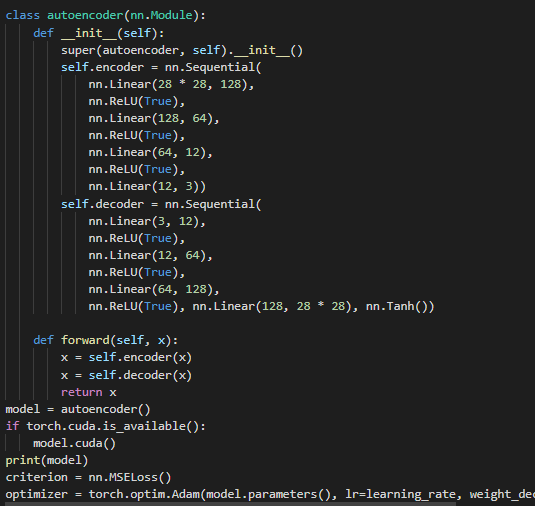
Exercise 3-2 Autoencoder

Jirayu Petchhan, D10907801

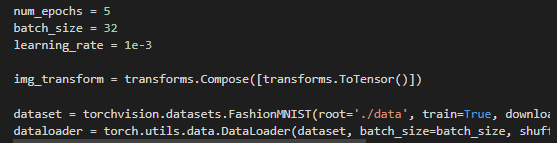
**Default provided autoencoder network**

Optimizer = Adam, cost function = MSE



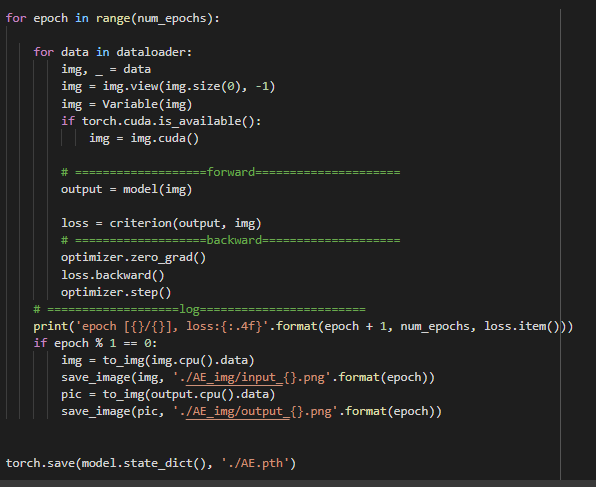
Training with the default value setting (epochs = 5)

Dataset usage = FashionMNIST (28 x 28 x 1 grayscale channel)

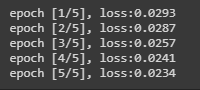


Training phase (processing in cuda device provided by colab)

* Backprop, gradient learning, model saving, accuracy estimation



The result of loss is 2.34%

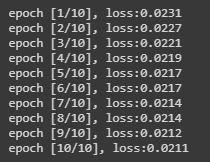


|  |  |
| --- | --- |
| Input image | Reconstructed image |
|  |  |

Training with the epochs = 10 instead



The result of loss is 2.11%

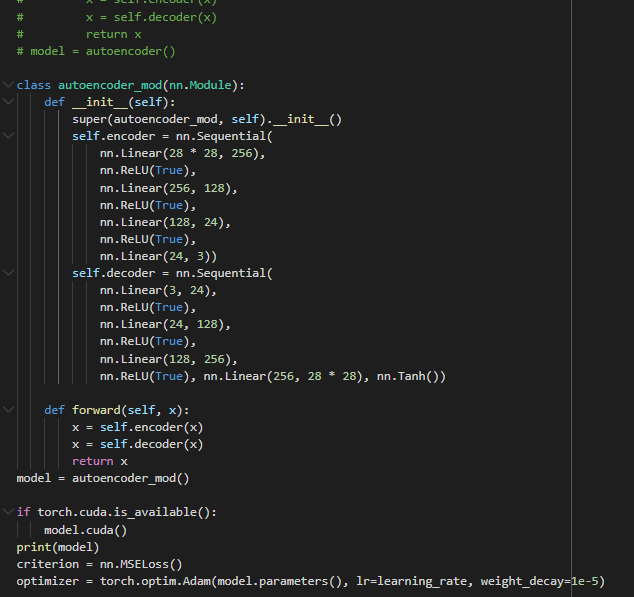


|  |  |
| --- | --- |
| Input image | Reconstructed image |
|  |  |

The result show that reconstructed image trained at epoch 5 still blur more than reconstructed image at epoch 10.

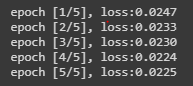
**Changing the amount of parameter in encoder and decoder architecture (according to the exercise)**

Model : autoencoder\_mod (the same loss and optimizer).



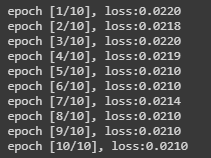
Training at 5 epochs

The result of loss is 2.25%



|  |  |
| --- | --- |
| Input image | Reconstructed image |
|  |  |

Training at 10 epochs



The result of loss is 2.10%

|  |  |
| --- | --- |
| Input image | Reconstructed image |
|  |  |

A changing of number of features still cause of blurring both training in 5 and 10 epochs

**Conclusion**

All of result (i.e. 5 and 10 epochs training for existing default model) show that when we add epochs for learning can minimize the loss value. That make the benefit to more recognizing, resulting in less loss. Nevertheless, adding the number of epochs help to increase the accuracy but should not add too much. The model might cause overfitted learning.

In the other hands, new model, the reconstruction of new images does not follow the old model since might adding the amount of feature makes learning to compress (encoding) and reconstruct (decoding) images with a greater gap that might cause of blurring both reconstructed images.