Anomaly Detection with Auto-encoder

Yuncheng Yuan

January 19, 2023

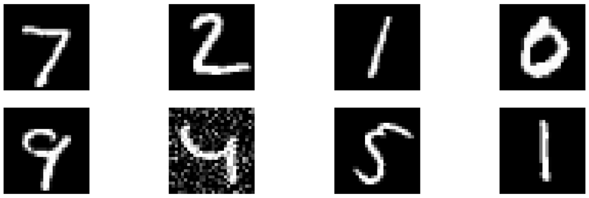
1 Questions

In this section, an overview of requirements and questions to be answered for your report is provided.

2 Some examples to get started

Simply use the section and subsection commands, as in this example document! With Overleaf, all the formatting and numbering is handled automatically according to the template you've chosen. If you're using Rich Text mode, you can also create new section and subsections via the buttons in the editor toolbar.

1. Take a subset (10 percentage) of the MNIST test dataset and add excessive noise to corrupt images. Then, visualize four samples for original and corrupted ones. Note: The type of noise is optional. Be sure that the amount of noise is enough for finding outliers (anomalies) in question 8 with the help of the threshold value.



2. Train your autoencoder architecture by using the MNIST training dataset. Compare the autoencoder output generated from corrupted input images with original and corrupted images. Comment on the results. What did you expect, and what did you observe? What are the reasons for similarities and differences?

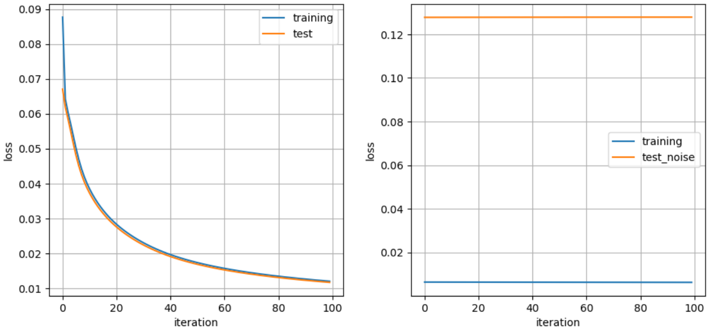


Figure 1: Output image

As the figure 1, left one is trained with original training and testing dataset, right one is trained with original training and corrupted testing dataset. It is totally different for the original image and corrupted image. The loss function result is totally different. The autoencoder will decrease the feature of input image. But for human being, it's hard to interpret the output of the autoencoder. Therefore, it's not likely to see the similarities with the input picture.

3. What happens if your dataset includes corrupted images not only in the testing dataset but also in the training dataset? How does it affect the performance of the model?

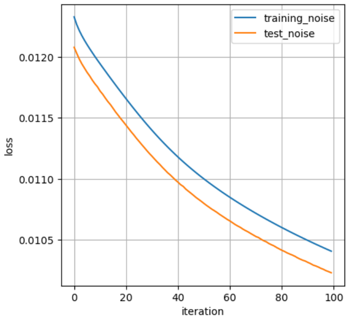


Figure 2: Loss distribution

It is obvious that the model cannot speify the correct image because the training dataset contain the noise. Model cannot identify the noise. Model will identify the corrupted image as original image. In the output result, the model will also resconsturct the noise.

4. Change the loss function of MSE with SSIM loss function. Explain the difference between them and compare the results. Please keep your loss function as MSE for the following questions.

   In general, two of them are used to evaluate the similarity of two images.

   MSE is a commonly used loss function for regression tasks, and it measures the average squared difference between the predicted and actual values.

   SSIM is based on the idea that the human visual system is more sensitive to changes in structure than changes in pixel intensity. It is a quality index that ranges between -1 and 1, with higher values indicating more similarity between images.

   After comparing these two loss function result, there is no too much difference between them.

5. Implement more complex autoencoder architecture to obtain better results. Compare the results with the first architecture you implemented and comment on the results.

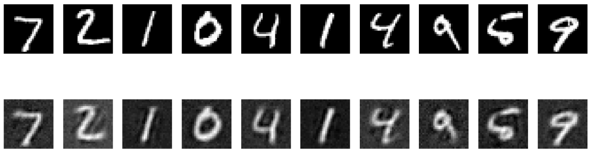


Figure 3: Simple Model

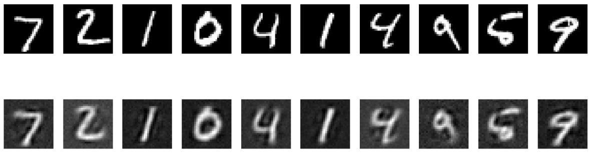


Figure 4: Complex Model

The image of Complex model is more clear than image of simple model but do not have too much difference.

6. Implement the convolution autoencoder model. Compare your results with the basic and complex architectures you implemented and comment on the results.

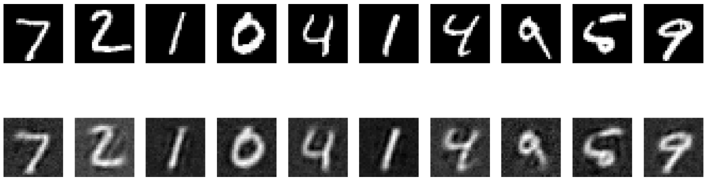


Figure 5: Simple model

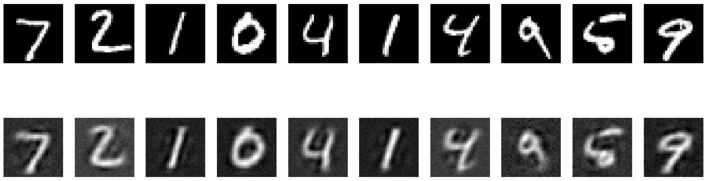


Figure 6: Complex model

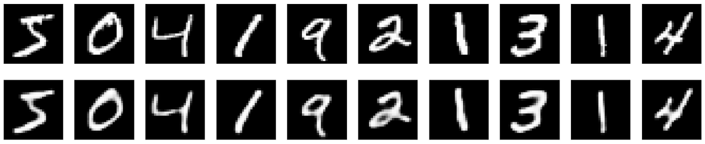


Figure 7: convolution model

Form Figure 5, Figure 6 and Figure 7, the convolution model have most clear edge compared to the simple model and complex model. As we can ss, all of these three model reconstructed the image successfully. we can see that the images of the convolution model have the best clear boundary and keep more details of image.

7. Choose the best implementation of them by explaining the reason for that decision. Plot the training and validation loss functions of your architecture to epoch numbers. What is over-fitting? How can one detect and avoid over-fitting?

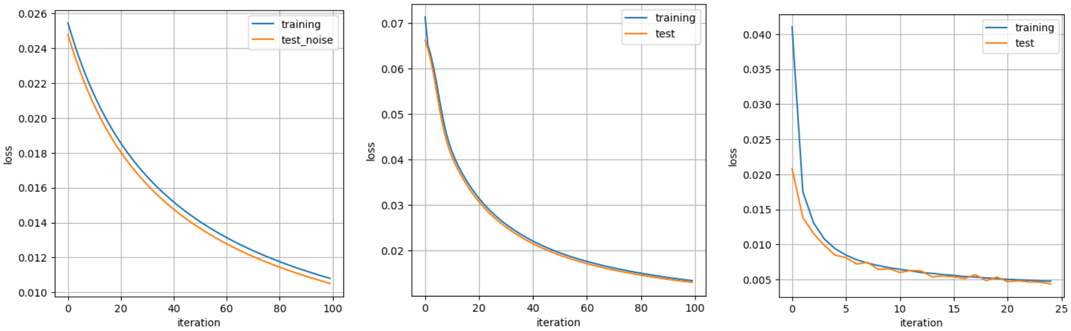


Figure 8: Simple loss

The best one is the convolution model. Overfitting means that the model perfectly perform on training dataset but have a bad result on testing dataset. There is a huge gap between the training dataset and testing dataset. To avoid overfitting, we can use the dropout method in training dataset.

8. Decide a threshold value in reconstruction error for detecting outliers (corrupted images) by drawing the loss distribution. This threshold value needs to be used to determine whether the input sample is an anomaly or not. Then, explain why you decided on that threshold value by comparing it with other threshold values.

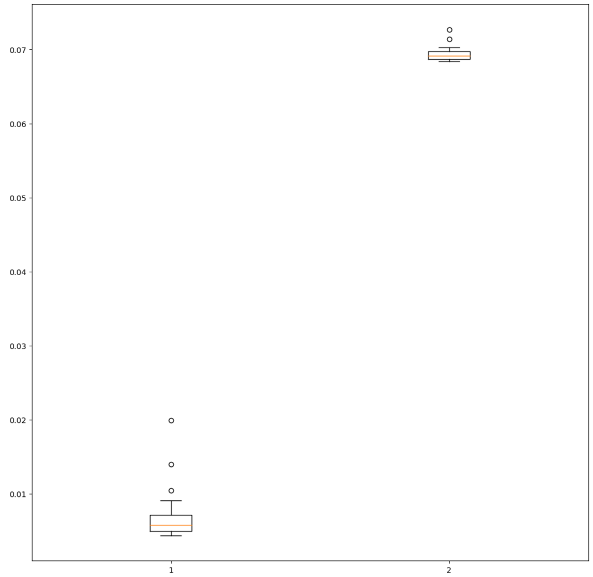


Figure 9: Loss distribution

In this image, 0 means the loss of original image and 1 means the loss of corrupted image. In this image, we could see that the threshold should be at 0.04. If we set the value too high like 0.07, we will not able to change identify the different of original image and corrupted image. it will be the same situation if we set the value to low like 0.01.

9. Provide true positives(TP), true negatives(TN), false positives(FP), and false negatives(FN) values for the threshold that you decided in the previous question. Plot the confusion matrix.

   FP = 148, TN = 7923, FN = 83, TP = 1844

10. Calculate precision, recall, and F1 score. Explain in detail how you can improve the F1 score. What is the effect of reconstruction error threshold value for precision and recall?

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 0.9767$$

$$Precision = \frac{TP}{TP + FP} = 0.9257$$

$$Recall = \frac{TP}{TP + FN} = 0.9567$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} = 0.941$$

11. What are the ROC curve and AUC? What is the difference between them? How can you use them to improve your model performance?

    The ROC curve is a graphical representation of the performance of a binary classifier as the threshold for classifying a sample is varied. The x-axis represents the false positive rate and the y-axis represents the true positive rate, also known as sensitivity or recall.

    The AUC curve is a metric that summarizes the performance of a binary classifier by measuring the area under the ROC curve. AUC ranges between 0 and 1, with a value of 1 indicating a perfect classifier and a value of 0.5 indicating a classifier no better than random guessing.

    The ROC curve and AUC can be used to improve model performance by comparing the performance of different models and selecting the one with the highest AUC. The ROC curve can also help to identify the optimal threshold for classifying samples based on the trade-off between the false positive rate and true positive rate.

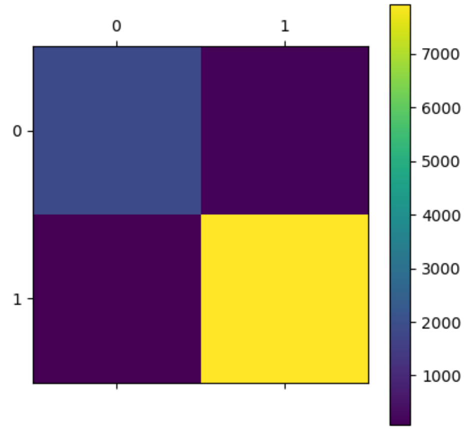


Figure 10: Confusion Matrix.

12. What is the false alarm rate? Calculate the false alarm rate for this problem. Explain in detail the importance of false alarm rate for anomaly detection problems and what should be the ideal value for that.

$$FAR = \frac{FP}{FP + TN} = 0.0183$$

The False Alarm Rate we get in the experiment is 0.0183. In ideal situation, we hope that the false alarm rate equal to zero. In this situation, we need to make FP = 0, which means the model not sensitive to the anomaly.

13. What are the cons of autoencoder structure? What could be another solution for the outlier detection problem? Explain it in detail.

Autoencoders are sensitive to the choice of architecture and hyperparameters. Choosing the wrong architecture or hyperparameters can result in poor outlier detection performance.

The architecture of the autoencoder, such as the number of layers and the number of neurons per layer, can affect the ability of the autoencoder to accurately capture the underlying structure of the data. Similarly, the choice of hyperparameters, such as the learning rate and the regularization strength, can also affect the performance of the autoencoder.