

머신러닝 / 딥러닝

– Introduction –

4차 산업혁명 - ABCD 인재의 중요성



- 2016년 스위스 다보스에서 개최된 세계경제 포럼에서 처음 언급됨
- 학자에 따라 정의는 조금씩 다르지만, 대체로 4차 산업혁명은 모든 것이 연결(Connectivity)되어 있는 환경에서 인공지능(Artificial Intelligence)에 의해 더욱 편리하고 지능적인 사회로의 혁신적 변화를 지칭함

⇒ ABCD 인재의 중요성이 부각됨

인공지능 머신러닝 딥러닝



- 인공지능 (Artificial Intelligence)

인간의 학습능력, 추론능력 등을 컴퓨터를 통해 구현하는 포괄적인 개념



- 머신러닝 (Machine Learning)

데이터를 이용하여 명시적으로 정의되지 않은 패턴을 학습하여 미래 결과(값, 분포)를 예측

※ 데이터마이닝(Data Mining): 데이터간의 상관관계나 속성을 찾는 것이 주목적

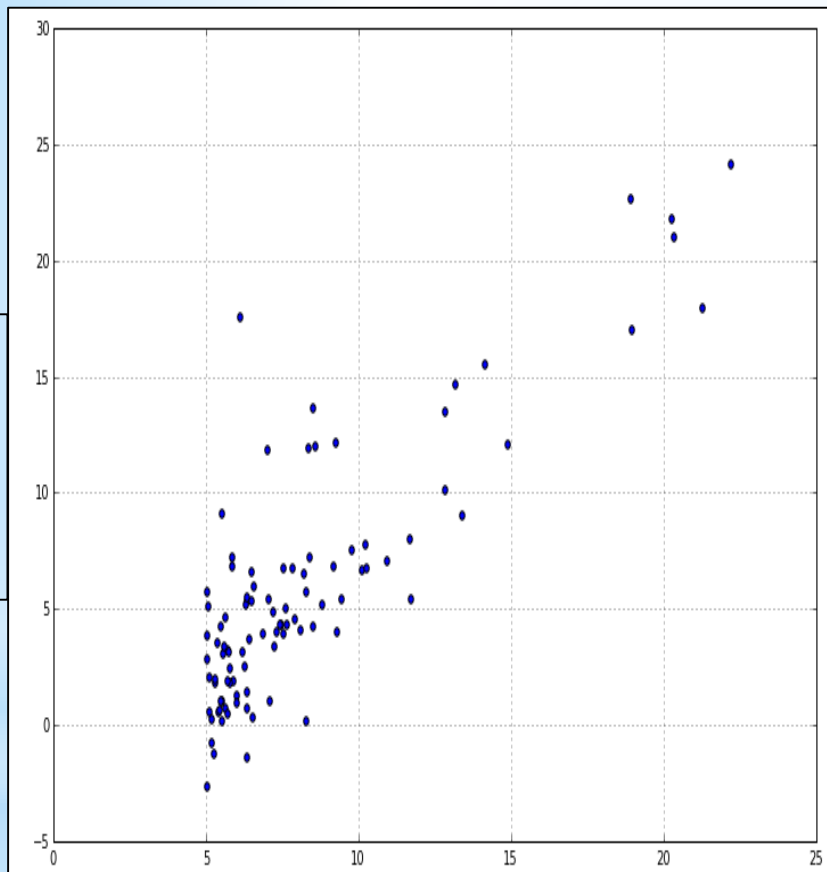


- 딥러닝 (Deep Learning)

머신러닝의 한 분야로서 **신경망(Neural Network)**을 통하여 학습하는 알고리즘의 집합

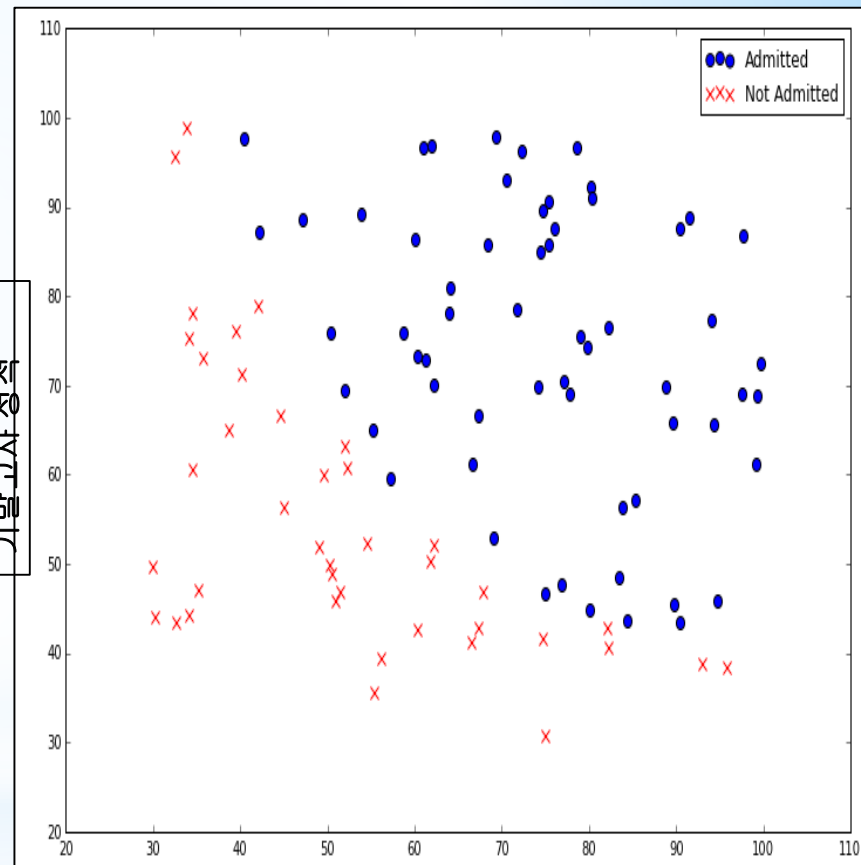
머신러닝 – Regression, Classification

회귀
(Regression)



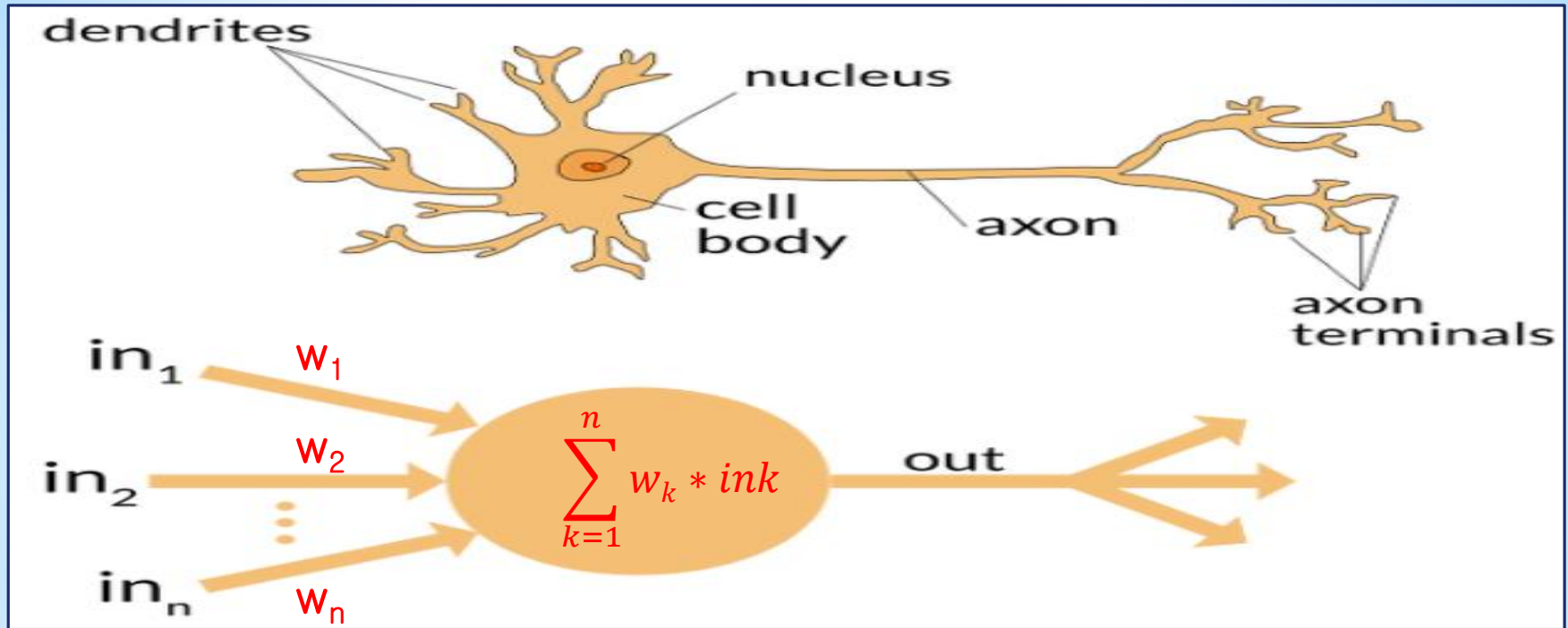
아파트 평수

분류
(Classification)



중간고사 성적

딥러닝 - Neural Network



- 신경망(Neural Network)은 인간의 뇌 세포(Neuron) 상호작용을 모방하여 회귀(Regression) 또는 분류(Classification) 을 수행함
 - 이전 뉴런 출력을 각각의 입력으로 받아, + 또는 - 가중치를 통해 전체의 합을 합하여 특정 임계치(threshold)를 넘으면 다음 뉴런으로 전달 하는 원리

Audience – Prerequisite

➤ 프로그래밍 개념 (No Python Experience)

- 변수, 데이터타입, 제어문, 함수, 클래스, 라이브러리 등의 개념

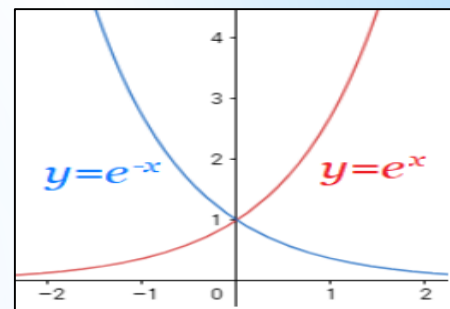
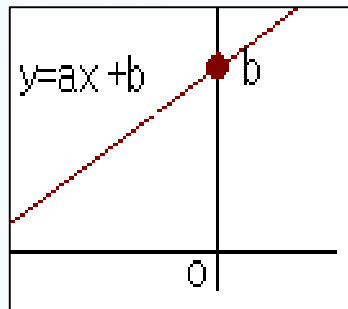
➤ 기본 수학 개념

$$y = ax + b$$

$$y = e^x$$

$$\sum_{k=1}^2 A_k = A_1 + A_2$$

$$\prod_{k=1}^2 A_k = A_1 \times A_2$$



➤ 행렬 (matrix) 연산

- 산술연산 (+, -, *, /)

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 2 \\ 3 & 5 \end{pmatrix}$$

$$(2 \times 2) + (2 \times 2) = (2 \times 2)$$

- 행렬 곱 (dot product)

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 0 \\ 3 & 4 & 0 \end{pmatrix}$$

$$(2 \times \boxed{2}) \cdot (\boxed{2} \times 3) = (2 \times 3)$$

Schedule

➤ Python 기초

- 데이터타입, 제어문, 함수, 클래스, 라이브러리(numpy, matplotlib,...)

➤ 수치미분 (Numerical Derivative)

- 미분, 편미분, 체인 룰 및 수치미분 구현

➤ 회귀 (Regression) / 분류 (Classification)

- 동작원리, 손실함수 및 python 예제 (수치미분 활용)

➤ 신경망 (Neural Network)

- Feed Forward / Back Propagation, MNIST 예제 (공식 활용)

➤ TensorFlow 기초

- 상수 / 변수 / 플레이스홀더, 회귀 • 분류 • MNIST 예제 (프레임워크 활용)
- Google Colaboratory (Colab)

➤ CNN (Convolutional Neural Network)

- 아키텍처 / 동작원리 / 99%이상 정확도 MNIST 예제 (프레임워크 활용)

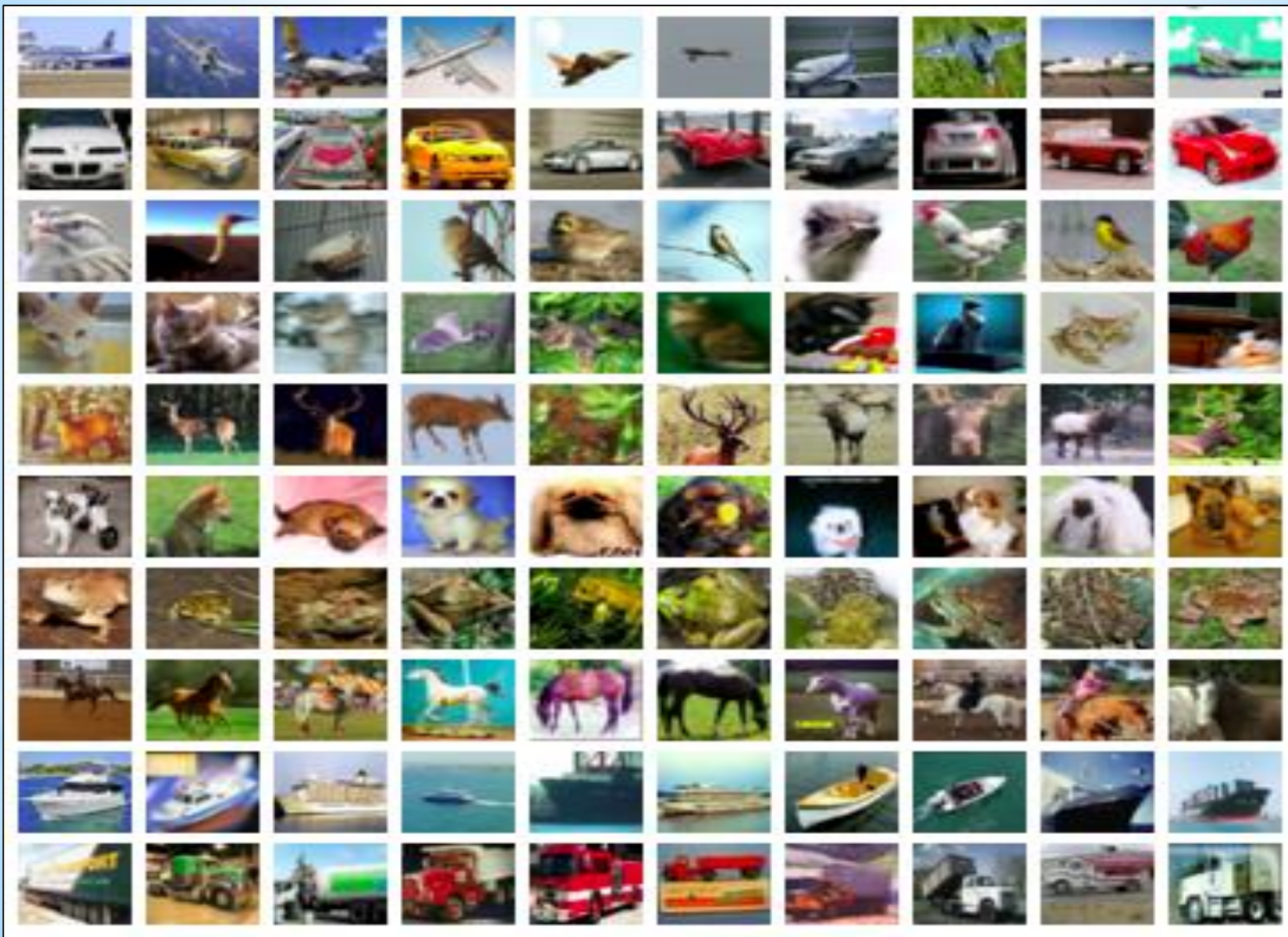
➤ Transfer Learning

- 아키텍처 / 동작원리 / 예제 (프레임워크 활용)

필기체손글씨 MNIST 인식

1 0 5 0 w w v 2 6 2 2 1
4 4 - 5 + 5 - 5 2 2 4
7 4 6 - 3 - 5 0 6
- 2 0 5 4 0 0 0 6
2 4 5 7 3 0 0 0 8
6 2 2 5 0 5 0 3
7 0 0 5 2 3 4 0 6
0 2 8 3 4 0 4 5 8
2 0 0 5 5 4 5 0
0 4 2 5 0 3 5 4

CIFAR 10 인식



나만의 이미지를 인식하는 시스템 구축



test1.jpg

```
(C:\Program Files\Anaconda3) C:\TransferLearning>python label_image.py --graph=/tmp/output_graph.pb --labels=p/output_labels.txt --input_layer=Placeholder --output_layer=final_result --image=./test1.jpg
```

```
dog 0.99960524  ←---- 입력데이터 test1.jpg 를 99.96% 확률로 dog 으로 예측함  
cat 0.00016304066  
squirrel 0.00015678526  
koala 7.49459e-05
```



test2.jpg

```
(C:\Program Files\Anaconda3) C:\TransferLearning>python label_image.py --graph=/tmp/output_graph.pb --labels=p/output_labels.txt --input_layer=Placeholder --output_layer=final_result --image=./test2.jpg
```

```
cat 0.9966613  ←---- 입력데이터 test2.jpg 를 99.66% 확률로 cat 으로 예측함  
squirrel 0.0020424824  
dog 0.00091571576  
koala 0.0003804999
```



test3.jpg

```
(C:\Program Files\Anaconda3) C:\TransferLearning>python label_image.py --graph=/tmp/output_graph.pb --labels=p/output_labels.txt --input_layer=Placeholder --output_layer=final_result --image=./test3.jpg
```

```
squirrel 0.999469  ←---- 입력데이터 test3.jpg 를 99.94% 확률로 squirrel 로 예측함  
dog 0.00034103458  
koala 0.00012662407  
cat 6.3346306e-05
```

예제 참고 사이트

- 전체 과정 및 심화 학습
YouTube에서 neowizard 검색 or <https://www.youtube.com/neowizard>
- 실습 예제 및 데이터 : <https://github.com/neowizard2018/neowizard>
- 강의를 바탕으로 한 서적 출간 예정 (2020년 2월 21일)
머신러닝을 위한 파이썬 한 조각, 비제이퍼블릭