

[2] # 파일 선택을 통해 예제 데이터를 내 컴퓨터에서 불러옵니다.
 from google.colab import files
 uploaded = files.upload()
 my_data = 'iris.csv'

) colab에서 CSV파일 가져오기

from tensorflow.keras.models import Sequential
 from tensorflow.keras.layers import Dense
 [from sklearn.preprocessing import LabelEncoder]

import pandas as pd
 [import seaborn as sns
 import matplotlib.pyplot as plt]
 import numpy as np
 import tensorflow as tf

실행할 때마다 같은 결과를 출력하기 위해 설정하는 부분입니다.
 np.random.seed(3)
 tf.random.set_seed(3)

데이터 안에 문자열이 포함되어 있으므로 numpy보다 pandas로

불러온 데이터를 적용합니다.

데이터를 불러와 X, Y 값을 구분하는게 좋음

df = pd.read_csv(my_data, names = ["sepal_length", "sepal_width", "petal_length", "petal_width", "species"])

그래프로 확인

[sns.pairplot(df, hue='species'); plt.show()] 데이터 전체를 한번에 보는 그래프 (속성별 상관도 그래프)



데이터 분류
 dataset = df.values
 X = dataset[:,0:4].astype(float)
 Y_obj = dataset[:,4]

문자열을 숫자로 변환 Y값이 문자열이므로 숫자로 바꿔야 함

e = LabelEncoder()
 e.fit(Y_obj)
 Y = e.transform(Y_obj)

→ 문자열을 숫자로 바꾸기

array(['Iris-setosa', 'Iris-versicolor', '...'])

Y_encoded = tf.keras.utils.to_categorical(Y)

→ 한정화 함수 적용을 위해

→ Y값을 0과 1로 바꾸어주기 만들기 array([1, 2, 3])

모델의 설정

model = Sequential()
 model.add(Dense(16, input_dim=4, activation='relu'))
 model.add(Dense(3, activation='softmax'))

array([[1., 0., 0.], [0., 1., 0.], [0., 0., 1.]])

모델 컴파일

최종 출력값이 세 중 하나 있음

model.compile(loss='categorical_crossentropy',
 optimizer='adam',
 metrics=['accuracy'])

* 원-핫 인코딩 (0과 1로 만들기)

→ 소프트맥스 함수

모델 실행

: 총합이 1인 형태로 바뀌서 계산해 주는 함수

model.fit(X, Y_encoded, epochs=50, batch_size=1)

→ 작은 값은 더 작아지고 큰 값은 두드러짐

결과 출력

print("\n Accuracy: %.4f" % (model.evaluate(X, Y_encoded)[1]))

입력 → 가중치 → 출력

3.0	→	0.7
1.5	→	0.2
0.3	→	0.1

→ 원-핫 인코딩

1.
0.
0.

원-핫 레이블

합 = 1