# MLDL2 Assignment 3
# (Due Date: Dec 7, 11:59PM)

# - Self-Supervised Learning

1. Get started by cloning the code in **https://github.com/htdt/self-supervised**, and install some requirements such as ***PyTorch*** and others to proper use the code for this assignment. Please also install ***wandb*** to easily visualize your results (the code already includes wandb for logging results) and create an account to use it (visit https://wandb.ai/).

2. In this assignment, the goal is to run some experiments related to a self-supervised learning method based on "*contrastive learning*". In particular, we will use ***SimCLR*** with a ***CIFAR10*** dataset. When we use a contrastive learning setting, it is important to properly define positive samples and negative samples.

   a. We will compute the contrastive loss that is based on 'N-pair loss (InfoNCE)', and the number of negative samples 'N-1' is an important factor to improve the quality of representation what we want to learn. Here, the number 'N' is exactly the mini-batch size.

   → The first goal is to observe *the effect of the number of negative samples in contrastive learning*. Please run at least two different experiment settings to show it, and use wandb to visualize the results. When visualizing there result, please consider how can we compare the quality of representation. Please use the next command as the default setting and compare other experiment settings with this one.

   ```
   python train.py --dataset cifar10 --epoch 1000 --lr 3e-3 --emb 64 --method contrastive --bs 128
   ```

   b. The positive samples are another important factor for contrastive learning. By using noise injection into the model or using data augmentation, it is able to obtain some meaningful positive samples.

   → The second goal is to observe *the effect of the choice of positive samples*. For this goal, use the best experiment setting from the previous goal as the baseline setting, and modify only the data augmentation setting or newly implement Dropout (noise injection) in the model. That way, please show how the choice of positive samples can effect the learning.

   (For data augmentation, you can modify the code here or change configurations related to data augmentation. For noise injection, you can implement Dropout layers in the model head part)

3. For the upcoming assignment, please *store the learned model parameters*. We will use it for "*transfer learning*" to learn some downstream tasks.