

## Homework #2

Due date: 10/17(목) 오후 11:59

CIFAR10 data set은 비행기, 자동차, 새 등 10개의 class로 labeling 되어 있는  $32 \times 32 \times 3$  픽셀의 컬러 이미지이다. 학습 데이터는 50,000개, 테스트 데이터는 10,000개로 구성되어 있다. CIFAR10 data set을 활용하여 컬러 이미지를 10개의 class로 분류할 수 있는 DNN 모델을 만들고자 한다.

**Problem 1.** 아래의 DNN 모델을 구현하라. 이 때 output layer의 activation function에는 적합한 함수를 지정하라.

Layer	Value
Flatten layer	Feature dimension: $32 \times 32 \times 3$
Hidden layer 1	Number of nodes: 100 Activation function: ReLU
Hidden layer 2	Number of nodes: 50 Activation function: ReLU
Output layer	Number of nodes: 10 Activation function: [알맞은 함수를 선택하라]

**Problem 2.** 수업 시간에 다룬 optimizer 중 최소 3가지와 두 가지의 batch size를 선택한 후 각 케이스에 대해 epoch에 따른 loss function을 그리고 분석하라. 그리고 test data를 사용하여 모델의 정확도를 확인하라. 각 모델의 hyperparameter와 epoch 횟수는 임의로 설정한다 (epoch는 최소 10 이상의 값을 설정하라). 단, 보고서에 모든 parameter 값을 명시해야 한다. 보고서에는 아래 6개 case에 대한 loss function 그래프와 test 결과가 포함되어 있어야 한다.

- (1) Optimizer 1, batch size 1
- (2) Optimizer 1, batch size 2
- (3) Optimizer 2, batch size 1
- (4) Optimizer 2, batch size 2
- (5) Optimizer 3, batch size 1
- (6) Optimizer 3, batch size 2

Test data를 사용한 모델 정확도 검증은 아래의 코드를 활용할 수 있다. (다른 방법도 사용 가능)

```
model.evaluate(X_test, Y_test, batch_size=100)
```

**Problem 3.** Problem 1의 각 layer에 dropout layer를 추가한 후 모델의 예측 성능이 개선되는지 확인하고 그 이유를 설명하라. Dropout 비율은 예측 성능이 개선되는 적당한 비율을 찾아야 한다.

## ※ 주의사항

1. 소스 코드와 보고서를 같이 제출해야 합니다. **둘 중 하나라도 미제출 시 0점 처리됩니다.**
2. Google colab으로 작성한 소스코드 제출 시 코드 URL을 보고서 댓글에 작성합니다. **제출 전 공개 여부 반드시 확인.**
3. 보고서는 워드 또는 한글 파일로 작성해야 합니다. (다른 문서 형식 사용 불가)
4. 보고서에는 실습 결과에 대한 충분한 설명이 포함되어 있어야 합니다. 결과에 대한 설명이 없을 경우 감점 발생합니다.
5. 보고서 파일명은 'hw2\_학번.hwp' 또는 'hw2\_학번.docx'을 사용합니다. (예시: hw2\_22010000.docx)  
**보고서 파일 명 및 양식 미준수 시 10점 감점됩니다.**
6. 보고서 및 소스 코드는 LMS의 과제 폴더에 제출합니다. **(제출기한: 10월 17일 목요일 오후 11시 59분, 시간 엄수)**
7. 과제 제출 기간을 엄수해야 하며, 제출 기간 이후의 제출은 허용하지 않습니다. (해당 과제 0점 처리)

## ※ library 및 data load

```
import tensorflow as tf
from tensorflow import keras
import matplotlib.pyplot as plt

(X_train, Y_train), (X_test, Y_test) = keras.datasets.cifar10.load_data()

X_train = X_train/ 255.0 # Normalize pixel values to the range [0, 1]
X_test = X_test/ 255.0 # Normalize pixel value to the range [0, 1]
```