Introduction to Deep Neural Networks (Spring 2021)

Final Project (100 Pts, Due Date: June 6)

Student ID ____________________

Name ____________________

Semi-supervised learning for image classification: The goal of our final project is to build a machine learning model for image classification, where a few data are only labeled and most of the data are unlabeled. Therefore, it is essential to utilize a large amount of unlabeled data to improve the accuracy of your model.

[Examples]

Label : Eyewear

~~Label : Sandals~~

~~Label : Watch~~

NOTE 1: **You should write your codes in 'EDIT HERE' signs.** It is not recommended to edit other parts.

NOTE 2: **Read the instruction in template codes VERY CAREFULLY.** We wrote detailed instructions, including the format of input and output data.

NOTE 3: We highly recommend implementing your code using PyTorch on Google Colab, which provides you with GPU environments, including various pre-installed packages. For your convenience, we also offer the guidelines and tutorials on Colab and PyTorch.

Kaggle & Colab Guide: Kaggle & Colab Manual 2021.pptx

Dataset:

# Train(labeled)/Train(unlabeled)/Test data: 5,000 / 35,551 / 10,000

Input: 32x32 image with RGB channels.

# Classes: 10 (Detailed information on labels is not provided.)

Competition (50 pts):

1. The competition is held with Kaggle in-class competition. Please make sure that you submit your answers on Kaggle. You can also find a link to the competition page on i-campus.
2. Prepare necessary files following guideline slides ("Kaggle & Colab Manual 2021.pptx").
   Write your answers on 'Answer.ipynb' **Link**.
3. You can use any name, but you should specify the name you used on Kaggle on top of your report to verify the score.
4. The score is measured with **classification accuracy** on test data.
5. **The public leaderboard you can see is calculated with 30% of the whole test data**. The final score includes the rest (70%) of the data. Therefore, the final rankings might be different. (This means you shouldn't overfit your model on 30% test data based on the public leaderboard.)
6. The competition score will be given based on the final standings.

Report (50 pts):

In addition to submission to Kaggle, you should **submit your code and report it to i-campus**. Write a report on your project with a **maximum of 3 pages.**

Final_Project_STUDENT_ID_Name.zip: compress 1) 'Answer.ipynb' and 2) Your document in pdf format.

Please follow the submission format. We will not be held responsible for any penalty caused by non-compliance.

The report does not have a specific format, but it should include the following:

1. Description of your method and reason for the choice (more than 1 page, most important)
2. Train and predict procedure (e.g., hyperparameter choice/tuning)
3. Any trial you made for performance
4. Final results on the public leaderboard (Screenshot of your leaderboard)

Rule:

1. You should only train your model with the given data. Any use of external data is prohibited.
2. You can write your code in 'Answer.ipynb' only.
3. Late submission on either Kaggle or the report will not be accepted.
4. Your submission to Kaggle is limited to 5 times a day.
5. Make sure you write your own codes, and do not copy other codes from anywhere and submit.

Tip: We introduce you to some simple methods you can try with the project.

Possible methods you can try:

1) **Data augmentation**: An image of ‘Sandals’ is still ‘Sandals’ even after cropping, rotating, flipping. Try to create more labeled images through various data augmentation.

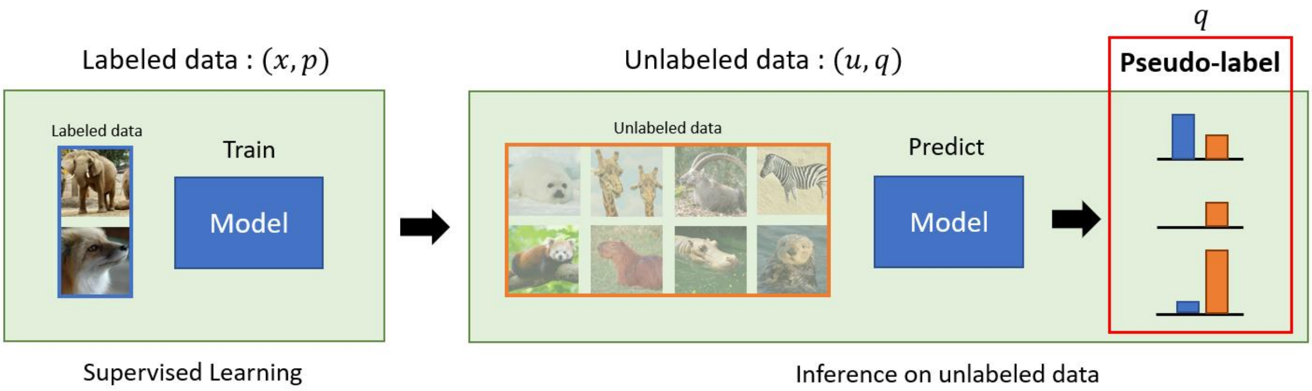
Original

Crop

Rotate

Horizontal Flip

2) **Pseudo-labeling:** Label the unlabeled data with the model trained with labeled data.



Steps of making pseudo-labels :

a) Train a model with labeled data

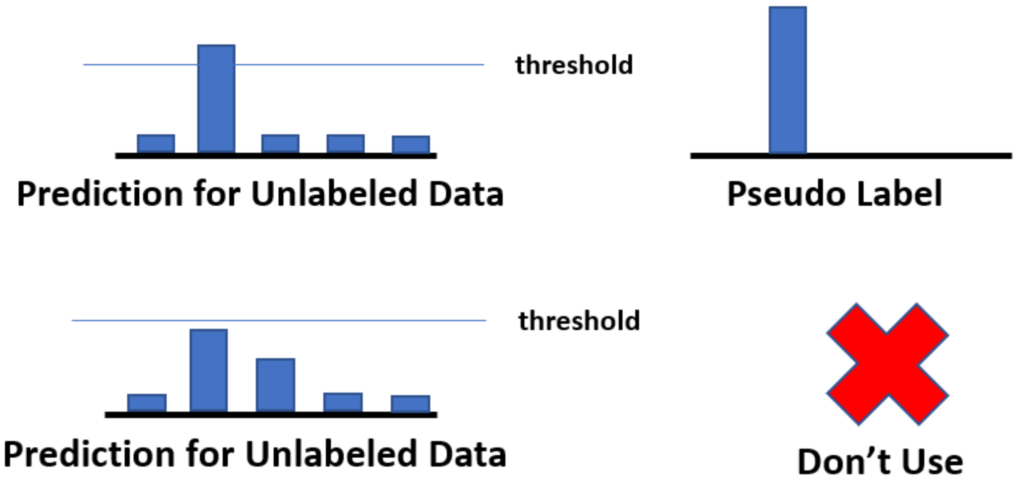
b) Make predictions on unlabeled data with the model.

c) Use the predictions of unlabeled data as the pseudo-label of its own image.

d) Train further with labeled + unlabeled data with pseudo-label

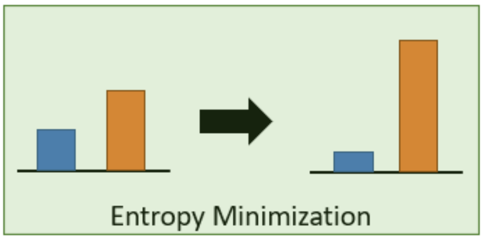
The pseudo-label related method you can try (it may or may not increase the performance):

- Select high confidence predictions: Only use pseudo-labels only if they have high confidence and throw it away if its confidence is lower than the threshold.



- Sharpening the pseudo-label: Most predictions have low confidence compared to the labels we use. So, sharpening the pseudo-label may enhance the performance.

$$\operatorname{Sharpen}(p, T)_i := p_i^{\frac{1}{T}} \bigg/ \sum_{j=1}^{L} p_j^{\frac{1}{T}}$$



e.g)

$$[0.4\ \ 0.6] \xrightarrow{\text{Sharpen (T=0.5)}} [0.307\ \ 0.693]$$

$$\text{Sharpen([0.4 0.6], 0.5)} = \left[\frac{0.4^2}{0.4^2+0.6^2}\ \ \frac{0.6^2}{0.4^2+0.6^2}\right]$$

$$= [0.307\ \ 0.693]$$

Here are articles in English/Korean to help you better understand the task.

https://amitness.com/2020/07/semi-supervised-learning/

https://blog.est.ai/2020/11/ssl/

You can also search several recent models on semi-supervised image classification with its ranks on commonly used image datasets.

https://paperswithcode.com/task/semi-supervised-image-classification