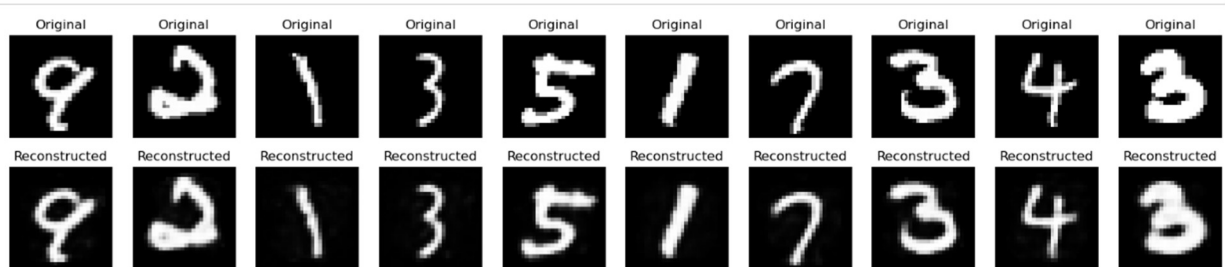


M22AI608-q1-Sparse auto-encoders

Implementing Code using Python for sparse auto-encoders

1. Installing Required Python Libraries
2. Download the MNIST dataset and saved in MNIST folder
3. Load and preprocess the data using torch vision
4. Define the Sparse Autoencoder class
5. Create the Sparse Autoencoder model
6. Define the loss function and optimizer
7. Train the model
8. Evaluate the model on the test set
Performance: Test Loss:0.003576
9. Visualize the reconstruction



10. Generate embeddings using the trained autoencoder
11. Perform k-means clustering on the embeddings
12. Evaluate the k-means clustering performance
13. Calculate accuracy and confusion matrix

Performance:

K-means clustering accuracy: 0.09985

Confusion matrix:

```
[ [ 638  409  927  567  949  451  373  503  426  680]
  [ 715  497 1055  657 1101  478  432  554  461  792]
  [ 668  387  945  544  955  443  381  445  437  753]
  [ 651  406  918  602 1040  464  372  476  412  790]
  [ 685  375  908  563  901  455  332  489  402  732]
  [ 602  386  863  501  883  450  300  403  378  655]
  [ 607  386  905  578  942  474  374  472  429  751]
  [ 705  386  969  608 1018  512  408  497  425  737]
  [ 608  394  921  580  946  499  346  468  405  684]
  [ 671  393  946  543 1031  450  385  460  388  682] ]
```

Finally:

Run the Python script using command: filename(m22ai608-Sparse auto-encoders-q1).py

References:

1. Sparse Autoencoders for MNIST Classification:
[Lec19 Sparse Autoencoders for MNIST classification \(Hands on\) - YouTube](#)
2. Making an Autoencoder:
<https://towardsdatascience.com/how-to-make-an-autoencoder-2f2d99cd5103>