

티처블머신과 함께 하는 인공지능 교육!

메카넘휠 AI 자율주행로봇 "단비"

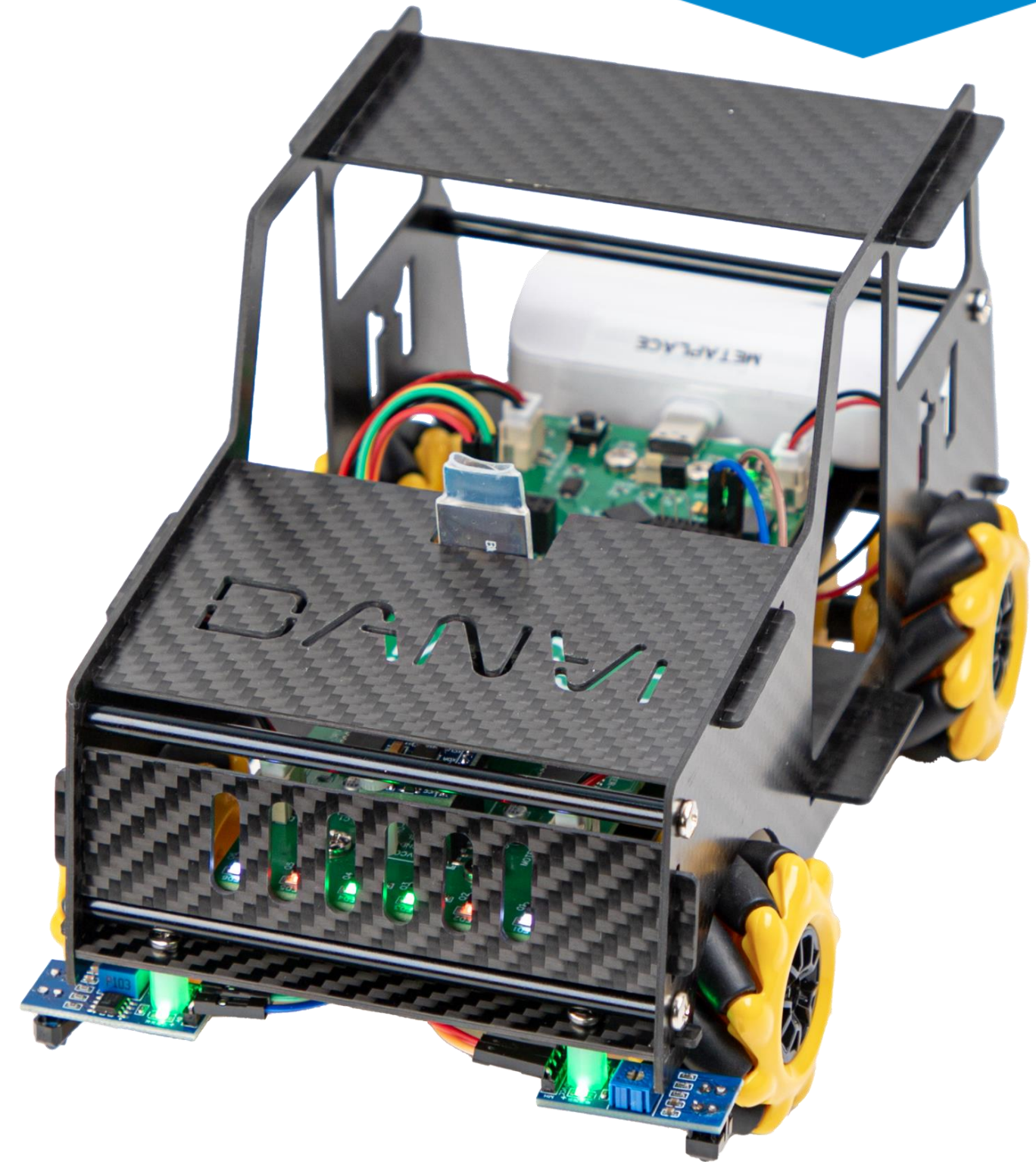
METAPLACE



(주)메타플레이스
강원특별자치도 춘천시 후석로 462번길 7 춘천ICT벤처센터
313호
T.033-252-4787

자유롭게 이동하는 메카넘휠 AI 로봇

단바 Mecanum Wheel Robot AI DanVI



METAPLACE

[제 1교시] - [학습목표]

- ① 파이썬 3.8 설치
- ② 파이참 설치 (프로젝트 및 가상환경 구축)
- ③ 드로이드 캠 설치 (크롬 브라우저 확인)
- ④ 인공지능 알고리즘 소개 (CNN, RNN)
- ⑤ 배경지식 전달 : 전산학개론

- ① 프레임 이해
- ② 파이썬 이해 (자료구조 - 리스트, 튜플, 딕셔너리)
- ③ OpenCV 라이브러리 설치 및 소개
- ④ 드로이드 캠 영상수신
- ⑤ 배경지식 전달 : 네트워크 이해

[제 3교시] - [학습목표]

- ① TensorFlow 라이브러리 설치 및 소개
- ② Pillow 라이브러리 설치 및 소개
- ③ 파이썬 라이브러리 import 방법 소개
- ④ 파이썬 이해 (함수, Class - Teachable Machine)
- ⑤ 배경지식 전달 : 파일관리자

[제 4교시] - [학습목표]

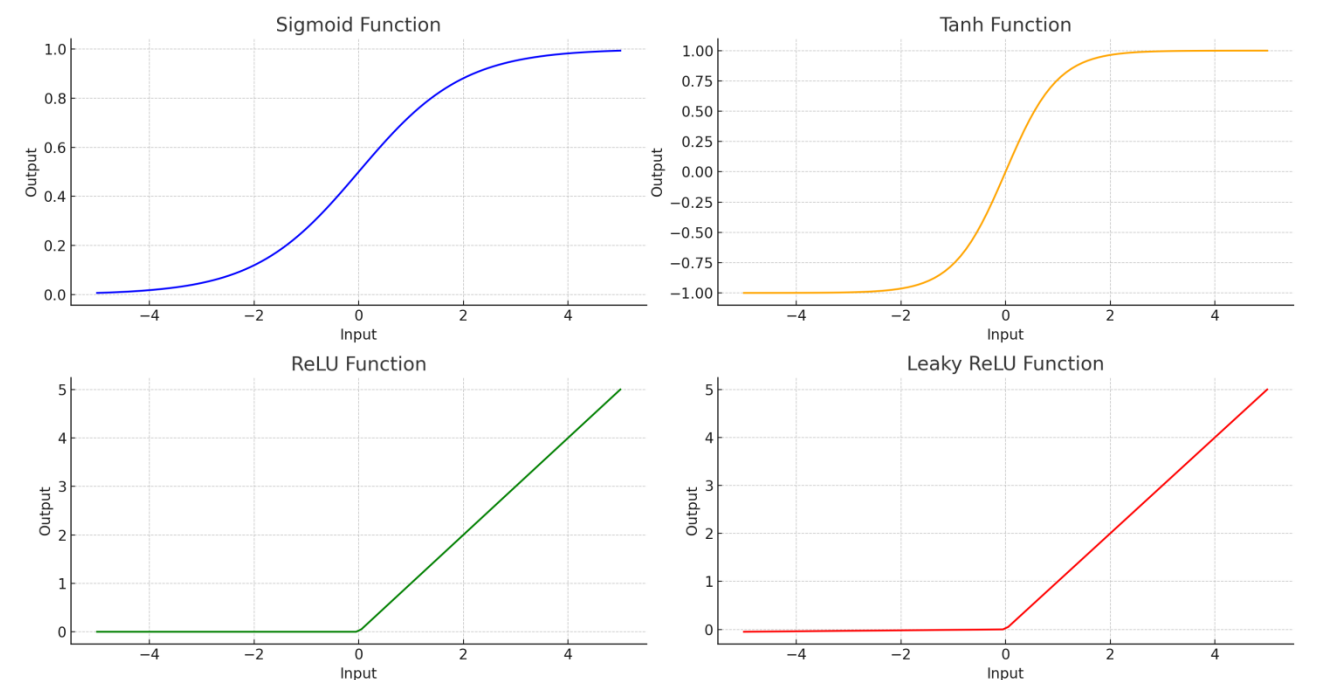
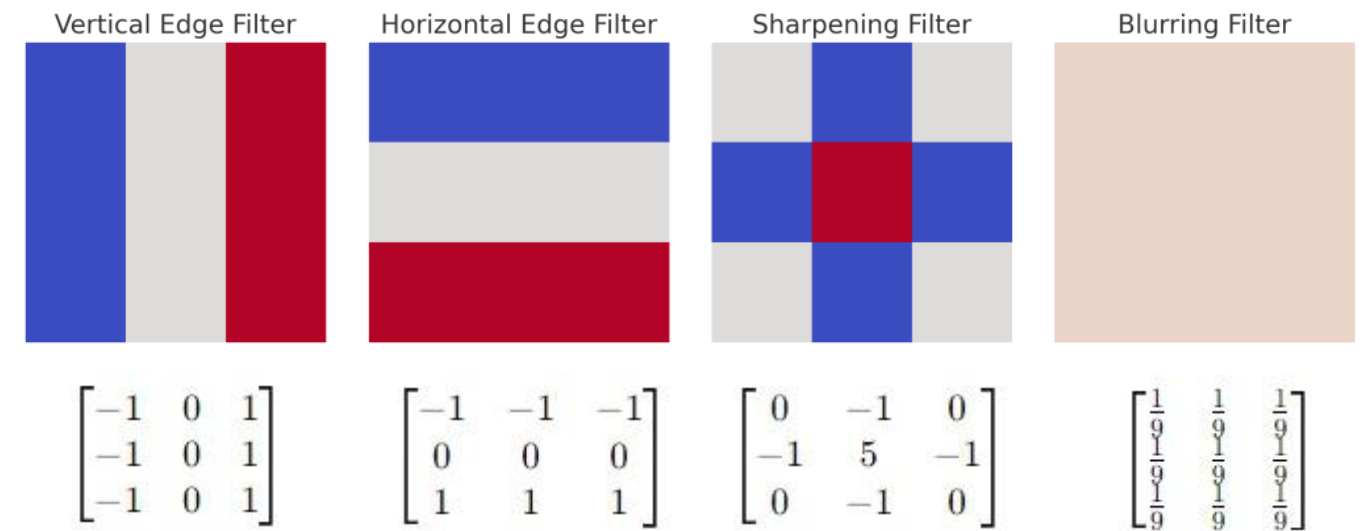
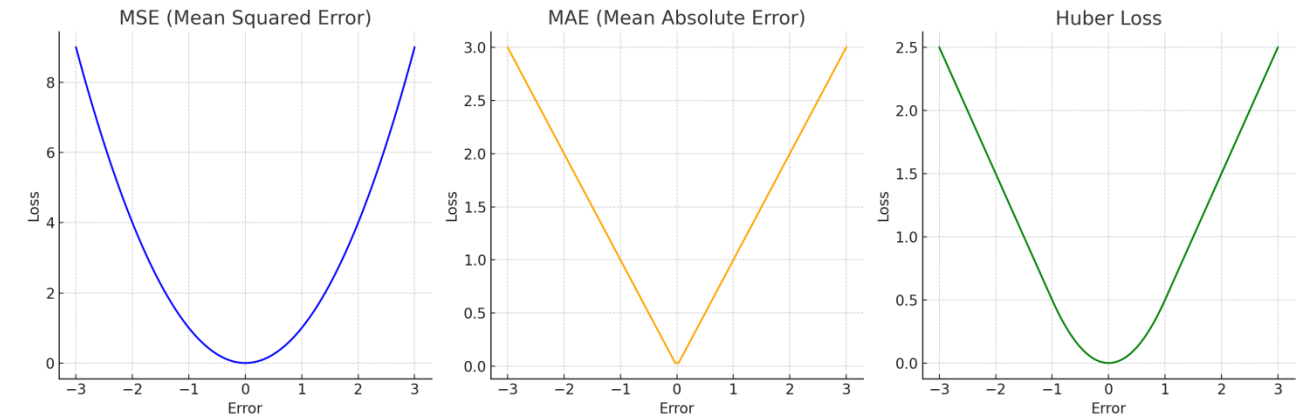
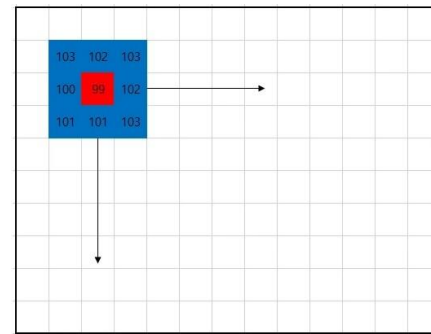
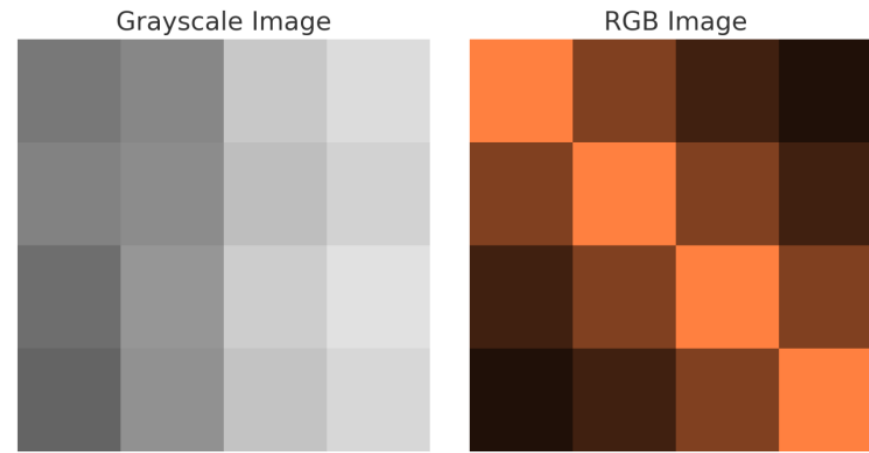
① 블루투스 소개

② bleak 라이브러리 설치 및 소개

③ 블루투스 검색 및 조종 실습

④ 인공지능 알고리즘 사용 (데이터셋, 모델학습/평가)

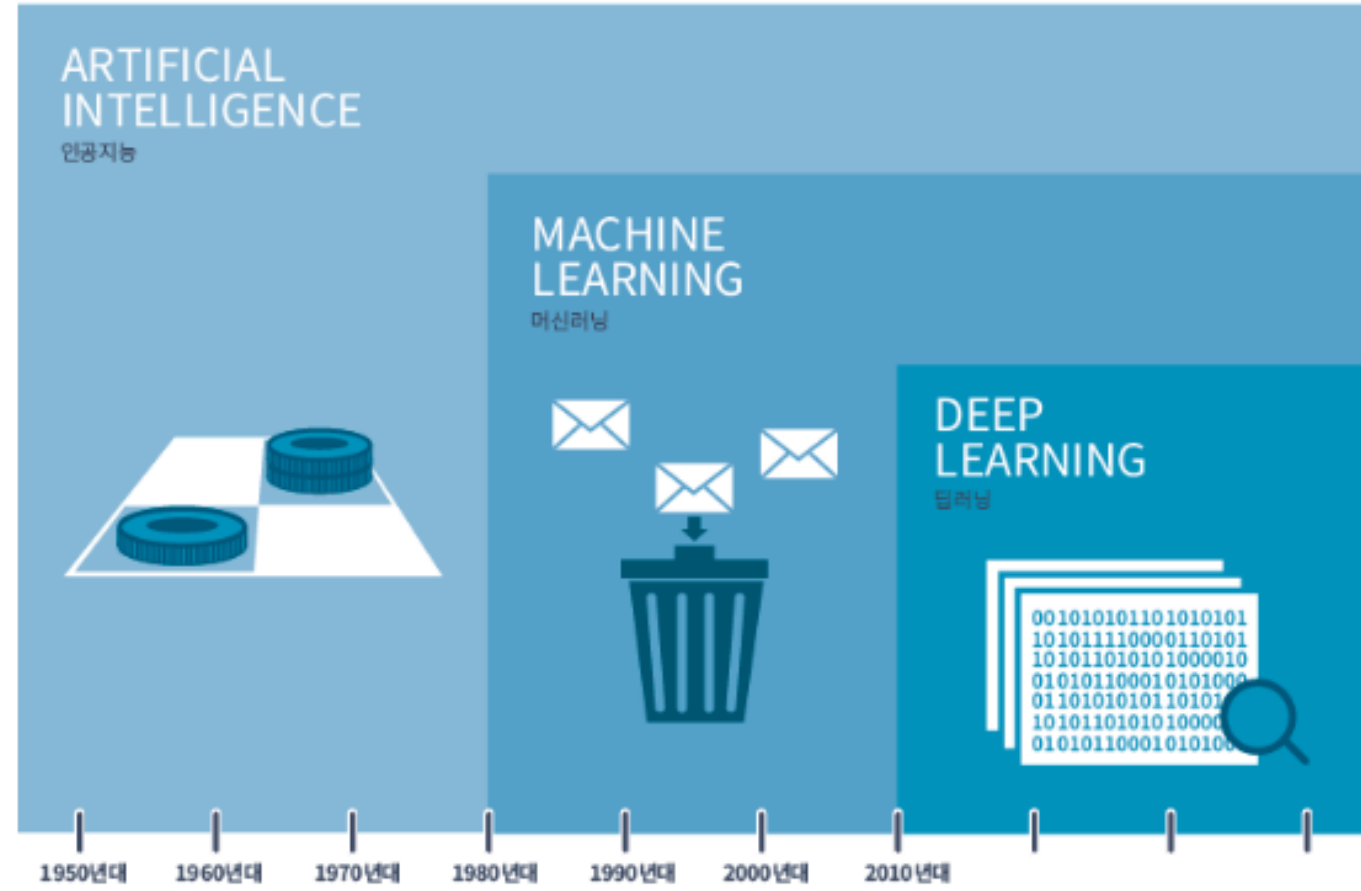
⑤ 배경지식 전달 : 무선통신 - 변복조기



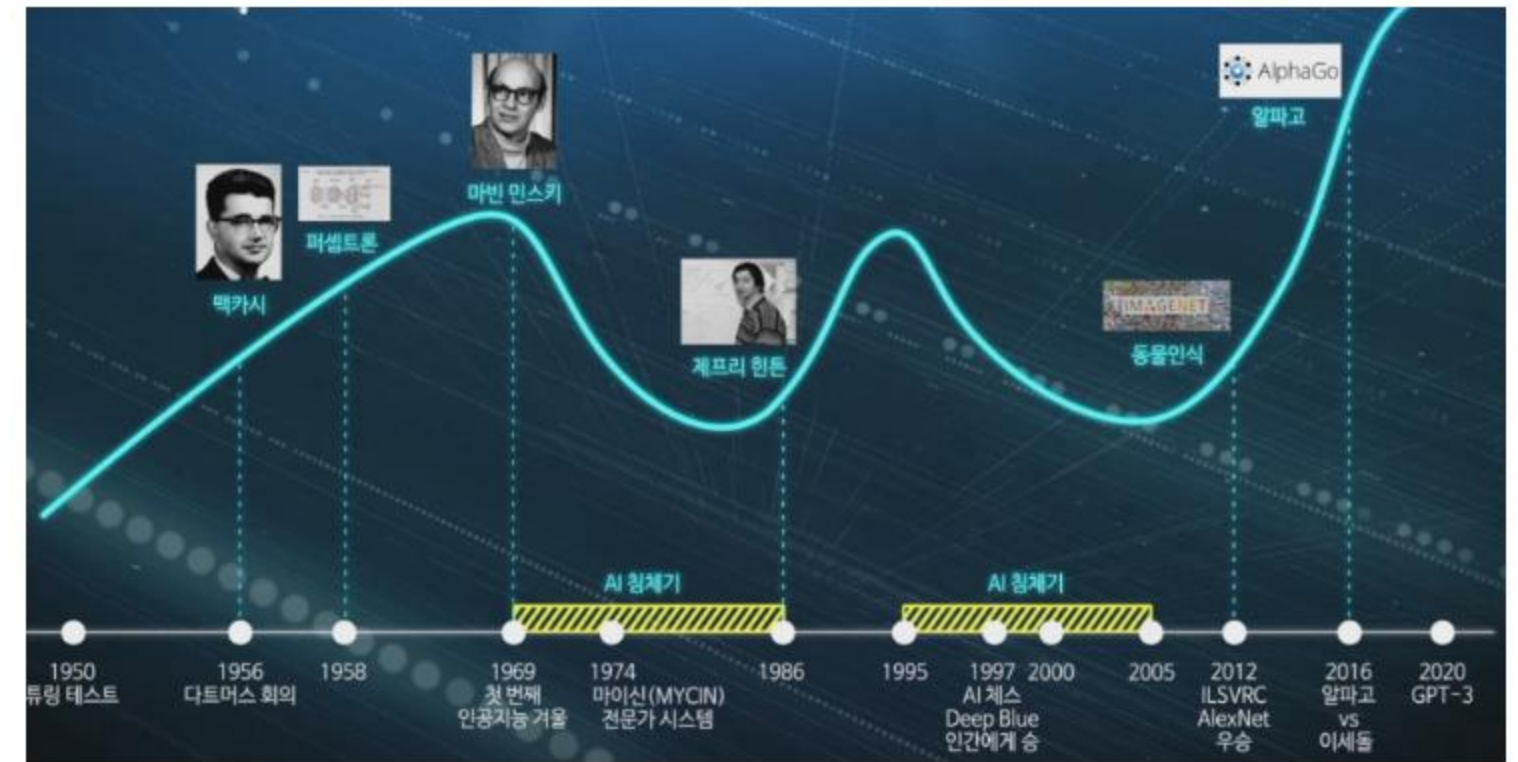
[제 5교시] - [학습목표]

- ① 파이썬 심화 (자율주행 실습)
- ② AWS DeepRacer League 소개
- ③ AWS DeepRacer 모델 만들기 (보상함수 셋팅)
- ④ AWS DeepRacer 시뮬레이션
- ⑤ 배경지식 전달 : ChatGPT

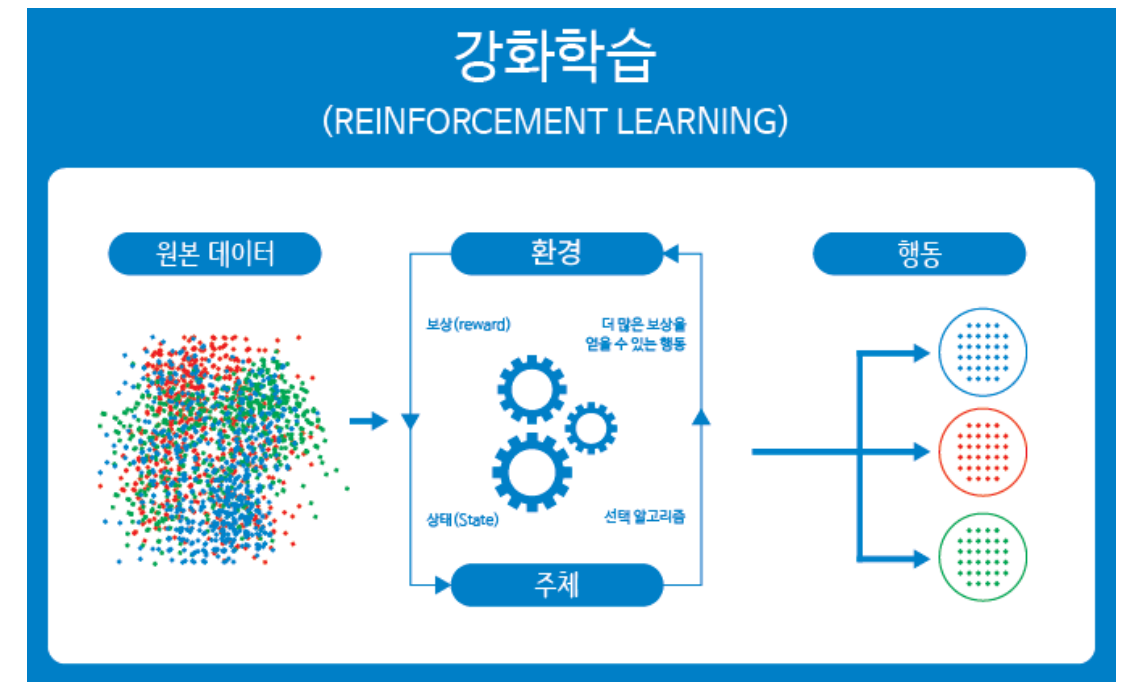
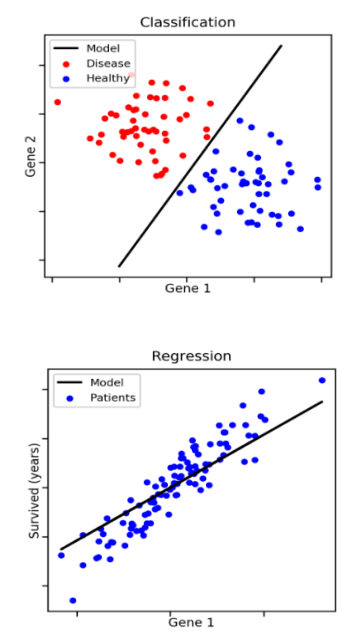
[파이선 인공지능 교육 - 인공지능]



인공지능 역사

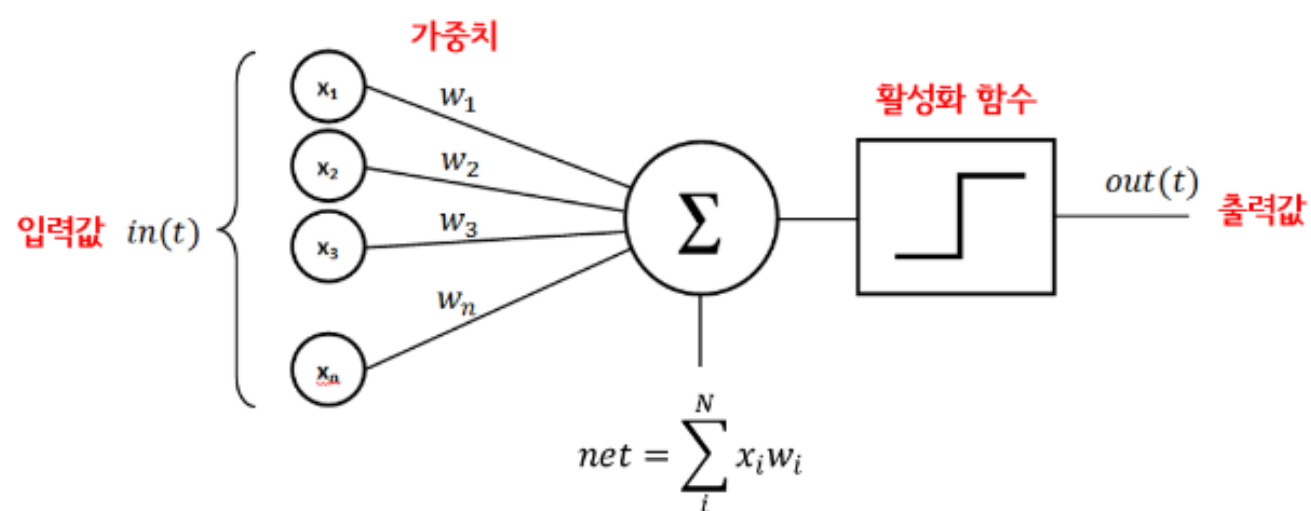


출처: EBS

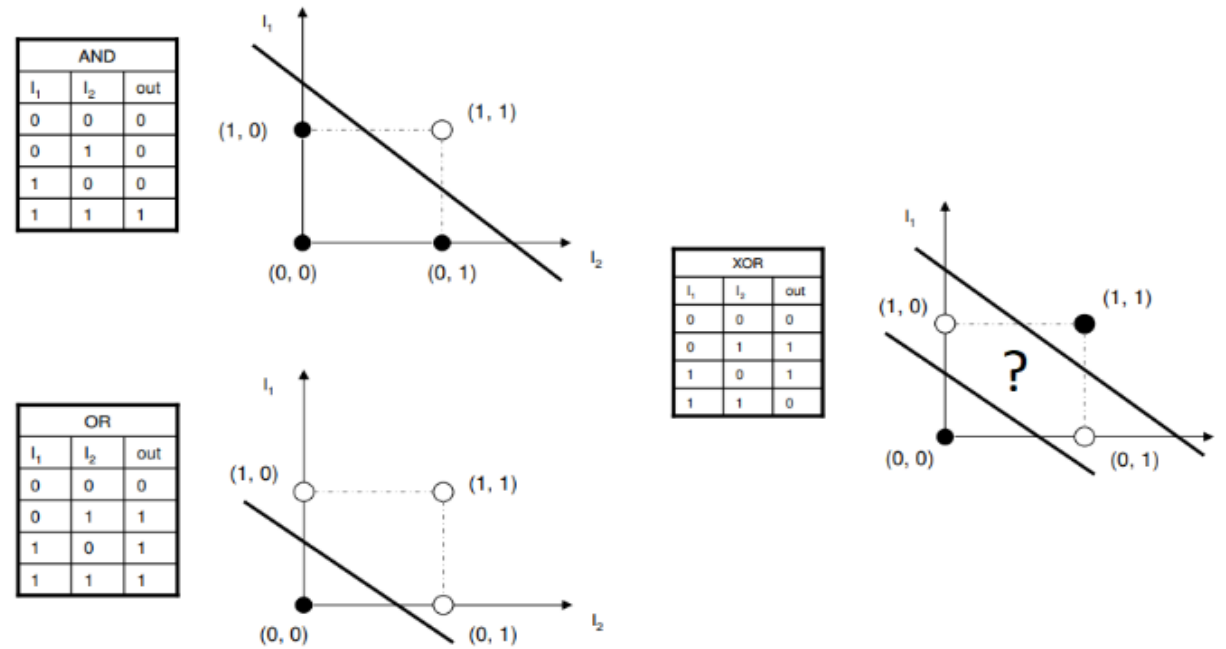


[파이선 인공지능 교육 - 퍼셉트론]

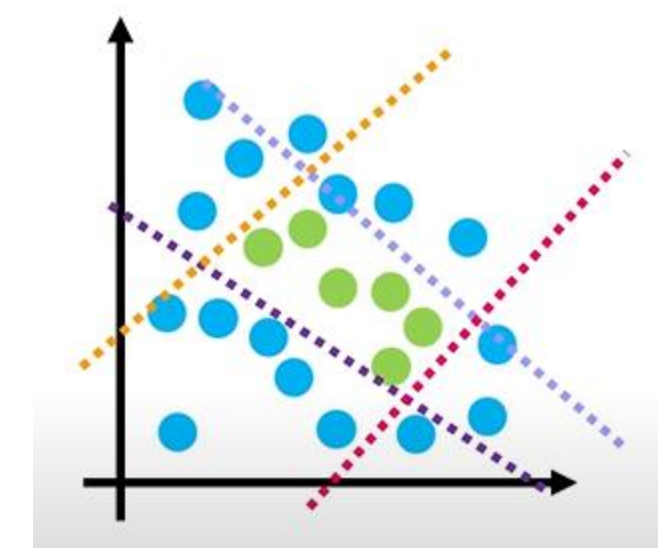
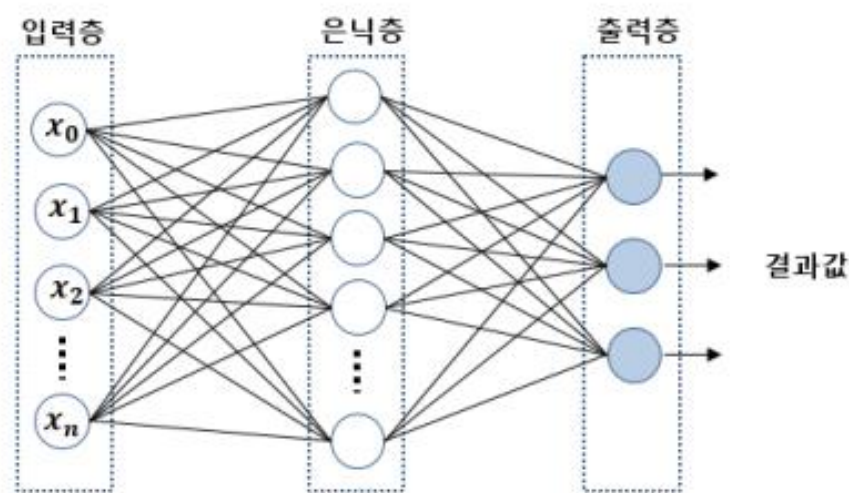
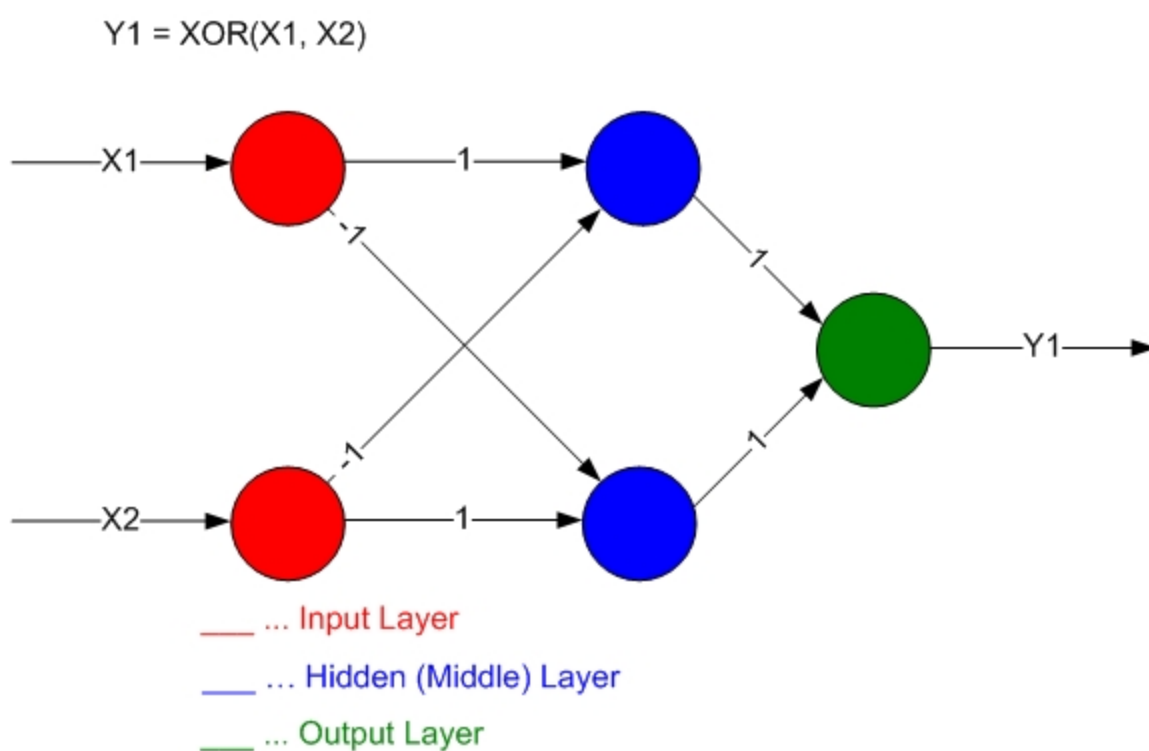
신경세포(=뉴런)의 동작 과정을 통계학적으로 모델링한 알고리즘



AND와 OR와 같은 선형 문제는 해결
XOR과 같은 비선형 문제는 해결 못함

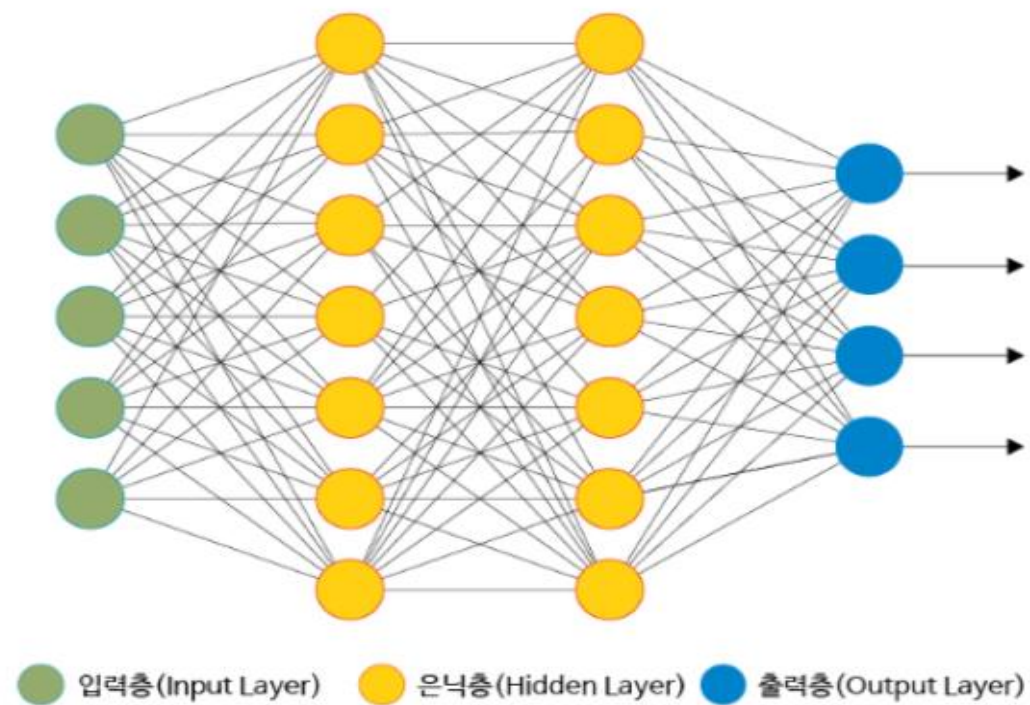


다층 퍼셉트론(Multi-Layer Perceptron)

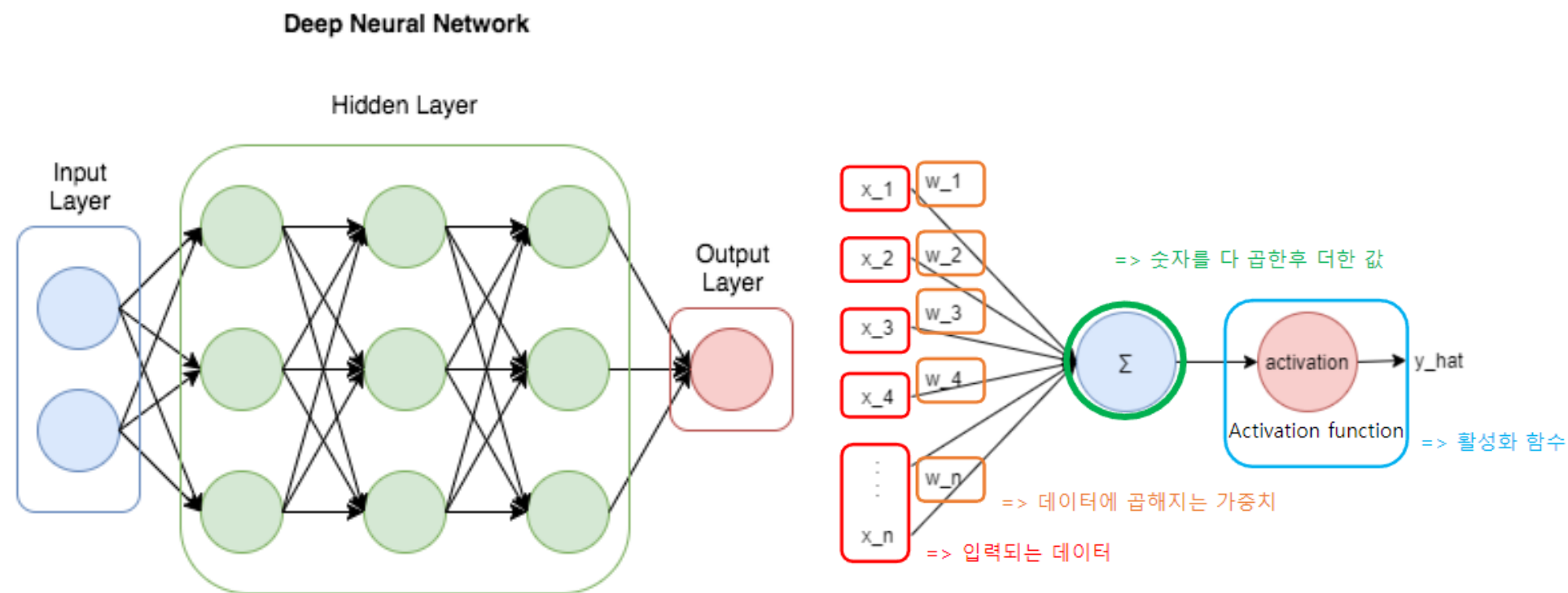


[파이썬 인공지능 교육 - 머신러닝 알고리즘]

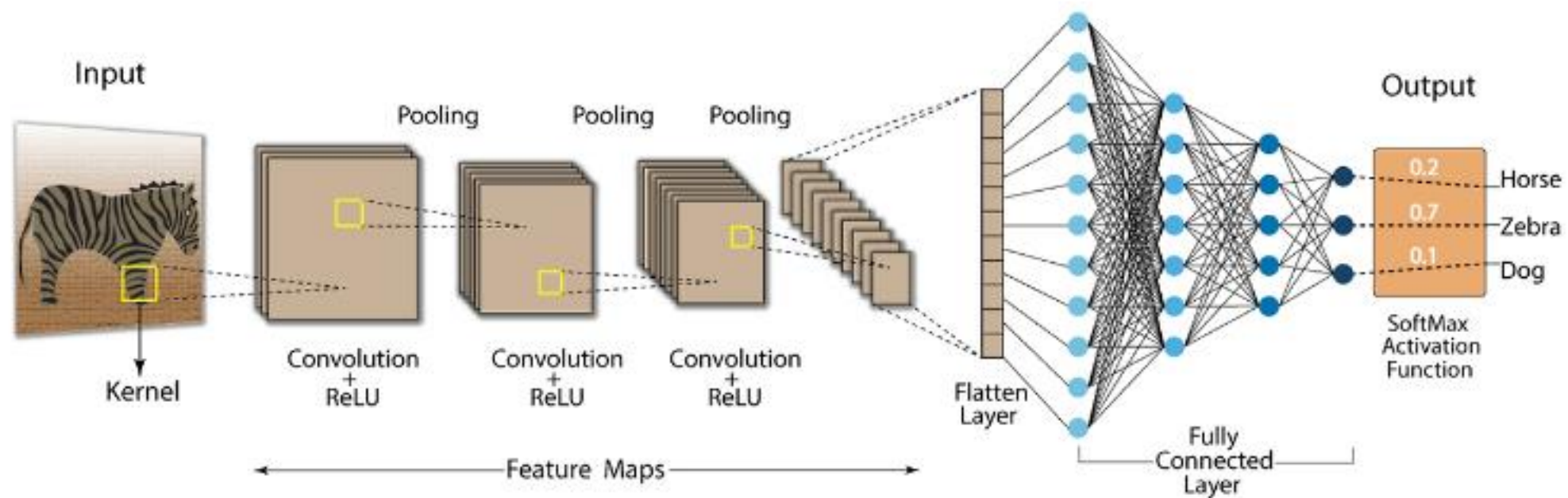
인공 신경망(ANN)



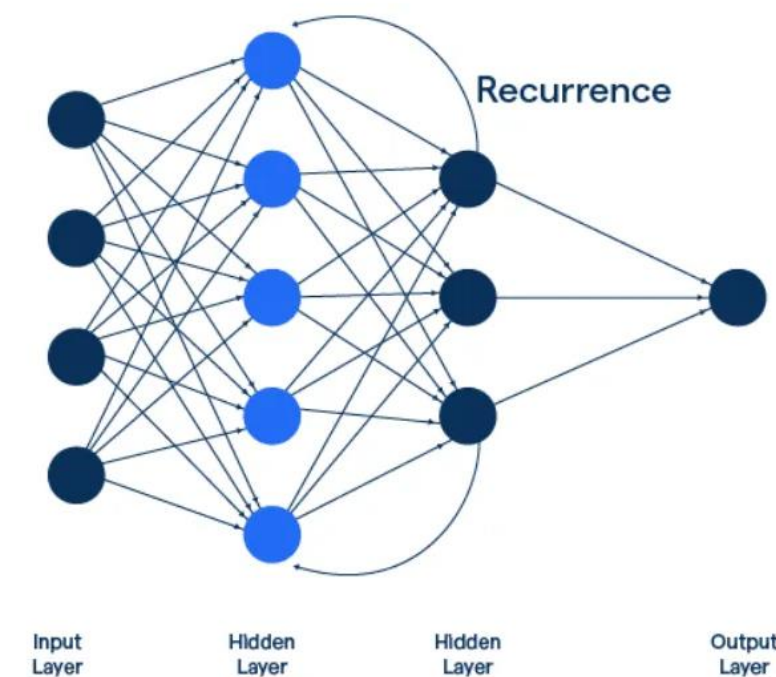
심층 신경망(DNN)



CNN (Convolutional Neural Network)

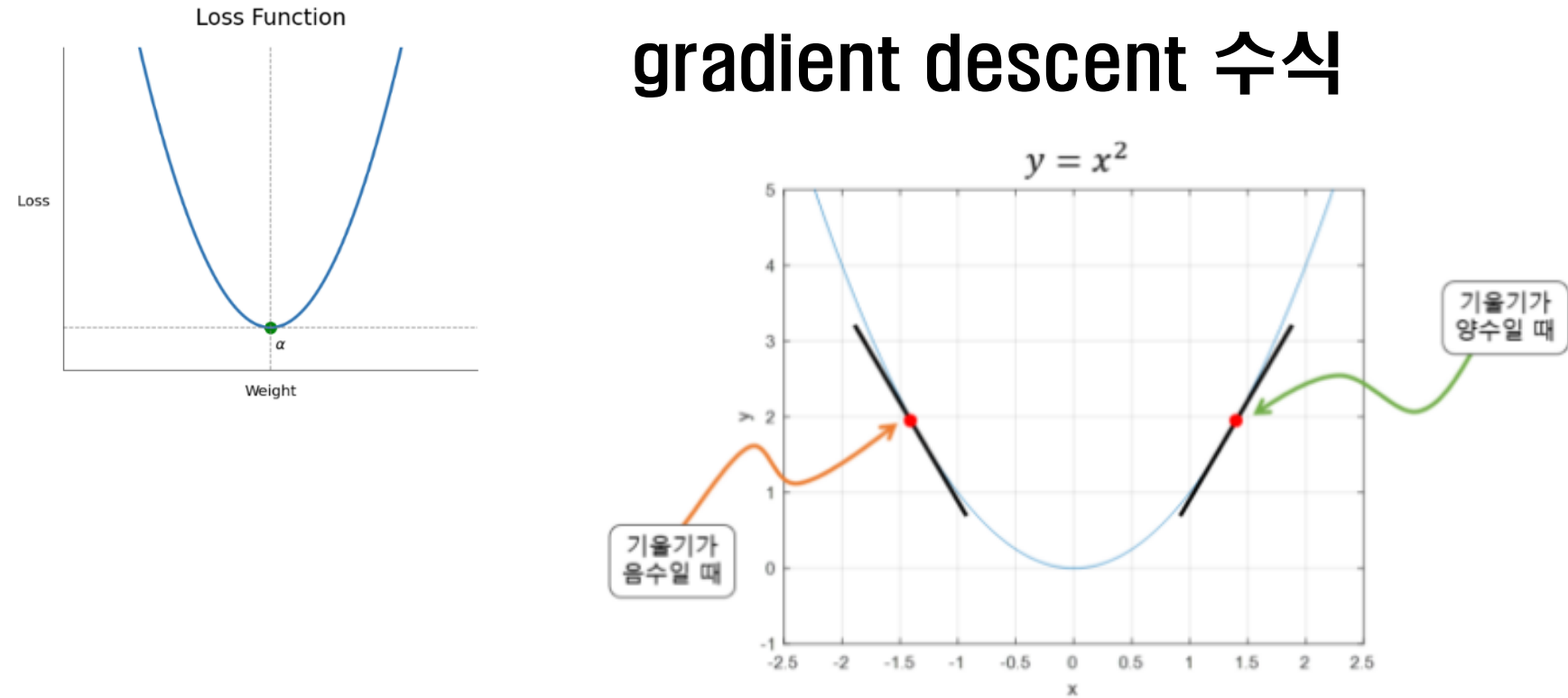


순환 신경망(RNN)

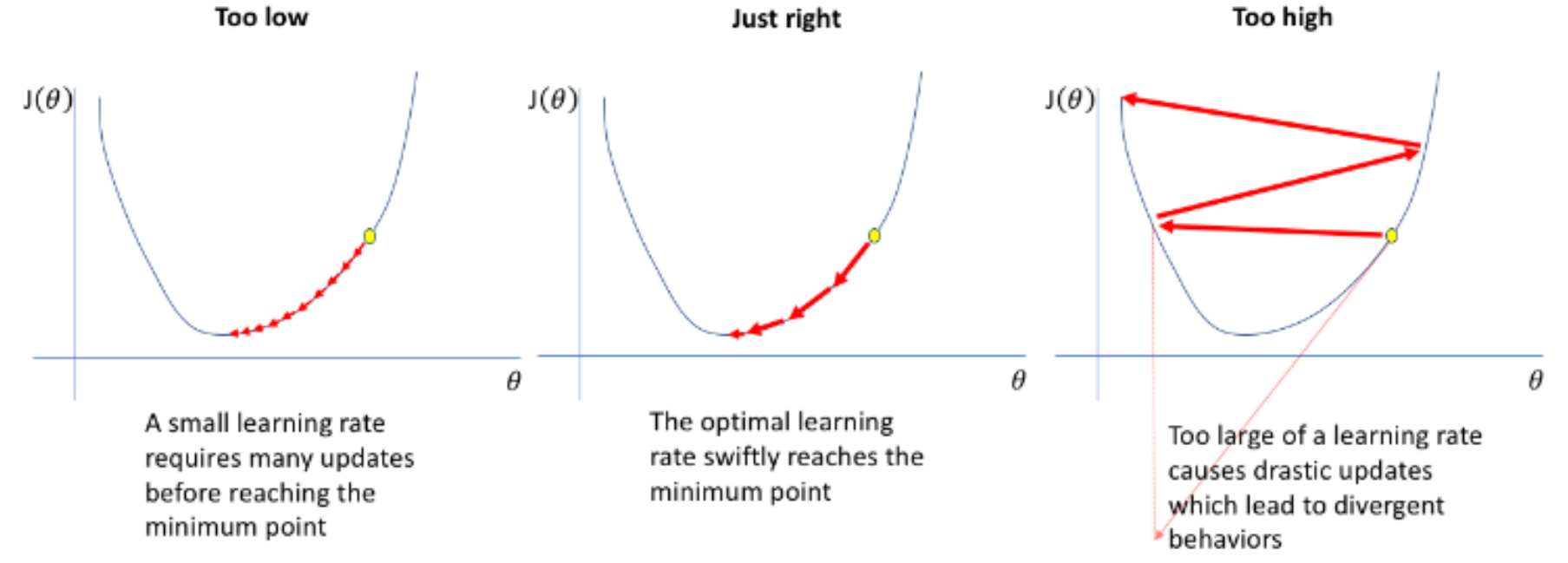


[파이썬 인식 인공지능 교육 - Loss function(손실 함수)]

gradient descent 수식

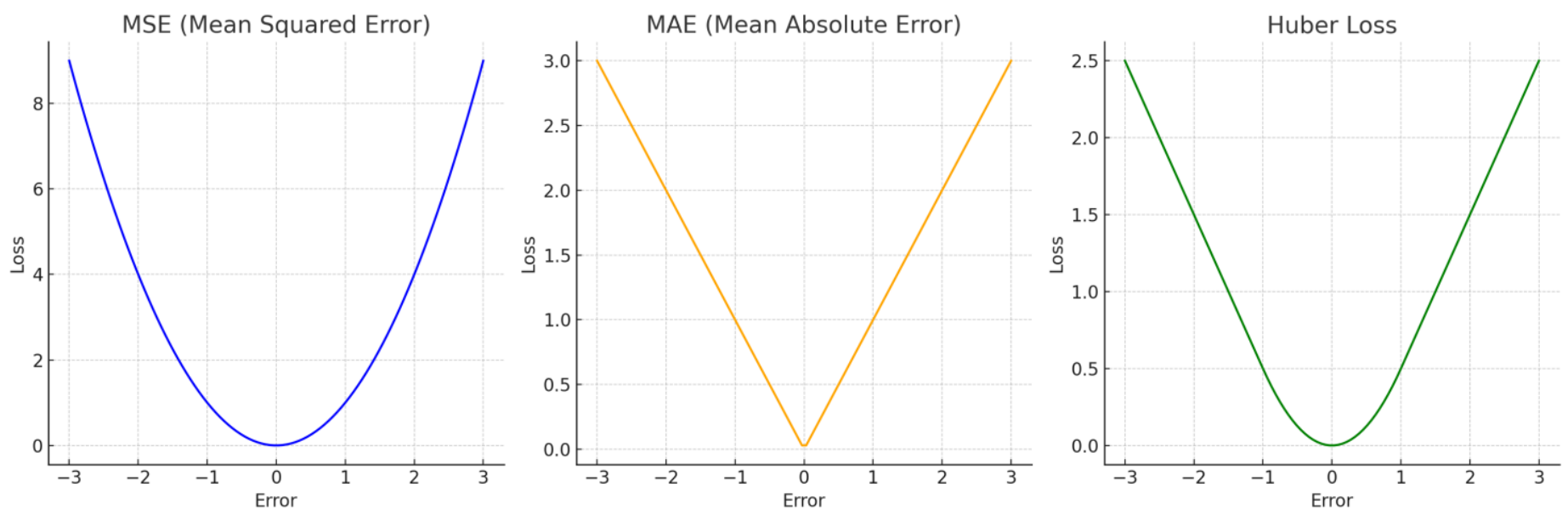


learning rate(step size)

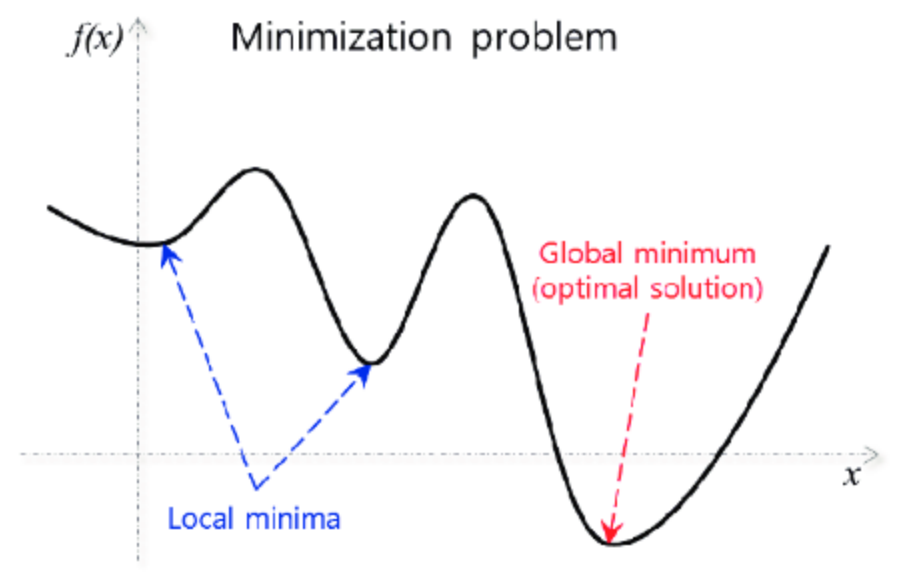


출처: jeremyjordan.me

Loss function(손실 함수) 종류



gradient descent의 한계점



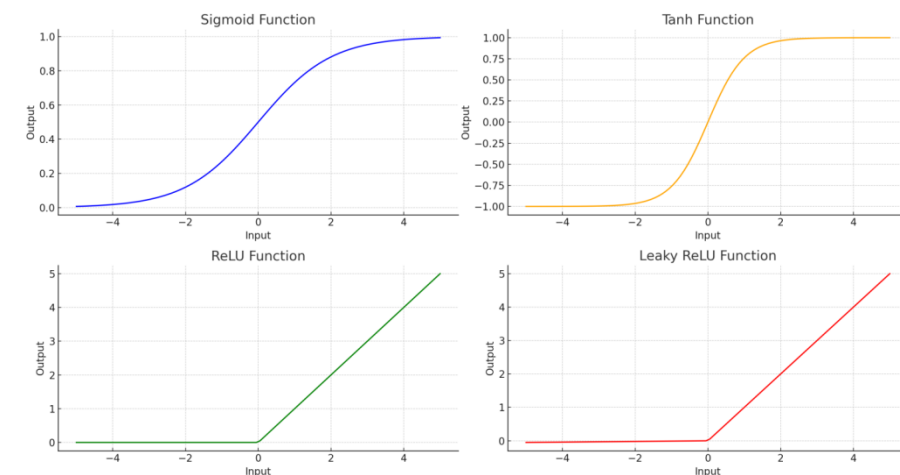
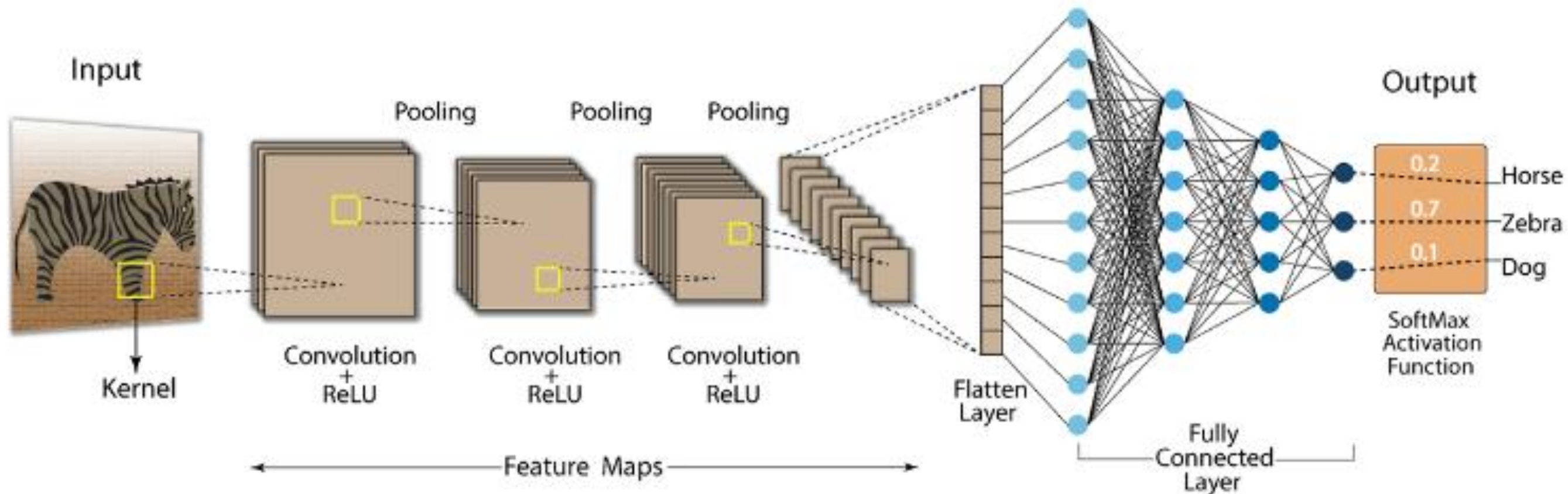
출처: ResearchGate.net

Optimization (최적화) : 신경망이 학습할 때 최적의 파라미터를 찾아가는 방식

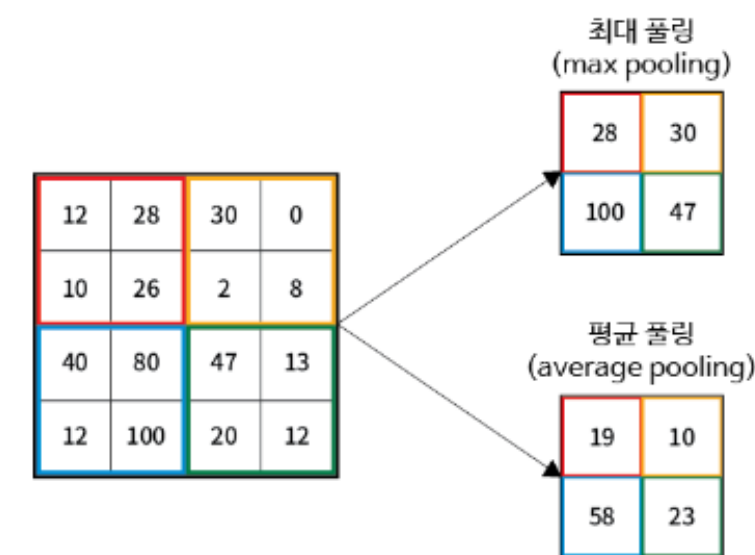
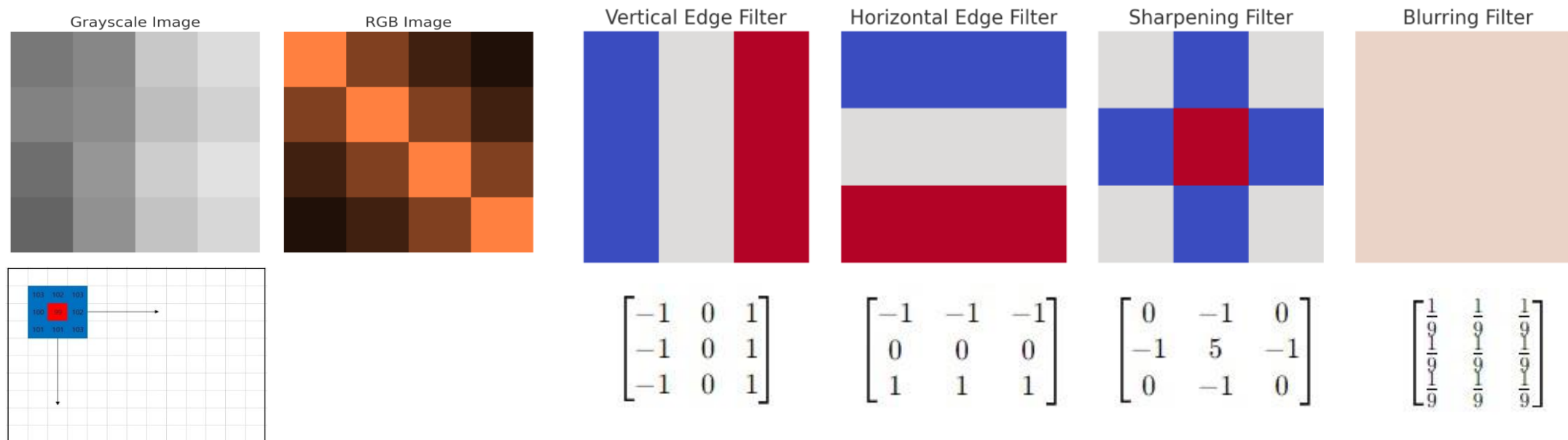
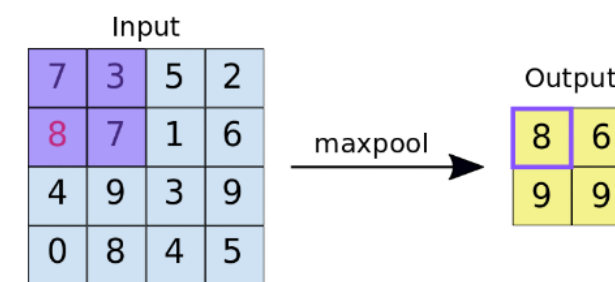
[파이썬 인공지능 교육 - CNN 알고리즘]

CNN (Convolutional Neural Network)

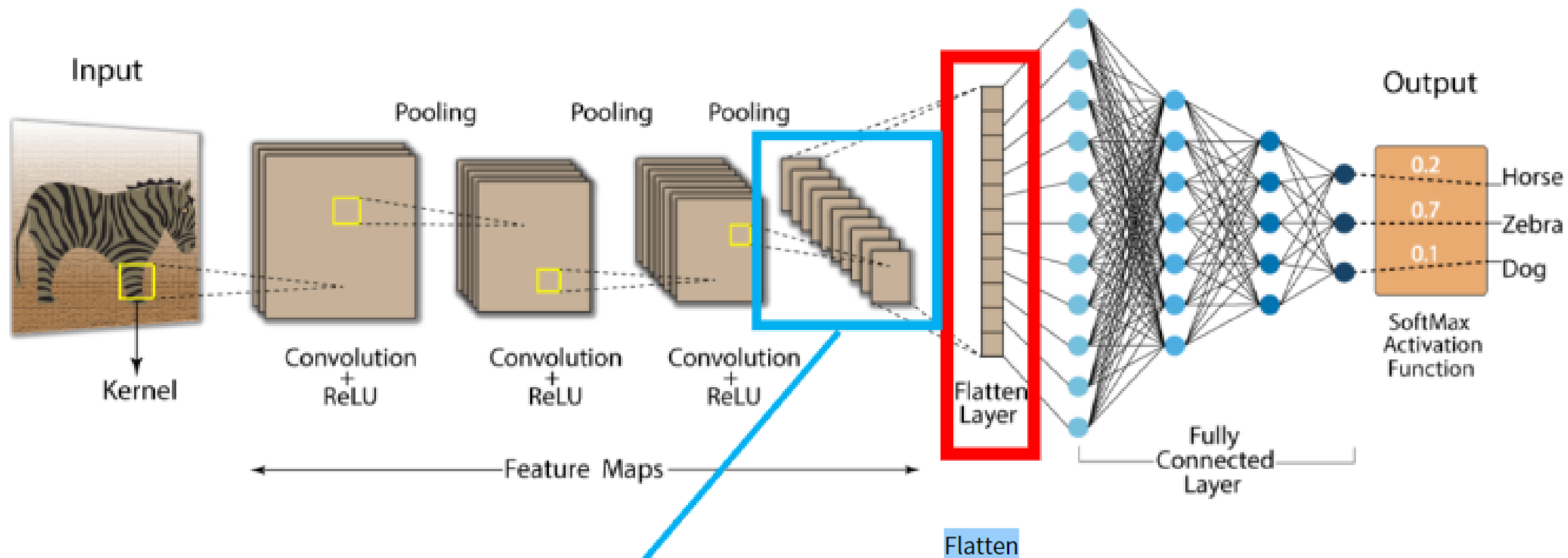
CNN은 인간의 시신경을 모방
컨볼루션과 풀링 연산을 반복적



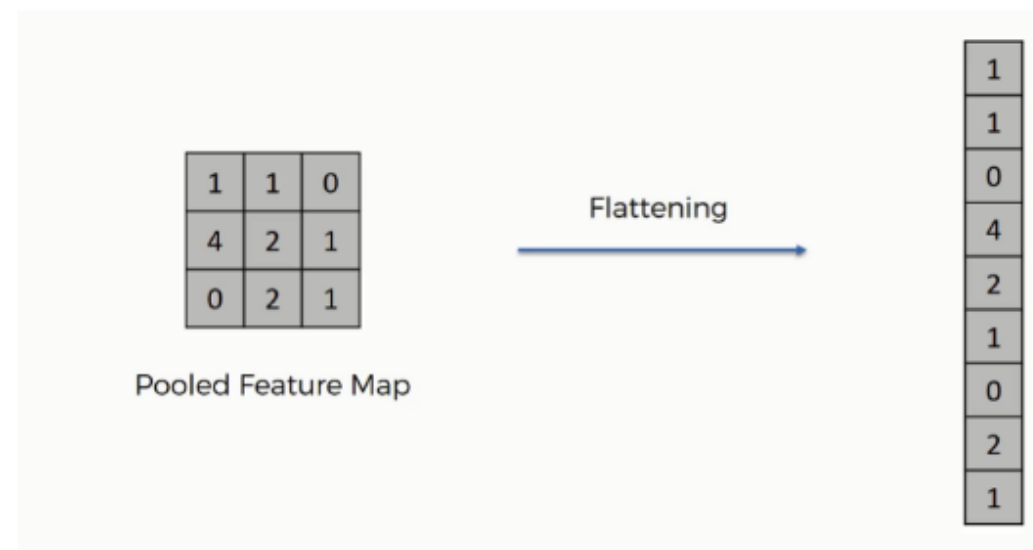
Pooling



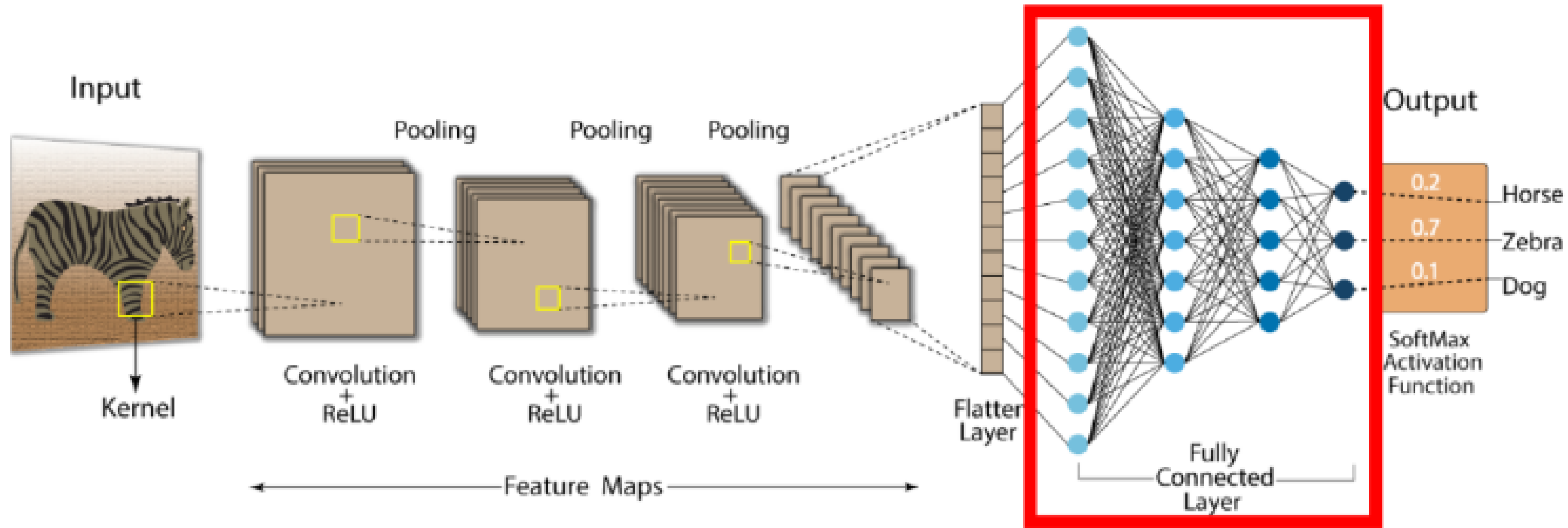
Convolution + pooling의 반복



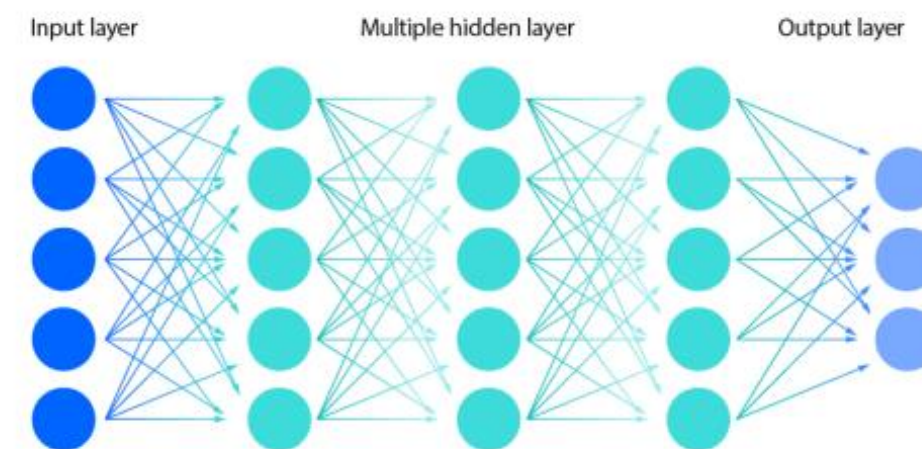
최종 이미지 주요 특이점 들
(feature maps)



Fully Connected Layers

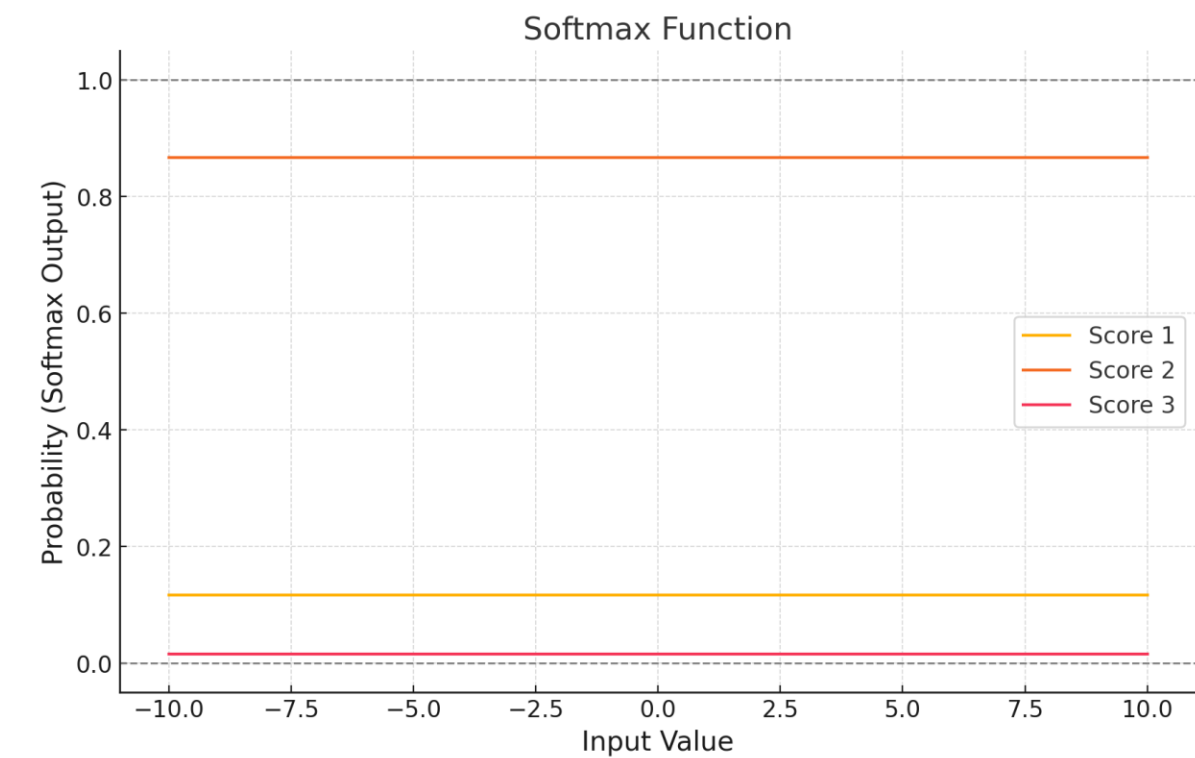
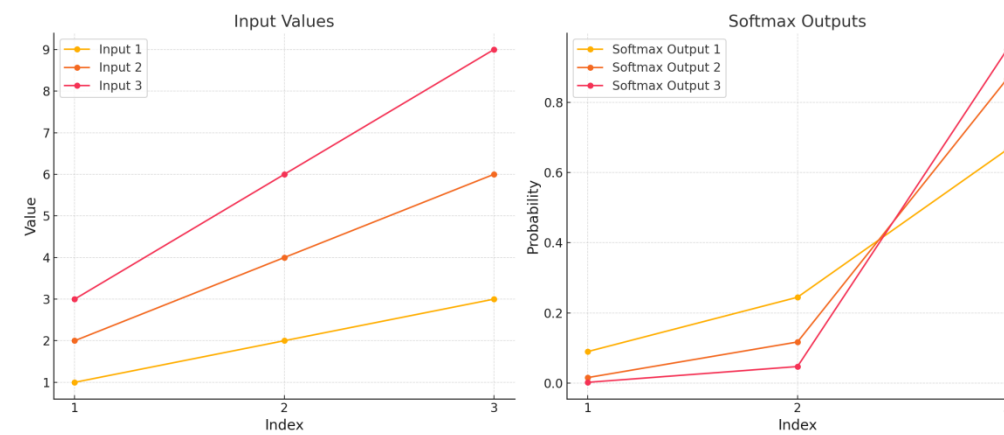
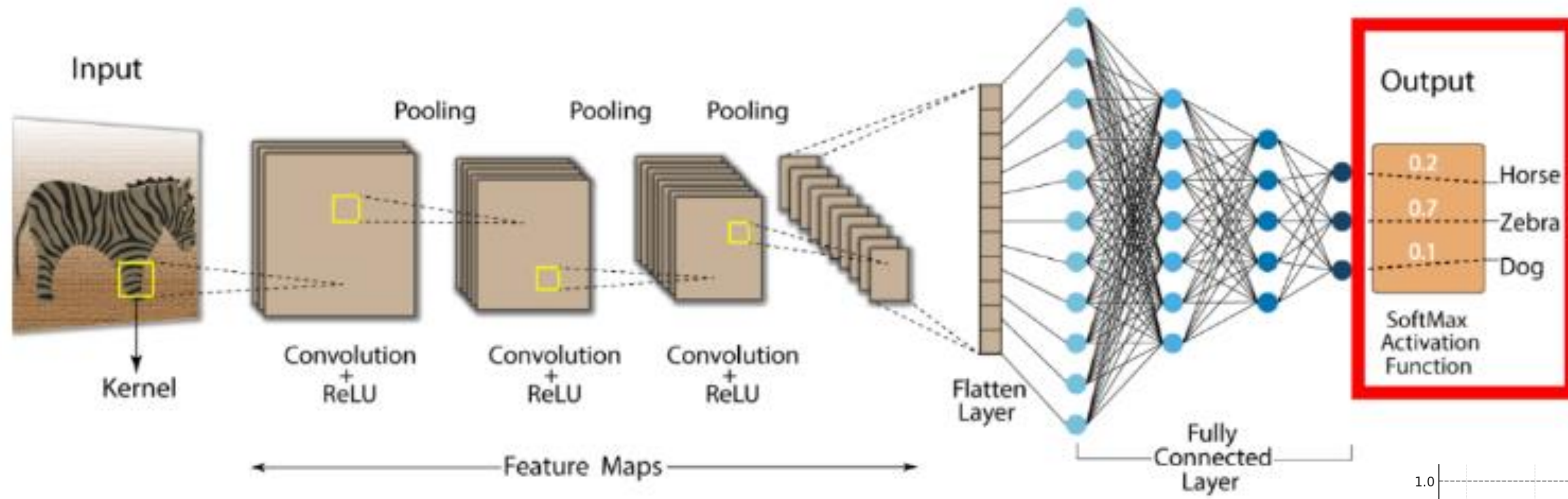


Deep neural network



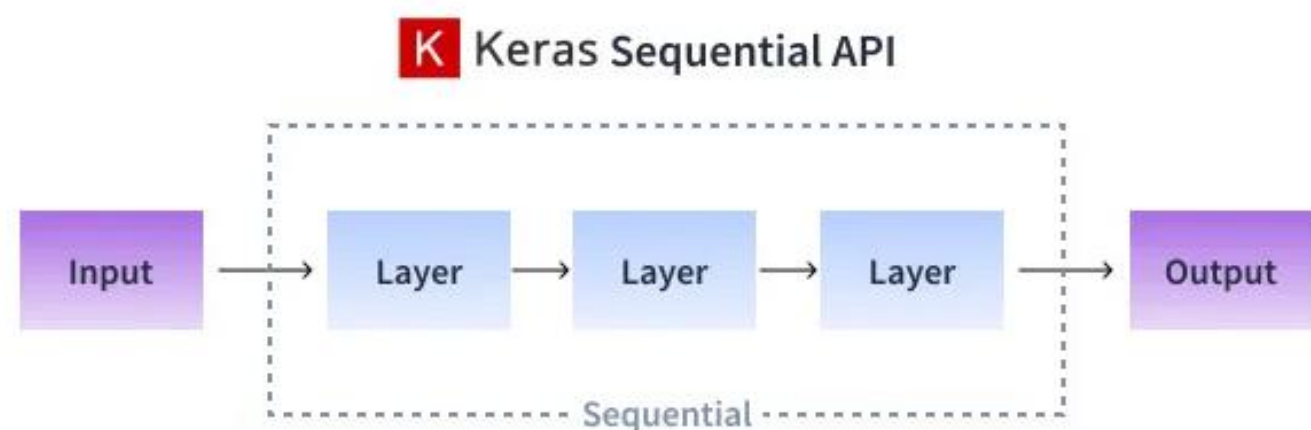
[파이선 인공지능 교육 - CNN 알고리즘]

SoftMax



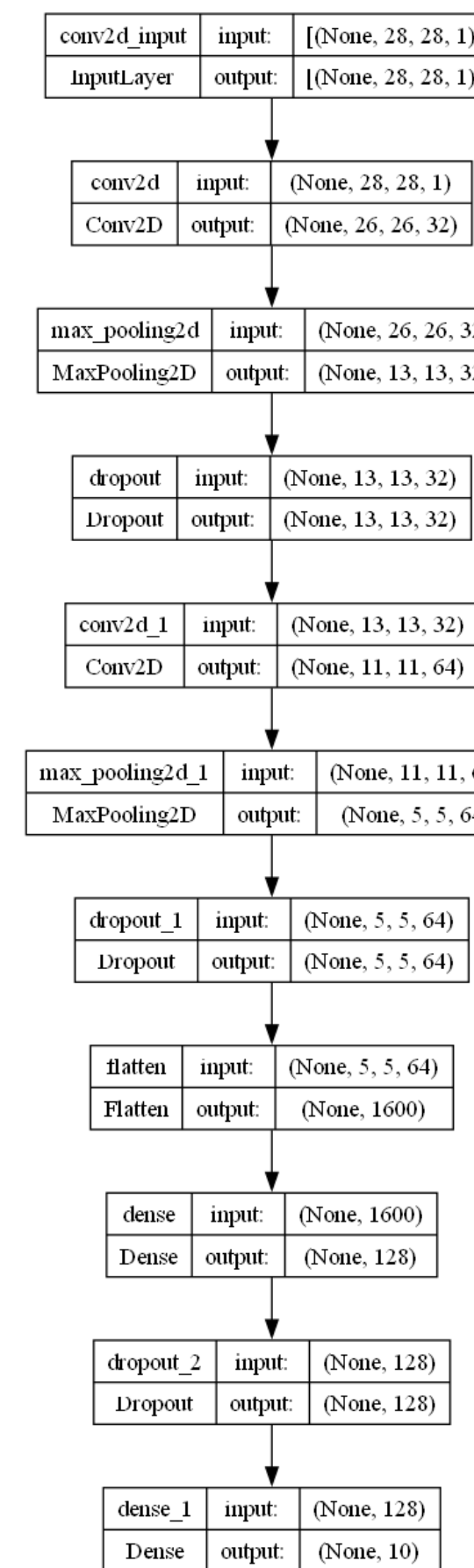
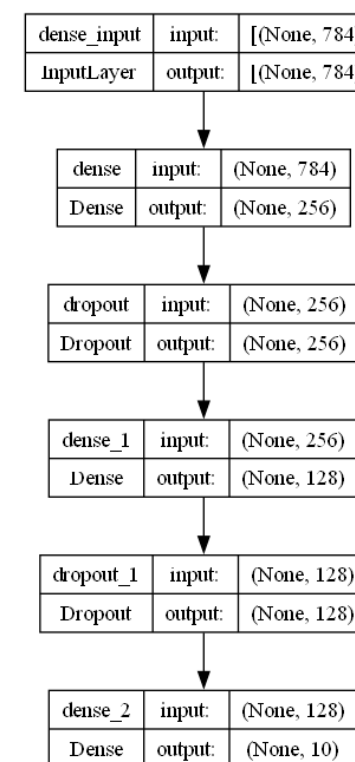
[파이썬 인공지능 교육 - CNN 모델 요약 해설]

Sequential API

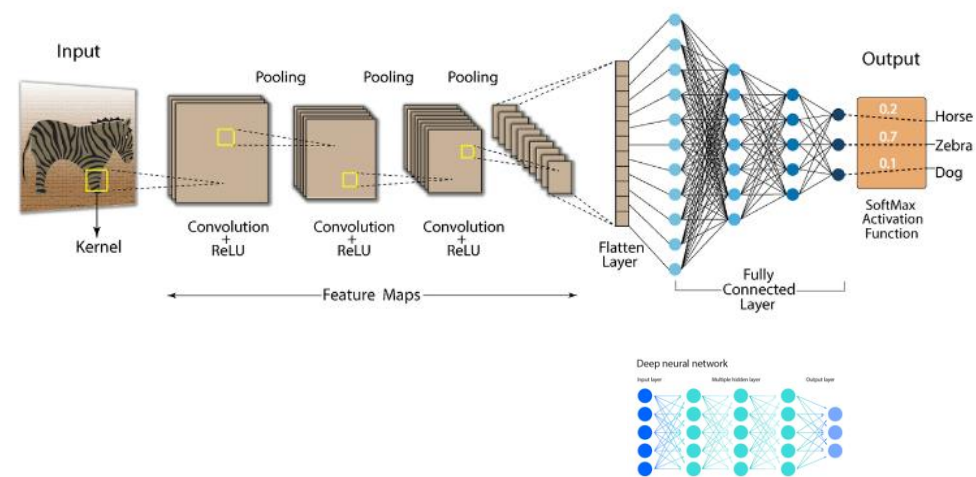


Model: "sequential"

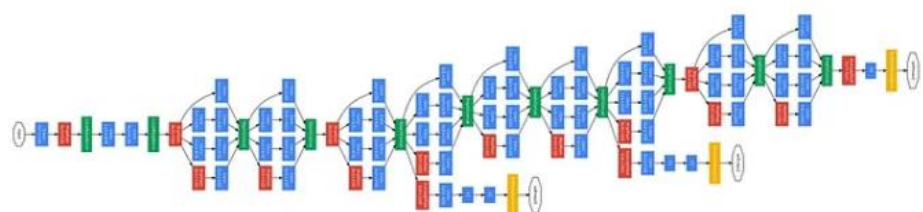
모델은 각 층을
순차적으로 쌓아 올리는
방식으로 모델을 설계



CNN (Convolutional Neural Network)



Functional API



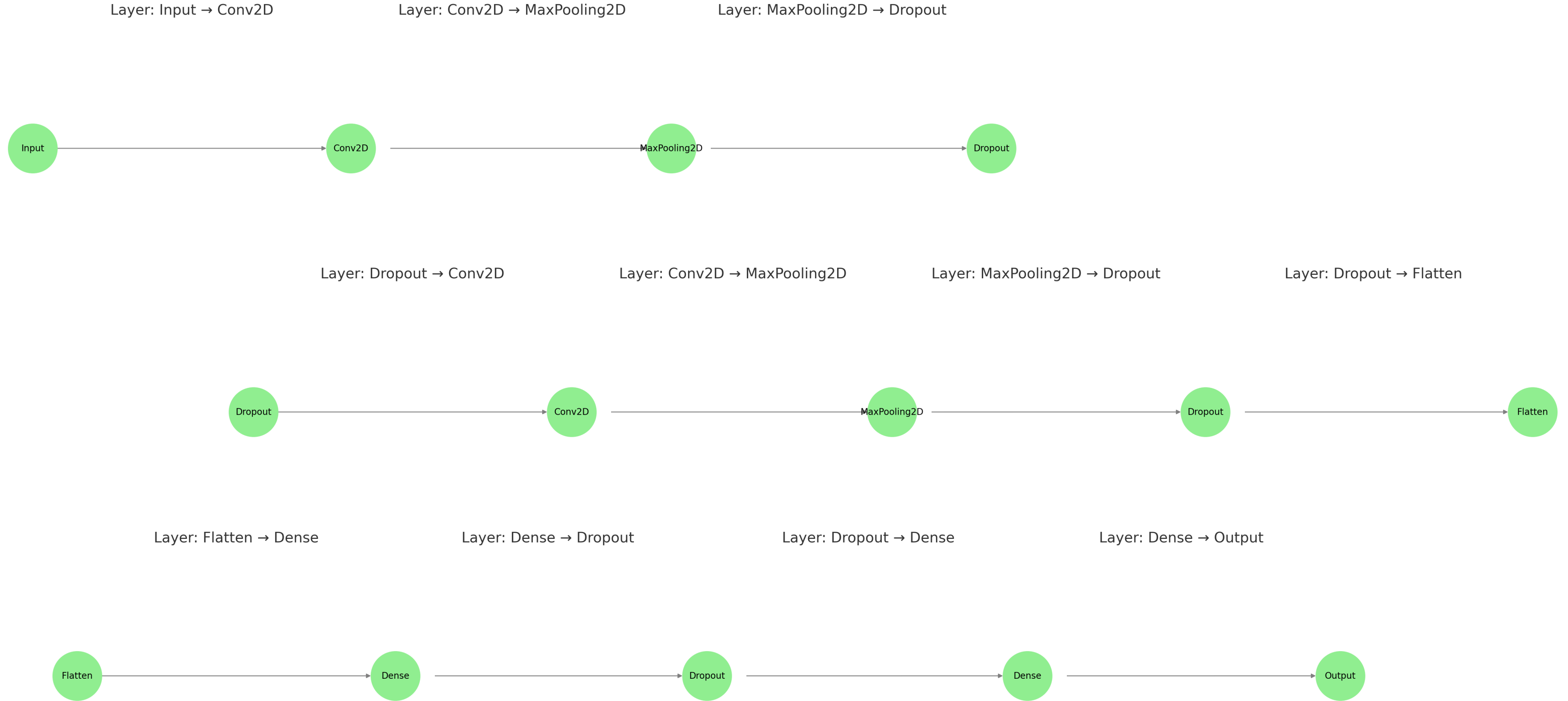
CNN 모델 설계

```
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)), # 합성곱 층
    MaxPooling2D(pool_size=(2, 2)), # 풀링 층
    Dropout(0.25), # 과적합 방지

    Conv2D(64, (3, 3), activation='relu'), # 추가 합성곱 층
    MaxPooling2D(pool_size=(2, 2)), # 추가 풀링 층
    Dropout(0.25), # 추가 과적합 방지

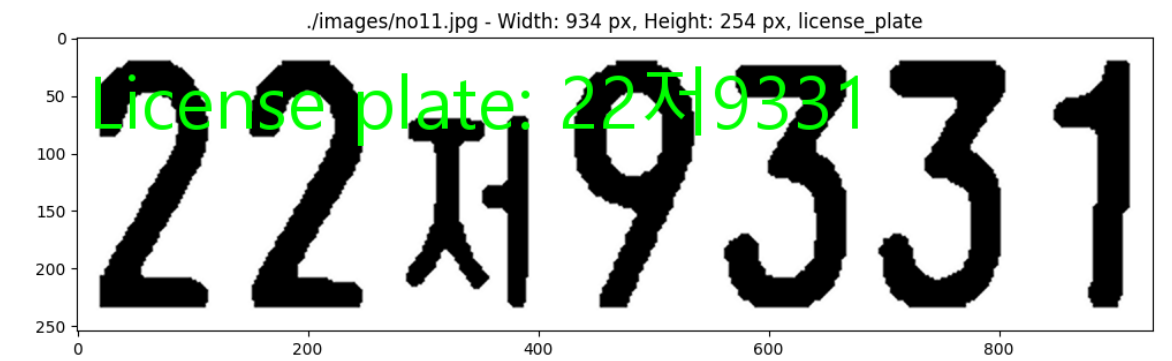
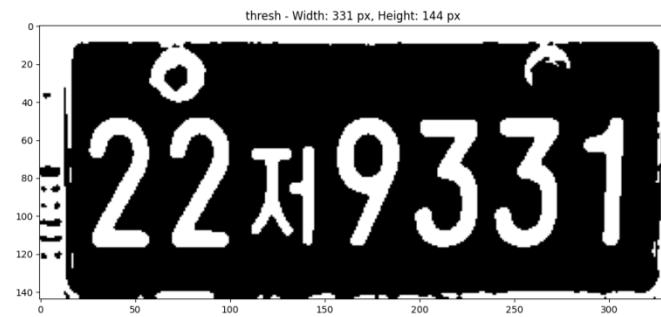
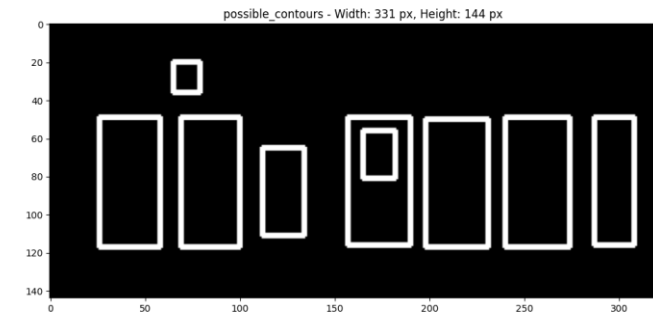
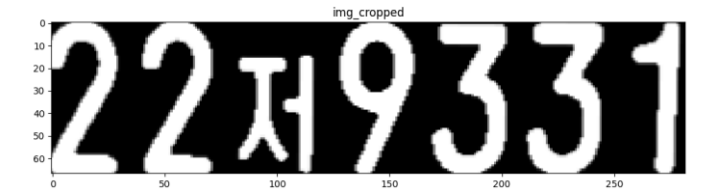
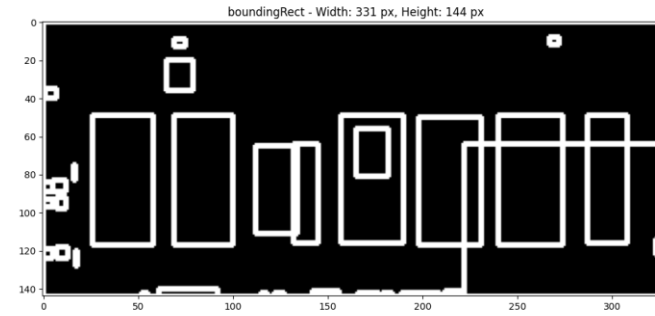
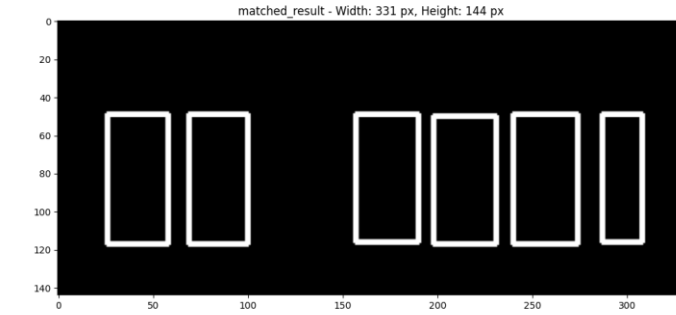
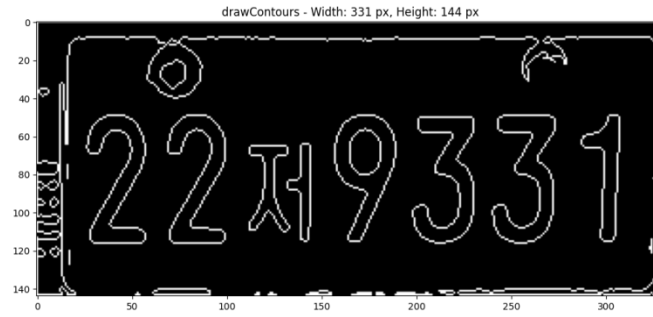
    Flatten(), # 1D로 펼침
    Dense(128, activation='relu'), # 완전 연결층
    Dropout(0.5), # 과적합 방지
    Dense(10, activation='softmax') # 출력층 (10개의 클래스)
])
```


[파이썬 인공지능 교육 – CNN 모델 요약 해설]



[파이썬 번호판인식]

./images/no11.jpg - Width: 1081 px, Height: 1440 px, license_plate



[제 1교시] - [Python 개발 환경 설정 / 객체 탐지]

- ① OpenCV[opencv-python]: 이미지 처리 및 YOLO를 사용하기 위해 필요
- ② matplotlib[matplotlib]: 데이터를 시각화하는 데 사용
- ③ YOLOv4 모델 파일 [yolov4.weights / yolov4.cfg / coco.names]
- ④ 이미지 파일 / 이미지 전처리 (Blob 생성) / 객체 탐지
- ⑤ 바운딩 박스 그리기



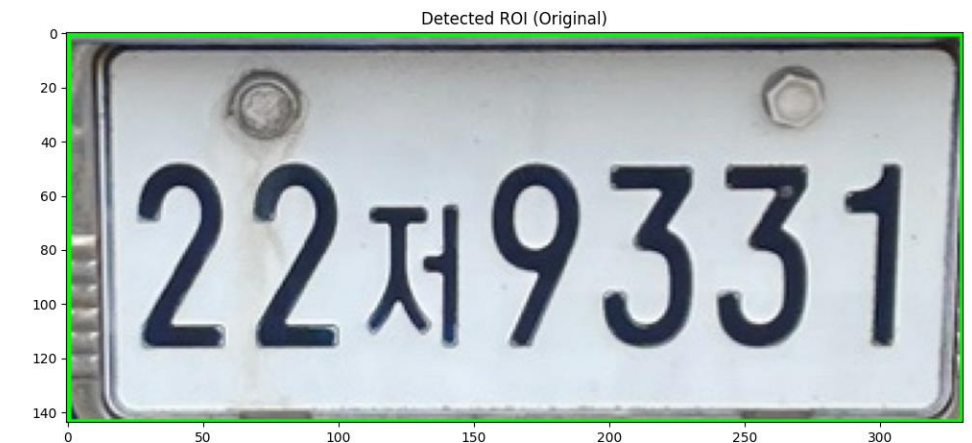
[제 2교시] – [OpenCV와 YOLO를 이용해 번호판 위치 찾기]

- ① YOLOv4 모델 파일 [custom.weights / yolov4_custom.cfg / license_plate.names]
- ② 클래스 이름 파일을 열어 객체 클래스 이름을 리스트로 저장
- ③ OpenCV의 dnn(Deep Neural Network) 모듈에서 제공하는 함수로, 학습된 모델을 불러오는 데 사용
- ④ 로드된 신경망 모델의 모든 레이어 이름을 리스트 형태
- ⑤ 모델의 출력 레이어 인덱스 반환



[제 3교시] – [ROI(Region of Interest) 관심영역 잘라냄]

- ① OpenCV 이미지를 RGB 형식으로 변환 (Matplotlib 호환)
- ② Blob은 Binary Large Object의 약자 – 다차원 배열 형태로 이미지 데이터를 변환
- ③ YOLO 모델에 Blob 데이터 설정
- ④ 모델 예측 실행
- ⑤ 비최대 억제(NMS, Non-Maximum Suppression) 적용하여 중복 박스 제거



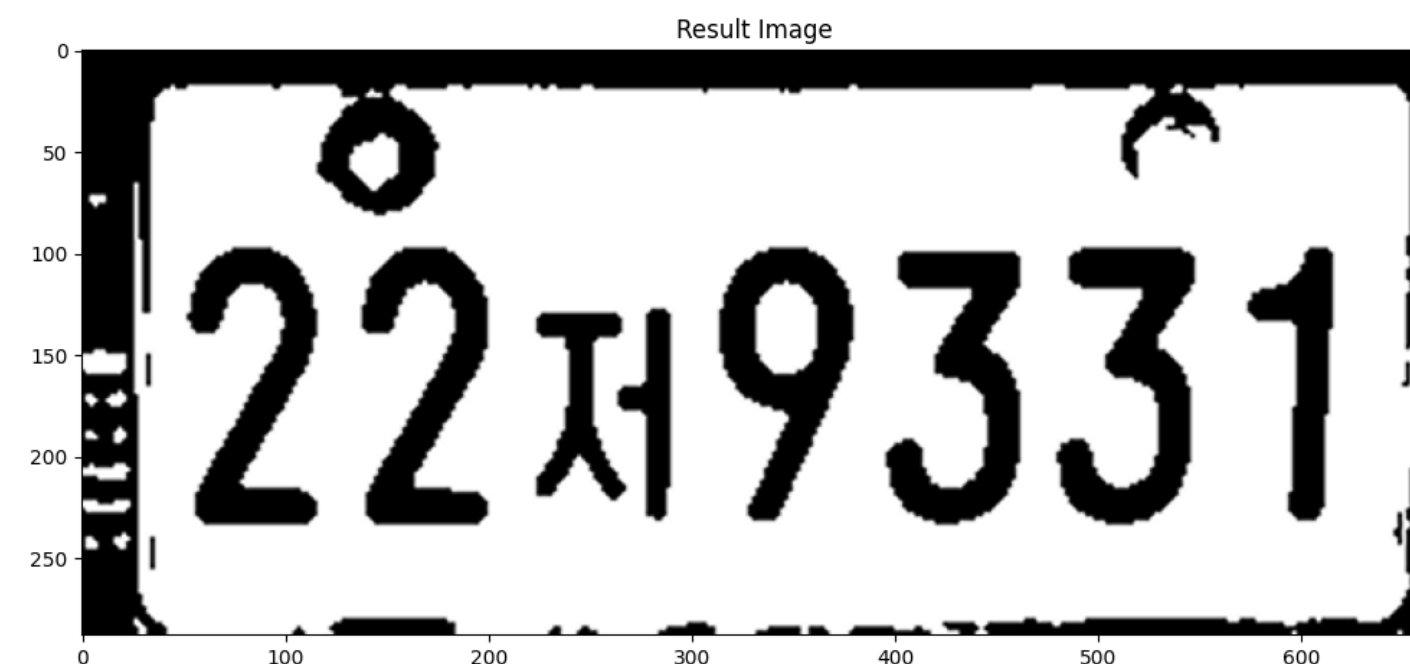
- ① Tesseract 다운로드
- ② pytesseract가 Tesseract 경로를 알 수 있도록 경로를 설정
- ③ 번호판 텍스트 추출 및 바운딩 박스 그리기
- ④ ROI(Region of Interest) 관심영역
- ⑤ OCR 수행 [Tesseract가 단일 라인의 텍스트로 인식]

- ① kor.traineddata 파일을 다운로드
- ② Tesseract-OCR\tessdata 폴더에 복사
- ③ lang = 'kor+eng'
- ④ RGB 이미지를 단일 채널 그레이스케일 이미지로 변환



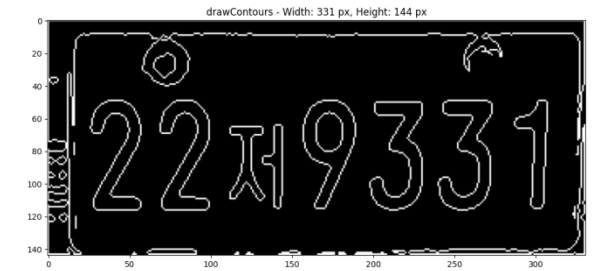
[제 6교시] – [GaussianBlur / threshold / inverted / resized / normalization]

- ① GaussianBlur 노이즈 제거
- ② Otsu의 이진화 적용하여 흑백 이미지 생성
- ③ 흰 글자가 검은 글자가 되도록 반전 – OCR 인식 성능 향상
- ④ 이미지 크기 조정 [스케일 업]
- ⑤ 한국어와 숫자만 남겨 저장할 문자열을 저장 [정규화]

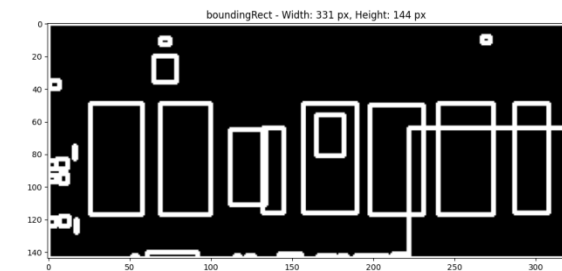


[제 7교시] – [Contours / boundingRect / possible_contours / img_cropped]

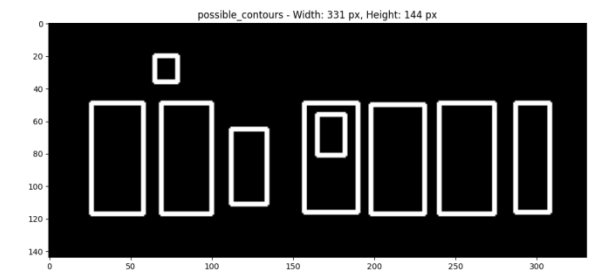
① Contours란 동일한 색 또는 동일한 강도를 가지고 있는 영역의 경계선을 연결한 선



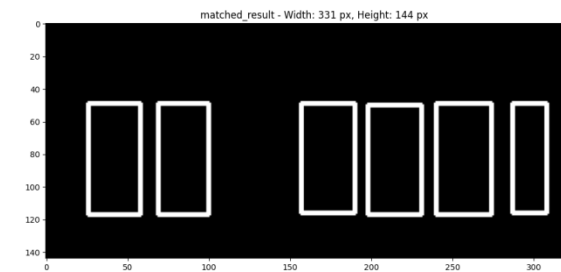
② 윤곽선은 특정 모양이나 경계에 해당하는 좌표로 구성된 배열



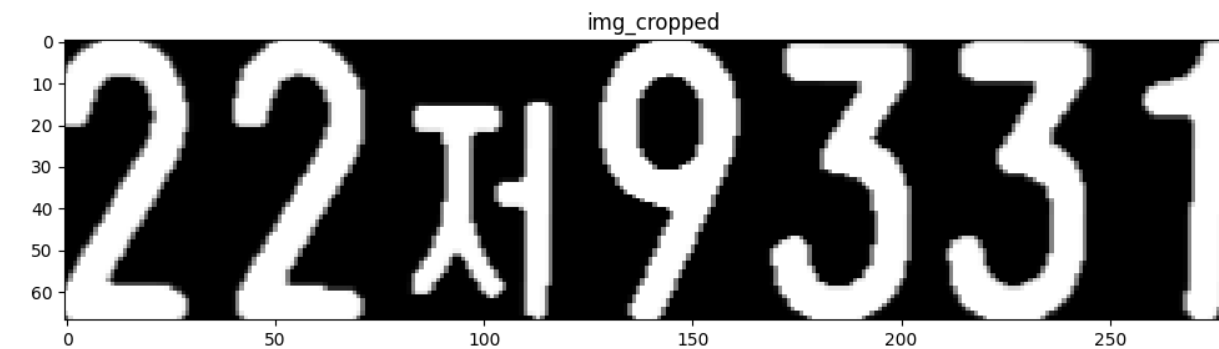
③ 윤곽선 정보에서 번호판 후보가 될 만한 윤곽선을 필터링



④ contours 중에 확실하게 번호판을 찾기 위해 기준을 강화

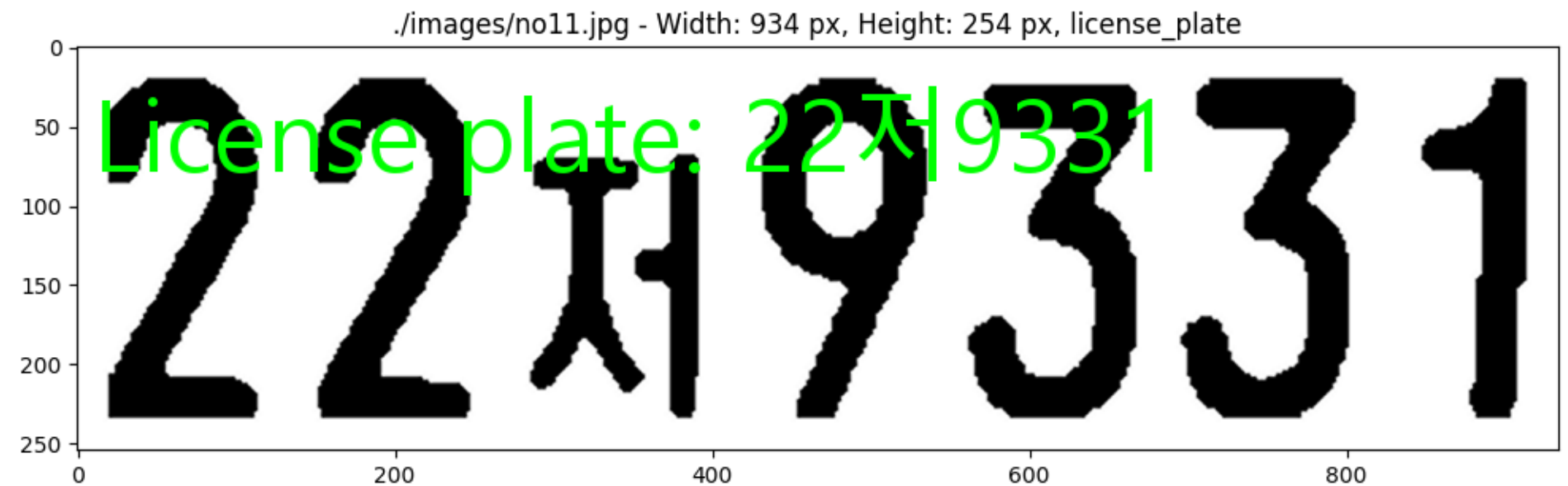


⑤ 기울어진 경우 Affine Transform 적용 Crop



[제 8교시] – [Another Thresholding]

- ① **resize** 함수로 이미지의 크기를 확대하여 OCR의 정확성을 높임
- ② **findContours**로 이진화된 이미지의 모든 윤곽선을 탐지
- ③ **GaussianBlur**로 노이즈 제거
- ④ 이미지에 여백 추가 (Padding)
- ⑤ 흰 글자가 검은 글자가 되도록 반전
- ⑥ 이미지 크기 조정 (스케일 업)



- ① 파이썬 함수 적용하기
- ② 파이썬 모듈 활용하여 소스 간단히 하기
- ③ 영상속 번호판 인식하기