Data Mining HW 1

3. *Explain how have you implemented clusterisation part of DNN (SGD optimisation)*
   Created a class `DeepEmbeddeddClustering`, which instance is a model, taking X (MNIST data) as input, then encodes it and clusters, the output is vector of clusters.
   Clustering is done with the help of additional layer class, which given embedded points predicts soft labels for each samples.
   The main algorithm is located in `def fit_predict` in `DeepEmbeddeddClustering` class. The method iteratively trains the model on batch (256) and each 10 iteration update q and p values. The process stops when either max iteration (10 000) is exceeded or tolerance is less than some minimum value (0.001).

4. *Explain how have you implemented encoder pretrain part.*
   Encoder is the first 4 layers of the autoencoder, which was trained the following way. The pertaining consists of two parts: layer-wise pertaining and fine-tuning. Layer-wise means that the auto encoder is splitter in four two-layers (encoder-decoder) autoencoders and each small autoencoder is trained separately taking as input output of previous encoder. Then create the whole autoencoder, consisting of trained encoders and decoders in the right order.
   While fine-tuning create new encoders and decoders without dropout layer and use weights from layer-wise pertaining. Then fit the whole new autoencoder.

   When layer-wise pertaining is done for 500 epochs, fine-tuning 100 epochs, learning rate is updated every 200 epochs the dec accuracy is about 74%. Epochs and hence results in code provided may vary somehow.

Weights for encoder: https://drive.google.com/open?id=1gPgy2fgxj6KLCuXjGyTEW9gNIcvVp4XZ
The code: https://colab.research.google.com/drive/1JlQ-Fvv2BqPdEKx4CS3Z3zA-w0SySA7S