

```
# Lab 6 Softmax Classifier
import tensorflow as tf
import numpy as np
```

다리 갯수 표기 여부 등에 따라

# Predicting animal type based on various features → 어떤 동물인지 예측하는 방법

```
xy = np.loadtxt('..\data-04-zoo.csv', delimiter=',', dtype=np.float32)
```

```
x_data = xy[:, 0:-1] → n개 data
```

```
y_data = xy[:, [-1]] → 1개 data
```

```
print(x_data.shape, y_data.shape)
```

```
'''
```

```
(101, 16) (101, 1)
```

```
'''
```

```
nb_classes = 7 # 0 ~ 6
```

```
# Convert y_data to one_hot
```

```
y_one_hot = tf.keras.utils.to_categorical(y_data, nb_classes)
```

```
print("one_hot:", y_one_hot)
```

클래스 벡터 (정수)를

이진 클래스 행렬로 변환

행렬로 변환할

총 클래스의 수

클래스 벡터

```
tf.model = tf.keras.Sequential()
```

```
tf.model.add(tf.keras.layers.Dense(units=nb_classes, input_dim=16, activation='softmax'))
```

```
tf.model.compile(loss='categorical_crossentropy', optimizer=tf.keras.optimizers.SGD(lr=0.1),
metrics=['accuracy'])
```

```
tf.model.summary()
```

```
history = tf.model.fit(x_data, y_one_hot, epochs=1000)
```

```
# Single data test
```

```
test_data = np.array([[0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0]]) # expected prediction == 3 (feathers)
```

```
print(tf.model.predict(test_data), tf.model.predict_classes(test_data))
```

```
# Full x_data test
```

```
pred = tf.model.predict_classes(x_data)
```

```
for p, y in zip(pred, y_data.flatten()):
```

```
print("{} Prediction: {} True Y: {}".format(p == int(y), p, int(y)))
```

다차원 배열을 1차원 배열로

꼭 퍼준다.

[ True] Prediction: 6 True Y: 6

[ True] Prediction: 1 True Y: 1

→ pred와 y\_data.flatten() 2개의 배열을 하나로 묶어서

표로 넘겨줌

: 데이터 로드

: 데이터셋 생성

: 모델 구성

: 학습과정 설정

: 모델 학습

: 예측

predict

: [[0.22520512]  
[0.9820919]  
[0.9612848]  
[0.02690611]]

predict\_classes

: [[0]  
[1]  
[1]  
[0]]