ECE 157A/272A - Homework 4

ECE 157/272 TAs

Fall 2022

Due date: Wednesday, November 23, 2022 at 11:59 PM

Introduction

In homework 3, you were introduced to neural network for classification. Neural network can be used to build a classifier when we have sufficient samples and labels for each pass/fail class. But, this might not be the case in the manufacturing setting. We will often see a large amount of good samples, but fail samples rarely occur. To make the matter worse, the fail patterns comes in all shapes and forms because there are many failing conditions and some edge-case fails are unpredictable. The question now is... With limited fail samples and large amount of good samples, are we able to learn the features associated with good samples and use them to detect fail samples when they occur?

The answer is "Yes." We will introduce you to the autoencoder neural network where it is trained for anomaly detection. Anomaly detection model are trained with good samples only and be able to detect fail samples at inference stage. The idea is this... the autoencoder learns the features and representations of good samples so it can use them to reconstruct the good samples. But, when a bad sample is presented, the autoencoder has no information of the fail features and unable to reconstruct it, which give us an indicator of a fail sample appearing.

Dataset

- **MNIST** dataset is a handwritten digit dataset. The dataset contains grayscale 28x28 images of single digit from 0 to 9. This dataset is a common dataset machine learning scientist used to test out their ideas. For anomaly detection, we will be using the 1s digit as our in-class samples and 9s digit as our out-of-class samples.
- **MVTEC** dataset[1] is a real-world dataset for unsupervised anomaly detection. It contains color 1024x1024 images of various objects with and without defects. In this homework, we will be using the transistor subset of the dataset for training anomaly detection model.

Tasks

MNIST

We will start by building a simple autoencoder model to learn the 1s digit as the in-class samples and use it to detect 9s digit which is the out-of-class samples.

- The train and validation loop is very similar to the previous homework. Note, in this homework, the image data themselves are the labels of the autoencoder model since we are trying to reconstruct the samples in the process of learning.
- We will implement a function to reconstruct the validation images. This function will be used in the `train_loop` to examine how well the autoencoder model has learned the features associated with the good samples.
- With the above setup, we will build and train an autoencoder to learn the in-class 1s digit. **Show the model architecture along with the layer parameters. Show a plot of the training and validation reconstuction loss vs epochs. Show the reconstruction images of one validation batch for every five epochs.**
- Now that the model is trained. We want to examine how well it reconstructs in-class 1s digit and how well it detects anomaly for the out-of-class 9s digit.
  - Use the model to reconstruct all validation samples.
  - Retrieve the original validation samples from the validation data loader. Compute the absolute difference map between the reconstructed validation samples and the original validation samples.
  - Using the difference map to compute the sum of difference for each validation sample.
  - We will now create a data frame from the reconstructed images, original images, difference maps, and the sum of difference. The data frame elements are sorted in descending order by the sum of difference.
  - We will plot the top ten erroneous reconstructed samples (based on sum of difference). **Show the top ten plots, each containing the reconstructed image, original image, and the difference map with the sum of difference as the title.**
  - **Repeat the above examination sub-steps (the ones with with -s for points) and show the results** for the 9s digit to investigate the reconstructed samples and the differences. **Compare the reconstruction performance between the 1s and 9s digits. Visually, which reconstructed digit looks more like the original sample? Could you tell what the digit is just by looking at the reconstructed image?**
- We can **plot the histogram of the 1s digit sum of difference (orange) with the 9s digit sum of difference (blue)** to see the distribution of reconstruction error. An ideal anomaly detector model should perfectly reconstruct the in-class 1s digit, (putting all 1s digits on the left side of the plot,) while poorly reconstructing the out-of-class 9s digit, (putting all 9s digits on the right side of the plot.) **Does your plot resemble this behavior? If you have to set a sum of difference threshold to separate the in-class 1s samples from the out-of-class 9s samples, what value will you draw the threshold at and why?**

MVTEC - Transistor

We will be building an autoencoder to learn the perfectly placed and non-defective transistors as the in-class samples and use the 10 misplaced placed or missing transistor as the out-of-class samples. We will repeat the steps in the MNIST task and train a model that can separate at least 7 of the out-of-class samples from the in-class validation samples. Note, the images are resize to 128x128 in the example code.

- Build and train an autoencoder to learn the in-class samples. **Show the model architecture along with the layer parameters. Show a plot of the training and validation reconstuction loss vs epochs. Show the reconstruction images of one validation batch for every five epochs.**
- Now that the model is trained. We want to examine how well it reconstructs in-class transistors and how well it detects anomaly for the out-of-class transistors.
  - Use the model to reconstruct all validation samples.
  - Retrieve the original validation samples from the validation data loader. Compute the absolute difference map between the reconstructed validation samples and the original validation samples.
  - Using the difference map to compute the sum of difference for each validation sample.
  - We will now create a data frame from the reconstructed images, original images, difference maps, and the sum of difference. The data frame elements are sorted in descending order by the sum of difference.
  - We will plot the top ten erroneous reconstructed samples. **Show the ten plots, each containing the reconstructed image, original image, and the difference map with the sum of difference as the title.**
  - **Repeat the above examination substeps (with -s for points) for the out-of-class transistors** to investigate the reconstructed samples and the differences. **Compare the reconstruction performance between the in-class and out-of-class transistors. Visually, which class has reconstructed samples that looks more like the original sample? Could you tell how the transistor looks and how it is placed just by looking at the reconstructed image?**
- **Plot the histogram of the in-class transistor sum of difference (orange) with the out-of-class transistor sum of difference (blue)** to see the distribution of reconstruction error. For this problem, we want the anomaly detector model to **perfectly separate at least seven of the out-of-class samples from the rest of the in-class samples (rebuild and retrain another model if this requirement is not satisfied)**. **If you have to set a sum of difference threshold to separate the in-class samples from the out-of-class samples, what value will you draw the threshold at?**

What to turn in

- For the report, make *slides* that thoughtfully answer every question and show the results in the Tasks section (highlighted in bold). Export and submit the slides as a PDF document.

References

[1] Paul Bergmann et al. “MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 9584–9592. DOI: 10.1109/CVPR.2019.00982.