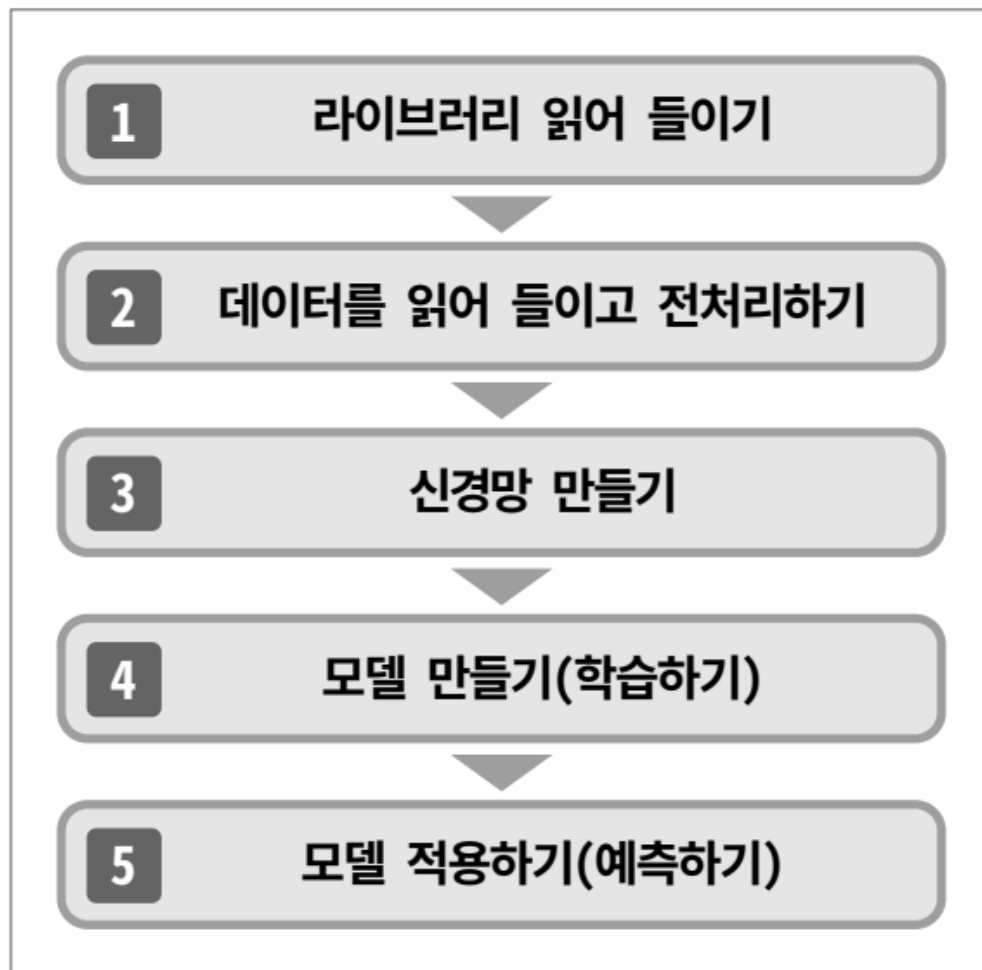


인공지능활용지도사 양성 과정(2일차)

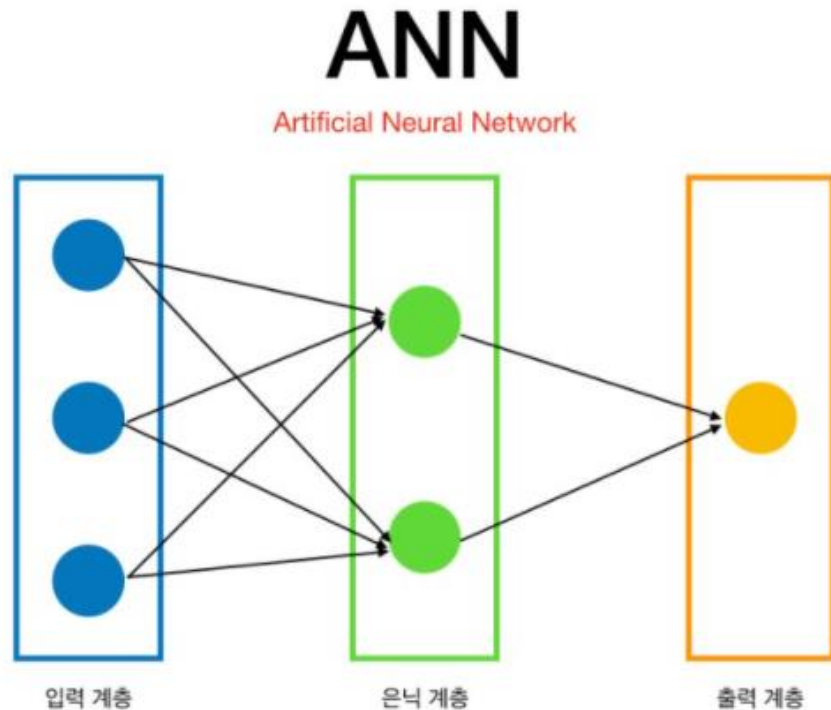
# 인공지능키트(AIIT) 실습 및 개발 실습

- (주)세울컴퍼니 -

## 딥러닝 처리절차



## ANN - Artificial Neural Network



### <신경망 특징>

- 생체의 신경망을 흉내 낸 인공지능
- 입력, 은닉, 출력층으로 구성, 은닉층은 한개이상 포함
- ANN은 인공지능을 총칭하는 용어
- 얇은 신경망(Shallow Neural Network, SNN) 구분

### ANN의 문제점

- 학습과정에서 파라미터의 최적값을 찾기 어렵다.

출력값을 결정하는 활성화함수의 사용은 기울기 값에 의해 weight가 결정되었는데 이런 gradient값이 뒤로 갈수록 점점 작아져 0에 수렴하는 오류를 낳기도 하고 부분적인 에러를 최저 에러로 인식하여 더이상 학습을 하지 않는 경우도 있습니다.

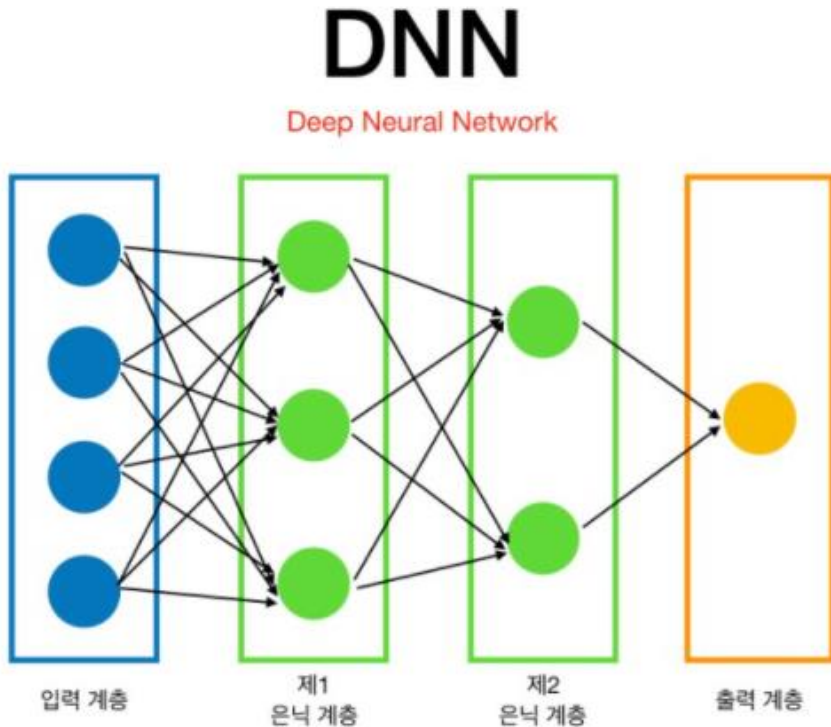
- Overfitting에 따른 문제

- 학습시간이 너무 느리다.

은닉층이 많으면 학습하는데에 정확도가 올라가지만 그만큼 연산량이 기하 급수적으로 늘어나게 됩니다.

하지만 이는 점점 해결되고 있습니다. 느린 학습시간은 그래픽카드의 발전으로 많은 연산량도 감당할 수 있을 정도로 하드웨어의 성능이 좋아졌고, 오버피팅문제는 사전훈련을 통해 방지할 수 있게 되었습니다.

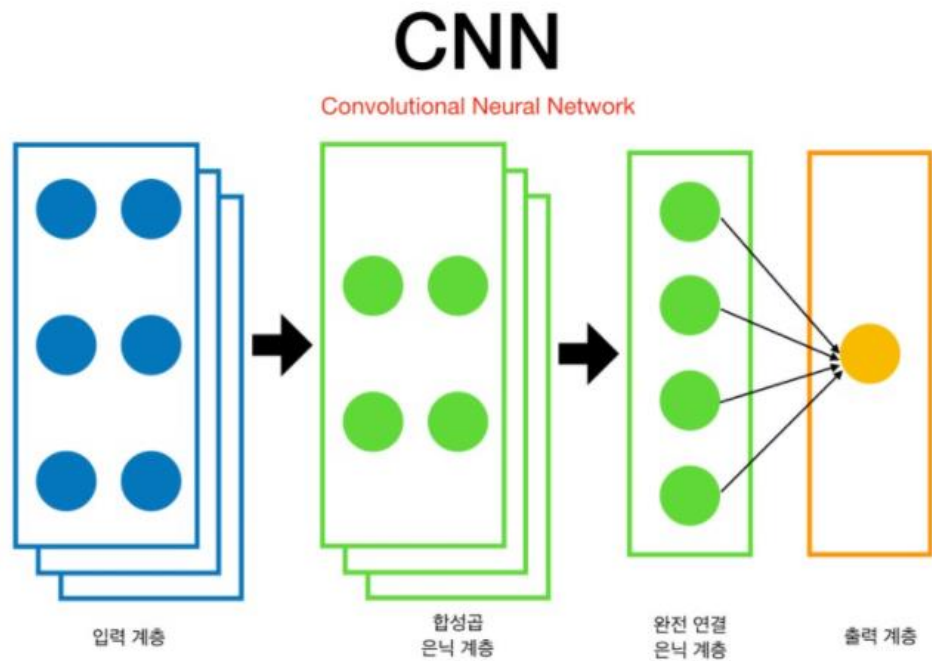
## DNN – Deep Neural Network



### <신경망 특징>

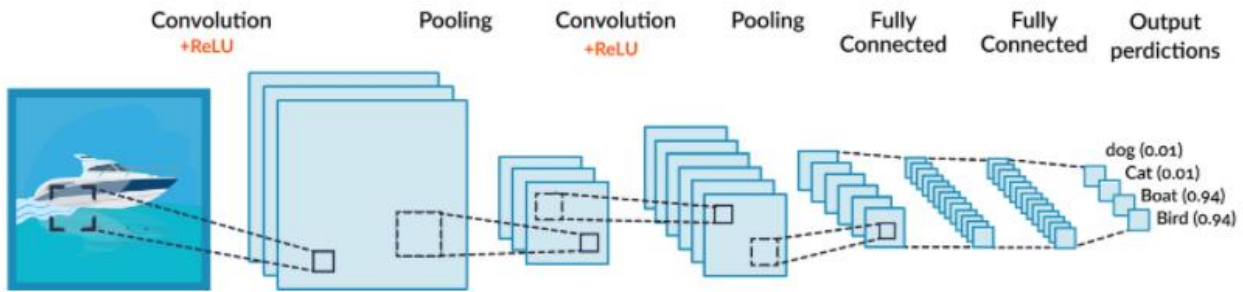
- ANN은 하나의 은닉층을 포함하지만 수백의 은닉층을 포함
- 컴퓨터가 스스로 분류레이블을 만들고, 공간을 왜곡하고, 데이터를 구분 짓는 과정을 반복하여 최적을 도출함
- 많은 데이터 반복학습, 사전학습과 오류역전파 기법을 통해 널리 사용
- DNN을 응용한 알고리즘 : CNN, RNN, LSTM, GRU등이 있음

## CNN(합성곱신경망) – Convolutional Neural Network

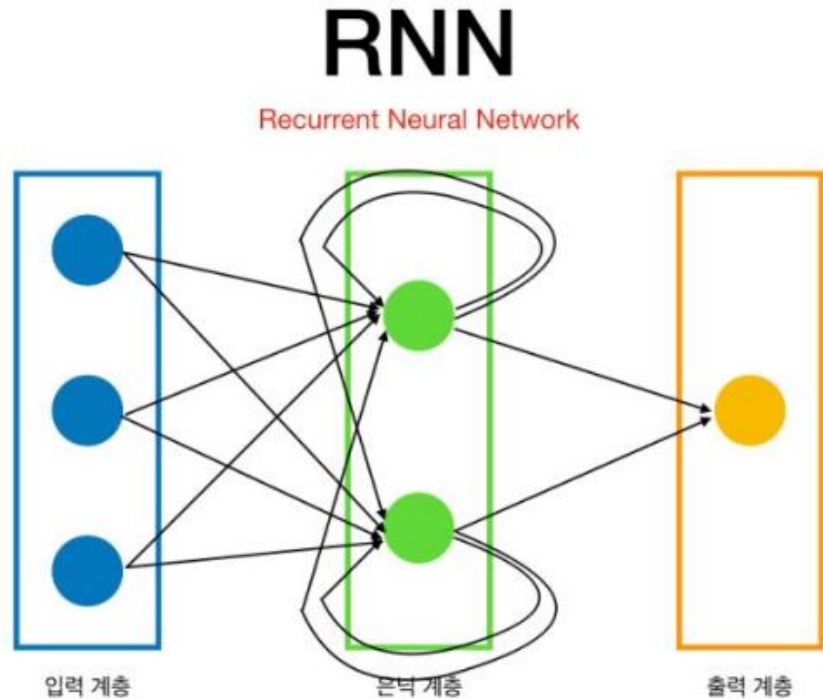


### <신경망 특징>

- 영상처리에 많이 활용되는 합성곱을 이용한 인공 신경망
- 합성곱을 이용해 가중치 수를 줄여 연산량을 줄이면서 이미지처리에 효과적
- 이미지의 특징점을 효율적으로 찾을 수 있는 인공 신경망
- 데이터의 특징들을 분석하여 패턴을 파악하는 구조
- Convolution과정과 Pooling과정을 통해 진행



## RNN(순환신경망) – Recurrent Neural Network



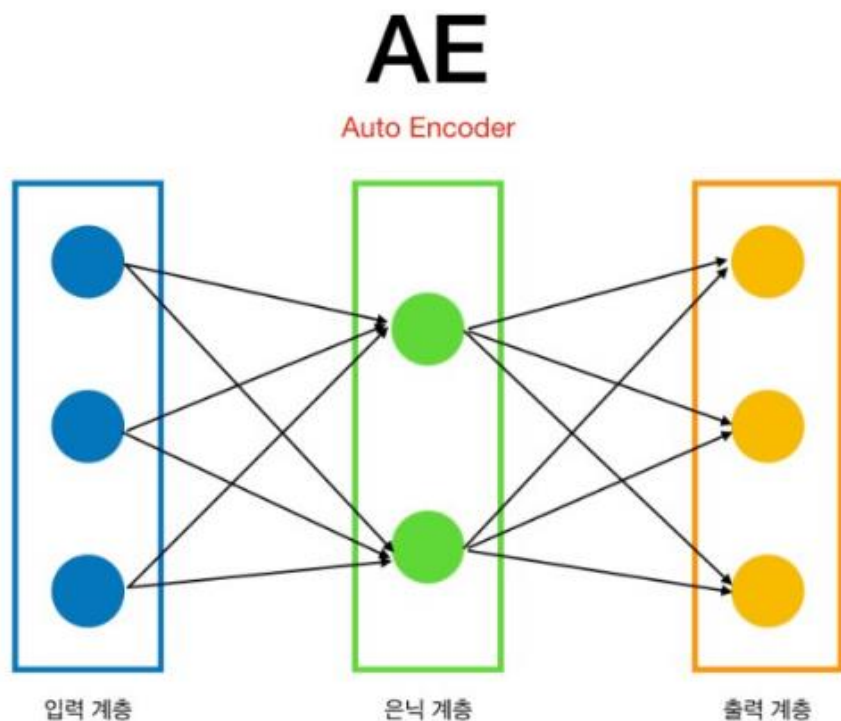
### <신경망 특징>

- 계층의 출력이 순환하는 인공신경망
- 순환방식은 은닉 계층의 결과가 다음 계층으로 넘어가며, 자기 계층으로 다시 돌아옴
- 시계열 정보 처리처럼 앞뒤 신호가 서로 상관도가 있는 경우 사용

### <RNN활용>

- 음성, 웨이브폼, 텍스트의 앞뒤를 분석하는등 언어처리

## AE - Auto Encoder



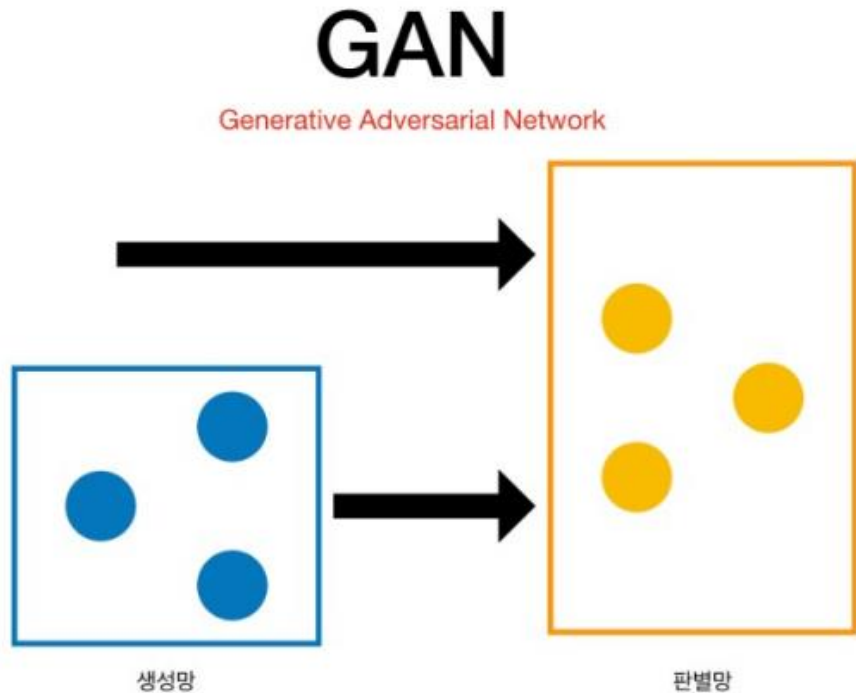
### <신경망 특징>

- 비지도학습 인공지능
- 레이블 정보가 없는 데이터 특성을 분석하거나 추출
- 목적은 입력 데이터의 특징점을 효율적으로 찾는 것
- 다차원 입력 데이터를 저차원 부호로 바꾸고 다시 저차원 부호를 처음 입력한 다차원 데이터로 바꾸면서 특징점을 분석

### <AE 활용>

- 데이터 압축, 저차원을 통한 데이터 관계 관찰, 배경 잡음 억제 등

## GEN(생성적 적대 신경망) – Generative Adversarial Network



### <신경망 특징>

- 신경망끼리 경쟁하여 최적화를 수행하는 생성형 신경망
- GAN내부의 두 신경망이 상호 경쟁하면서 학습
- 하나는 생성망, 하나는 판별망

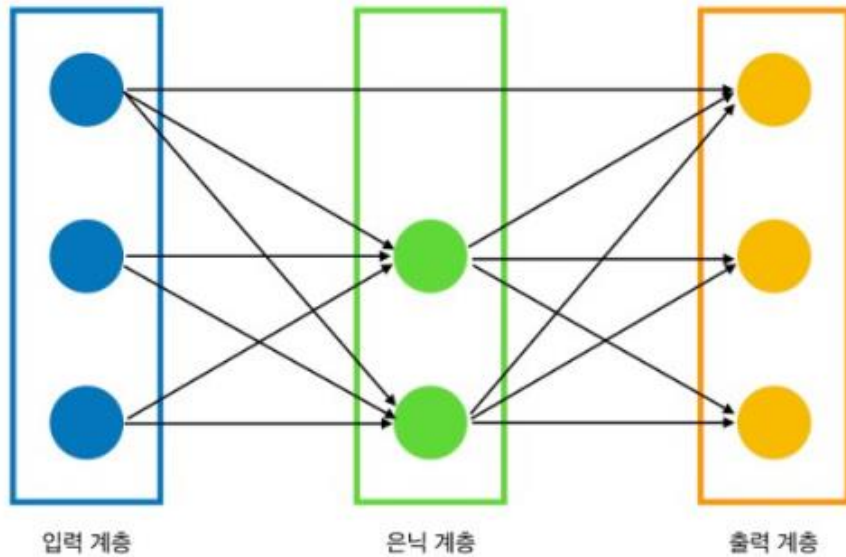
### <GAN활용>

- 이미지생성, 이미지복원, 동작흉내, 신약개발, 음성 생성, 편집, 변환, 복원



## U-NET

### U-NET



#### <신경망 특징>

- 저차원과 고차원 정보를 모두 사용하여 이미지의 특징 추출 인공신경망
- AE와 비슷
- AE는 디코딩의 저차원 정보만을 이용해 고차원 복원
- U-Net의 특징점도 함께 이용해서 이미지 특징점을 추출

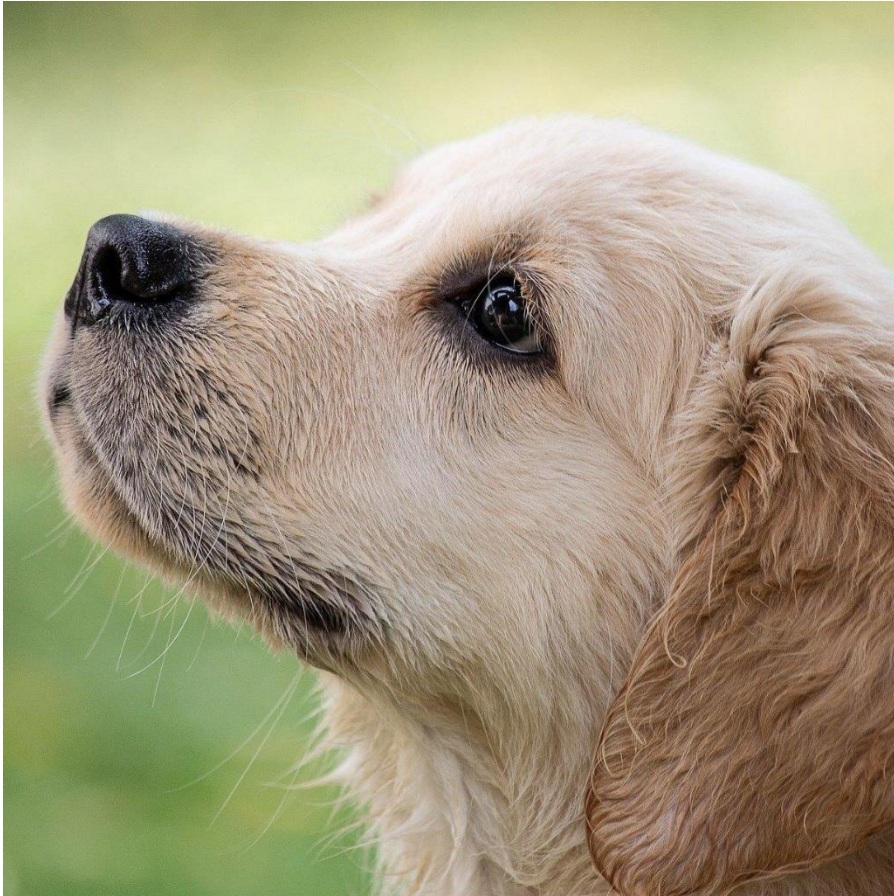
#### <U-NET활용>

- 2D-CNN분야, 이미지처리분야에서 사용

## CNN(합성곱신경망) – Convolutional Neural Network



## CNN(합성곱신경망) – Convolutional Neural Network





## CNN(합성곱신경망) – Convolutional Neural Network



CNN(합성곱신경망) – Convolutional Neural Network

## Convolutional Neural Network 합성곱신경망

**CNN = 합성곱**필터가 사진을 지나가면서 사진의 특징을 찾는 과정

(합성곱은 사람의 뇌에서 시각을 받아들이는 ‘시각피질’ 구조를 모방하여 만든 방법입니다.)

CNN(합성곱신경망) – Convolutional Neural Network

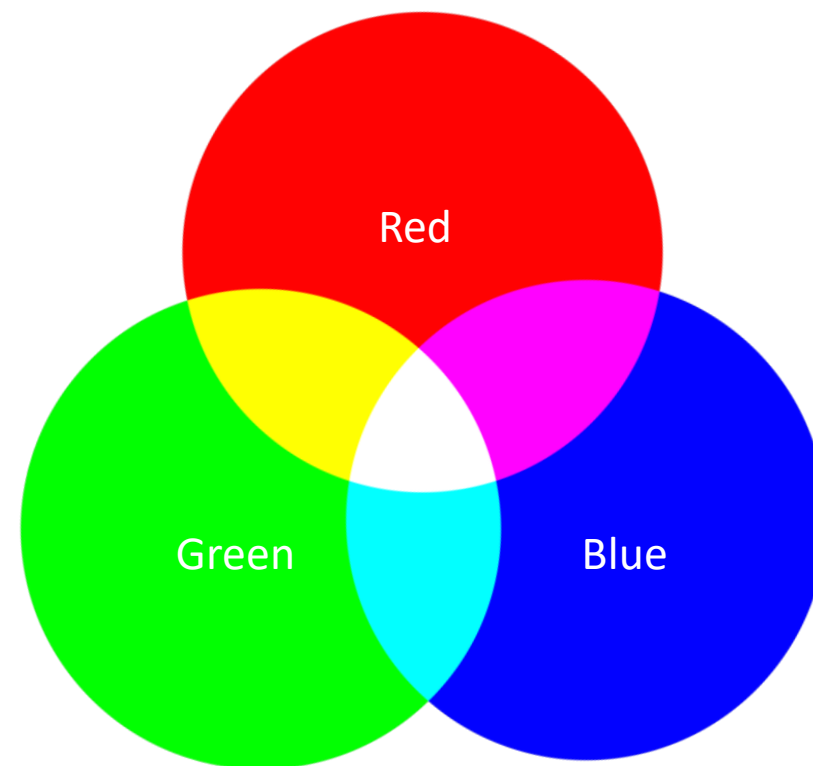
## 합성곱 신경망이란 ?

합성곱 신경망(CNN:Convolution Neural Network)은 시각적이미지분석에 가장 일반적으로 사용되는 인공신경망의 한 종류로, 입력 이미지로부터 특징을 추출하여 입력 이미지가 어떤 이미지인지 클래스를 분류하는 알고리즘

### <사용용도>

이미지 및 비디오 인식, 추천 시스템, 이미지 분류, 의료 이미지 분석 및 자연어 처리등에 응용됨

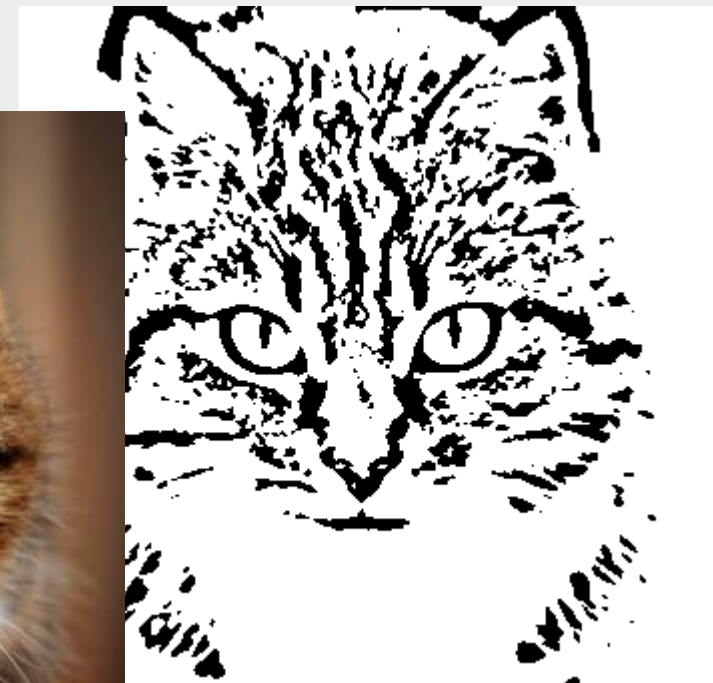
## CNN(합성곱신경망) – Convolutional Neural Network



RGB(빨강, 초록, 파랑의 세가지 채널로 인식)



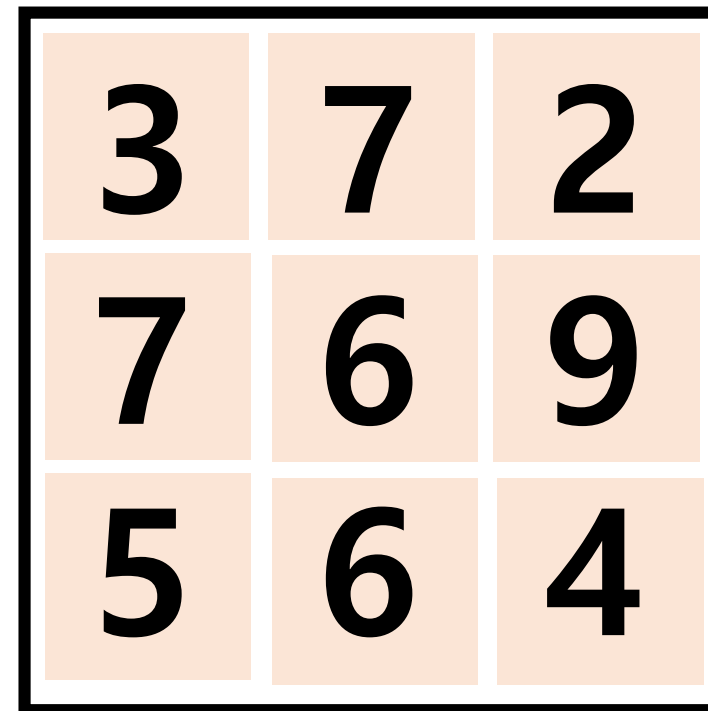
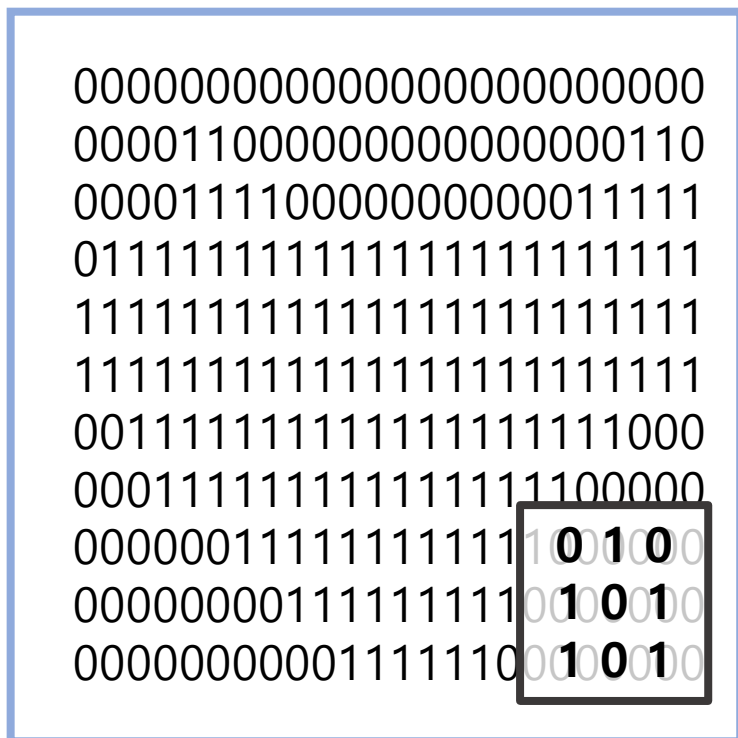
## CNN(합성곱신경망) – Convolutional Neural Network



원본 이미지에 필터를 씌워서 파악

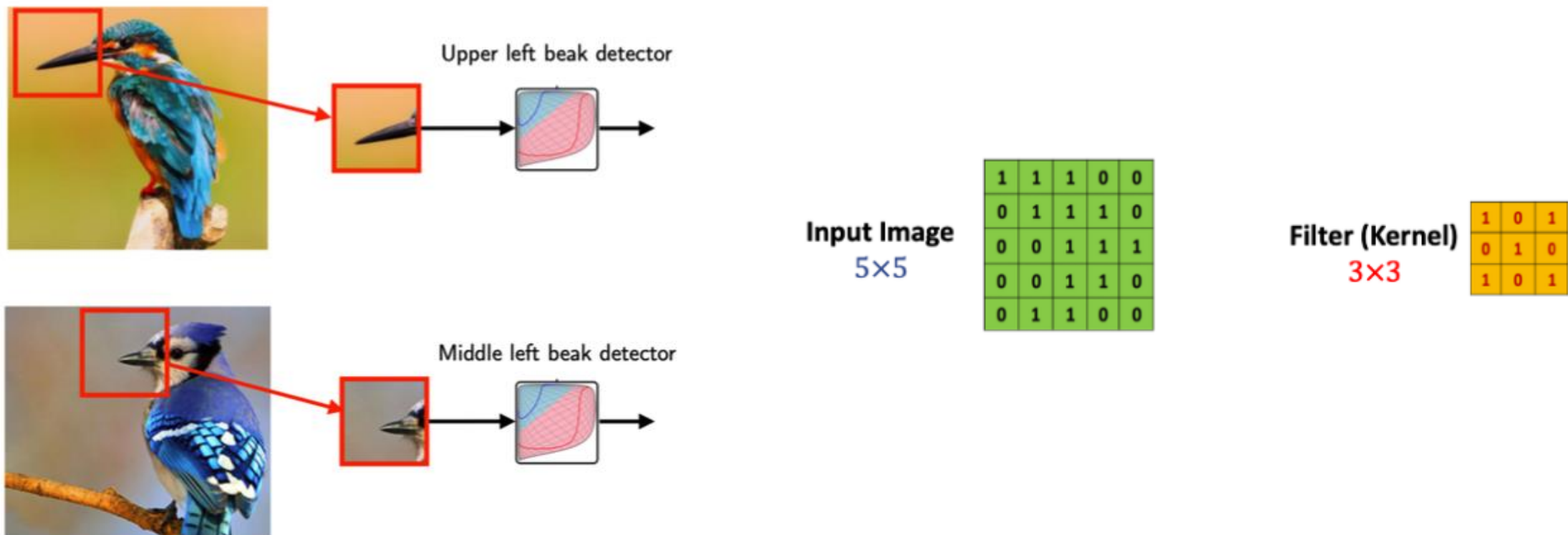


## CNN(합성곱신경망) – Convolutional Neural Network

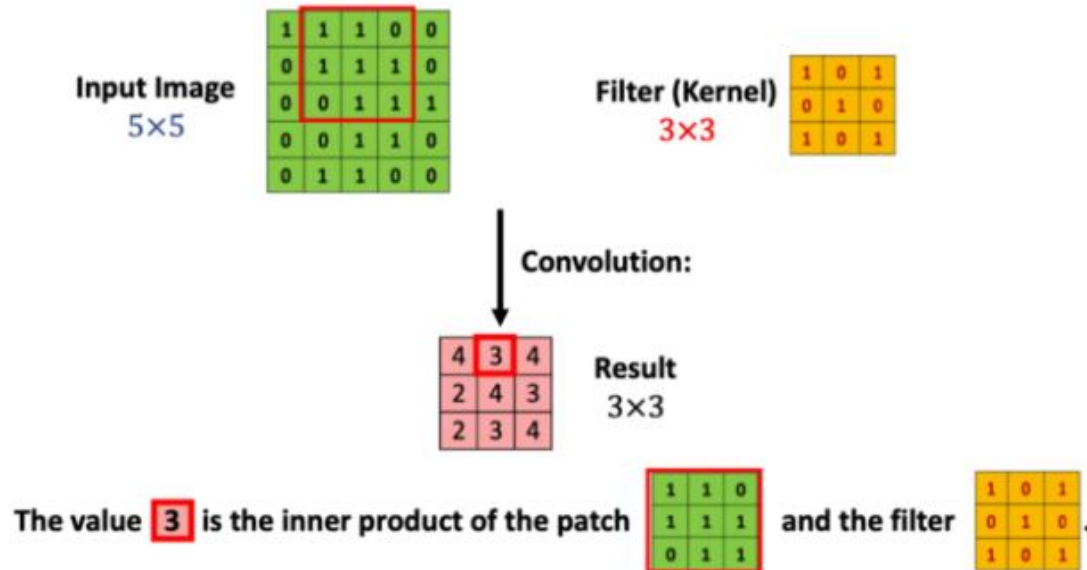
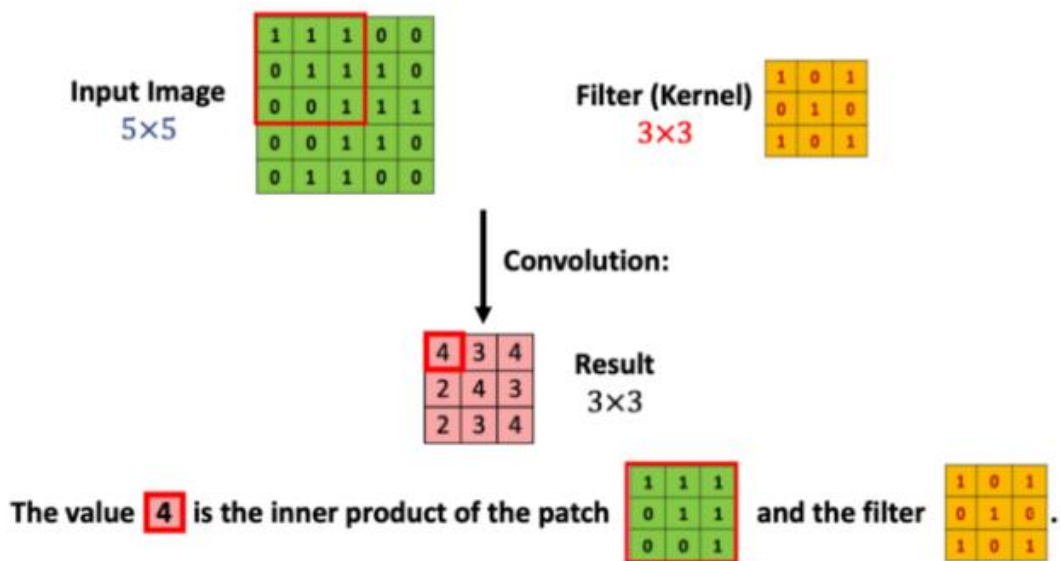


이렇게 나온 결과물을 **특성 지도**라고 합니다.

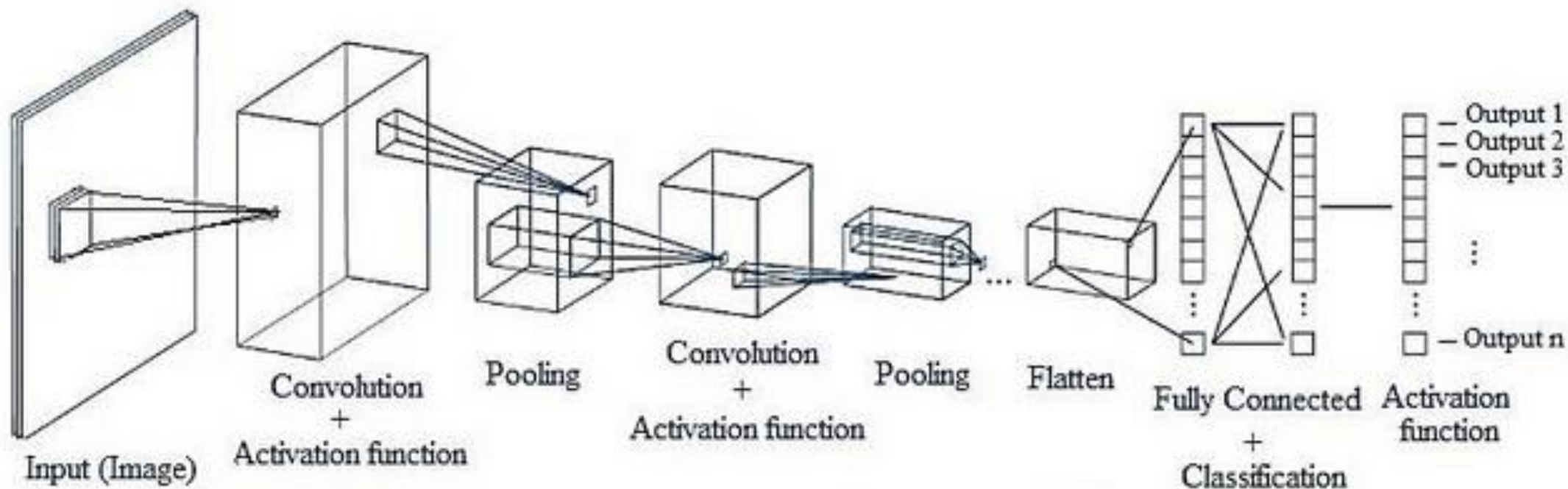
## CNN(합성곱신경망) – Convolutional Neural Network



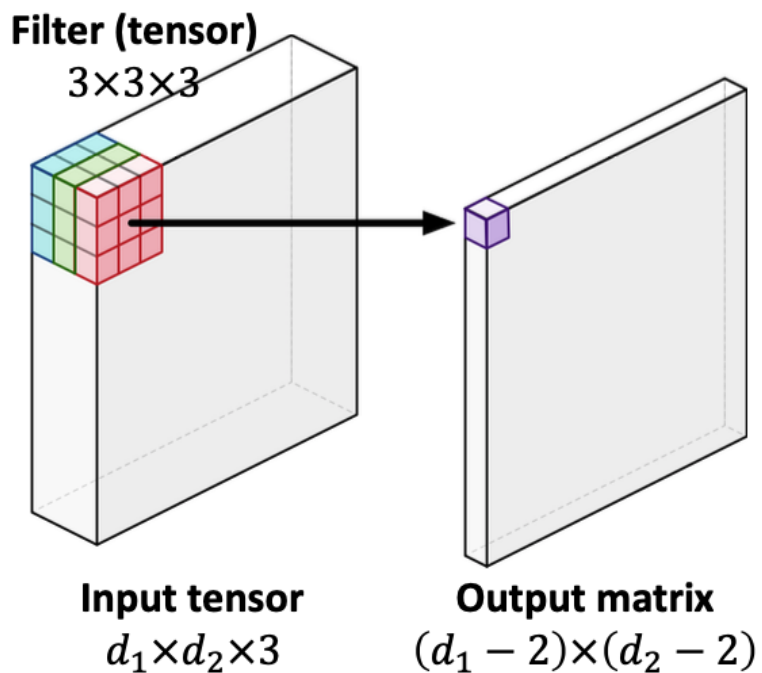
## CNN(합성곱신경망) – Convolutional Neural Network



## CNN(합성곱신경망) – Convolutional Neural Network



### CNN – Convolutional Neural Network



#### <Convolution 이란 ?>

- 데이터의 특징을 추출하는 과정으로 데이터의 각 성분의 인접 성분들을 조사
- 특징을 파악하고 특징을 한장으로 도출 -> Convolution Layer
- 이과정을 통하여 하나의 압축되며, 파라미터의 개수를 효과적으로 줄여주는 역할

#### <Pooling 이란 ?>

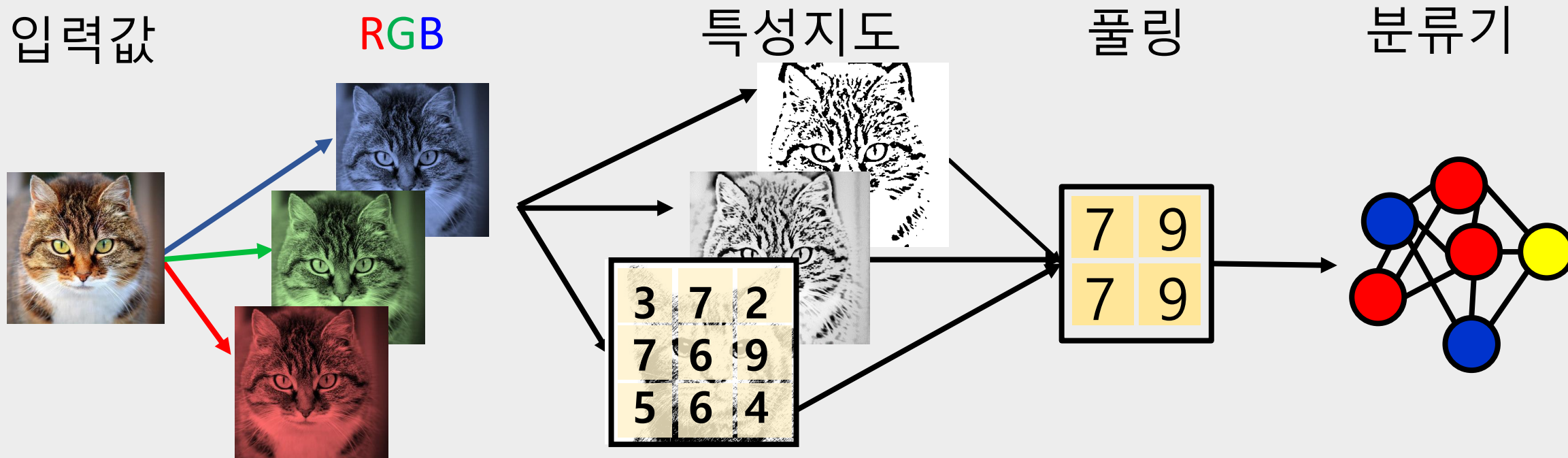
- Convolution 과정을 거친 레이어의 사이즈를 줄여주는 과정
- 단순히 데이터의 사이즈를 줄여줌
- 노이즈를 상쇄시키고 미세한 부분에서 일괄적인 특징을 제공

#### <CNN활용>

- 정보추출, 문장분류, 얼굴인식

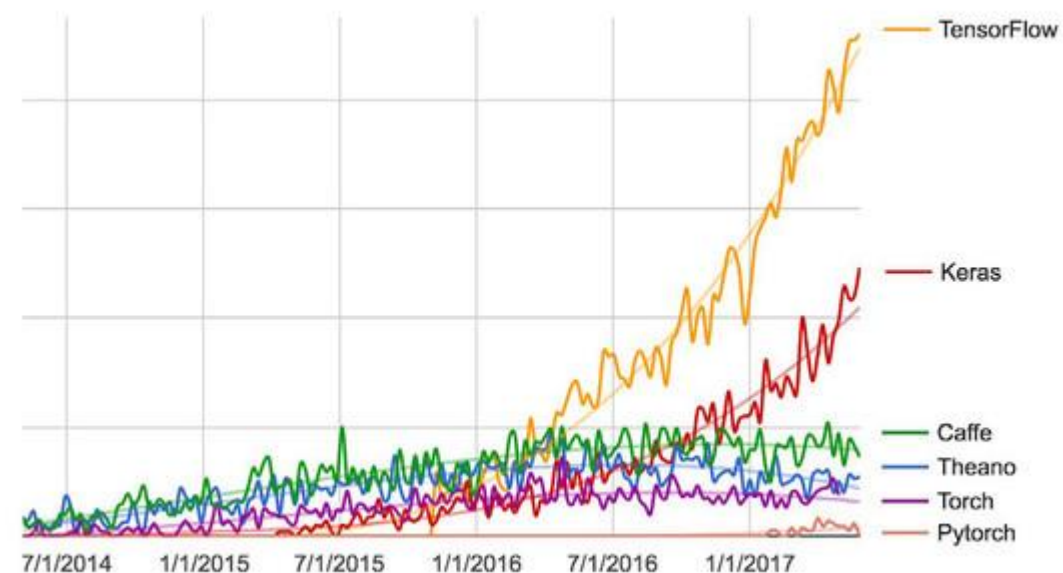
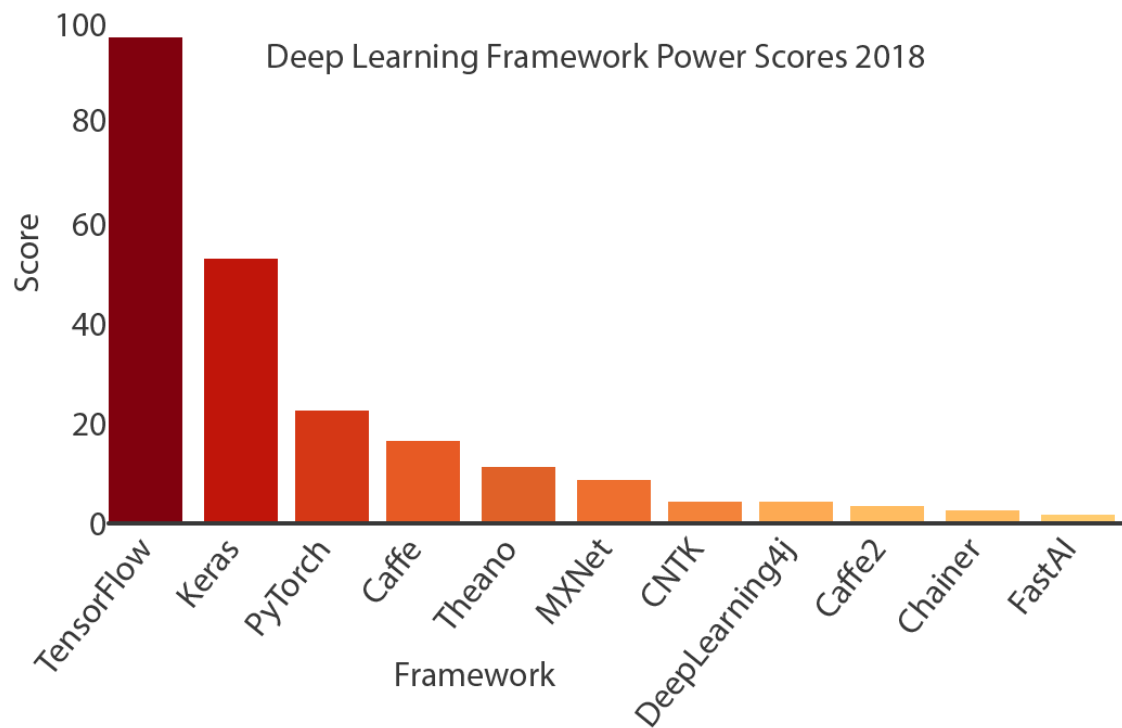
# 인공지능 알고리즘분석 합성곱신경망(CNN)

CNN – Convolutional Neural Network

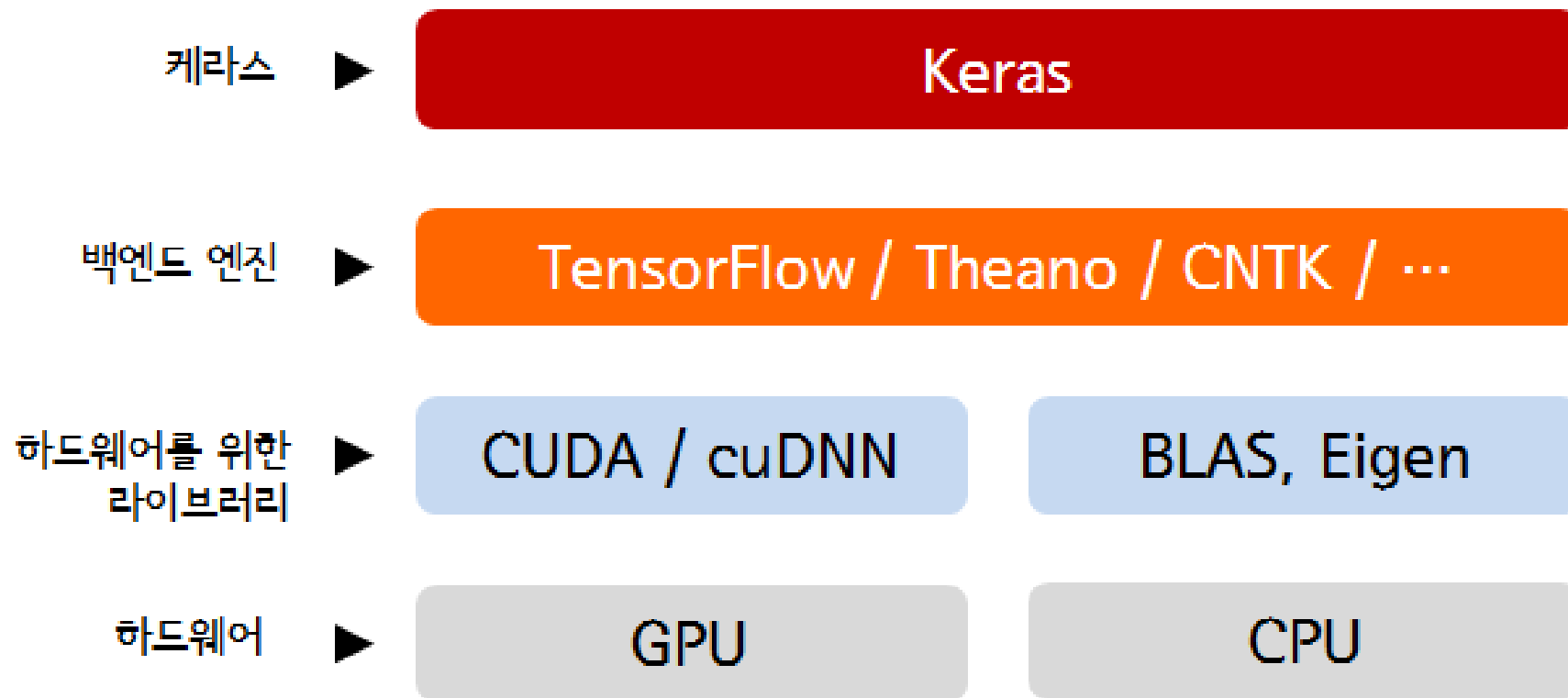


## 딥러닝 라이브러리

| 라이브러리명       | 개발 단체   | 지원 OS  | 언어   | 라이선스                           |
|--------------|---------|--|--|--------------------------------|
| 카페(Caffe)    | UC버클리   | <ul style="list-style-type: none"> <li>• macOS</li> <li>• 우분투</li> </ul>                                   | <ul style="list-style-type: none"> <li>• C++</li> <li>• 파이썬</li> </ul>                             | BSD 2-Cause license            |
| 체인너(Chainer) | 프리퍼드네트웍 | <ul style="list-style-type: none"> <li>• 우분투</li> <li>• Cent OS</li> </ul>                                 | <ul style="list-style-type: none"> <li>• 파이썬</li> </ul>  | MIT License                    |
| H2O          | H2O.ai  | <ul style="list-style-type: none"> <li>• Cent OS</li> <li>• 우분투</li> </ul>                                 | <ul style="list-style-type: none"> <li>• 자바</li> <li>• 파이썬</li> <li>• R</li> <li>• 스칼라</li> </ul>  | AGPL                           |
| 텐서플로         | 구글      | <ul style="list-style-type: none"> <li>• macOS</li> <li>• 우분투</li> <li>• 윈도우</li> </ul>                    | <ul style="list-style-type: none"> <li>• C++</li> <li>• Go</li> <li>• 자바</li> <li>• 파이썬</li> </ul> | Apache 2.0 open source license |
| 테아노(Theano)  | 몬트리올 대학 | <ul style="list-style-type: none"> <li>• macOS</li> <li>• 우분투</li> <li>• Cent OS</li> <li>• 윈도우</li> </ul> | <ul style="list-style-type: none"> <li>• 파이썬</li> </ul>  | 오픈 소스                          |







## MNIST 데이터셋 불러오기

```
In [25]: import tensorflow as tf
from tensorflow.keras import datasets, layers, models

# 1. MNIST 데이터셋 불러오기
(train_images, train_labels), (test_images, test_labels) = datasets.mnist.load_data()
```

## 데이터 전처리하기

```
In [26]: # 2. 데이터 전처리하기
train_images = train_images.reshape((60000, 28, 28, 1))
test_images = test_images.reshape((10000, 28, 28, 1))
train_images, test_images = train_images / 255.0, test_images / 255.0
```

## 합성곱 신경망 구성하기

```
In [27]: # 3. 합성곱 신경망 구성하기
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))

model.summary()
```

Model: "sequential\_1"

| Layer (type)                   | Output Shape       | Param # |
|--------------------------------|--------------------|---------|
| =====                          |                    |         |
| conv2d_2 (Conv2D)              | (None, 26, 26, 32) | 320     |
| -----                          |                    |         |
| max_pooling2d_1 (MaxPooling2D) | (None, 13, 13, 32) | 0       |
| -----                          |                    |         |
| conv2d_3 (Conv2D)              | (None, 11, 11, 64) | 18496   |
| -----                          |                    |         |
| max_pooling2d_2 (MaxPooling2D) | (None, 5, 5, 64)   | 0       |
| -----                          |                    |         |
| conv2d_4 (Conv2D)              | (None, 3, 3, 64)   | 36928   |
| -----                          |                    |         |
| max_pooling2d_3 (MaxPooling2D) | (None, 1, 1, 64)   | 0       |
| =====                          |                    |         |
| Total params: 55,744           |                    |         |
| Trainable params: 55,744       |                    |         |
| Non-trainable params: 0        |                    |         |
| -----                          |                    |         |

## Dense 층 추가하기

```
In [28]: # 4. Dense 층 추가하기
model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(10, activation='softmax'))

model.summary()
```

Model: "sequential\_1"

| Layer (type)                   | Output Shape       | Param # |
|--------------------------------|--------------------|---------|
| conv2d_2 (Conv2D)              | (None, 26, 26, 32) | 320     |
| max_pooling2d_1 (MaxPooling2D) | (None, 13, 13, 32) | 0       |
| conv2d_3 (Conv2D)              | (None, 11, 11, 64) | 18496   |
| max_pooling2d_2 (MaxPooling2D) | (None, 5, 5, 64)   | 0       |
| conv2d_4 (Conv2D)              | (None, 3, 3, 64)   | 36928   |
| max_pooling2d_3 (MaxPooling2D) | (None, 1, 1, 64)   | 0       |
| flatten_1 (Flatten)            | (None, 64)         | 0       |
| dense_2 (Dense)                | (None, 64)         | 4160    |
| dense_3 (Dense)                | (None, 10)         | 650     |
| Total params: 60,554           |                    |         |
| Trainable params: 60,554       |                    |         |
| Non-trainable params: 0        |                    |         |

## 모델 컴파일하기

```
In [29]: # 5. 모델 컴파일하기
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

## 훈련하기

```
In [30]: # 6. 훈련하기
model.fit(train_images, train_labels, epochs=5)

Epoch 1/5
1875/1875 [=====] - 16s 8ms/step - loss: 0.4579 - accuracy: 0.8519
Epoch 2/5
1875/1875 [=====] - 15s 8ms/step - loss: 0.0808 - accuracy: 0.9749 0s - loss:
Epoch 3/5
1875/1875 [=====] - 15s 8ms/step - loss: 0.0576 - accuracy: 0.9819
Epoch 4/5
1875/1875 [=====] - 15s 8ms/step - loss: 0.0468 - accuracy: 0.9859
Epoch 5/5
1875/1875 [=====] - 15s 8ms/step - loss: 0.0356 - accuracy: 0.9889
```

```
Out [30]: <tensorflow.python.keras.callbacks.History at 0x2374d27d408>
```

## 모델 평가하기

```
In [31]: # 7. 모델 평가하기
loss, acc = model.evaluate(test_images, test_labels, verbose=2)

313/313 - 1s - loss: 0.0467 - accuracy: 0.9864
```

# 크롤링 실습(TEXT)

## 네이버의 지식iN 크롤링 예제(파이썬)



```

web_crawling.py
D: > python_exam > Django > web_crawling.py > ...
1 import requests
2 from bs4 import BeautifulSoup
3
4 url = 'https://kin.naver.com/search/list.nhn?query=%ED%8C%8C%EC%9D%B4%EC%8D%AC'
5
6 response = requests.get(url)
7
8 if response.status_code == 200:
9     html = response.text
10    soup = BeautifulSoup(html, 'html.parser')
11    ul = soup.select_one('ul.basic1')
12    titles = ul.select('li > dl > dt > a')
13    for title in titles:
14        print(title.get_text())
15 else :
16     print(response.status_code)
17
18
    
```

파이썬 크롤링 프로그램(소스)

```

Anaconda Prompt (anaconda3)
C:\python_exam>python web_crawling.py
파이썬 기초 질문입니다.
파이썬 인강으로 공부 해보려고해요
파이썬의 한계
파이썬 기초 질문
파이썬 온라인강의 퀄리티 좋은 곳 알
파이썬으로 어떻게 하나요?
파이썬인강 어디서 볼까요?
파이썬 Folium 사용방법 질문
파이썬 문제
파이썬 질문입니다.(채택 바로 해드림)
(django) D:\python_exam>Django>
    
```

파이썬 크롤링 실행(데이터 취합)



네이버 지식인 정보

## 데이터수집 - 크롤링 라이브러리 설치

- pip install beautifulsoup4
- pip install selenium
- pip install requests

위의 4가지의 라이브러리를 설치하면 크롤링을 위한 기본 준비완료



## 텍스트 크롤링 실습


<프로그램 다운로드>  
<https://url.kr/gaew2q>

# 크롤링 실습(이미지)

## 이미지 크롤링 실습




<프로그램 다운로드>  
<https://url.kr/qjx3ka>


## 이미지 크롤링 실습





크롬드라이브 다운로드


×





 전체

 이미지

 뉴스

 동영상

 지도

 더보기

도구

검색결과 약 153,000,000개 (0.45초)

수정된 검색어에 대한 결과: 크롬 드라이버 다운로드  
다음 검색어로 대신 검색: 크롬드라이브 다운로드

<https://chromedriver.chromium.org/downloads>

ChromeDriver - WebDriver for Chrome - Downloads

If you are using **Chrome** version 93, please **download** ChromeDriver 93.0.4577.15 ... Resolved issue 3734: **Driver** should Error Attempting to set Cookie on Wrong ...  
[Version Selection](#) · [ChromeDriver Canary](#) · [Need help?](#)

함께 검색한 항목

크롬드라이버 확인

ChromeDriver

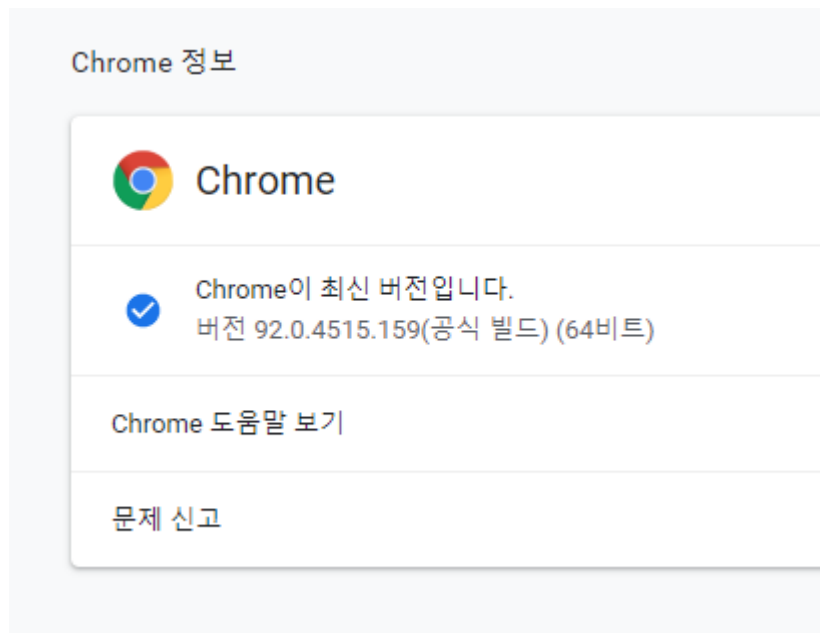
크롬드라이버 버전 확인

크롬드라이버 사용법

크롬드라이버 64비트

Chromedriver win64

## 이미지 크롤링 실습



### Current Releases

- If you are using Chrome version 93, please download [ChromeDriver 93.0.4577.15](#)
- If you are using Chrome version 92, please download [ChromeDriver 92.0.4515.107](#)
- If you are using Chrome version 91, please download [ChromeDriver 91.0.4472.101](#)
- For older version of Chrome, please see below for the version of ChromeDriver that supports it.

If you are using Chrome from Dev or Canary channel, please following instructions on the [ChromeDriver Canary](#) page.

For more information on selecting the right version of ChromeDriver, please see the [Version Selection](#) page.

#### **ChromeDriver 93.0.4577.15**

Supports Chrome version 93

인공지능활용지도사 양성 과정(2일차)

# 인공지능키트(AIIT) 실습 및 개발 실습

- (주)서울컴퍼니 -

- 질문과 답변 -