Anomaly Detection on FashionMNIST

Davide Esposito Pelella, 1886977

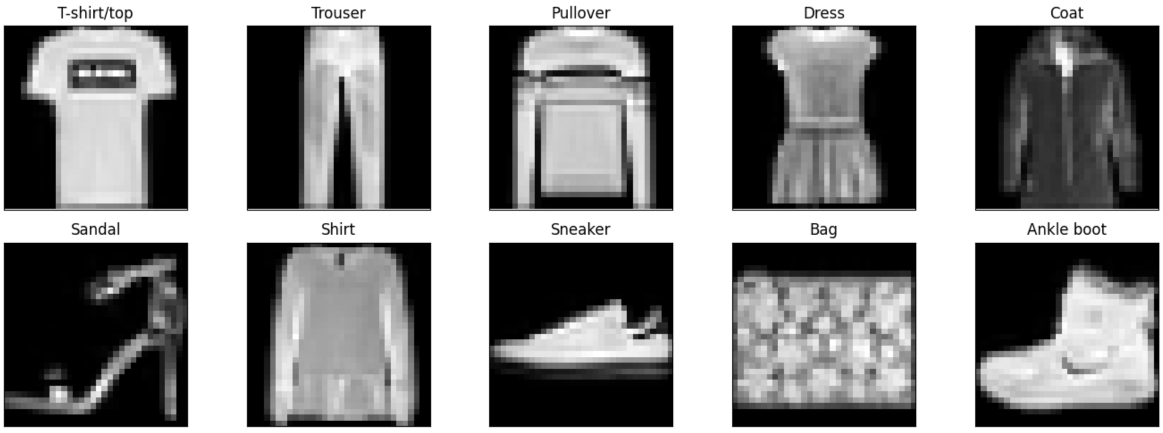
Introduction

1.1 Project Overview

The report which follows is the result of the project of the course of Deep Learning. The aim of this work is focused on the Anomaly Detection in Imbalance Problems, dealing with gap of samples and classes distributions, based on the paper: [1] *GAN-based Anomaly Detection in Imbalance Problems.* In particular this work implement the SOTA architecture explored in the paper's research and 2 more baseline to compare the results.

1.2 Dataset

The Dataset used is the FashionMNIST [2] , a dataset of Zalando's article images of 10 classes, with a total of 60,000 train samples and 10,000 test samples. Each sample is a 28x28 grayscale images (784 px) containing only 1 class. The dataset in particular is perfectly balanced, and every class has the same number of elements for training and test.



Dataset

1.3 Data Augmentation

For the training was applied some Data Augmentation techniques to the FashionMNIST dataset, in particular the images were vertically and horizontally flipped, randomly cropped 0~2 pixels from the boundary, randomly rotated of 90, 180, or 270 degrees and resized to 32x32 dimension.

1.4 Preprocessing and Unbalance

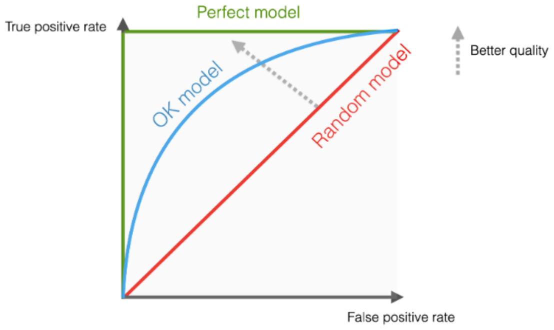
The dataset in the proposed implementation is loaded by using a specific realized class called *AnomalyDetectionDataModule*, which load the dataset and provide the training and test dataloaders. The DataModule is realized in a flexible way, allowing to perform a training/test also only on a subset of the available data in order to perform tests and to deal with the limitations in computational power. Furthermore the model will need to deal with the imbalance problems typical of anomaly detection.

Class imbalance is a well known problem for which, usually, there is a class imbalance between the normal and the anomal class. For this specific task with the FashionMNIST, each class is once considered as the 'normal class' at time, and the remaining 9 as 'anomal class', in this way, 10 experiments are performed. Due to samples availability this division lead to an unbalanced situation, to deal with this problem is performed a *sampling* of thesample from each class recognized as 'anomal' in order to get the same number of the 2 classes, both now with 6000 samples.

Furthermore, if the data are sampled randomly, without considering their particular distribution, the resulting dataset won't be balanced in the distribution and representation of starting dataset. To solve this condition was applied a *k-means clustering* approach, in order to sample the same number of data within each cluster of each group, now then considering the distribution of the samples. This is obtained by using the *kmeans_sampling* function in the DataModule, which performs both the k-means clustering and sampling according to these requirements.

1.5 Metric

In order to evaluate the performance of the models on the dataset during development, the main metric was the Area Under the Receiver-Operating Characteristics (ROCAUC). This metric, indicating the area under the ROC curve, sums up how well a model is able to produce relative scores to discriminate between positive or negative instances, which in our case is represented by the normal and anomalous samples of interest. It ranges from 0 (bad) to 1 (perfect), and 0.5 indicates the performance of random guessing.



ROC AUC score

1.6 Limitations

For the implementation was used pytorch lightning and all the training and test phases were performed by using the hardware resources provided by the free version of Google Colab. Unfortunately the free version of Google Colab has some limitation in computational power and resource continuative disponibility. These limitations have negatively impacted the overall performances of the work, in particular the total number of samples that could be used for training and the maximum time for for which was possible to train and test the models. The Google Colab policy indeed, after some hours of use will progressively limit the resources provided to the user, forcing into using a subset of the dataset and limit the train not allowing to perform the same training showed in the paper.

Models

2.1 Baseline 1: Random Guessing

The first choosen model as baseline is a random guessing type. The architecture is simple and straightforward, This model represent so the behaviour of 0.5 of the ROC AUC score metric.

2.2 Baseline 2: Simple CNN

The

2.3 Baseline 2: Autoencoder

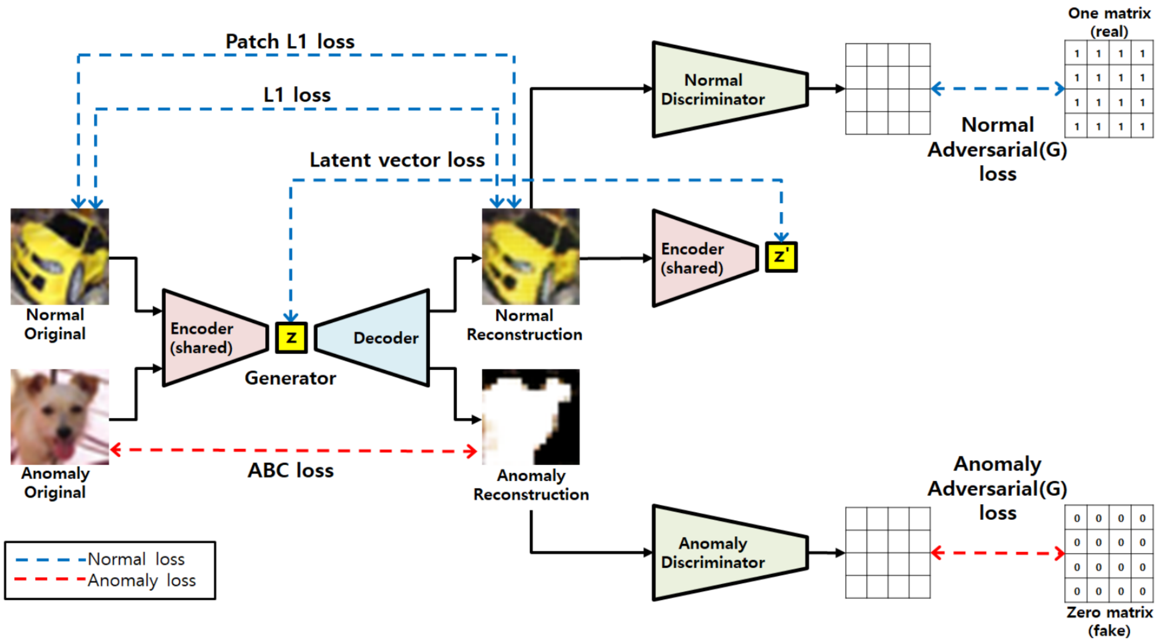
The

2.4 Baseline 3: GAN-based

The last implemented model is a novel GAN-based anomaly detection model, the state-of-art for the Benchmark datasets for the AUROC(%). The network architecture is based on a GAN structure, consisting of a generator and a discriminator.

- *Generator* = It is a modified U-Net structure, in the form of an autoencoder. It consistsof four $4x4$ convolutions with stride 2, then followed by 4 transposed $4x4$ transposed convolutions;
- *Discriminator* = The discriminator is a more simple convolutional network, consisting of three $4x4$ convoluitiond with stride 2 then followed by two $4x4$ convolutions with stride 1.

During the training phase the Discriminator in particular is composed of 2 identical structures which separately process the input data when the class is the normal or the anomal one. In this way the model will learn to minimize reconstruction error when normal data is provided to Generator, while will try to maximize the reconstruction error when input is anomal.



GANarchitecture

As in the paper, for the training where used 6 types of loss functions to train the generator and 2 for the discriminators (1 for each), used and combined with a weighted summation as in the research mentioned. The double discriminator allow to solve the discriminator distributional bias, because the generator will be trained to output 1 from normal data and 0 from anomaly, while, if it was trained with only 1 discriminator, the model will be trained to only classify well normal images.

The model was trained for 20 epochs with Adam optimizer, a batch size of 1, a learning rate of 0,002

Bibliography

[1] Junbong Kim, Kwanghee Jeong, Hyomin Choi, and Kisung Seo (2020). *GAN-based Anomaly Detection in Imbalance Problems*.

[2] FashionMNIST dataset. *https://github.com/zalandoresearch/fashion-mnist*.

[3] Area Under the Receiver-Operating Characteristics. *https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html*.