CSE 555 (Fall 2021): Project Description

Department of Computer Science and Engineering
University at Buffalo, The State University of New York

Instructor:	Shu Hu	**TAs:**	Lipisha Chaudhary, Enjamamul (Akib) Hoq, and Meng Ding
Email:	shuhu@buffalo.edu	**Emails:**	lipishan@buffalo.edu and ehoq@buffalo.edu mengding@buffalo.edu
Office hours:	Zoom links in Piazza	**Office hours:**	Zoom links in Piazza

1 Introduction

The goal of this project is to develop a robust learning system based on what you have learned in the pattern recognition class, which operates on a given dataset. Please feel free to confront and solve issues that are not covered in class. There is only one project (described below), which consists of 2 parts. For the first part, you are expected to solve several baselines, which is for you to get familiar with the whole pipeline and build necessary fundamentals. The second part should be your main focus, which is an extension to your baselines.

2 Collaboration

The project should be carried out by teams of 2, 3, or 4 students (no more than 4). We do expect that projects done with 3 and 4 students have more impressive write-up, results, and novelty.

3 Freedom

Each team will need to inform the corresponding TA (your team name, team member details) before 11:59 PM 09/10/2021. Teammates should be from the same class group. You will have significant freedom in designing what you will do for your baseline and extension. The techniques in terms of data pre-processing, what algorithms to use, training strategy, and evaluation are up to you. Don't be afraid to think outside of the box. There will be no strict answer to the project. You are welcome to come to the course TA's office hours for discussion any related questions.

4 Project Details

4.1 Project Description

Modern real-world large-scale datasets often have long-tailed label distributions. On these datasets, conventional machine learning models and deep neural networks have been found to perform poorly on less represented classes. This is particularly detrimental if the testing criterion places more emphasis on minority classes. For example, accuracy on a uniform label distribution or the minimum accuracy among all classes are examples of such criteria. These are common scenarios in many applications due to various practical concerns such as transferability to new domains, fairness, and so on.

On the other hand, traditional machine learning models and deep neural networks (DNNs) are often requiring large-scale datasets with clean label annotations for proper training. However, labeling large-scale datasets is a costly and error-prone process, and even high-quality datasets are likely to contain noisy (incorrect) labels. Therefore, training accurate traditional machine learning models and DNNs in the presence of noisy labels has become a task of great practical importance in machine learning and deep learning.

Dataset Type / Models	Balanced	Imbalanced (Original MNIST)	Balanced & Symmetric Noise	Balanced & Asymmetric Noise	Imbalanced & Symmetric Noise	Imbalanced & Asymmetric Noise
ML-1 (SVM)						
ML-2 (Logistic Regression)						
LDAM-DRW [3]						
SL [5]						
Proposed ML						
Proposed DL						

Table 1: Sample output table. ML represents machine learning. DL means deep learning.

4.2 Models

For the above problems, the goal for you is to explore some machine learning-based classification methods (such as SVM and Logistic Regression.) and deep learning-based baseline methods discussed in [3,5]. Based on these existing methods, You should design your own robust classifiers that can deal with a dataset that has imbalanced and noise problems. More specifically:

- Baselines:
  - At least two machine learning-based classification methods. For example, SVM and Logistic Regression. Since the dataset is a multi-class dataset, you should consider how to use these methods in a multi-class task.
  - You will implement two deep learning-based baseline methods described by the authors in [3,5]. In paper [3], the authors proposed the LDAM-DRW method for learning a robust model that only can handle the imbalanced dataset. On the other hand, in paper [5], the authors proposed the SL method for learning robust models that only can handle the datasets that contain noisy (symmetric noise and asymmetric Noise) data points. The codes of these two methods can be found in [1,2]
- Extension: After implementation of the baselines, you should understand how to process the data, generate the data that you need, and use data to train a model. Based on your understanding, you are expected to improve or come up with a different learning structure that can work on an imbalanced and noisy dataset simultaneously. Note that even the model from [5] can work on the imbalanced and noisy dataset, you should propose a new model that differs from it. Some necessary steps are as follows:
  - You should generate a balanced data set, some noisy (symmetric and asymmetric) datasets, and some imbalanced-noisy datasets.
  - Try to design two robust models. The first one is based on your selected machine learning models. The second one should be designed based on [3,5].

4.3 Dataset

- MNIST dataset [4]. Training set size 60,000 and testing set size 10,000.

4.4 Evaluations

You can select some evaluation metrics such as accuracy, precision, recall. Display a chart like the one shown in Table 1 in the report to compare all models with various combinations and explain the results.

5 General Requirements & Notes

- It is better to go through relevant literature and resources listed above first and then implement the baselines before doing the extension part.
- Please clearly state how to generate the different datasets in your report and how to design your models.
- Be careful to keep your final testing set uncorrupted, by setting it aside at the beginning. In other words, the testing data set cannot be modified.
- You can use CPU or GPU (Google Colab) to run your code but you should give details about the computation environment in your final report.

6 Deliverables

Your final report is required to be between 5-6 pages using the ICCV 2021 provided template. You can add other necessary supplementary materials (not counted toward the report page limit). You will submit your report as a PDF file, your supplementary material as a separate PDF or ZIP file, and your source code as another ZIP file. All will be submitted to UBlearns. We will provide more submission instructions as the deadline nears. Examples of components to put in your report and supplementary material are listed below:

Report

- Title, Author(s)
- Abstract
- Introduction
- Description of the Baseline methods and how your proposed methods improve over those baselines
- Experiments and Results
- Conclusion
- Contributions: Please include a section that describes what each team member has contributed to the project and it is not counted toward your report page limit.
- References: Do not miss any references and it is not counted toward your report page limit

Supplementary Material

- More analysis
- More experiments

7 Grading Criteria

Since the project is open-ended, and the work you will do is largely up to you. Grading criteria will include:

- understanding and interpretation (of approach, algorithms used, and results) of the problem which will be evaluated from your write up (weight: 20%).
- Preparedness and clarity in Presentation (weight: 20%).
- Code implementation (weight: 25%) ;

- Final performance and significance of the work (weight: 25%).
- Novelty of the new idea introduced to improve the baseline (weight: 10%).

The final presentation will be in class, when you would be asked to present your work. Some requirements for presentation as follows,

- It should consist of less than 8 slides.
- Use bullet points instead of full sentences.
- Each team should submit one 8 min video-recorded presentation (just send a UB Box link to TA). According to the submitted videos, we will select 5 best teams from each group (Total: 10 teams). Each of the 10 teams will present their works in class. The presentation session will include Q&A. 12 min for presentation and 3 min for Q&A.

8 Honor Code

Your report and code should reflect your own work done specifically for this class. You may consult any materials like papers, books, or any publicly available resources for implementation, code, and ideas that you might want to use as a basis for your projects, as long as your clearly quote and reference in your write-up and code.

References

[1] Code 1. https://github.com/kaidic/LDAM-DRW. 2

[2] Code 2. https://github.com/YisenWang/symmetric_cross_entropy_for_noisy_labels. 2

[3] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 1567–1578, 2019. 2

[4] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2

[5] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330, 2019. 2