# Machine Learning Course 2024 Spring: Experiment 3

April 21, 2024

## 1 Overview

In this experiment, you will implement neural networks on CIFAR-10, and deepen the knowledge and understanding of the networks through multiple tricks.

## 2 Dataset Description

The CIFAR-10 is labeled subsets of the 80 million tiny images dataset. It was collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The CIFAR-10 dataset consists of 60000 $32 \times 32$ colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. You can refer to the page for more information about the dataset.

## 3 Task Decription

| Layers | Hyper-parameters |
|---|---|
| Convolution 1 | Kernel size=(5, 5), stride=1, followed by ReLU |
| Maxpool 1 | Kernel size=(2, 2), stride=2 |
| Convolution 2 | Kernel size=(3, 3), stride=1, followed by ReLU |
| Maxpool 2 | Kernel=(2, 2), stride=2 |
| Fully Connected Layer 1 | Out Features=128, followed by ReLU |
| Fully Connected Layer 2 | Out Features=10, followed by ReLU |

Table 1: CNN architecture

## 3.1   Task 1

Implement a simple CNN model for CIFAR-10 classification. Your model should have the architecture in 1. Plot and observe the loss curve during training. Test your model on CIFAR-10 test set.

## 3.2   Task 2

In this task, you are required to implement a CNN for CIFAR-10 classification with data augmentation. Plot and observe the loss curve during training. Test your model on CIFAR-10 test set.

## 3.3   Task 3

In this task, you are required to implement a CNN for CIFAR-10 classification with dropout. Plot and observe the loss curve during training. Test your model on CIFAR-10 test set.

## 3.4   Task 4

In this task, you are required to implement LA loss functions on imbalanced CIFAR-10. Plot and observe the loss curve during training. Test your model on CIFAR-10 test set.

# 4   Requirements

- Implement learning algorithms with **Pytorch**.

- Record the procedure of experiments in **Jupyter Notebook**. It is recommended to use **Markdown** in Notebook to properly explain the procedure.

- If you find training extremely slow, you can reduce the number of epochs or the size of the dataset. This can be due to the limited computational resources provided by your personal computer. You can also use **Google Colab** to run your code on a GPU.

- Your submission includes:

  - source code: python notebook (.ipynb);
  - 'report.pdf' file : your report. Refer to the report template for the format.