CSCE 421 – Fall 2020
Final Project – Dealing with Imbalance in Computer Vision

Guidelines

In image recognition, there's usually different kinds of imbalance issues, where majority class can have one or more orders of magnitude more instances than the minority class. The situation is worsened further in multi-class classification problem, which is often found in CV task. The ratio of training instances among different classes can be drastically different. However, it can be coarsely categorized in two types of imbalance, i.e. step imbalance and linear imbalance [1].

Project Implementation

You will take the MNIST Dataset – uploaded for you – and create digit recognition models. You will provide code that properly handles data imbalance to accurately model the hand-writing digits. In particular, you will be provided a training set and a testing set, and we will evaluate your model performance on a held out testing set.

Project Details

First, in step imbalance, within majority classes and minority classes the instances are roughly the same, but the difference between them majority and minority can be large. Second, in linear imbalance, the different classes have different instances but the number can be coarsely interpolated linearly from smallest class to biggest class.

These two imbalances are idealized from different real-world problems. The algorithm needed should be able to tackle both of them. In terms of implementation, it can be two sets of networks or one that can distinguish the two.

The evaluation metric to be used is AUROC since the traditional metric might be biased to majority class. In multi-class scenario, the 'macro' AUROC is used, where each metric is obtained independently for each class and then averaged among them all.

You will implement a Convolutional Neural Network technique, a manual feature extraction technique with a higher-order model (either SVM, Random Forest, or XGBoost, your choice) and linear model (logistic regression).

We will evaluate on a hidden test set

In the project, test set will test a variety of imbalance scenarios, so we urge you to take your training data and create additional test sets beyond the one provided to you.

Project Code + Report (due November 23rd 11:59 pm)
You will provide your code (Jupyter notebook)

You will also write an individual project report (due November 23rd 11:59 pm)
This will describe all the details about your problem, approach, and results. The report should follow the standard IEEE conference **double column format** and include at least 4 pages. The report will contain the following parts:

- **Introduction and Literature Review:** Briefly describe the problem and the dataset (at least 5 citations)
- **Methods Differences:** describe your implemented solutions in terms of features, machine learning methods, system evaluation techniques, i.e. system settings, implementation details, data, etc. Explain the implementation differences for CNN vs. Nonlinear model (SVM, RF, or XGBoost), vs. LR
- **Test Results:** describe and explain (interpret) your testing for confidence in final performance
- **Comparison of Techniques:** Explain the result differences for CNN vs. Nonlinear (SVM, RF, or XGBoost) vs. LR
- **Conclusions:** main take-away messages and future work

[1] Buda, Mateusz, Atsuto Maki, and Maciej A. Mazurowski. "A systematic study of the class imbalance problem in convolutional neural networks." Neural Networks 106 (2018): 249-259