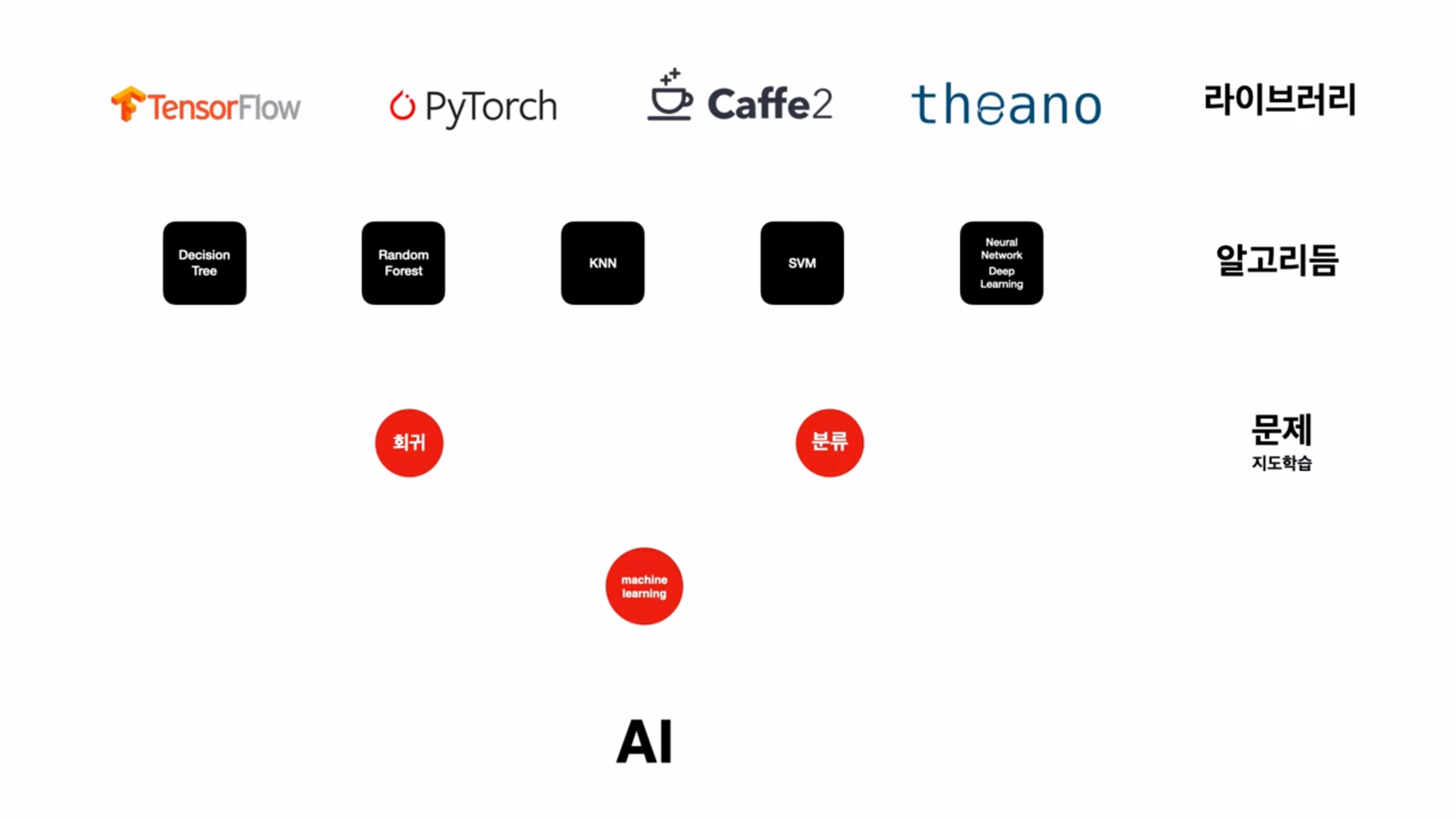
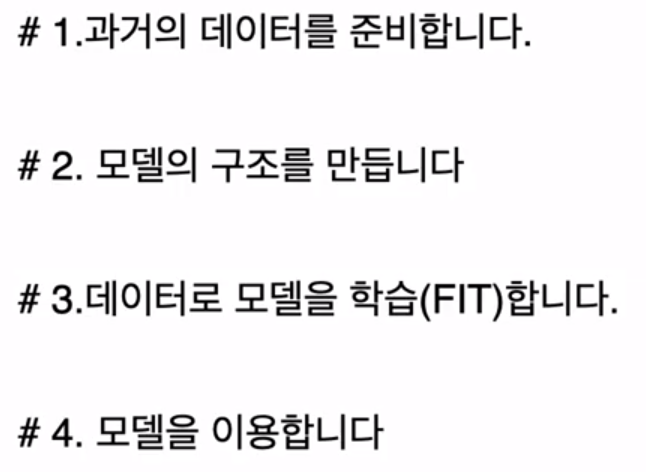
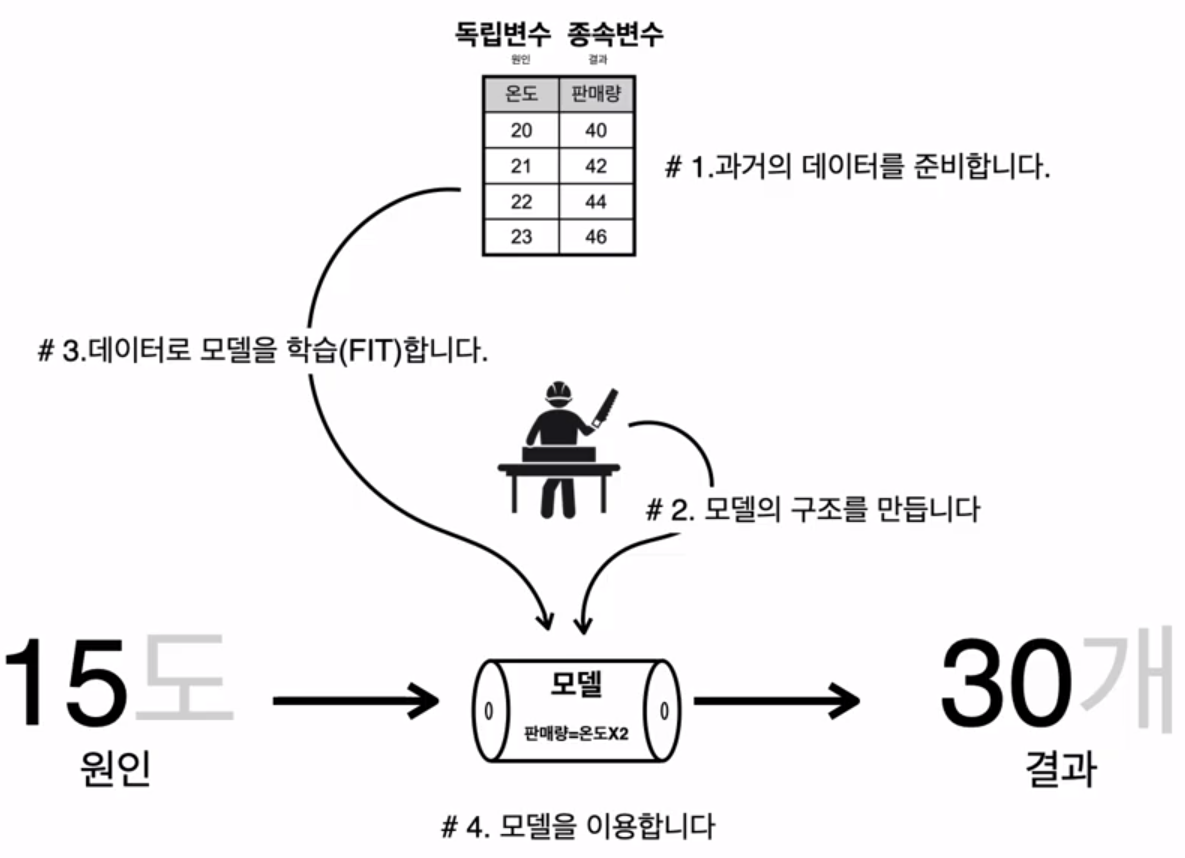
10/2 월

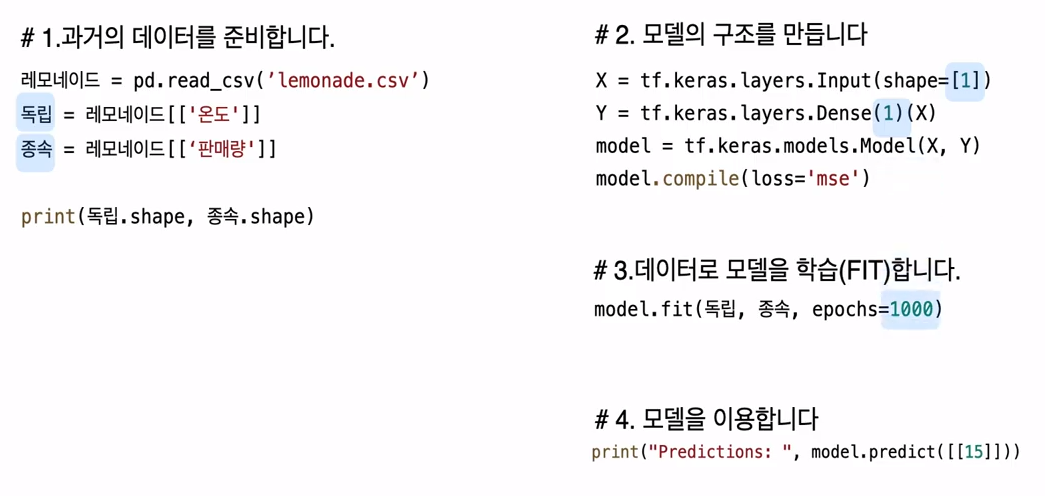
텐서플로우



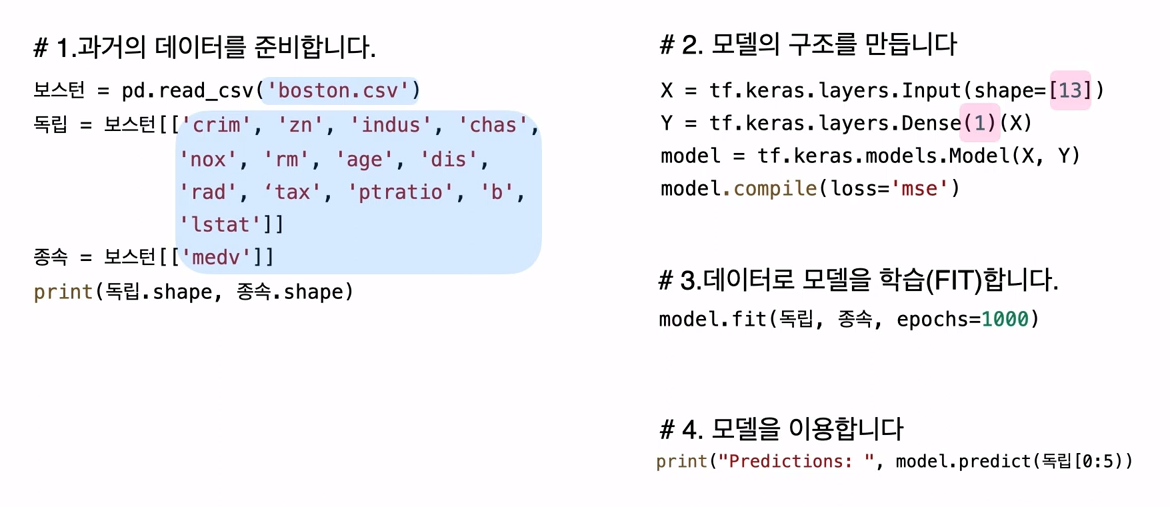
지도학습 과정



레모네이드 판매 예측

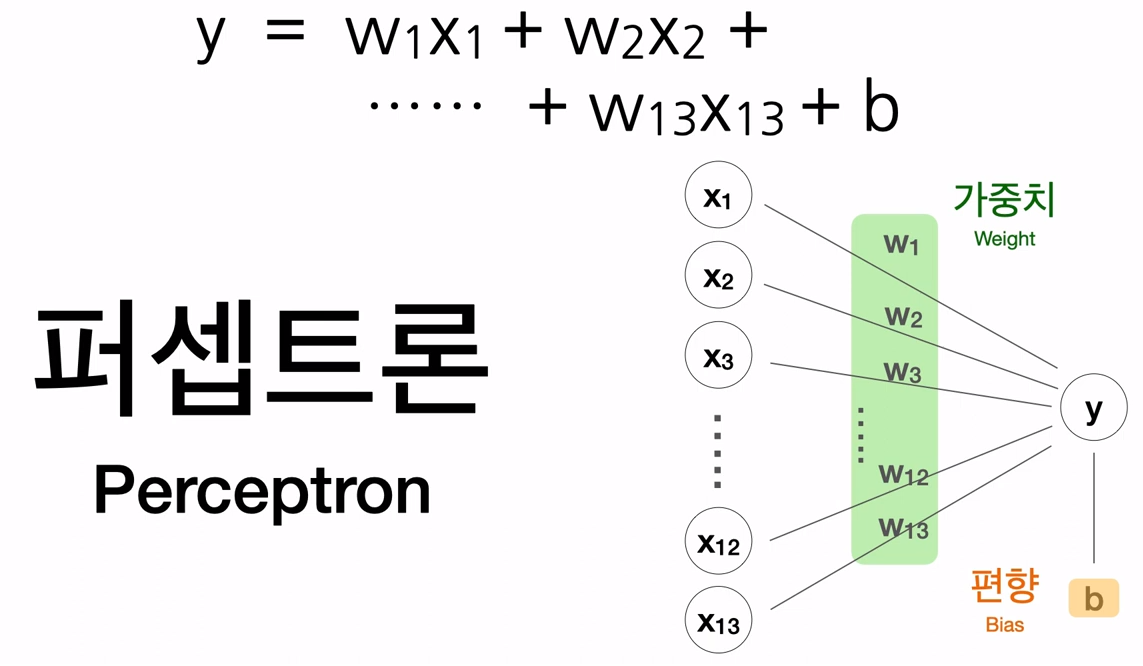


**보스턴 집값 중앙값 예측**

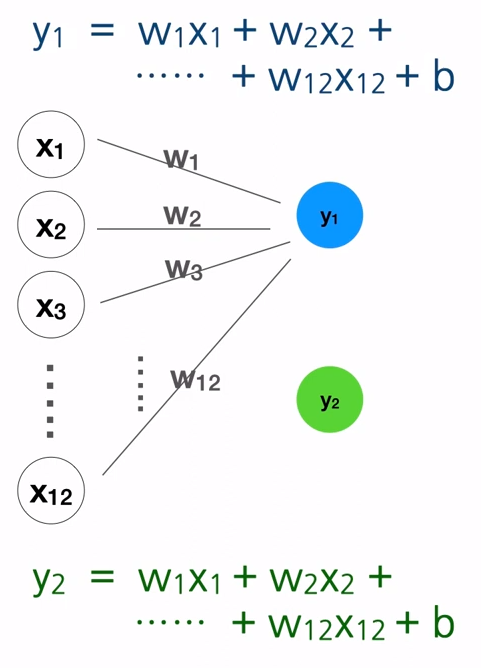


**퍼셉트론** : **인공신경망 모형**

입력 레이어/ 출력 레이어/ 가중치/ 편향 … #2 모델 구조 만들기 그림으로 표현!

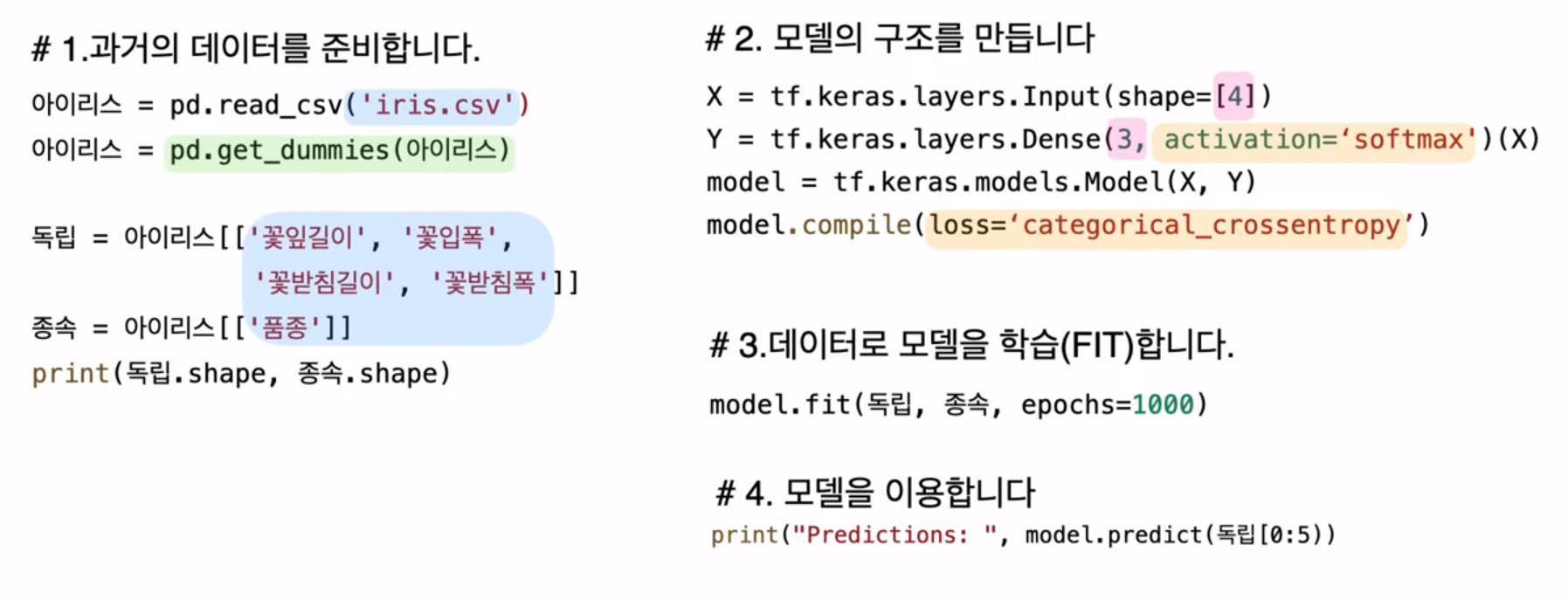


퍼셉트론 두 개가 병렬로 연결된 모델



**아이리스 품종 분류**

* 종속변수가 범주형
* 뉴런 하나로 만든 모델



데이터준비(원핫인코딩) -> 종속변수 3개

모델구조(활성화함수, loss 설정) 부분에서 회귀모델과 차이점 발생

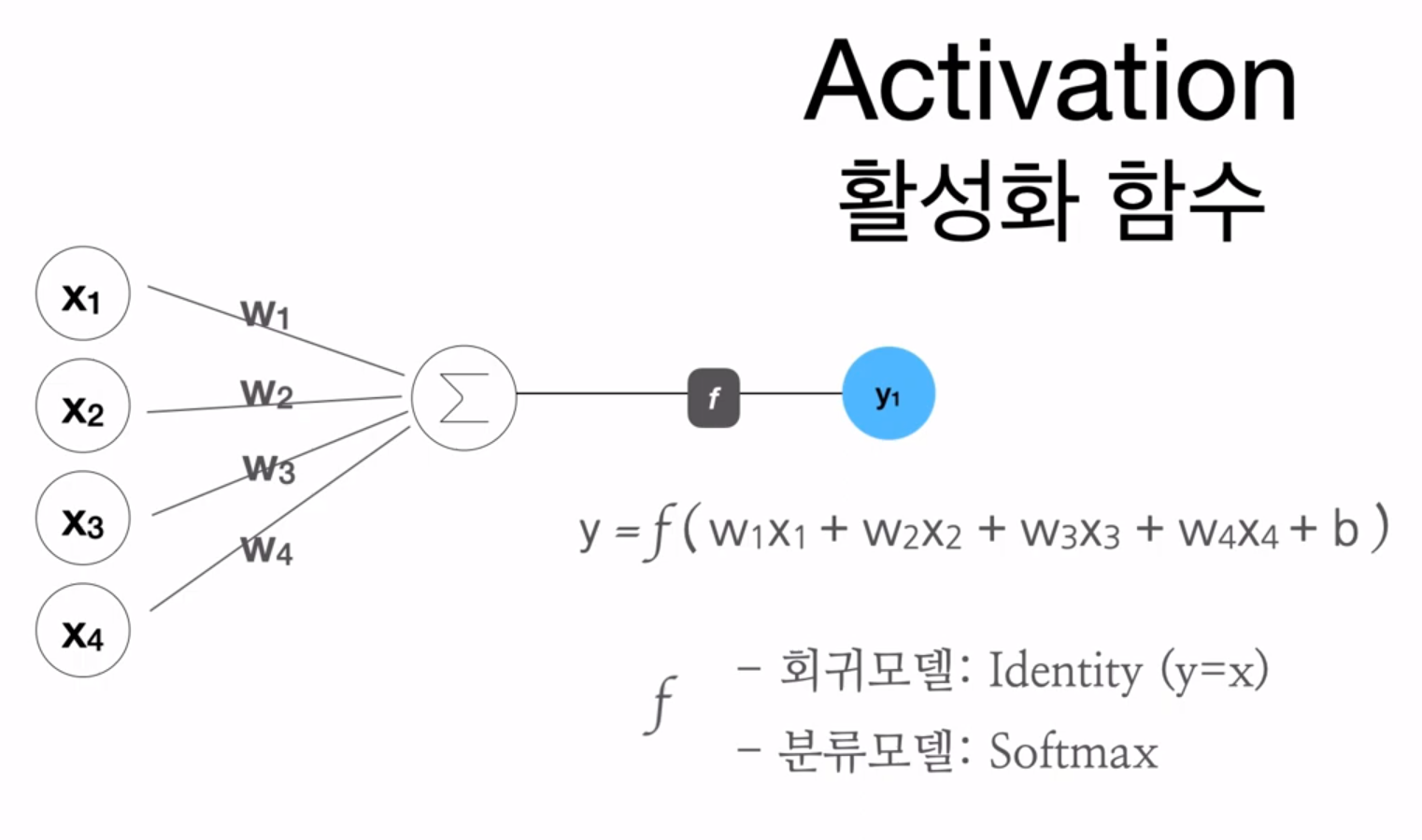
**원 핫 인코딩**

* 범주형 데이터를 0,1 로 변환
* **Pd.get\_dummies(**아이리스) -> 종속변수가 3개가 됨.

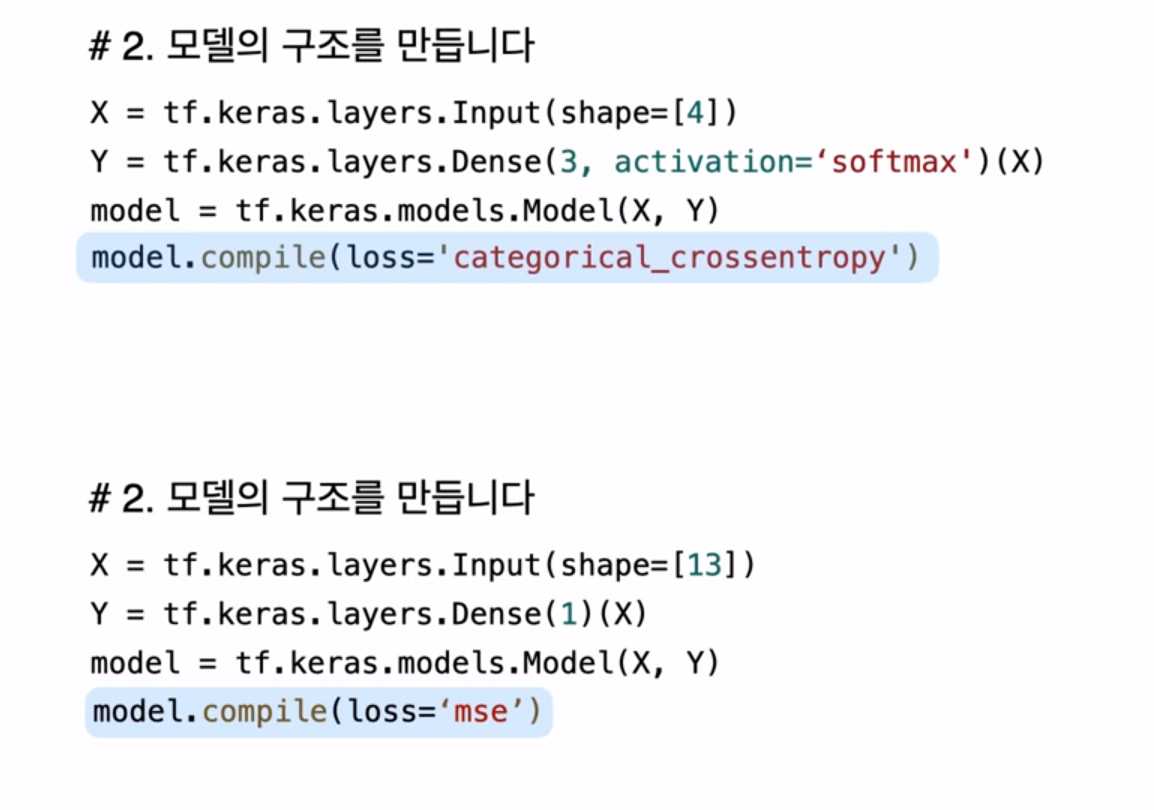


sigmoid, softmax함수: 분류를 0~1의 확률로 표현해 예측.

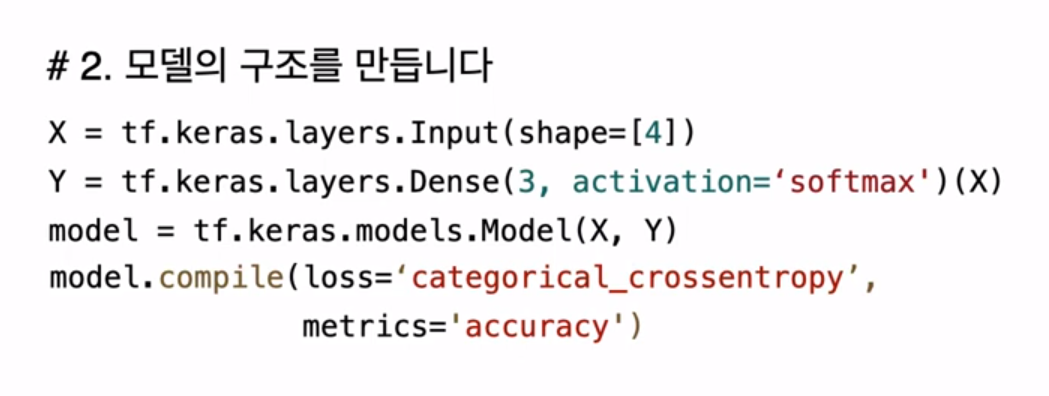
**활성화 함수**: 퍼셉트론의 출력이 어떤 형태로 나갈지 조절



Loss: 분류-cross entropy, 회귀-mse

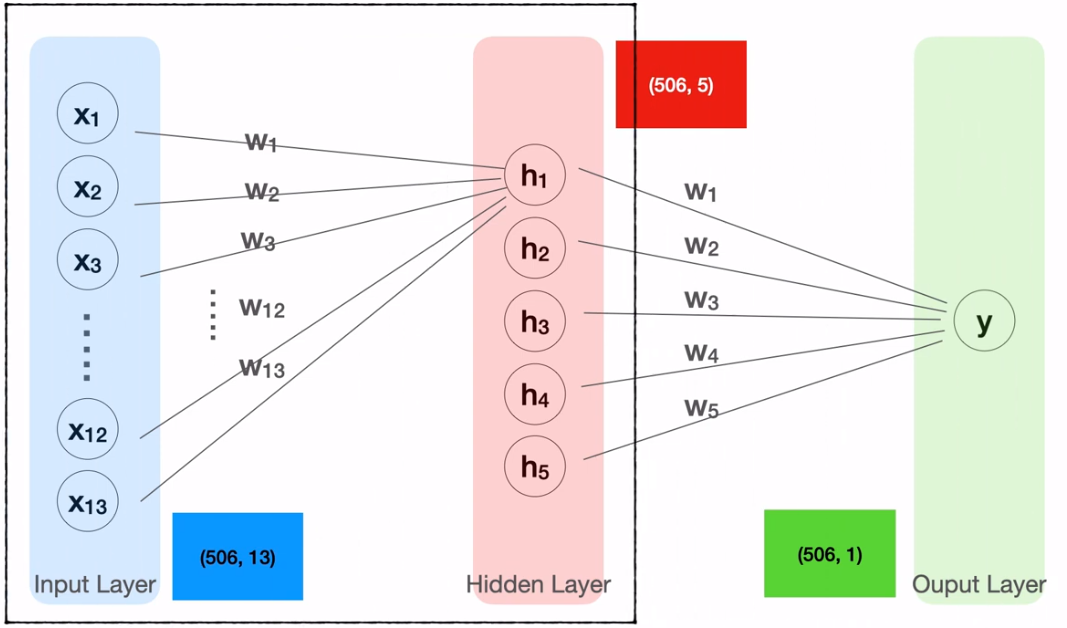


분류 평가지표 – 정확도, F1 Score 가 loss보다 보기 좋음.



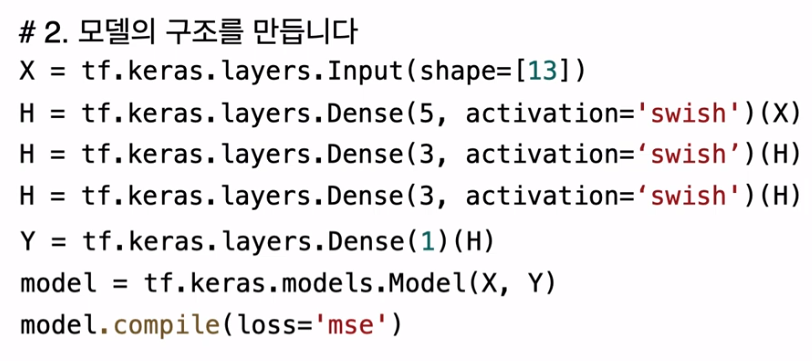
**딥러닝 모델: 멀티레이어 신경망**

* 퍼셉트론을 여러 개 사용해 깊게 연결하여 거대한 신경망을 만듦
* by using 히든 레이어



**히든 레이어**

* 출력층에 X대신 H를 넣어야 함
* Swish 함수 사용



데이터 타입 조정

* 변수(칼럼) 타입 확인: 데이터.dtypes
* 변수를 범주형으로 변경:
  + 데이터['칼럼명'].astype('category')
* 변수를 수치형으로 변경:
  + 데이터['칼럼명'].astype('int')
  + 데이터['칼럼명'].astype('float')
* NA 값의 처리
  + NA 갯수 체크: 데이터.isna().sum()
  + NA 값 채우기: 데이터['칼럼명'].fillna(특정숫자)

학습이 잘 되는 모델

* 사용할 레이어
  + tf.keras.layers.BatchNormalization()
  + tf.keras.layers.Activation('swish')

H = tf.keras.layers.Dense(8, activation='swish')(X) 이거를

H = tf.keras.layers.Dense(8)(X)

H = tf.keras.layers.BatchNormalization()(H) # Dense layer와 activation 사이에 넣어야 효과적.

H = tf.keras.layers.Activation('swish')(H)

이렇게 변형.