Metric Learning with Adaptive Density Distribution

Team Members -
Anushka Agarwal(201530064)
Eavanshi Arora (201501115)
Nikita Agarwal (201501041)
Sailaja Nimmagadda(20153008)

Team No - 1
Mentor - Abhijeet Kumar

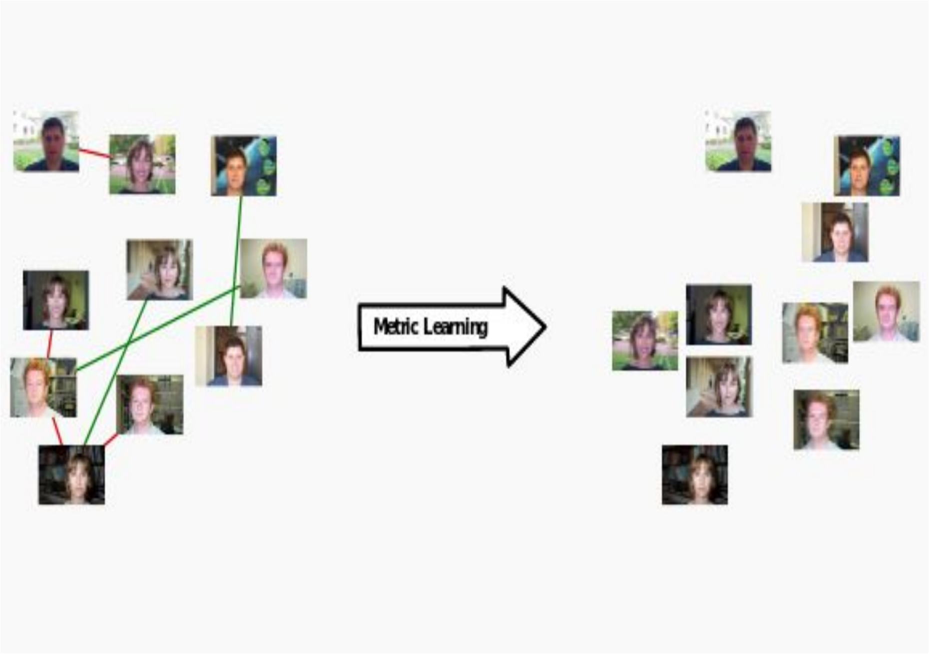
Objective

To understand and implement the distance metric learning algorithm using magnetic loss model which maintains a distribution of classes in representation space and then employs this knowledge to adaptively assess similarity,and achieve local discrimination by penalizing class distribution overlap.

Distance Metric Learning

Distance Metric Learning

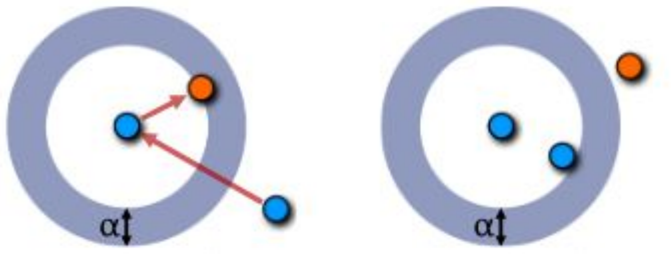
- ➢ It learns transformation in a space where distance corresponds to a notion of similarity which is predefined.
- ➢ Intra-class similarity and Inter-class variation are maximised.
- ➢ All information except the class label is discarded.



Motivation:
Challenges in Distance Metric Learning

Challenges in Distance Metric Learning

- Similarity function is determined a-priori.
- Each class should be captured by a single node. This harms intra-class variance
- In local similarity original input space is used and are never updated.
- Most popular DML algorithms- Triplet Loss and Contrastive Loss exhibit short-sightedness.They penalise pairs or triplets, thus do not imply sufficient insight of neighbourhood structure.

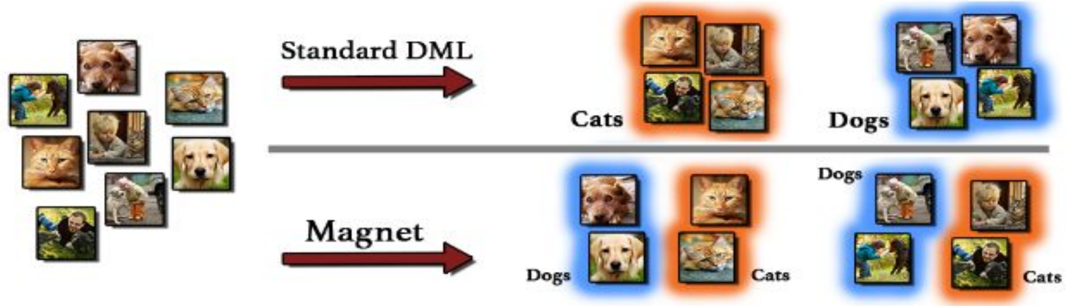


(a) Triplet: before. (b) Triplet: after.

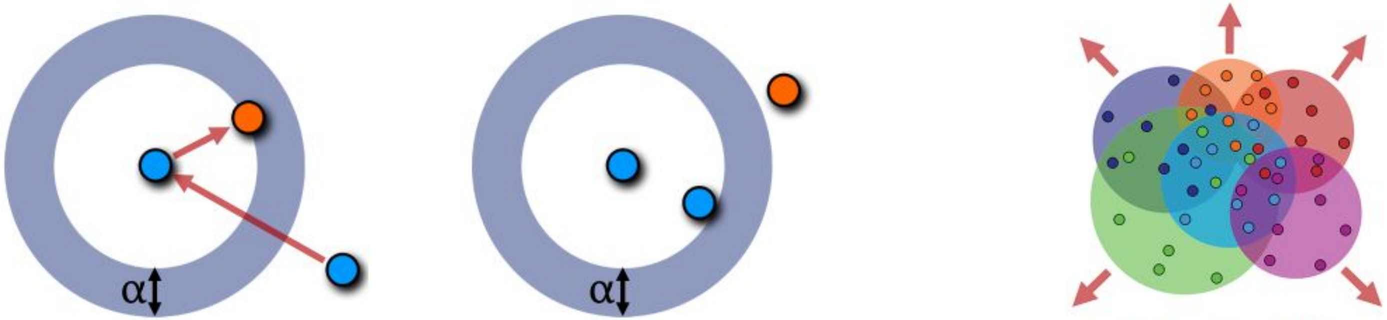
Magnet Loss

Magnet Loss

- Similarity function is characterised as a function of current representation structure.
- Local separation is pursued instead of global, thus unimodularity and prior-target-neighbourhood assignment is not required.
- Nearest neighbours retrieval is a two-step process(cluster retrieval then example retrieval), this is computationally easier.



Magnet Loss v/s Triplet Loss



(a) Triplet: before. (b) Triplet: after. (c) Magnet: before. (d) Magnet: after.

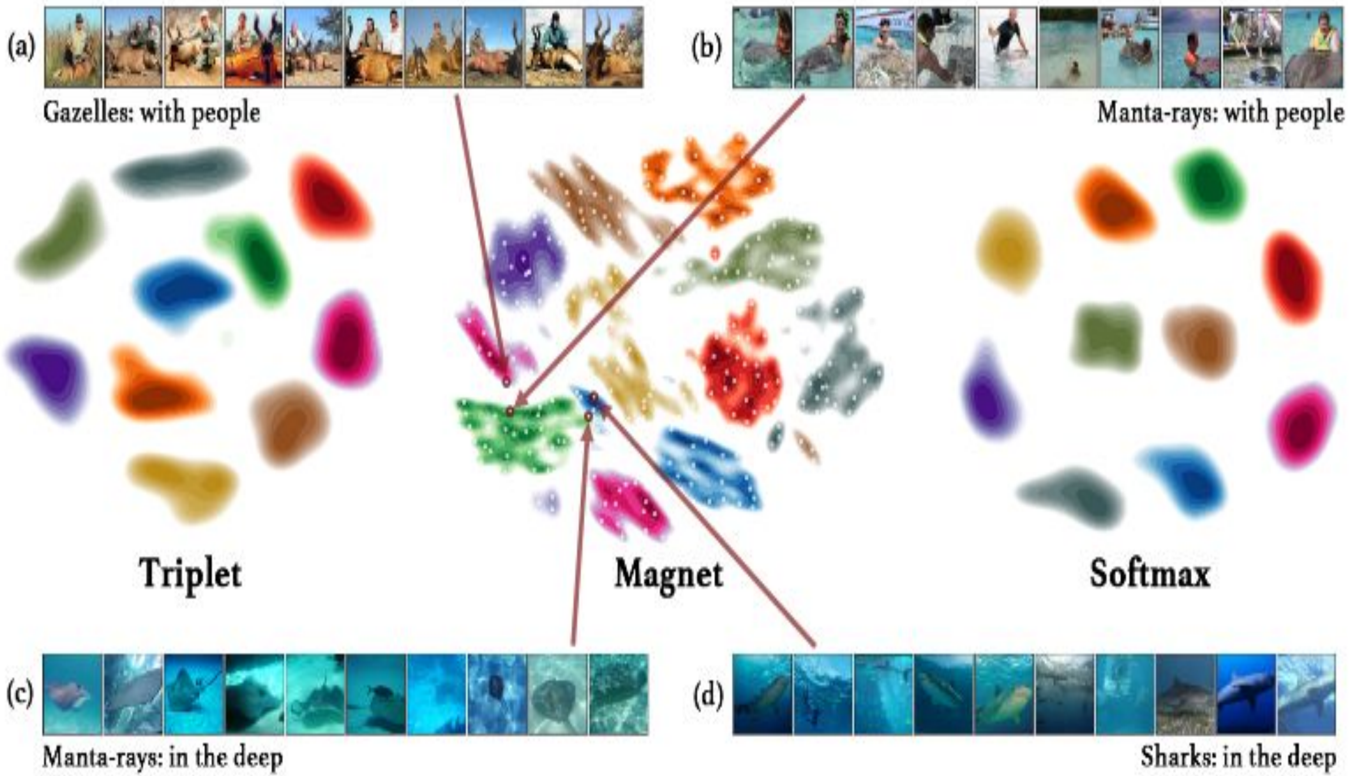
Magnet Loss v/s Triplet Loss Efficiency

1. Triplet loss only considers a single triplet at a time, resulting in reduced performance and training inefficiencies.

2. In contrast, in Magnet Loss, at each iteration an entire local neighbourhood of nearest clusters is retrieved.Class overlaps are penalized. Instead of independent examples, whole neighbourhood is sampled, this considerable improves training efficiency.

Insight into representation distribution permits adaptive similarity characterization, local discrimination and a globally consistent optimization procedure.

Magnet Loss v/s Triplet Loss



Model Formulation

Model Formulation

- Representation space is learnt using GoogLeNet CNN.
- Each class has K clusters (I^c_1 , I^c_2 ,..., I^c_k) which are chosen to minimise intra-class distances.

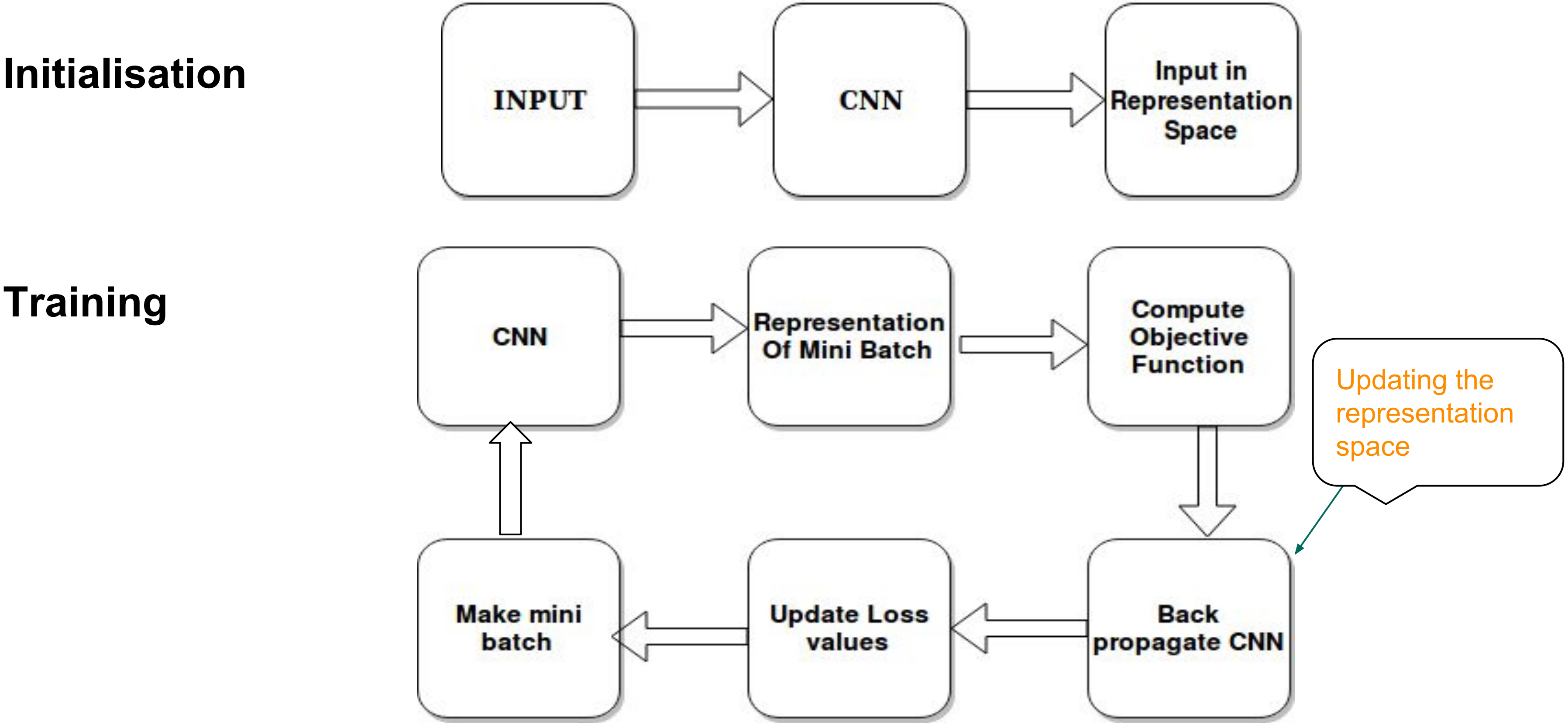
$$I^c_1 , I^c_2 ,..., I^c_k = \arg(I^c_1 , I^c_2 ,..., I^c_k)\min \Sigma^K_{k=1} \Sigma_{r\epsilon Ik,c} || r-\boldsymbol{\mu}^c_k ||_2^2$$

- Objective function:

$$L(\Theta) = \frac{1}{N} \sum_{n=1}^{N} \left\{ - log\ (exp(- \frac{1}{\sigma^2} \left| r_n - \mu(r_n) \right|^2 - \alpha)) \,/\, \sum_{c \neq C(r_n)} \sum_{k=1}^{K} exp(- \frac{1}{\sigma^2} \left| r_n - \mu_k^c \right|^2) \right\} +$$

- Variance Standardisation in this approach makes the objective invariant to characteristic length scale of the problem.

Training Procedure



Evaluation Procedure

➢ Label of each example Xn is a function of its representation {r(n)} softmax similarities to its L closest clusters.

$$c_n^* = \arg\max_{c=1,\ldots,C}\left(\sum_{\mu_l : C(\mu_l)=c} \exp\left(-\frac{1}{\sigma^2}\left|r_n - \mu_l\right|^2\right)\right) / \left(\sum_{l=1}^{L} \exp\left(-\frac{1}{\sigma^2}\left|r_n - \mu_l\right|^2\right)\right)$$

➢ Efficiency increases monotonically with L as more neighbourhood information is being used, but after a certain value further increase in L is of no use since the new points are farther away from the lengthscale defined by σ.

Related to Other Models

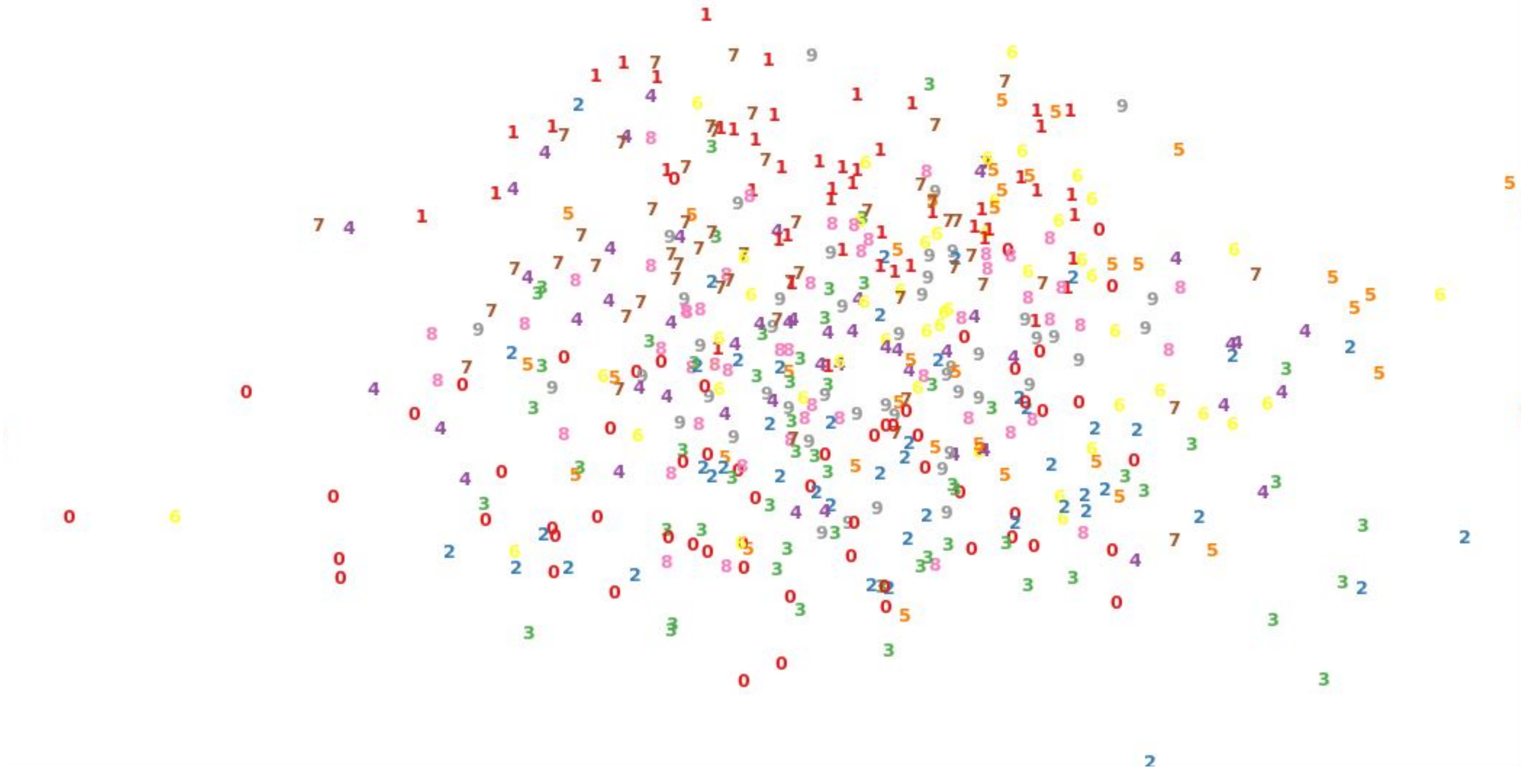
Experiments and Results

Dataset

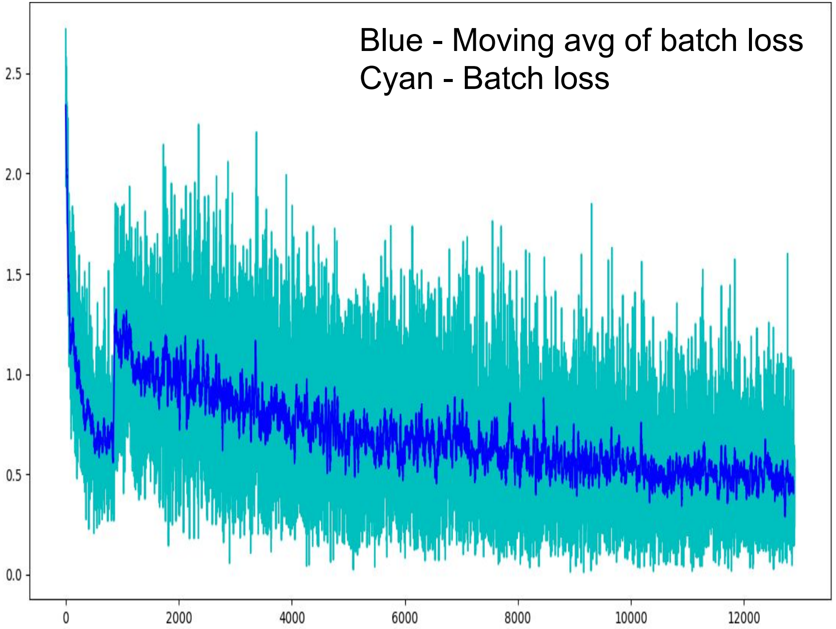
- ➢ MNIST is a computer vision database consisting of handwritten digits, with labels identifying the digits. As mentioned earlier, every MNIST data point has two parts: an image of a handwritten digit and a corresponding label.
- ➢ Each image is 28 pixels by 28 pixels. We can interpret this as a big array of numbers. We can flatten this array into a vector of 28×28 = 784 numbers.
- ➢ Number of Training Examples : 55000
- ➢ Number of Testing Examples : 10,000
- ➢ Parameters :

k = Number of clusters in each class m = Total number of clusters in a minibatch
d = Number of examples from each cluster α = Desired cluster separation or variance

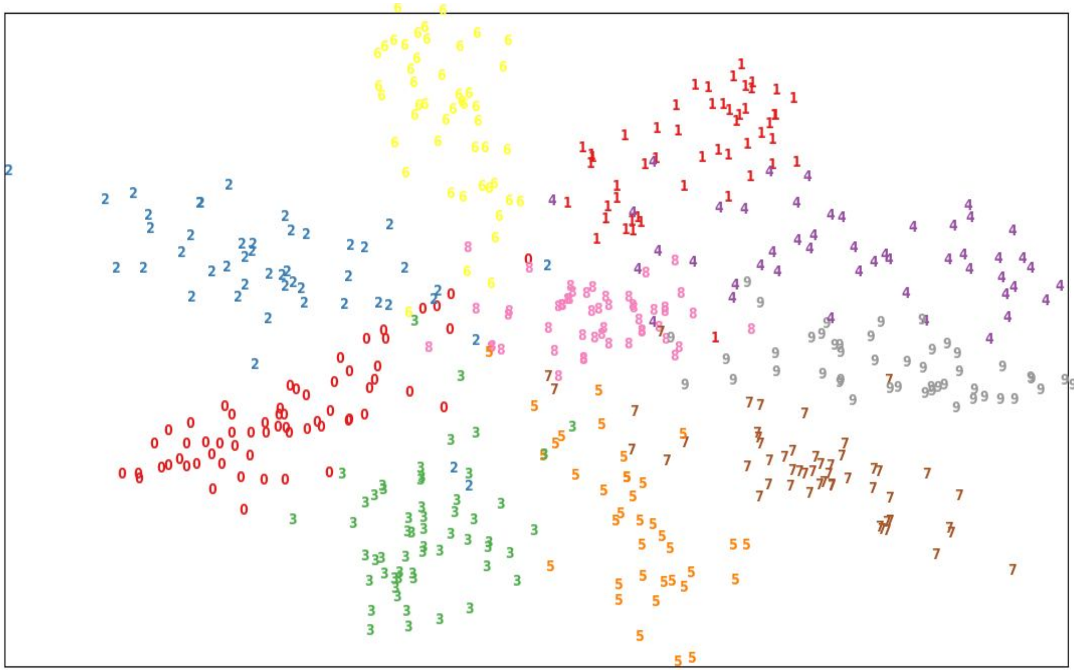
Initial Distribution Of Points



Results (Given variables)
Epochs = 15, m=8, d=8, k=3, alpha=1

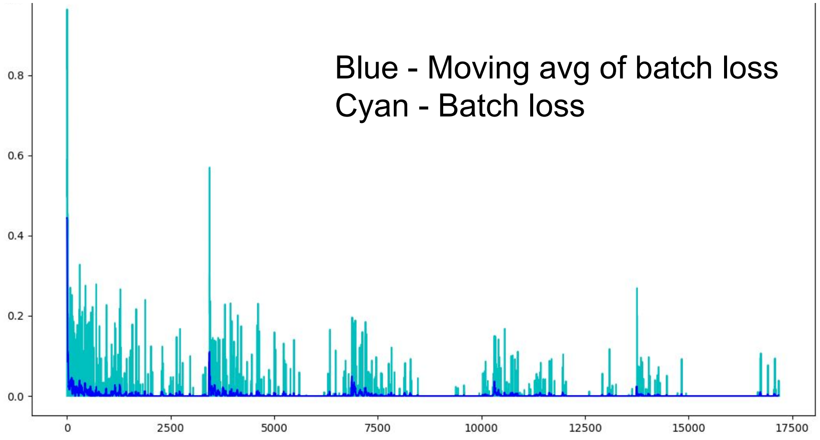


Loss v/s Iterations

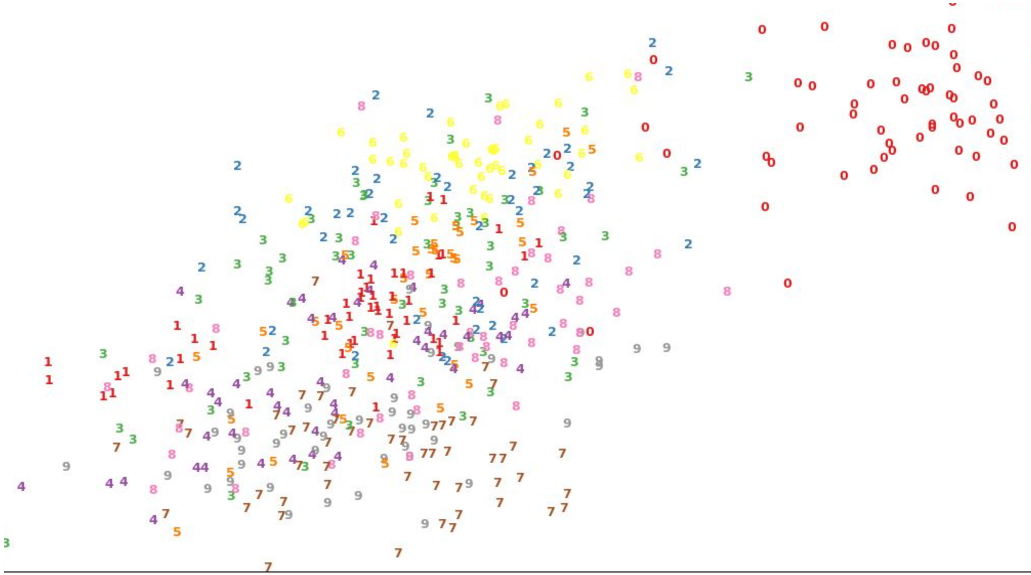


Final Distribution

Results (Epochs = 5, m changing)
Epochs = 5, m=2, d=8, k=3, alpha=1

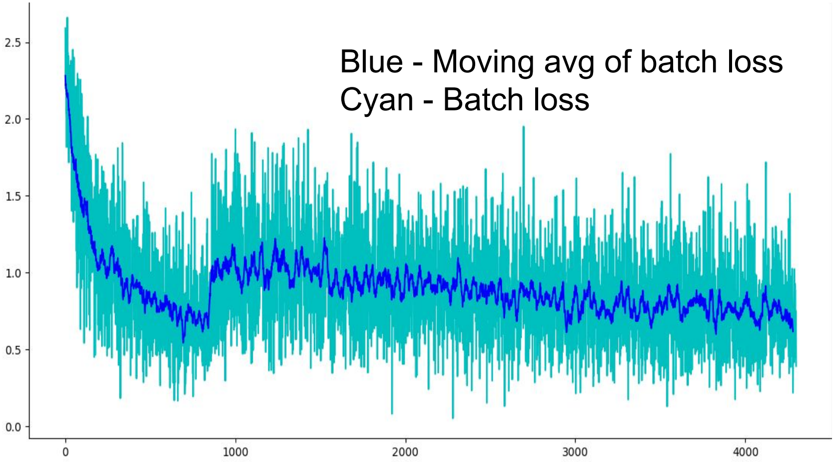


Loss v/s Iterations

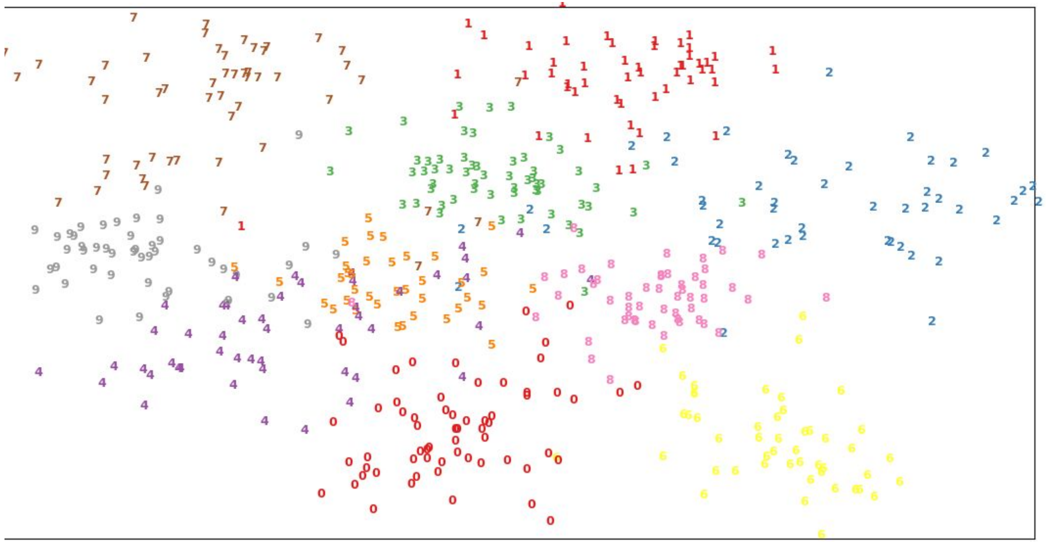


Final Distribution

Results(Epochs = 5, m changing)
Epochs = 5, m=8, d=8, k=3, alpha=1

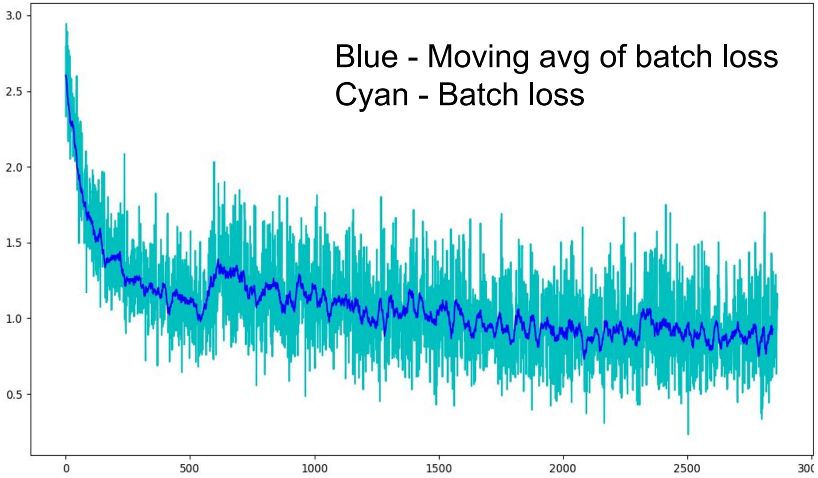


Loss v/s Iterations

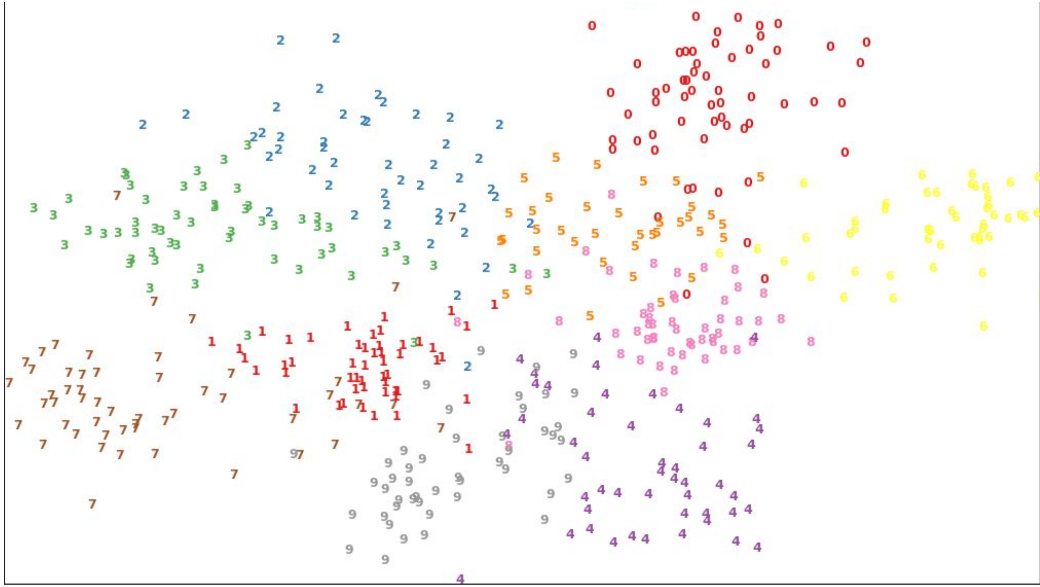


Final Distribution

Results(Epochs = 5, m changing)
Epochs = 5, m=12, d=8, k=3, alpha=1

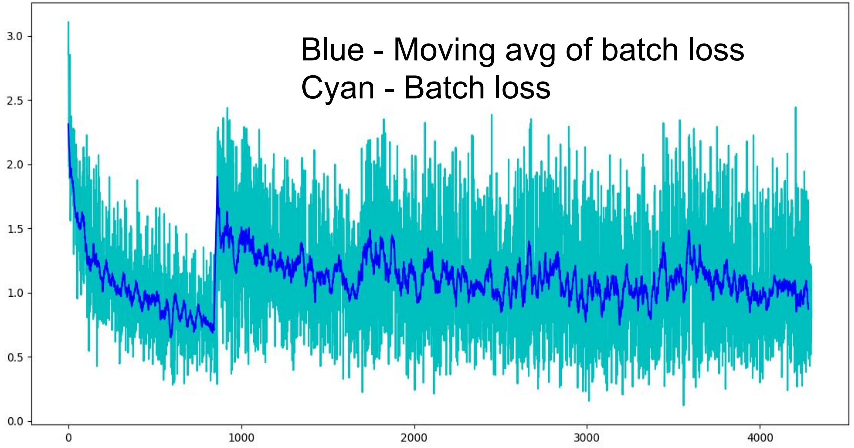


Loss v/s Iterations

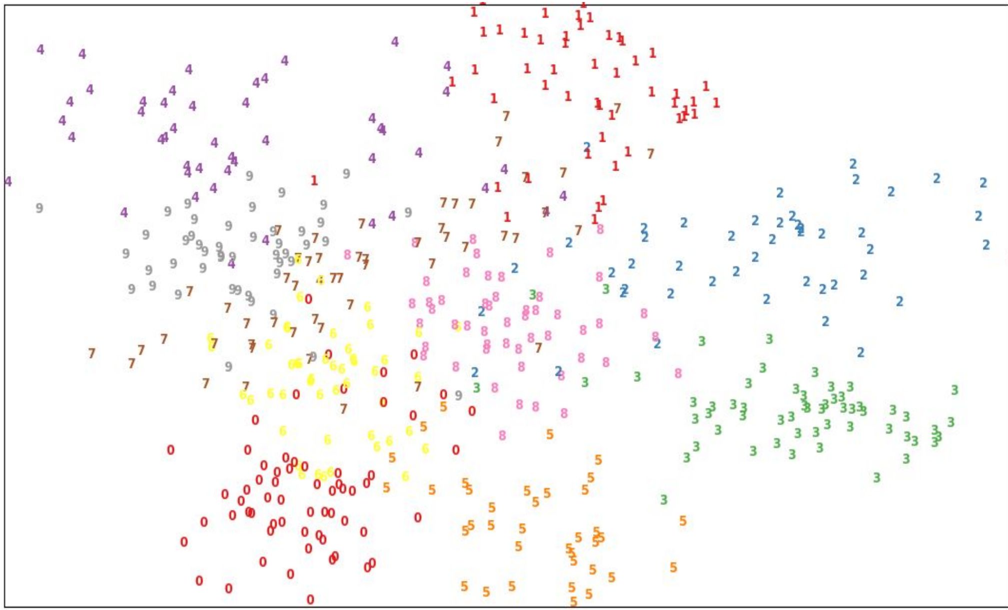


Final Distribution

Results(Epochs = 5, k changing)
Epochs = 5, m=8, d=8, k=6, alpha=1



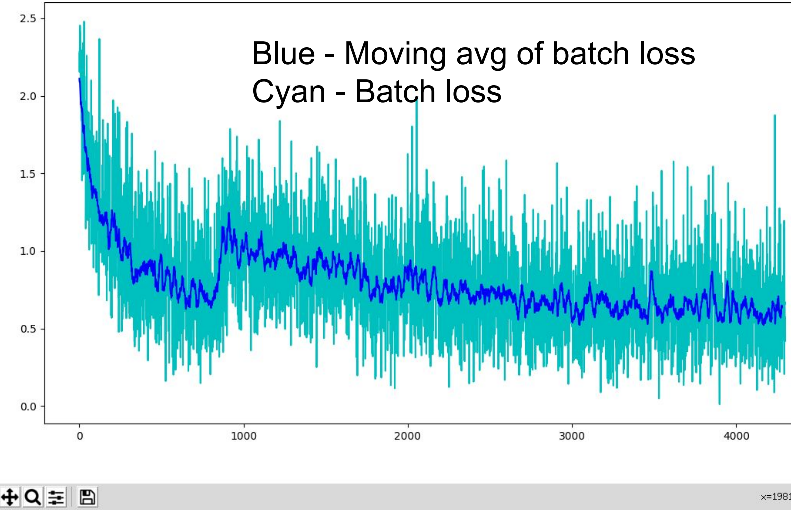
Loss v/s Iterations



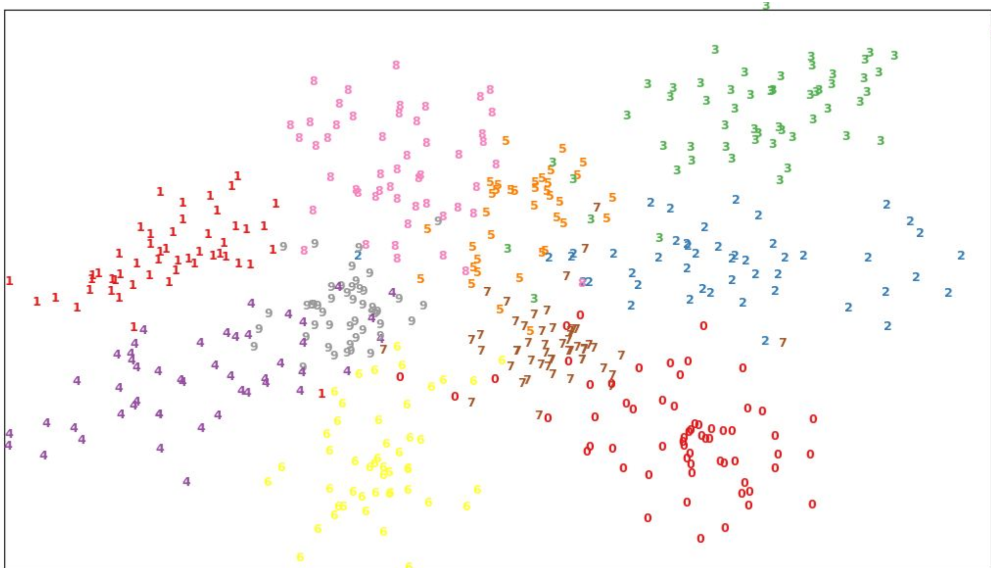
Final Distribution

Results(Epochs = 5, k changing)

Epochs = 5, m=8, d=8, k=2, alpha=1

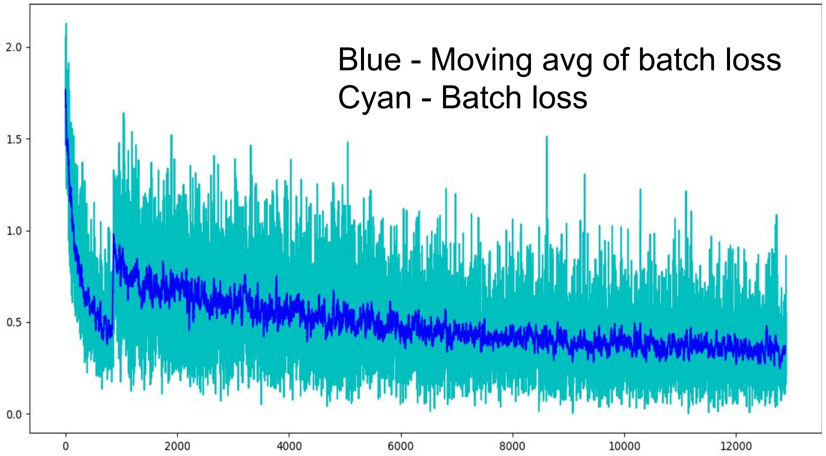


Loss v/s Iterations

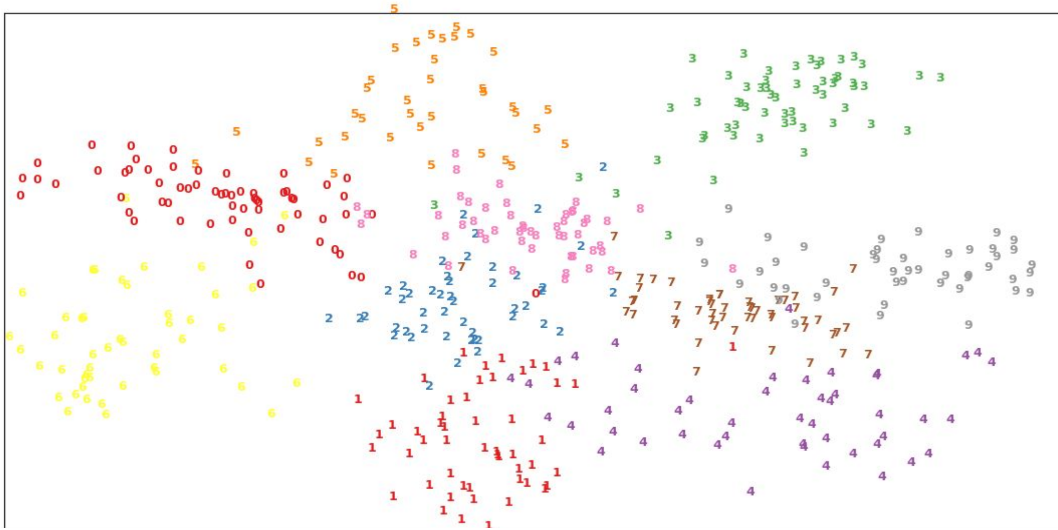


Final Distribution

Results(Epochs = 5, alpha changing)
Epochs = 15, m=8, d=8, k=3, alpha=0.5



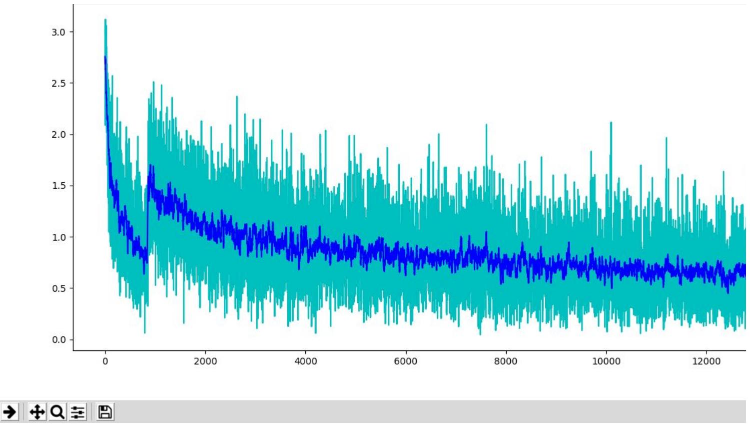
Loss v/s Iterations



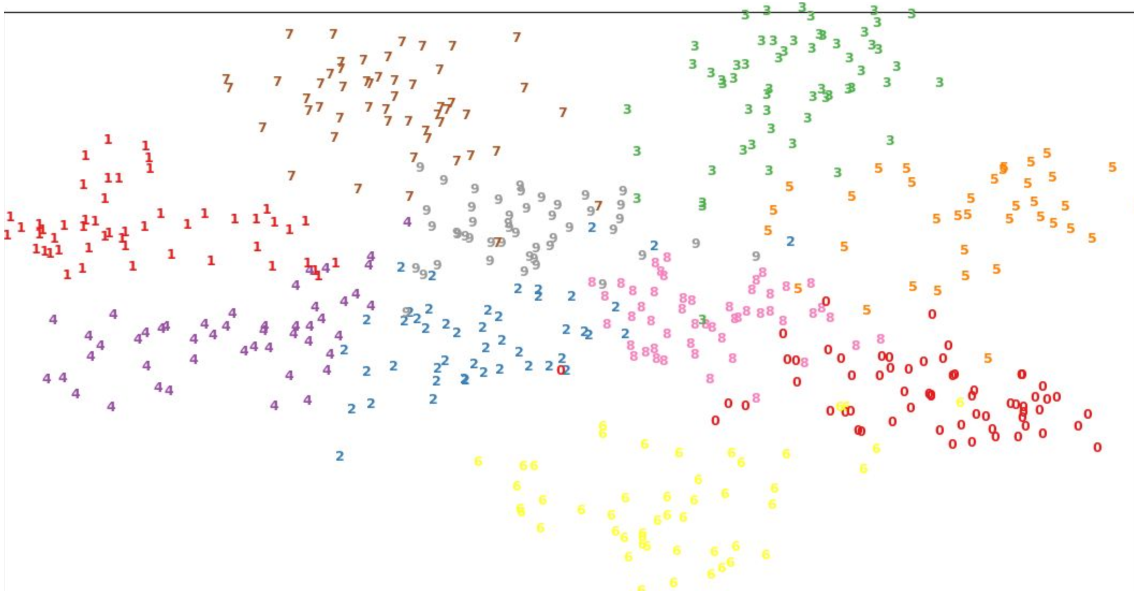
Final Distribution

Results(Epochs = 5, alpha changing)
Epochs = 15, m=8, d=8, k=3, alpha=1.5

Blue - Moving avg of batch loss
Cyan - Batch loss



Loss v/s Iterations



Final Distribution

Scope for Future Work

- ➢ In this work we chose the number of clusters K per class as uniform across classes, and refreshed our representation index at a fixed rate. We believe that adaptively varying these during training can enhance performance and facilitate computation.

- ➢ We can replace the density estimation and indexing component with an approach more sophisticated than K-means. One natural candidate would be a tree-based algorithm. This would enable more efficient and more accurate neighbourhood retrieval.

Thank you!