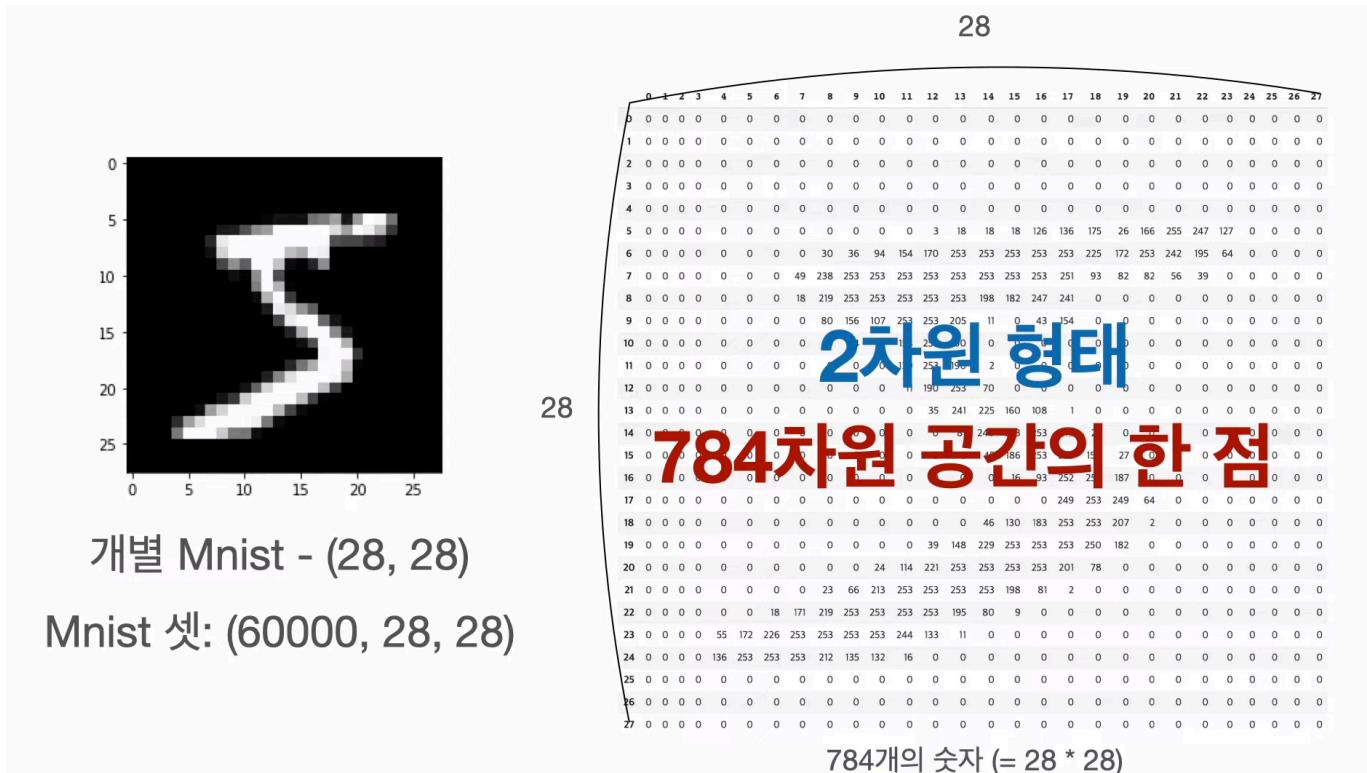


이미지 데이터셋 구경하기

이번에는 우리가 앞으로 사용할 이미지들을 먼저 구경해보겠습니다. 구경하면서 앞서 배운 차원의 관점으로 이미지 데이터를 바라보고 이해하는 연습을 겸해서 할 것입니다. 우리는 전체과정에서 mnist 라고 하는 손글씨 숫자 이미지 셋과 cifar10 이라고 하는 10가지 사물 분류를 위한 이미지셋을 사용해 보겠습니다.



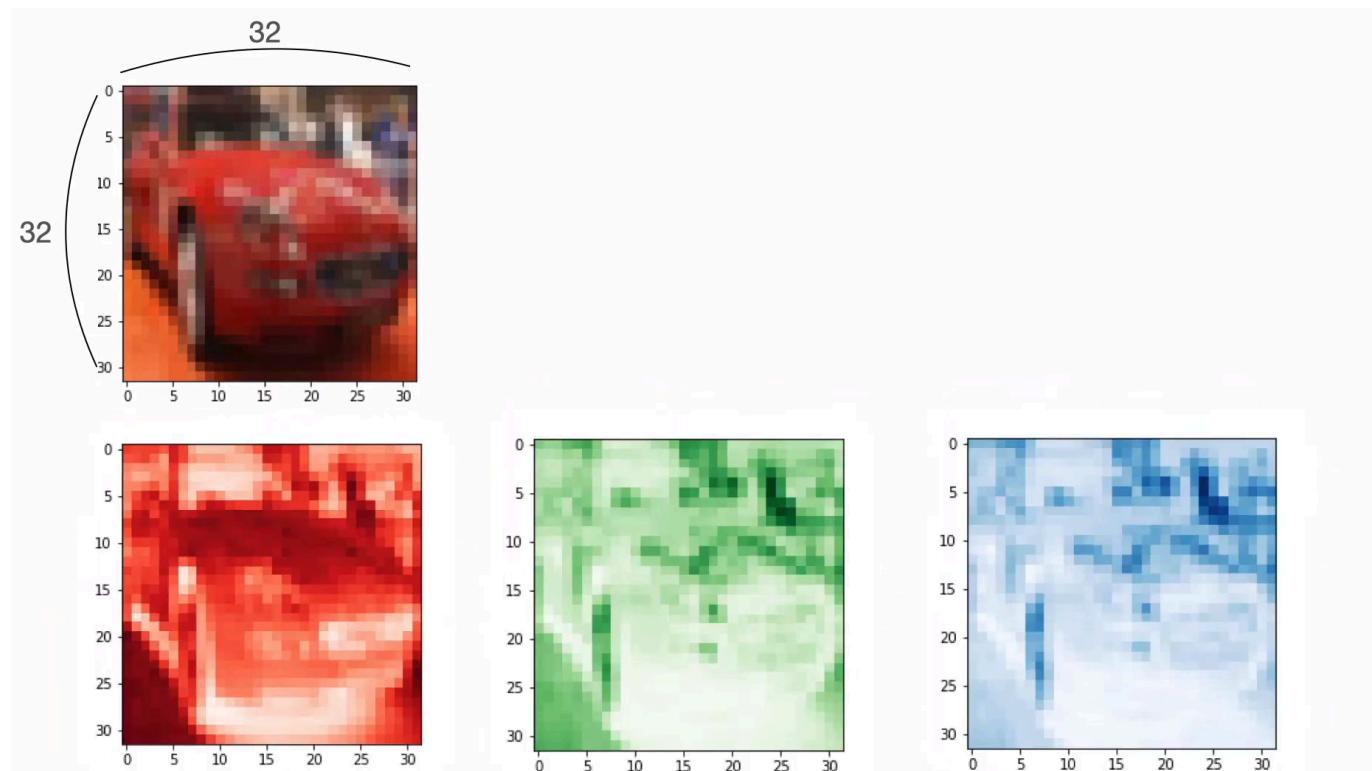
엠니스트는 흑백 이미지로 싸이파10은 컬러 이미지입니다. 여기서 엠니스트의 이미지 하나를 가져와보겠습니다. 5라는 숫자가 쓰여 있습니다. 컴퓨터에게 이미지는 이미지가 아닌 숫자들의 모임일 뿐입니다. 다음과 같이 말이죠.



5라는 숫자가 보이시나요 실눈을 뜨고 보시면 숫자들의 뭉치 사이에 5라는 이미지가 보일 것입니다. 숫자들의 집합이 2차원 형태인 28, 28에 모양으로 이루어져 있습니다. 가로로 28개의 칸, 세로로 28개의 칸으로 채워진 숫자들입니다. 각 숫자 하나는 점에 까맣고 하얀 정도를 나타냅니다. 이 까맣게 되어 있는 부분은 0으로 하얗게 되어 있는 부분은 255에 가까운 수로 되어 있습니다. 각 숫자는 검정인 0과 흰색인 255 사이의 흑백의 정도를 표현하고 있습니다. 이 안의 숫자가 몇개 있을까요? 가로 28개, 세로 28 곱하면 784 입니다. 바로 784개의 숫자가 이 안에 있다는 것입니다.

우리가 조금 전에 차원의 개념을 배웠습니다. 차원의 개념을 알고 있는 우리는 이 이미지를 이렇게 말할 수 있습니다. 이 이미지 하나는 2차원 형태로 되어 있고, 784 차원 공간에 1점으로 표현할 수 있다고 말이죠. 이 이미지가 6만장 준비되어 있다면 이미지셋 데이터의 형태는 (60000, 28, 28)이 되겠습니다.

싸이킷10이미지 하나를 가져와 봤습니다. 빨간색 자동차 이미지입니다.

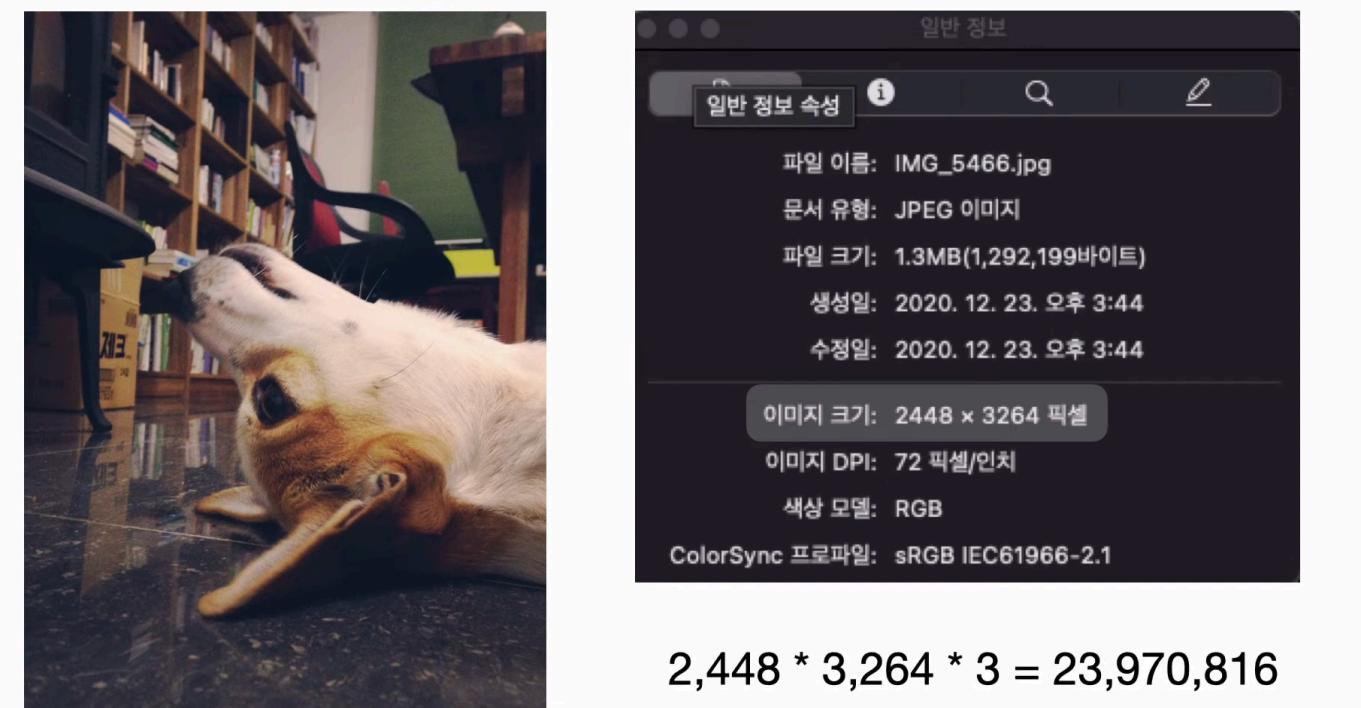


이 이미지 역시 컴퓨터에게는 이미지가 아니라 숫자들의 모음으로 인지 됩니다. 가로로 32개의 칸, 세로로 32개의 칸에 채워진 숫자들입니다. 컬러 이미지는 흑백과 다른점이 있는데 (32, 32) 2차원 숫자 집합이 3개 존재 합니다. 각각 아래 그림에 있는 것처럼 빨간 점에 대한 숫자, 녹색 점에 대한 숫자, 파란 점에 대한 숫자입니다. 여기서 잠깐 앞에 그림에서 본 숫자 데이터를 보면 흑백은 각 칸의 흑백에 대한 숫자가 하나씩 들어 있는데요. 컬러는 그 칸에 숫자가 3개씩 들어 있다고 생각하면 됩니다. 빨강과 녹색 파랑에 대한 숫자가 들어 있는 것입니다. 즉 (32, 32)에 각 점이 세개의 값을 가지게 되는 것입니다. 0에서 255사이의 값으로 하나는 빨간색의 농도, 하나는 녹색의 농도, 하나는 파란색의 농도를 나타내는 것입니다.

그래서 개별 사이파10의 이미지는 (32, 32, 3)의 모양이 되고 3차원 형태입니다.



32 X 32 X 3을 하면 3072 입니다. 사이파10 이미지 하나에는 3072개의 숫자가 들어 있고 이미지 하나는 3072차원 공간의 한점으로 표현된다고 말할 수 있겠습니다. 만약 이 이미지가 5만장 준비되어 있다면 이미지셋 모양은 4차원인 (5000, 32, 32, 3)이 됩니다.



속성중에 이미지 크기라는 속성을 보면 2448 X 3264 픽셀이라고 되어 있습니다. 이 이미지에는 얼마만큼 숫자가 들어 있을까요? 2448의 3264를 곱하고 컬러 이미지이기 때문에 3을 또 곱합니다. 총 23,970,816 개의 숫자가 들어 있습니다. 이미지 한장 안에는 실로 어마어마한 양의 숫자가 들어 있습니다.

컴퓨터에게 이미지를 학습 시킨다고 할 때 당연히 이 큰 이미지를 그대로 사용하지는 않습니다. 충분히 사이즈를 줄인 뒤에 사용해 줘야 합니다. 이제 여러분은 이미지 데이터가 어떻게

구성되고 컴퓨터가 이미지를 어떻게 해석하고 사용하는지 알게 되었습니다. 이미지에 대해 충분히 알게 되었으니 이제 모델을 만들고 학습 시키는 방법을 배우며 되겠습니다. 다음 단계로 가기 전에 배운 내용에 대해 코드 구경과 함께 실습을 할 건데요.

엠니스트와 사이파10 이미지 데이터는 텐서플로어에서 학습을 위해 샘플 이미지를 준비한 것을 사용할 것입니다.

The screenshot shows the TensorFlow API documentation for Python. The URL is tensorflow.org/api_docs/python/tf/keras/datasets. The left sidebar has a tree view of modules: Overview, Input, Model, Sequential, activations, applications, backend, callbacks, constraints, datasets, Overview (selected), boston_housing, cifar10, cifar100, fashion_mnist, imdb, mnist, reuters, estimator, experimental, initializers, layers, losses, metrics, mixed_precision, models, and optimizers. The main content area is titled 'Module: tf.keras.datasets' and contains a 'Modules' section with links to boston_housing, cifar10, cifar100, fashion_mnist, imdb, mnist, and reuters. Each link has a brief description: boston_housing module: Boston housing price regression dataset., cifar10 module: CIFAR10 small images classification dataset., cifar100 module: CIFAR100 small images classification dataset., fashion_mnist module: Fashion-MNIST dataset., imdb module: IMDB sentiment classification dataset., mnist module: MNIST handwritten digits dataset., and reuters module: Reuters topic classification dataset.

사용 하는 코드는 다음과 같습니다.

- MNIST

```
(독립, 종속), _ = tf.keras.datasets.mnist.load_data()  
print(독립.shape, 종속.shape)  
(60000, 28, 28)(60000,)
```

독립변수와 종속변의 `shape`에 집중해서 보시면 되겠습니다. 독립변수의 `shape`은 아까 우리가 보았던 숫자와 동일하게 돼 있는 걸 알수 있습니다. MNIST 는 28, 28 이미지 6만장으로 이루어져 있고 각 이미지마다 그 이미지가 어떤 숫자인지에 대한 정답이 6만개 있습니다.

- CIFAR10

```
(독립, 종속), _ = tf.keras.datasets.cifar10.load_data()  
print(독립.shape, 종속.shape)  
(50000, 32, 32, 3)(50000, 1)
```

사이파10 은 (32, 32, 3) 이미지 5만장으로 되어 있고 각 이미지마다 하나씩 총 5만개인 종속변수가 있습니다. 다만 종속변수에 `shape`은 MNIST 는 1차원 인데 사이파10은 2차원으로 조금 다르게 되어 있는 걸 볼수 있습니다. 그이유는 실습을 통해 알아 보겠습니다.

<https://www.cs.toronto.edu/~kriz/cifar.html>

```
1 # -*- coding: utf-8 -*-
2 import tensorflow as tf
3
4 (mnist_x, mnist_y), _ = tf.keras.datasets.mnist.load_data()
5 print(mnist_x.shape, mnist_y.shape)
6 (cifar_x, cifar_y), _ = tf.keras.datasets.cifar10.load_data()
7 print(cifar_x.shape, cifar_y.shape)
8
9 print(mnist_y[0:10])
10
11 import matplotlib.pyplot as plt
12 plt.imshow(mnist_x[4], cmap='gray')
13
14 print(cifar_y[0:10])
15
16 import matplotlib.pyplot as plt
17 plt.imshow(cifar_x[0])
18
19 import numpy as np
20
21 d1 = np.array([1, 2, 3, 4, 5])
22 print(d1.shape) # 1차원
23
24 d2 = np.array([d1, d1, d1, d1])
25 print(d2.shape) # 2차원
26
27 d3 = np.array([d2, d2, d2])
28 print(d3.shape) # 3차원
29
30 d4 = np.array([d3, d3])
31 print(d4.shape) # 4차원
```

```
32
33 print(mnist_y.shape) # 1차원
34 print(cifar_y.shape) # 2차원
35
36 x1 = np.array([1, 2, 3, 4, 5])
37 print(x1.shape) # 1차원
38 print(mnist_y[0:5])
39 print(mnist_y[0:5].shape)
40
41 x2 = np.array([[1, 2, 3, 4, 5]])
42 print(x2.shape) # 2차원
43
44 x3 = np.array([[1], [2], [3], [4], [5]])
45 print(x3.shape) # 2차원
46 print(cifar_y[0:5])
47 print(cifar_y[0:5].shape)
```