딥러닝 이미지 분류

tensorflow에서 제공하는 샘플 이미지를 다운로드받는다.

(독립,종속), \_ = keras.datasets.mnist.load\_data()

Print(독립.shape,종속.shape)

#(60000,28,28)(60000,)

Mnist는 흑백이미지 인데, 독리변수는 총 6만장, 28\*28크기이다

종속변수. 정답을 의미 하는 6만개 숫자 1차원 배열

(독립,종속), \_ = keras.datasets.cifar10.load\_data()

Print(독립.shape,종속.shape)

#(50000,32,32,3)(50000,1)

Cifar는 컬러이미지 5만장에 32\*32크기, rgb 3비트이다.

정답을 의미하는 5만개숫자, 행이 5만이고 열이 1개인 50000\*1 2차원배열

\_ 부분은 또다른 반환값이 있지만 예제에선 다루지 않으므로 생략한다.

코드 최초 실행시 다운로드를 받고 이후에 다시 실행하면 다운로드 없이 수행된다.

이미지를 한장 출력해본다.

Matplotlib 라이브러리의 모듈 pyplot을 사용하기위해

import matplotlib.pyplot as plt

print(mnist\_y[0:10])

plt.imshow(mnist\_x[0],cmap='gray')

print(cifar\_y[0:10])

plt.imshow(cifar\_x[0])

mnist는 흑백이미지 이므로 cmap =’gray’를 명시해주어야 한다

차원을 확인한다.

import numpy as np

d1 = np.array([1,2,3,4,5])

print(d1.shape)

d2 = np.array([d1,d1,d1,d1])

print(d2.shape)

d3 = np.array([d2,d2,d2])

print(d3.shape)

d4 = np.array([d3,d3])

print(d4.shape)

결과

(5,)

(4, 5)

(3, 4, 5)

(2, 3, 4, 5)

즉 원소가 5개인 1차원 배열 d1,d1을 원소로 4개갖는 2차원 배열d2

D2를 원소로하고 3개를 갖는 3차원 배열 d3, d3를 원소로하는 2개갖는 4차원 배열 d4

이전에 mnist\_y.shape (60000,) 은 숫자 6만개인 1차원 배열

[1,2,3,…60000]

Cifar\_y.shape (50000,1)은 숫자 1개를 원소로 갖는 1차원 배열을 원소로하는 5만개 2차원 배열이다.

[[1],[2],[3]……[50000]]