

220207

★ input layer을 따로 명시하기

`keras.layers.Dense(1, input_shape=(1,))` ⇒ 기준.

(추가) `model.add(keras.layers.Input(shape=(1,)))`

✓ `verbose=1` → epoch 과정 시각화.

1. 모델 생성과정 (Sequential Model)

① 청소 : `keras.backend.clear_session()`

② 모델 선언 : `model = keras.models.Sequential()`

③ Layer 선언 : `model.add(keras.layers.Input(shape=(1,)))`
`.Dense(1))`

㉔ 컴파일 `model.compile (loss = 'mse', optimizer = 'adam')`

☆ 시작점을 고정시키지 않는다면 성능이 다르게 나올 수 있다.

딥러닝 → 블랙박스 모델 ⇒ 결과의 향상을 위해 노력.

→ 연결주의 알고리즘.

(이미지	자율행
	텍스트	자연어(번역)

☆ y : sigmoid 0 ~ 1 사이로 축소시켜주어라.

ex : 아이스크림을 구매할 확률.

⇒ $y = \text{Dense}(1, \text{activation} = \text{'sigmoid'})$

⊕ binary_crossentropy

⇒ sigmoid 미분함수의 loss.

Activation 함수미기입시 작동은 되나, loss / accuracy의 수치변화X

Sequential API

Functional API

* `il = Input(shape = ...)`

`o1 = Dense(1)(il)`

① 변수명기입 → `il`

② `Dense(1)(input layer)`

거의 비슷하지만 위의 2가지가 다르다.

→ ③ `model = keras.models.Model(il, o1)`

⇒ 모델링 하는 방법도 다르다.

★ `model.fit` = 반복실행시 여러번 쌓이는 구조.

⇒ `session clear`가 이를 방지.

Function style → Layer의 연결을 조절. 자유롭게 연결가능.

Sequential style → 반드시 순서대로 연결.

★ import 부분에서 from - import를 세부들 다 불러와도 된다.