Boyu Jiang
20697921

MATH 6380O Project 2

Introduction

Anomaly detection is a common machine learning problem, the main challenge of this problem being the imbalanced data between the normal (good) class and the abnormal (bad) class. When facing an extremely unbalanced set, the network tend to overfit to the class with abundant data and not well trained for the class with limited data. There are two main reasons for this problem. Firstly, the reward for getting the normal class right is too big because of its high percentage in the dataset. Secondly, the fact that there is a lot more data in the normal class produce a setup that is similar to training the normal model a lot more epochs compared to the abnormal class. These reasons combined, forms the major obstacle in anomaly detection models.

In the problem setup we are facing, 27000 normal data and 3000 abnormal data are given, making there a 9 to 1 normal to abnormal ratio. In this project, I will discuss methods to belance for the abnormal class and improvement it produces.

Method

The first method attempted is to simply balance the data by using adding a sampler. The sampler assign the possibility of each data being selected at training, and the training data with abnormal label is 9 times more likely to be selected than data with normal label. This produces a balanced training set with 50% for data belongs to each class. However, under this setup, because the train data was used multiple times in every epoch, it decreases the generalizability of the model.

To improve on this shortcoming, data augmentation is needed to make the training data different each time it is used. In the implementation, I uses random flip and random affine to augment the training dataset.

The image model is built following ResNet50 architecture. The first convolution layer configured as 1 input channel to fit the grayscale image. The output layer was configured to have two classes to represent the normal and abnormal data.

Result

The training configuration are: {optimizer: Adam, learning_rate: 1e-3, batch_size: 128; epoch: 50}.

The result we got are as following, with unbalanced data being very difficult to train over the base rate of guessing all normal of 90%. With balanced data, the base rate it 50% as the distribution of normal and abnormal data are 50-50. The result converge quickly on training and validation, however, the final validation is still quite high and the testset accuracy was far from what is in training and validation. This is possibly due to overfitting to the training set, especially for the abnormal class. By using simple data augmentation, the result became much better, and we can see the test result being fairly similar to the training and validation set.

	Unbalanced	Balanced	Balanced with Augmentation
Train	90.82%	93.07%	96.51%
Validate	88.31%	91.28%	95.38%
Test	67.37%	78.42%	91.14%

Table 1: train, validate, and test result for model with different data loader.

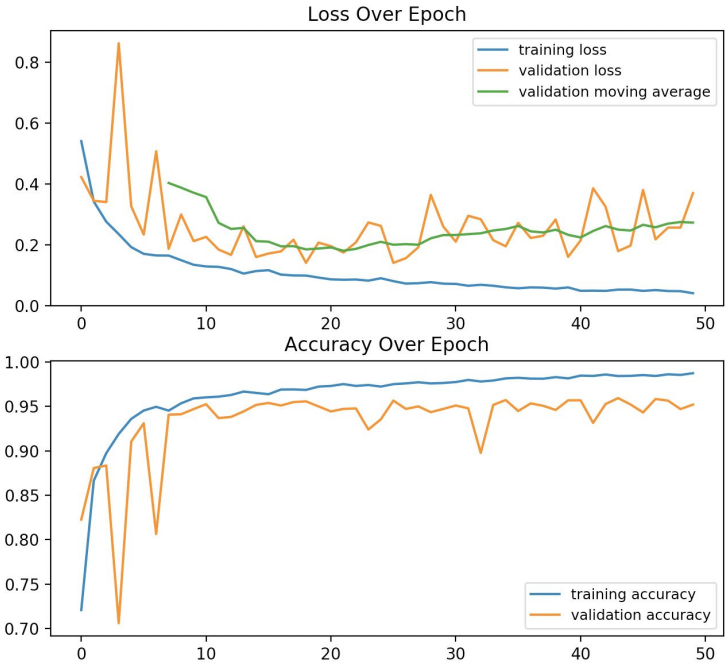
Future Work

In the work presented, the data augmentation method is simple, and we see the training loss decrease pretty fast in the initial epochs. This means we can increase the amount of data augmentation. More advanced method can be using generative network to produce better data augmentation.

A though is to use ACGAN to generate image with class label, and use it along with the original data to produce a balanced dataset for the training.

Code

https://github.com/Boyu1997/MATH-6380O/tree/master/Project-2



Plot 1: Training loss and accuracy for the best model.