2️⃣

분류

분류

분류란 좁게는 데이터를 통해 레이블을 구분해내는 것을 의미한다. 이는 이진 분류와 다항 분류로 나누어볼 수 있는데, 이진 분류란 2개의 즉, 0과 1(True or False)을 구분하는 것이다. 다항 분류란 항이 여러개란 의미로 앞으로 MNIST 데이터와 같이 여러 종류의 카테고리가 존재하는 데이터를 구분하는 것을 의미 한다.

자. 그렇다면 인공지능을 활용한 여러 task 중에서 가장 기본이 되는 task 중 하나인 이미지 분류를 진행해볼 것이다. 분류 중 가장 기본이 되는 MNIST 데이터셋을 사용하여 따라해보자.

분류분류분류란 좁게는 데이터를 통해 레이블을 구분해내는 것을 의미한다. 이는 이진 분류와 다항 분류로 나누어볼 수 있는데, 이진분류란 2개의 즉, 0과 1(True or False)을 구분하는 것이다. 다항 분류란 항이 여러개란 의미로 앞으로 볼 MNIST 데이터와 같이 여러 종류의 카테고리가 존재하는 데이터를 구분하는 것을 의미한다.자, 그렇다면 인공지능을 활용한 여러 task 중에서 가장 기본이 되는 task 중 하나인 이미지 분류를 진행해볼 것이다. 분류 중 가장 기본이 되는 MNIST 데이터셋을 사용하여 따라해보자.

아래에서 간단한 분류 코드를 작성해볼 것이다. Colab 노트북을 하나 열어 코드를 작성해보자.MNIST 분류

> 💡 MNIST는 0부터 9까지의 손으로 쓴 숫자 이미지 데이터이다. 기초적인 인공지능 코드 및 논문에서도 자주 볼 수 있으니 기억해두자!!

```
from sklearn.datasets import fetch_openml

mnist = fetch_openml("mnist_784", version=1) # 데이터 불러오기

X, y = mnist["data"], mnist["target"] # X는 이미지 데이터, y는 레이블

print(f"mnist data shape : {X.shape}, mnist label length : {y.shape}")
```

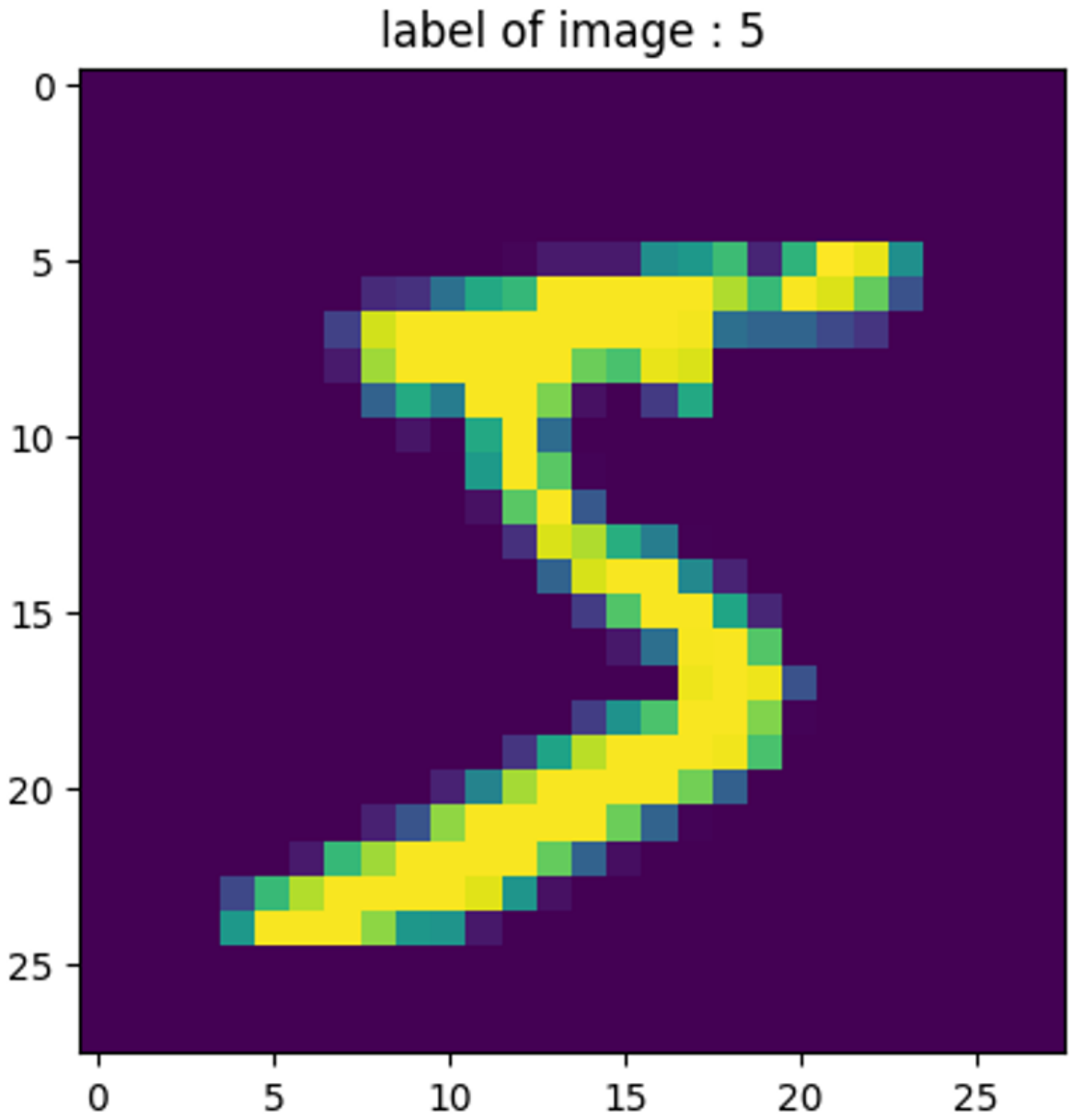
```
>> mnist data shape : (70000, 784), mnist label length : (70000,)
```

데이터의 형상을 보았을 때에 (70000, 784)인 것을 알 수 있다. label의 길이가 70000인 것을 토대로 보았을 때에, 70000장의 이미지가 각각 784개의 데이터를 가진다고 생각할 수 있다. 조금 더 나아가 이미지는 784 = 28 * 28이기에, 크기가 28px인 것을 예상해볼 수 있다. 이미지를 간단하게 출력해보자.

```
import matplotlib.pyplot as plt

img = X.iloc[0, :].to_numpy().reshape(28, -1)

plt.imshow(img)
plt.title(f"label of image : {y[0]}")
plt.show()
```



이미지는 숫자 5에 대한 손글씨라는 것을 확인할 수 있다.

그렇다면 이미지를 train 이미지와 test 이미지로 나누어 학습시켜보자.

```
X_train, X_test, y_train, y_test = X[:60000], X[60000:], y[:60000], y[60000:]
```

위 코드에서는 학습 이미지를 60000장, 테스트 이미지를 10000장으로 구분하였다. 하지만 이 수를 나누는 것은 절대적으로 개인에게 달려있다(잘 모르겠다면 학습 80%, 테스트 20%가 무난하다).

학습 코드는 다음과 같다.

```
from sklearn.linear_model import SGDClassifier

sgd_clf = SGDClassifier(random_state=42) # 모델 선언
sgd_clf.fit(X_train, y_train) # 학습
```

학습이 완료되었다면 모델이 잘 작동하는지 확인해보자.

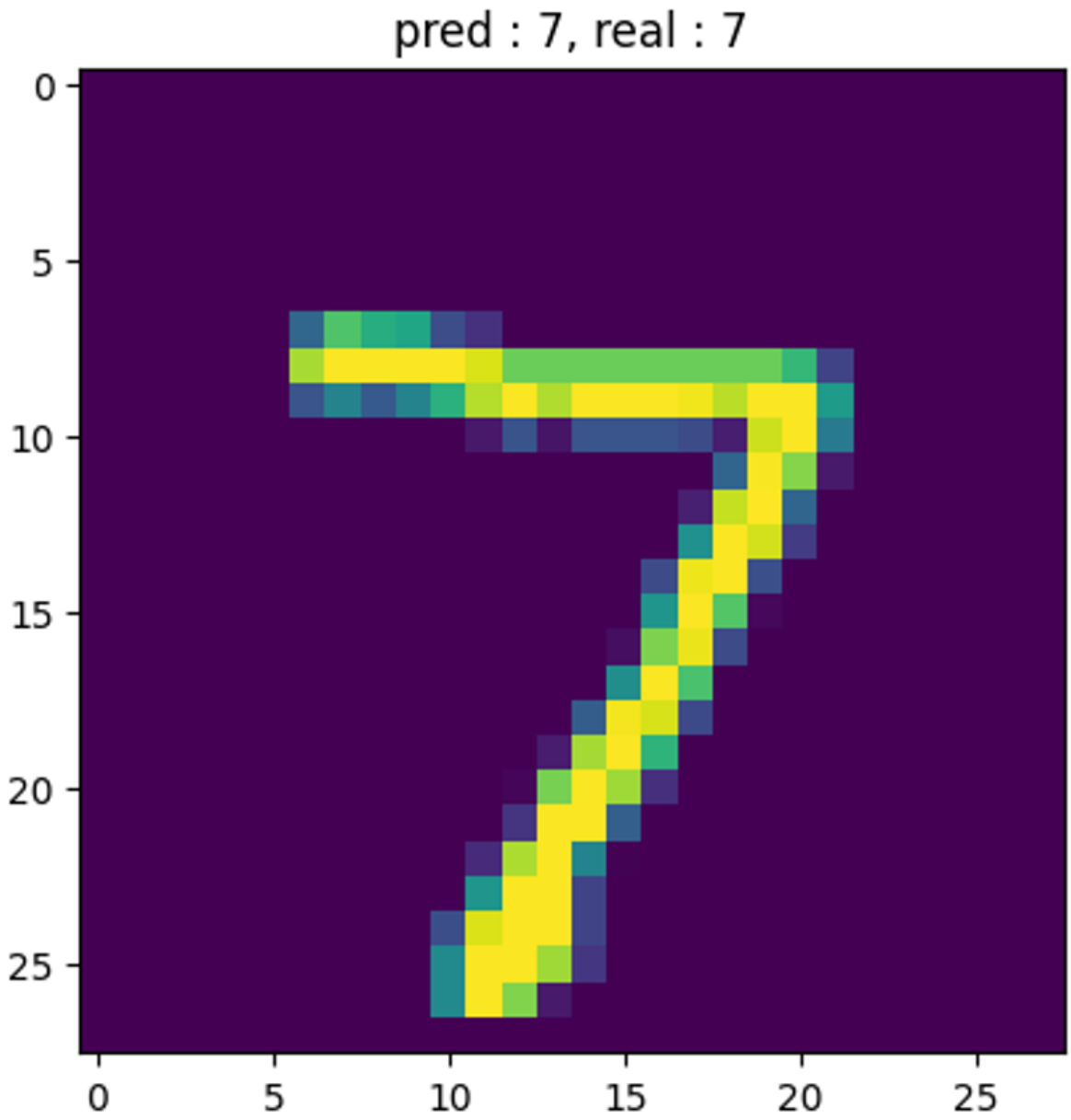
```
sgd_clf.score(X_test, y_test)

>> 0.874
```

정확도는 약 87%정도가 나오는 것을 확인할 수 있다.

```
img = X_test.iloc[0, :]
pred = sgd_clf.predict([img]).item()

plt.imshow(img.to_numpy().reshape(28, -1))
plt.title(f"pred : {pred}, real : {y_test.iloc[0]}")
plt.show()
```



7이란 이미지를 넣어주었을 때, 이미지를 잘 예측해내는 것을 확인할 수 있다.

Conclusion

이처럼 분류는 많은 task에 사용된다. 여러 task 중 분류를 사용하는 것은 정답 레이블이 카테고리 형태인가(i.e.고양이, 개, 소 …)에 따른다. 즉, 회귀 형태인 레이블(주식 값 등)에서가 아닌 형태의 데이터라고 생각하면 편할 것이다.