

프로그래밍_인공지능1

2022학년도 2학기

만약,
예지 능력이 생긴다면
무엇을 예측하고
싶은가요? ˊ ˋ ˋ

예측 모델을 구현하고자 할 때 가장 고려해야할 점?

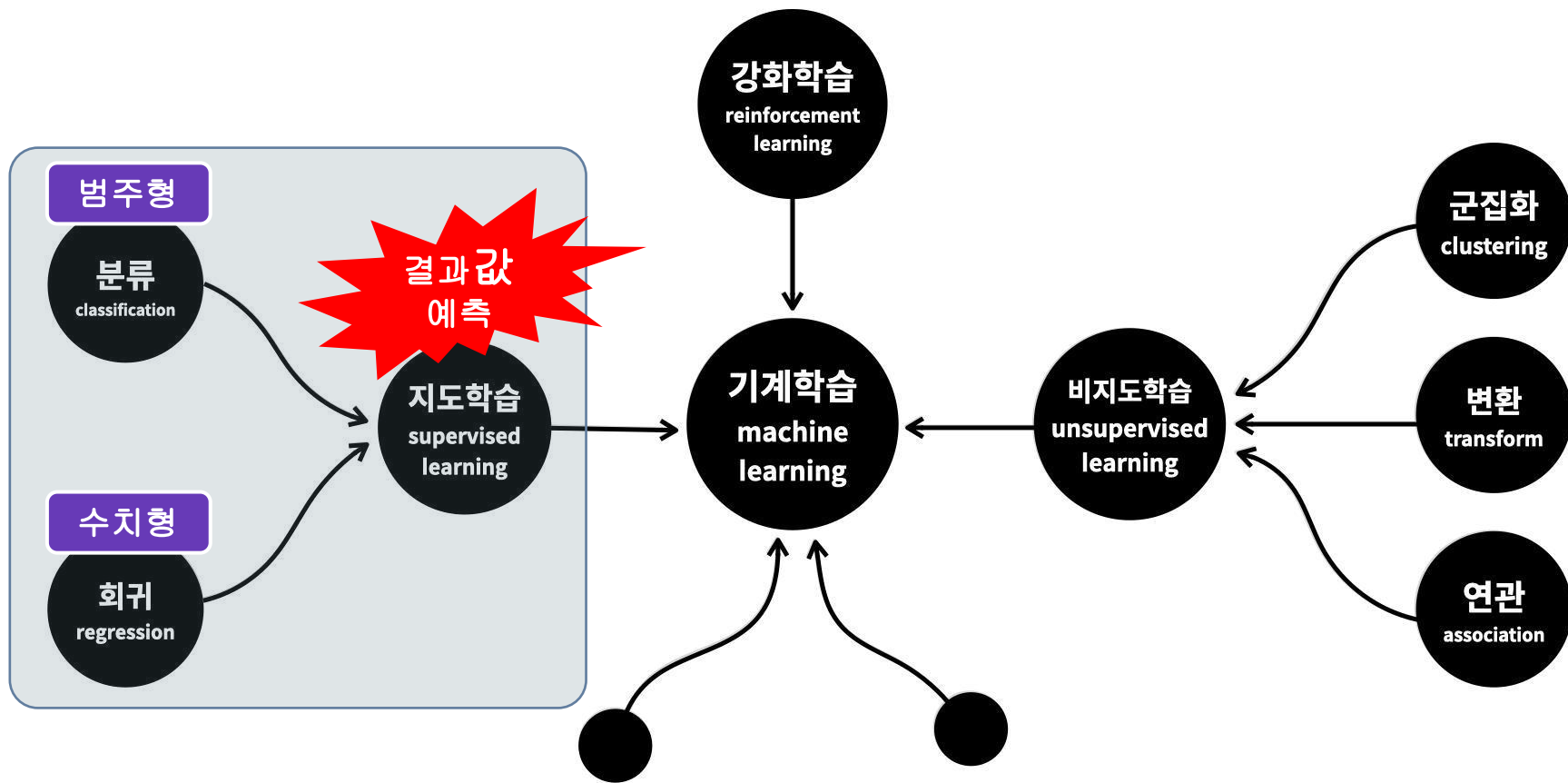
예측하고자 하는 값

수치형

인구수, 평균기온,
매출액, 신장,
체중, 혈압...

범주형

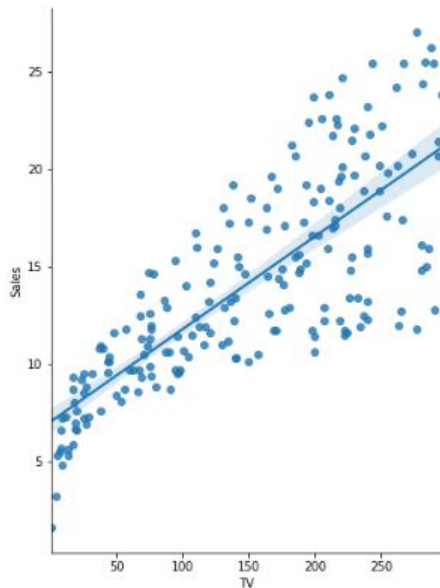
성별(남/여), 성공여부
(성공/실패), 효과
(없음/조금있음/매우있음),
혈액형(A/B/O/AB)...



회귀(Regression) 사례

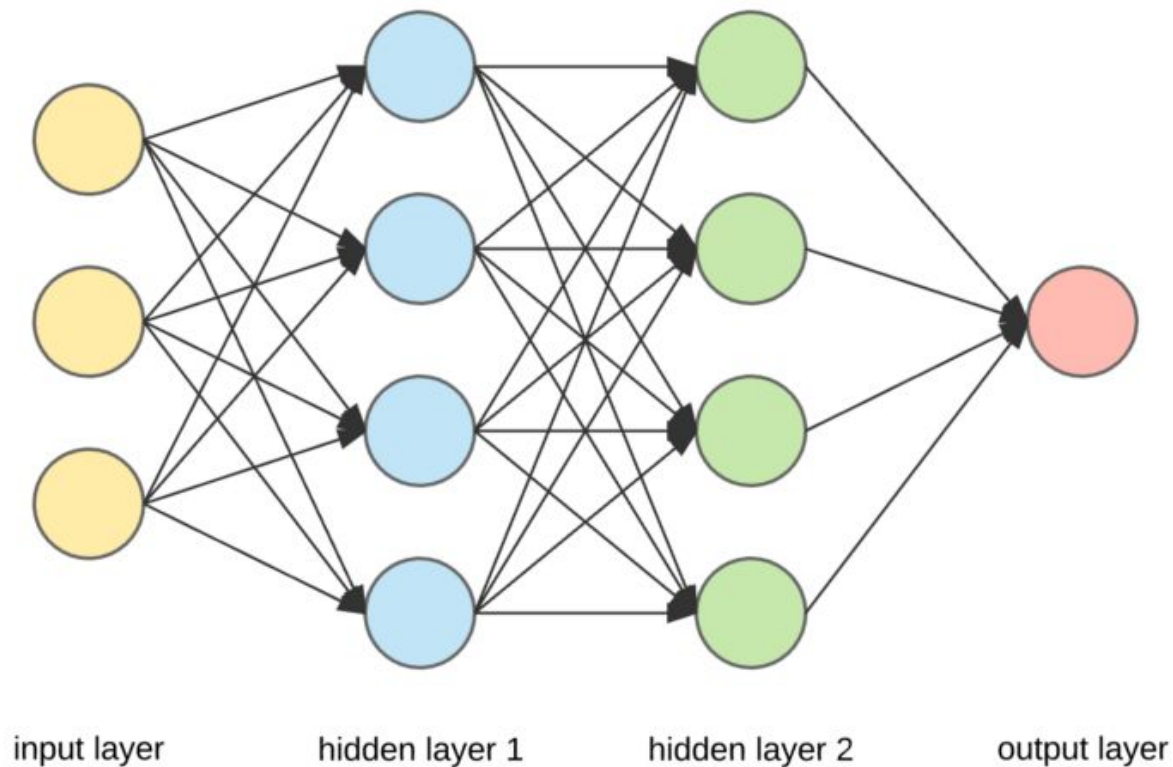
독립변수	종속변수
공부시간	시험점수
온도	레모네이드 판매량
역세권, 조망	집 값
온실 기체량	기온 변화량
자동차 속도	충돌 시 사망 확률
나이	키

→ 학습을 통한
선형모델 찾기 →

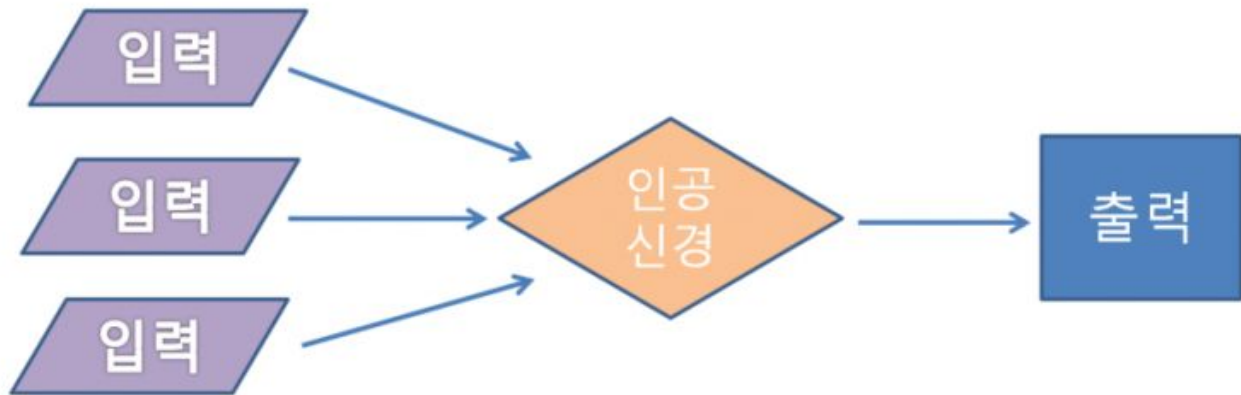
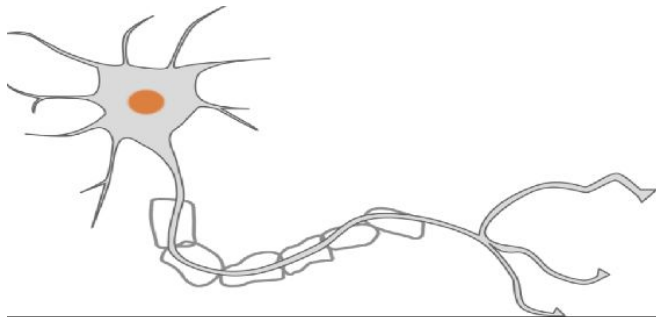


인공지능

인공지능이란?



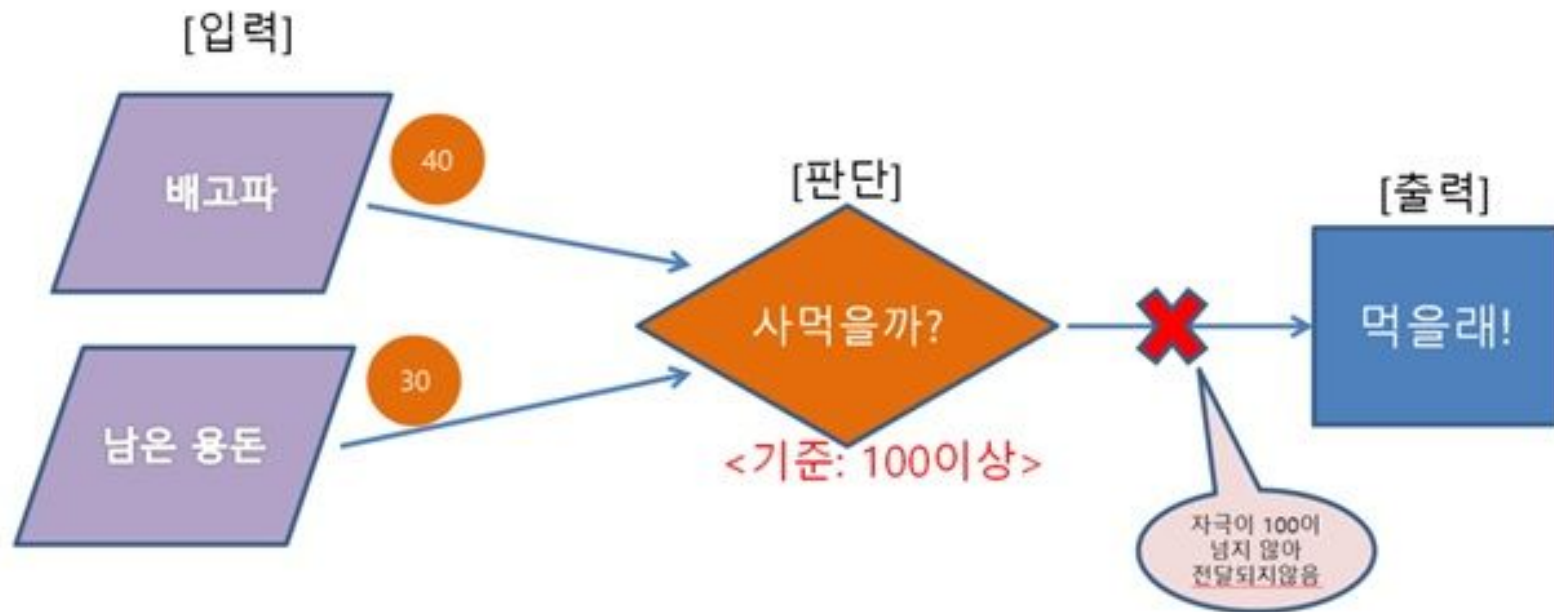
사람이라면? (1)



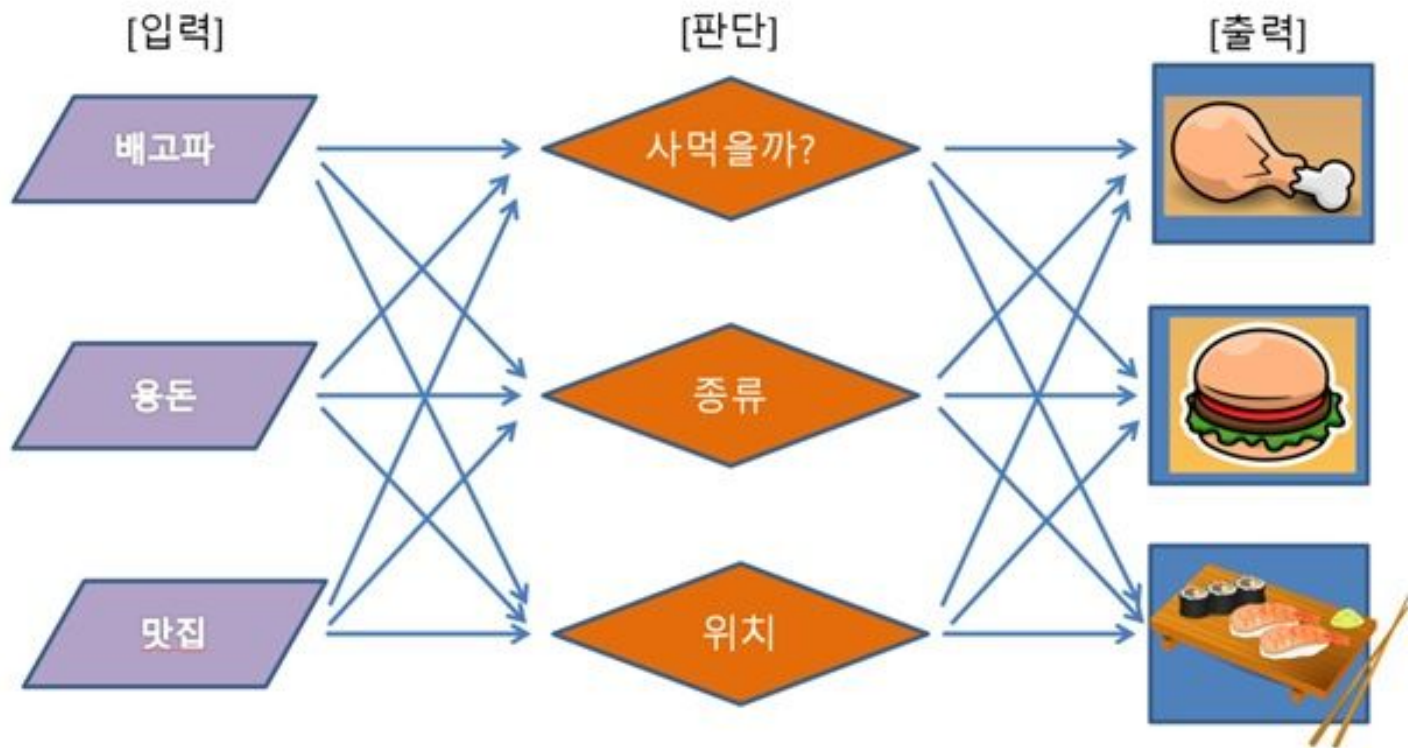
사람이라면? (2)



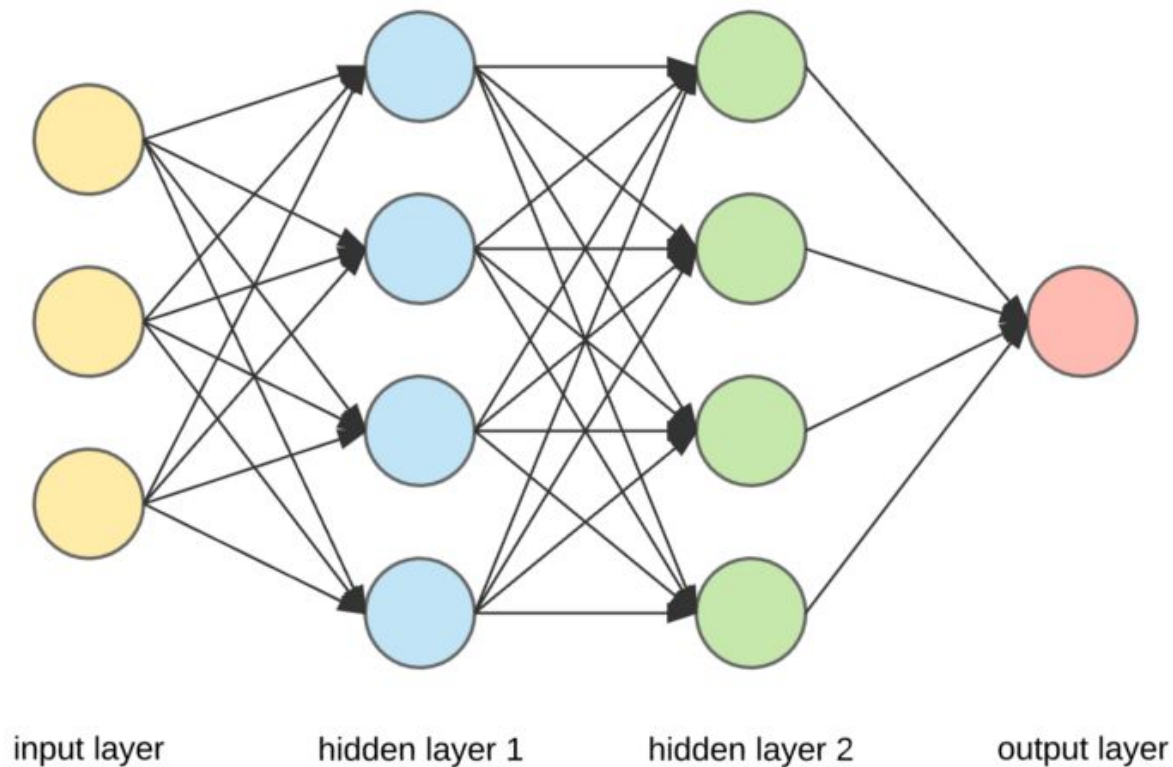
사람이라면? (3)



사람이라면? (4)



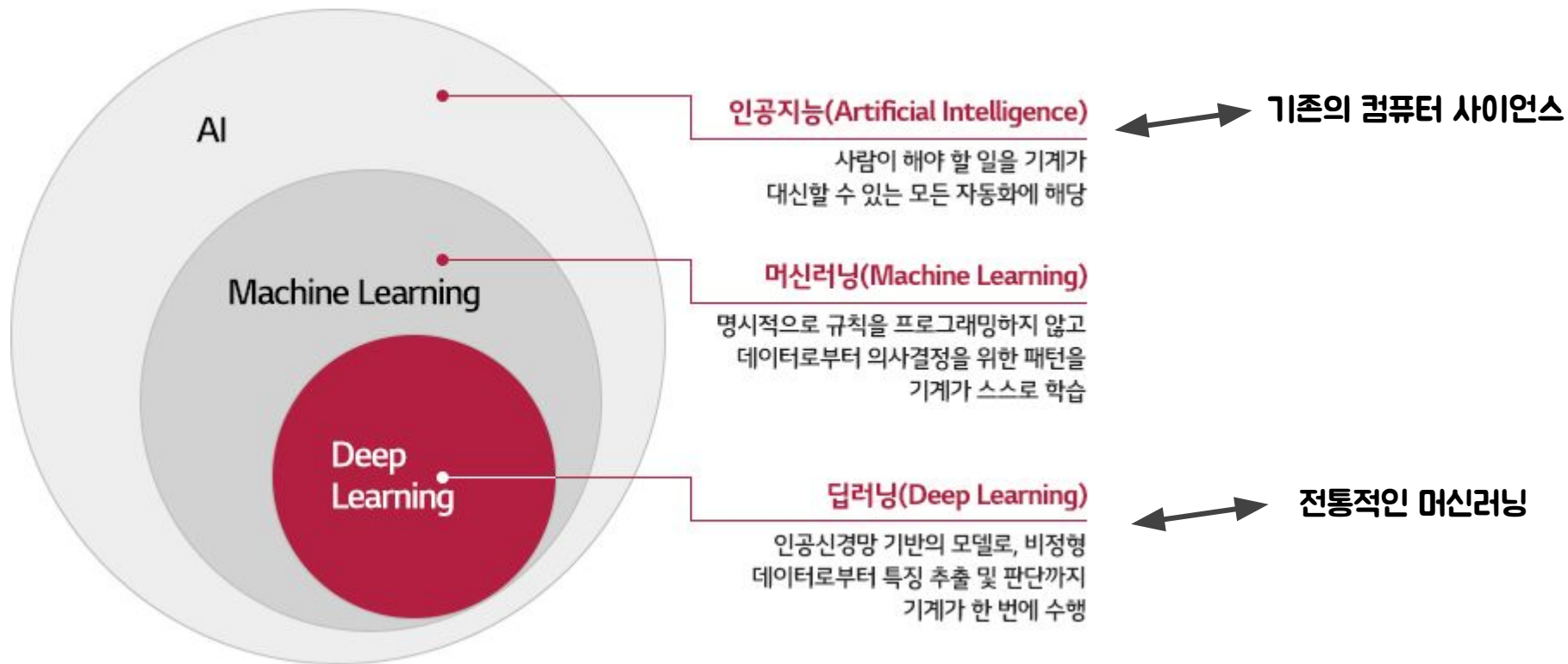
인공신경망



다층 신경망(다층 퍼셉트론) 종류

단층 신경망		입력층 - 출력층
다층 신경망	얇은 신경망	입력층 - 은닉층 - 출력층
	심층 신경망	입력층 - 은닉층 n개 - 출력층

딥러닝의 개념



[퀴즈] 신경망과 딥러닝

1. 인공신경망은 사람의 뇌세포 뉴런의 구조를 모방하는 것으로 시작하였다?
2. 인공신경망의 층이 얇을수록 간단해서 의사 결정에 유리하다?
3. 인공지능과 딥러닝, 머신 러닝은 다 같은 말이다?
4. 딥러닝은 각광 받은 최신기술이므로 데이터의 양이 적을 때에도 가장 좋은 성능을 가진다?
5. 자율 주행 자동차는 객체 인식에 활용된 딥러닝이 적용되었다?

간단하게 해보기

간단한 머신러닝 제작하기

$$y = (2 * x) - 1$$

```
float func(float x){  
    float y = ( 2 * x ) - 1;  
    return y;  
}
```

파이썬 사용

- 인공지능 관련 라이브러리 중 특징에 따라 선택해서 사용하면 됨

	장점	단점
텐서플로우	파라미터 변화 양상이나 DNN의 구조 알 수 있음	메모리를 효율적으로 사용하지 못함
사이킷런	머신러닝용 / 탄탄한 학습 알고리즘	딥러닝이나 강화 학습 다루지 않음
파이토치	간단하고 직관적으로 학습 가능	텐서플로우에 비해 사용자층 낮고 예제 구하기 힘들
케라스	배우기 쉽고 모델 구축하기 쉬움 (텐서플로우 2.0의 API)	오류가 발생할 경우, 이유를 알 수 없는 경우가 있음

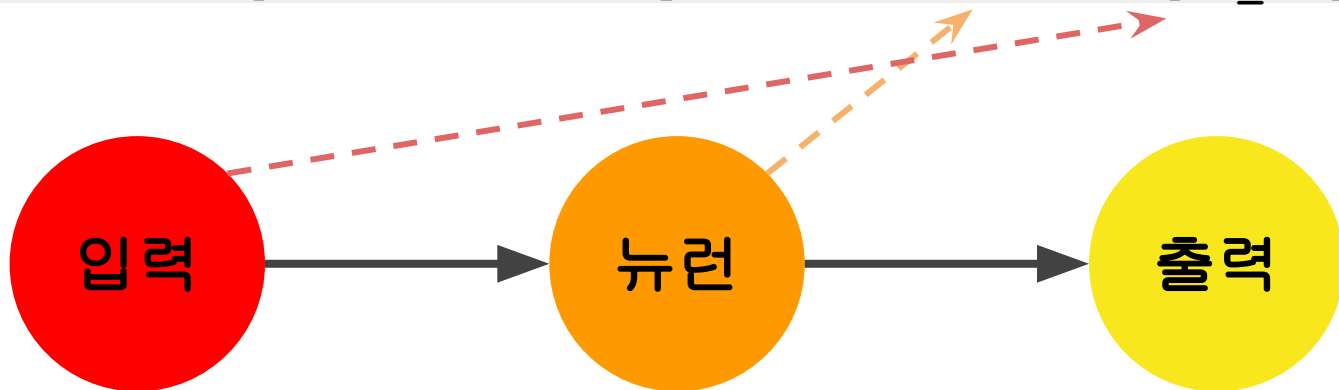
- 우리는 텐서플로우, 케라스 사용 할 예정

라이브러리 가져오기

```
import tensorflow as tf #텐서플로우를 사용할 예정이며, 애칭 tf로  
import numpy as np #numpy(연산 관련 라이브러리)  
from tensorflow import keras #텐서플로우에서 케라스 가져옴
```

인공신경망 정의, 컴파일

```
model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
```



인공신경망 정의, 컴파일

```
model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
```



```
model = Sequential()
```

```
model.add(Dense(input_dim=5, units=25, activation='relu'))
```

```
model.add(Dense(units=25, activation='relu'))
```

```
model.add(Dense(units=1, activation='linear'))
```

손실함수 LOSS

```
model.compile(optimizer = 'sgd', loss = 'mean_squared_error')
```

- 인공지능이 학습하며 발생하는 오류(손실)을 측정
- 입력/출력을 바탕으로 규칙을 파악(생성)하는데, 예측이 얼마나 잘 됐는지 평가함
- loss가 적을수록 수렴 = 학습이 잘 되고/되어 있음
- mse 평균 제곱 오차
 - 실제값과 예측값의 차이를 계산함

최적화 Optimizer

```
model.compile(optimizer = 'sgd', loss = 'mean_squared_error')
```

- 학습 시 발생한 오류(손실)을 수정하여 loss가 적도록 만들어야 함
- loss함수로 예측이 얼마나 잘 됐는지 평가하고 모델을 최적화하기 위해 어떻게 할 것인지 의사결정을 함
- sgd 확률적 경사 하강방식
 - 데이터의 경사를 계산하여 다음 추측값을 찾음

훈련데이터

```
xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype = float)
ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0], dtype = float)
```

- 인공지능이 학습을 할 때 사용할 데이터
- 원래는 스프레드 시트(주로 csv파일)이나 사진, 음성 등 자료 필요

인공신경망 훈련하기

```
model.fit(xs, ys, epochs=500)
```

- x(입력)와 y(출력) 간의 관계를 학습함
- 여기에서는 아래 x, y의 관계를 학습함
 - `xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype = float)`
 - `ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0], dtype = float)`
- fit함수는 1)추측 2)오류(손실) 측정 3)다른 추측 ... 반복
- epochs: 훈련 횟수

학습 끝! 예측해보기

```
model.predict([10])
```

- 훈련의 결과, x 값을 입력하면 y 값을 예측해준다!
- 아래 훈련데이터를 기반으로 $y = 2x - 1$ 과 비슷하다고 예측한 것
 - `xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype = float)`
 - `ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0], dtype = float)`

실습 해보기