



Feature Map을 구한 뒤 편향(Bias)까지 더해지면 합성곱의 최종 Feature Map이 완성됨

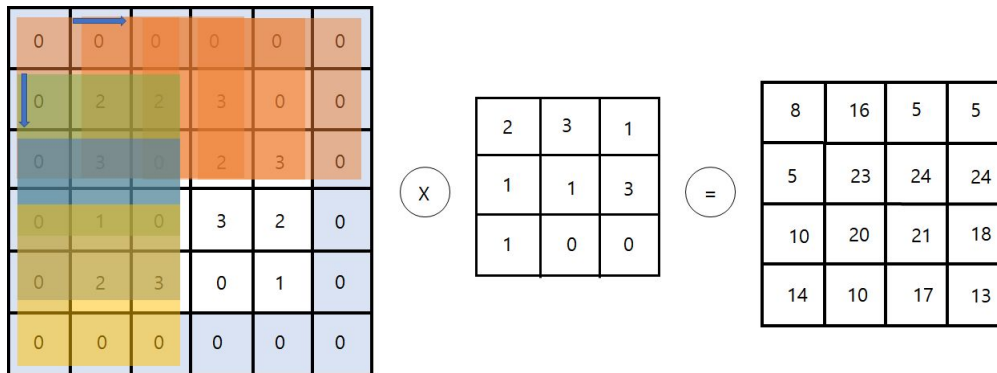
# CNN 구조 - Feature Extraction



## ■ 패딩(Padding)

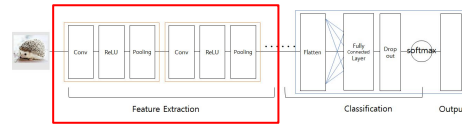
: 합성곱 연산을 수행하기 전 입력데이터 주변을 특정 값(주로 '0' : zero padding)으로 채우는 것

- 입력데이터와 출력데이터의 크기를 맞추기 위해 사용하며 원본데이터의 손실을 막음



제로 패딩 예시

# CNN 구조 - Feature Extraction



## ■ 패딩(Padding)

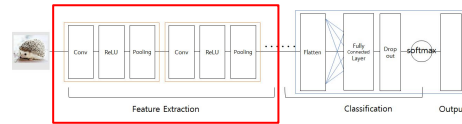
: 합성곱 연산을 수행하기 전 입력데이터 주변을 특정 값(주로 '0' : zero padding)으로 채우는 것

- 입력데이터와 출력데이터의 크기를 맞추기 위해 사용하며 원본데이터의 손실을 막음

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

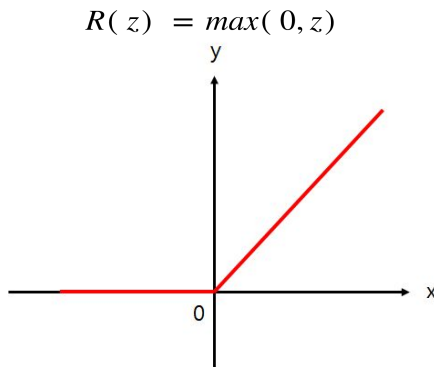
패딩 예시

# CNN 구조 - Feature Extraction



## ■ 활성화함수 - ReLU

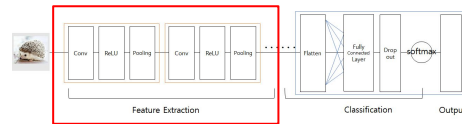
- CNN 초기 모델인 LetNet에서는 활성화함수로 시그모이드를 사용함.
- 하지만, 시그모이드의 경우 역전파를 할 경우 Gradient Vanishing 문제가 발생하기 때문에 부적합하기 때문에 LetNet이후 모델인 AlexNet부터는 활성화함수로 ReLU를 사용



활성함수 ReLU



# CNN 구조 - Feature Extraction

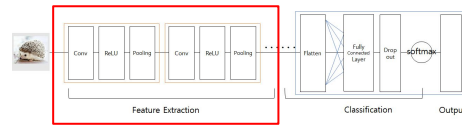


## ■ 풀링(Pooling)

: 필터를 통과하여 특징이 추출된 데이터의 크기를 인위적으로 줄이는 작업

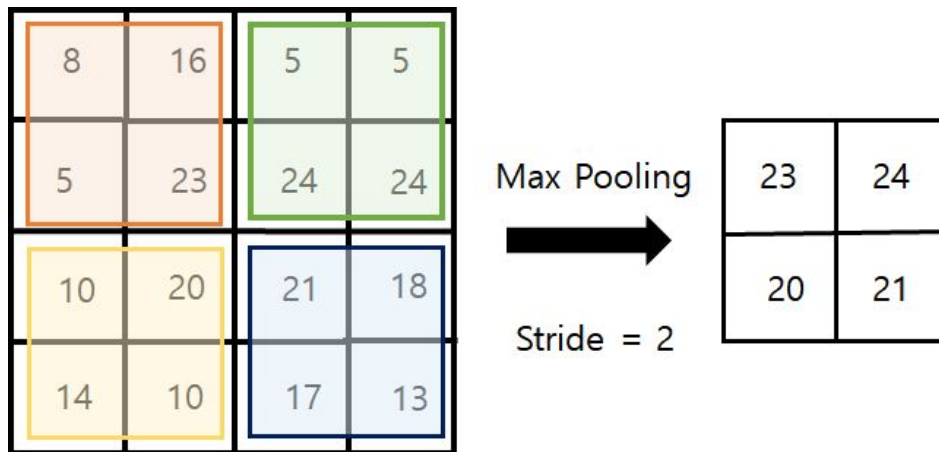
- 데이터중 일부를 선정하여 크기를 줄이기 때문에 subsampling이라고 부르기도 함
- 풀링을 하는경우 데이터 크기를 줄임으로써 프로그램의 연산을 줄일수 있고, 데이터의 일부가 사라지기 때문에 과적합을 방지할 수 있음
- 종류 : Max Pooling, Average Pooling, L2-Norm Pooling
  - > 과거에는 해당하는 영역의 평균을 계산하는 Average Pooling을 많이 사용했지만, 현재는 해당 영역의 최대값을 계산하는 Max Pooling이 더 좋은 성능을 보이면서 Max Pooling이 더 많이 사용됨

# CNN 구조 - Feature Extraction



## ■ 풀링(Pooling)

- Max Pooling 예시
- 2x2 크기의 필터를 사용하고, 스트라이드를 2로 하는 경우
- 각 필터에 속한 데이터 중 가장 큰 값을 출력함



# 합성곱 신경망의 여러가지 종류

## ✓ ZFNet

2013년에 실시한 ILSVRC에서 우승한 Matthew Zeiler와 Rob Fergus가 만든 모델로 저자의 이름을 따서 ZFNet이라고 이름 붙여짐  
ZFNet은 AlexNet 모델의 합성곱 계층의 크기를 조절하는 등 하이퍼 파라미터를 수정해 만듦

## ✓ GoogLeNet

2014년에 실시한 ILSVRC에서 우승한 구글이 제안한 모형으로 각 필터의 결과를 합쳐서 표현한 Inception module이라는 개념을 도입함.  
그 결과 Inception Module을 통해 아주 깊은 신경망에서도 1x1 합성곱 계층을 적극적으로 사용해 연산량을 줄이고, 합성곱 계층 마지막에 완전 연결 계층 대신 평균 풀링(Average Pooling)을 사용해 파라미터의 개수를 줄이는 것이 특징

## ✓ VGGNet

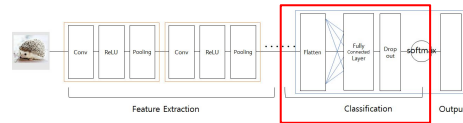
2014년에 실시한 ILSVRC에서 2등을 차지한 모델로 Karen Simonyan과 Andrew Zisserman이 만들어 VGGNet라고 이름 붙였짐  
VGGNet는 16개의 합성곱 계층과 완전 연결 계층으로 이루어져 있으며 모든 합성곱은 3x3 크기로, 모든 풀링 계층은 2x2 크기로 구성됨  
이미지 분류에서는 GoogleNet 보다 낮은 성능을 보이지만 Transfer Learning과 같은 과제에서 더 좋은 성능을 보이는 것이 밝혀서 최근 이미지 특징 추출에 많이 사용됨. 하지만 VGGNet의 단점은 연산시 많은 메모리가 사용됨

## ✓ ResNet

2014년에 실시한 ILSVRC에서 우승한 Kaining He et al.이 만든 모델로 Skip-connection이라는 구조를 사용해 batch normalization을 많이 사용한 것과 합성곱 신경망의 마지막 부분에 완전 연결 계층을 사용하지 않은 것이 특징

참고 : <http://aikorea.org/cs231n/convolutional-networks/#fc>

# CNN 구조 - Classification



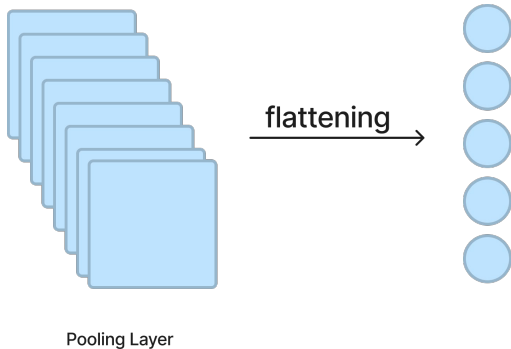
## ✓ Classification

: 분류단계는 Feature Map과 완전연결 신경망을 연결해주는 Flatten 계층과 일반적인 인공신경망 구조를 가지는 완전연결 신경망, 과적합을 막기위한 드롭아웃 그리고 소프트맥스로 구성됨

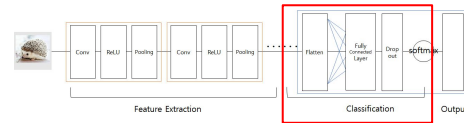
### ■ Flatten

: 합성곱 계층을 거친 결과 값을 완전 연결 신경망에 입력될 수 있는 값으로 변경해주는 계층

- 즉, Flatten 계층은 데이터의 모양(shape)을 변경해주는 계층

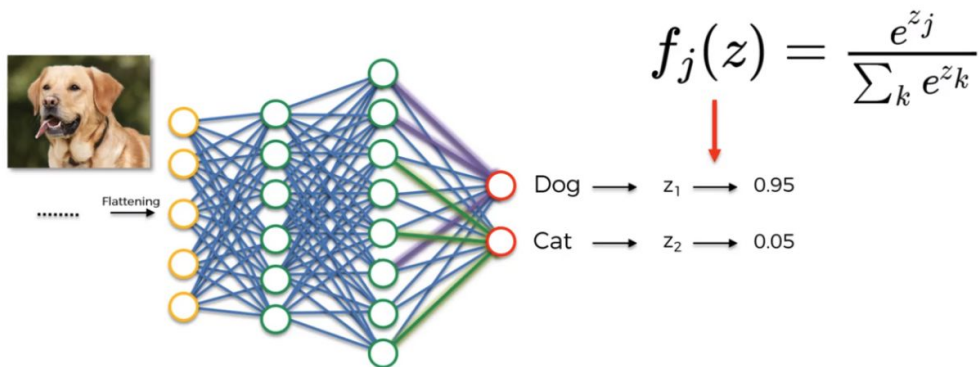


# CNN 구조 - Classification



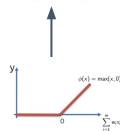
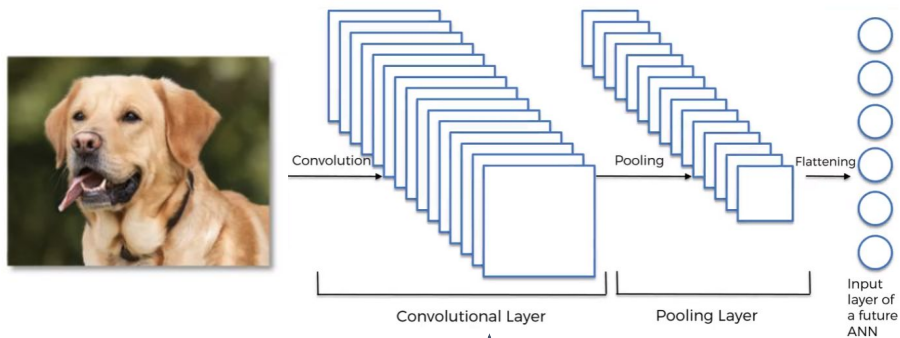
## ■ 활성화함수 - softmax

- 완전연결 계층을 통과한 데이터를 softmax에 넣으면 모든 데이터의 합이 1이 되는 확률값으로 결과값이 변환되며 각 확률값 중 가장 높은 값을 가지는 카테고리 분류하면 이미지 분류가 완성됨

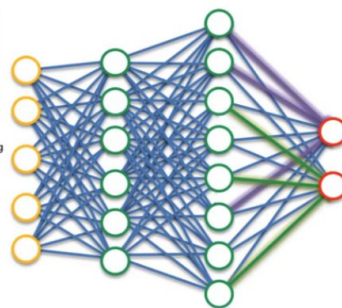


이미지출처 : <https://www.andreaperlato.com/aipost/cnn-and-softmax/>

# CNN 구조



Flattening



$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}}$$

↓

Dog  $\rightarrow z_1 \rightarrow 0.95$   
 Cat  $\rightarrow z_2 \rightarrow 0.05$

이미지출처 : <https://www.andreaperlato.com/aipost/cnn-and-softmax/>

# 실습1

데이터명 : MNIST 이미지 손글씨

데이터설명 : 컴퓨터 비전 데이터 세트로, 손으로 쓰여진 이미지로 구성된 데이터

- 60,000개의 train, 10,000개의 test

**? 손글씨 예측 모델을 만들어보세요**

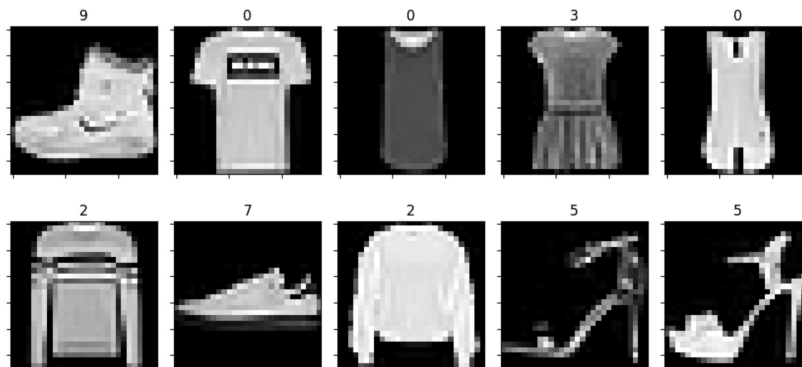


## 실습2

데이터명 : Fashion MNIST

데이터설명 : 28 x 28 픽셀의 이미지 70,000개로 총 10개의 카테고리  
(운동화, 셔츠, 샌들 등) 이미지 데이터셋

**? 의류 이미지 예측 모델을 만들어보세요**



- 0 : T-shirt/top
- 1 : Trouser
- 2 : Pullover
- 3 : Dress
- 4 : Coat
- 5 : Sandal
- 6 : Shirt
- 7 : Sneaker
- 8 : Bag
- 9 : Ankel boot



## 실습3

데이터명 : RPS

데이터설명 : 2,892개의 가위, 바위, 보 손동작 제스처 이미지

- 다양한 손동작, 인종, 나이, 성별에 대한 이미지가 포함됨
- 모두 흰색 배경에 24비트 색상의 300 x 300픽셀로 이루어진 이미지 데이터

**? 손모양 예측 모델을 만들어보세요**

