Deep Learning Homework #6

Indra Imanuel Gunawan / 20195118

No. 2 – Anomaly Detection using Autoencoder

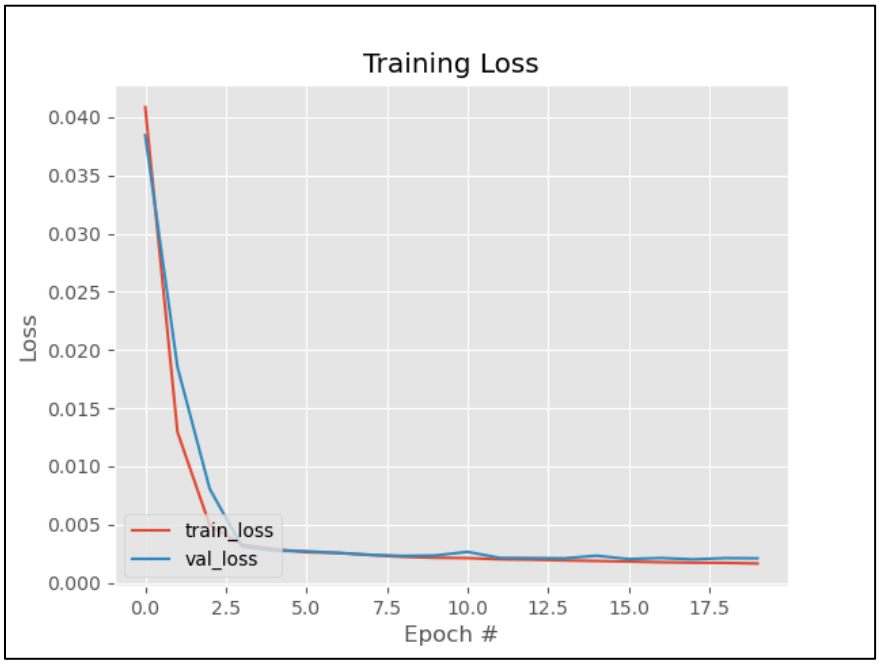
The code for this experiment can be found in my github repository “DL_Homework6”, under the folder “No2”

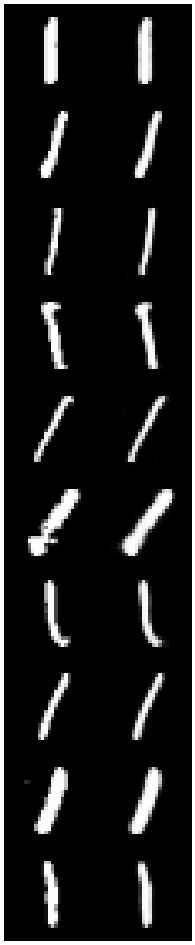
The experiment done in this question is basically running autoencoder to find anomalies/outliers in a dataset. The dataset chosen for this experiment is the MNIST dataset. The reconstruction loss of the autoencoder is calculated using the MSE (Mean Squared Error) loss. By anomalies, it means that the reconstructed images that have a high MSE with respect to the original images. If it has a high MSE, it means that the reconstructed image is very different than the original image (possibly a different label with the original, although from the result, it may not always be the case).

To run the code, type in “python train_unsupervised_autoencoder.py --dataset output/images.pickle --model output/autoencoder.model” in the terminal. This will run the script for creating the dataset with anomalies in it (saved under the name “images.pickle” in the folder “output”), create the autoencoder model (saved under the name “autoencoder.model” in the folder “output”), and train that model also (training is done for 20 epochs).

```
(deeplearningenv) D:\Master Studies\2nd Semester\Deep Learning\HW6\No 2\autoencoder-anomaly-detecti
utput/autoencoder.model
[INFO] loading MNIST dataset...
[INFO] creating unsupervised dataset...
[INFO] building autoencoder...
2020-06-16 21:26:34.752772: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports
Train on 5447 samples, validate on 1362 samples
Epoch 1/20
5447/5447 [==============================] - 9s 2ms/sample - loss: 0.0409 - val_loss: 0.0385
Epoch 2/20
5447/5447 [==============================] - 7s 1ms/sample - loss: 0.0129 - val_loss: 0.0185
Epoch 3/20
5447/5447 [==============================] - 8s 1ms/sample - loss: 0.0050 - val_loss: 0.0081
```

Here are the training history of the model and example of the reconstructed images from the model.



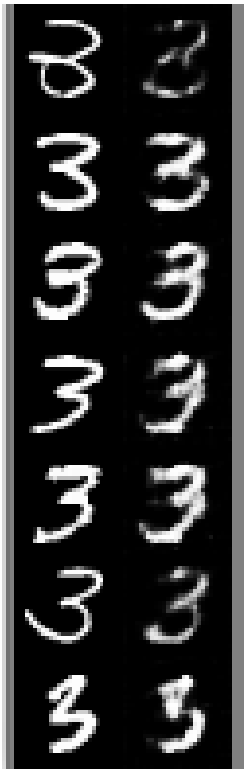


Comparison of the reconstructed images (left), and the original images (right)

As we can see, the trained autoencoder model is able to reconstruct the image pretty well. Now we want to see if there are any anomalies in which the autoencoder didn't reconstruct the image well enough. To do so, type in "python find_anomalies.py --dataset output/images.pickle --model output/autoencoder.model". This will run the script for finding the anomalies in the images.pickle dataset using the autoencoder.model model. The output of this will be the MSE threshold, and the number of outliers found in the dataset, along with images of the outlier.

```
(deeplearningenv) D:\Master Studies\2nd Semest
r.model
[INFO] loading autoencoder and image data...
2020-06-16 21:50:31.787511: I tensorflow/core/
[INFO] mse threshold: 0.025195109680295
[INFO] 7 outliers found
```

As we can see, there are 7 outliers found, here are those 7 outliers :



The 7 outliers found in the dataset. This means that these 7 data have a high MSE compared with the other. The MSE threshold used is ~0.0252. If we see the reconstructed images (right), we can see that some of them differ quite a lot when compared with the original image (left), which would explain why they have a high MSE. The bigger the difference between the original and reconstructed image, the higher the MSE is. MSE is calculated by the division of the sum of squared errors and the degree of freedom. The error is calculated by subtracting the original image by the reconstructed image. If the image is similar, that means that the error will be closer to 0, as the original image will be subtracted by an image with similar value. If the reconstructed image is different on the other hand, the error will be larger, resulting in larger MSE as well.