과목 명: 시스템프로그래밍

담당 교수 명: 소 정 민

<<Assignment 5>>

**서강대학교 컴퓨터공학과**

**20171601**

**강주형**

**목 차**

1. 프로그램 개요 3

2. 프로그램 설명 3

2.1 흐름도 3

3. 모듈 정의 5

4. 코드 5

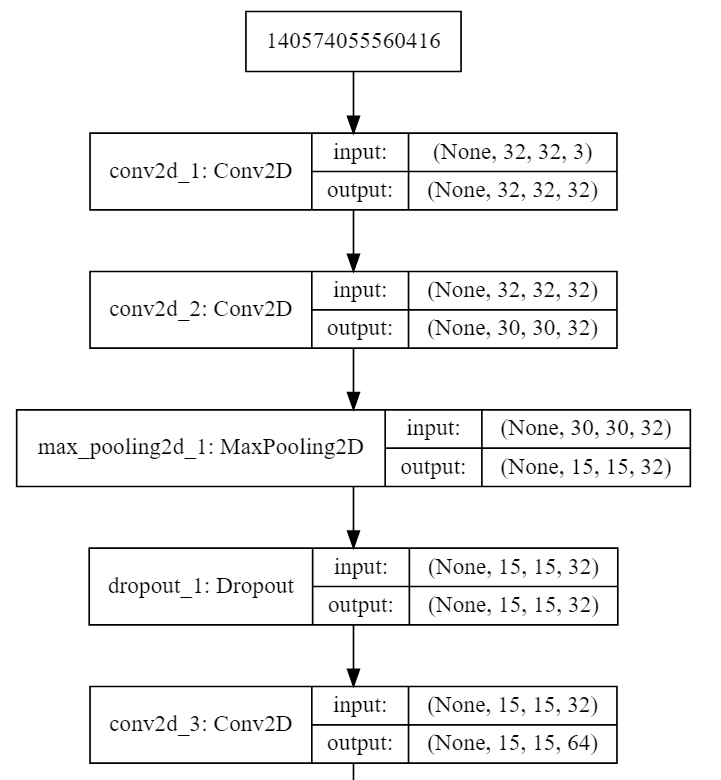
1. **프로그램 개요**

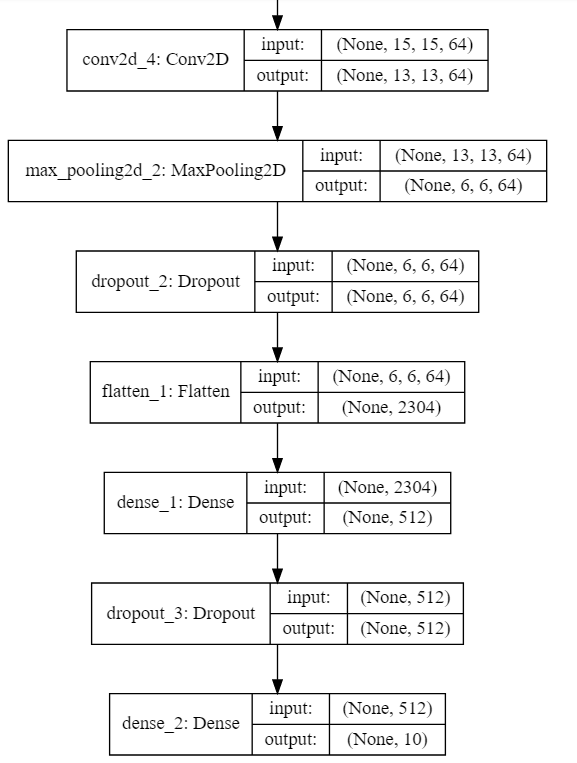
이 프로젝트는 파이썬 프로그래밍을 통해 머신 러닝과 클라우드 서비스의 필요성에 대한 이해를 향상하고자 하는 과제이다.

1. **프로그램 설명**

CIFAR-10 데이터집합은 10개의 클래스 안에 60000개의 이미지로 구성되어 있고 50000개의 훈련 이미지와 10000개의 테스트 이미지가 있다.

이 프로젝트는 CIFAR-10 데이터집합 분류에 대해서 75% 이상의 정확도를 얻는 프로그램을 작성하는 것이다.





[흐름도]

1. **모듈 정의**

좋은 정확도를 얻기 위해 몇 가지 방법을 사용하였다.

첫 번째로, 훈련 횟수를 증가시켰다. 초기 훈련 횟수(epoch)의 값은 10이었고, 이를 100으로 키워주었다. 훈련 횟수를 늘려줄수록 훈련하는 케이스가 많아지기 때문에 정확도가 높아진다.

두 번째로는 hidden layer를 추가하였다.

세 번째로는 활성화 함수 실행 후에 dropout을 추가하여, overfitting을 방지하였다.

1. **코드**

#import modules

import keras

from keras.datasets import cifar10

from keras.models import Sequential

from keras.layers import Dense, Dropout, Flatten, Activation

from keras.layers import Conv2D, MaxPooling2D

from keras import backend as K

from keras.utils.vis\_utils import model\_to\_dot

from IPython.display import SVG

%matplotlib inline

import matplotlib.pyplot as plt

from sklearn.metrics import confusion\_matrix

import pandas as pd

import seaborn as sns

#Tuning parameters

epochs = 100 //epoch값을 증가해준다

learning\_rate=0.01

batch\_size = 128

num\_classes = 10

#plot image

def plot\_images(x, y\_true, y\_pred=None, size=(5, 5)):

assert len(x) == len(y\_true) == size[0] \* size[1]

fig, axes = plt.subplots(size[0], size[1])

fig.subplots\_adjust(hspace=0.5, wspace=0.1)

for i, ax in enumerate(axes.flat):

if x[i].shape[-1] == 1:

ax.imshow(x[i].reshape(x[i].shape[0], x[i].shape[1]))

else:

ax.imshow(x[i])

if y\_pred is None:

xlabel = "True: {0}".format(y\_true[i].argmax())

else:

xlabel = "True: {0}, Pred: {1}".format(y\_true[i].argmax(),

y\_pred[i].argmax())

ax.set\_xlabel(xlabel)

ax.set\_xticks([])

ax.set\_yticks([])

plt.show()

#load dataset

(x\_train, y\_train), (x\_test, y\_test) = cifar10.load\_data()

#reshaping the data

if len(x\_train.shape) < 4:

x\_train = x\_train.reshape(x\_train.shape[0], x\_train.shape[1], x\_train.shape[2], 1)

x\_test = x\_test.reshape(x\_test.shape[0], x\_train.shape[1], x\_train.shape[2], 1)

x\_train = x\_train.astype('float32')

x\_test = x\_test.astype('float32')

x\_train /= 255

x\_test /= 255

print('x\_train shape:', x\_train.shape)

print(x\_train.shape[0], 'train samples')

print(x\_test.shape[0], 'test samples')

#applying one hot encoding for the data

y\_train = keras.utils.to\_categorical(y\_train, num\_classes)

y\_test = keras.utils.to\_categorical(y\_test, num\_classes)

#show data

plot\_images(x\_train[:25], y\_train[:25])

#creating the DNN model

model = Sequential()

#applying layers to the model // hidden layer 추가

model.add(Conv2D(32, (3, 3), padding = 'same',activation = 'relu',

input\_shape=x\_train.shape[1:]))

model.add(Conv2D(32, (3, 3),activation='relu'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Dropout(0.25)) //overfitting 방지

model.add(Conv2D(64, kernel\_size=(3, 3), activation='relu',padding='same'))

model.add(Conv2D(64, (3, 3),activation='relu'))

model.add(MaxPooling2D(pool\_size=(2, 2)))

model.add(Dropout(0.25)) //overfitting 방지

model.add(Flatten())

model.add(Dense(512,activation='relu'))

model.add(Dropout(0.5)) //overfitting 방지

model.add(Dense(num\_classes,activation='softmax'))

#visualization the model

SVG(model\_to\_dot(model, show\_shapes=True).create(prog='dot', format='svg'))

#optimizer

optimizer = keras.optimizers.SGD(lr=learning\_rate)

#compiling the model

model.compile(loss=keras.losses.categorical\_crossentropy,

optimizer=optimizer,

metrics=['accuracy'])

#training the model

model.fit(x\_train, y\_train,

batch\_size=batch\_size,

epochs=epochs,

verbose=1,

validation\_data=(x\_test, y\_test))

#evaulating the model

score = model.evaluate(x\_test, y\_test, verbose=1)

print('Test loss:', score[0])

print('Test accuracy:', score[1])

#predictioin the model

y\_pred = model.predict(x\_test)

plot\_images(x=x\_test[:25], y\_true=y\_test[:25], y\_pred=y\_pred[:25])

y\_result = confusion\_matrix(y\_test.argmax(axis=1), y\_pred.argmax(axis=1))

sns.heatmap(pd.DataFrame(y\_result, range(10), range(10)), annot=True, fmt='g')