E0:270 - Machine Learning - Clustering using Deep Learning

Mayank Singh [1] **Shivank Gupta** [1] **Tapan Bhardwaj** [1]

Abstract

Clustering is central to many data-driven application domains and has been studied extensively in terms of distance functions and grouping algorithms. Relatively little work has focused on learning representations for clustering. In this project, we have explored recent deep learning approaches like Deep Embedded Clustering(Xie et al., 2016), Improved Deep Embedding Clustering(Xifeng Guo, 2017)and Towards K-means-friendly Spaces(Yang et al., 2016). We have implemented these three approaches and test their performance on one synthetic and two real world data.

1. Problem Statement

Unsupervised clustering is a vital research topic in data science and machine learning. However, when the dimension of input feature space (data space) is very high, the clustering becomes ineffective due to unreliable similarity metrics.

Thus the objective is to implement techniques for transforming data from high dimensional feature space to clustering friendly lower dimensional space, perform clustering on it and do comparative analysis of these techniques on various data-sets.

2. Literature Review

We have gone through the following Deep Learning architecture for Clustering

2.1. DEC : Deep Embedded Clustering(Xie et al., 2016)

This method simultaneously learns feature representations and cluster assignments using deep neural networks. It learns a mapping from the data space to a lower dimensional feature space in which it iteratively optimizes a clustering objective without including the reconstruction cost. The authors define the objective as a KL divergence loss between the soft assignments q_i and the auxiliary distribution p_i as follows:
$L = KL(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{i,j}}$

2.2. Towards K-means-friendly Spaces(Yang et al., 2016)

The paper introduces us to a Deep Clustering Network which models the relationship between observable data and its k-mean clustering friendly latent representation. It contains of two halves, the first half converts the observable data of high dimensionality into clustering friendly latent representation, and the second half reconstructs the latent vector (output of first half) close to the original data of high dimensionality. This helps in avoiding trivial solutions. The cost function consists of reconstruction as well as clustering losses as given below:-

$\min_{W,Z,M,\{s_i\}} \sum_{i=1}^{N} (l(g(f(x_i)), x_i) + \frac{\lambda}{2}\|f(x_i) - Ms_i\|_2^2)$

$s.t. \quad s_{i,j} \in \{0,1\}, 1^T s_i = 1 \quad \forall i, j$

where authors have simplified the notation $f(x_i; W)$ and $g(h_i; Z)$ to $f(x_i)$ and $g(h_i)$, respectively, for conciseness. Function f projects high dimensional data point to clustering friendly lower dimensional space and function g tries to approximate the lower dimensional data point to original high dimensional. Function $l(.)$ measures this reconstruction loss from x_i to $g(f(x_i))$. The second term takes into account the clustering loss, here M is the matrix which consists of vectors representing centers of clusters in low dimensional space, s_i is the assignment vector of data point x_i which maps it to one on the centers in matrix M. $\lambda \geq 0$ is a regularization parameter which balances the reconstruction error versus finding Kmeans-friendly latent representations

Initialization of network parameters(W, Z) is done using layer-wise pre-training(Bengio et al., 2007) Optimization is done in subproblems keeping either Network Parameters(W, Z) or Clustering Parameters(M, s_i) as constant.

[1]Department of Computer Science and Automation, Indian Insitute of Science, Bangalore. Correspondence to: Mayank Singh <mayanksingh@iisc.ac.in>, Shivank Gupta <shivankgupta@iisc.ac.in>, Tapan Bhardwaj <tapanb@iisc.ac.in>.

Final Project Report, E0:270, Machine Learning, 2018.

2.3. IDEC : Improved Deep Embedding Clustering(Xifeng Guo, 2017)

Some pioneering work proposes to simultaneously learn embedded features and perform clustering by explicitly defining a clustering oriented loss.But considering only clustering loss may corrupt feature space and which can lead to poor clustering performance. To address this issue, Improved Deep Embedded Clustering (IDEC) algorithm take care of data structure preservation. To maintain the local structure of data generating distribution, an auto-encoder is applied.The loss function is given by:
$L = L_r + \gamma L_c$,where
L_c is the clustering loss as in DEC (Xie et al., 2016) and
$L_r = \sum_{i=1}^{N} \|x_i - g_{W'}(z_i)\|_2^2$, where $z_i = f_W(x_i)$ and f_W and $g_{W'}$ are encoder and decoder mappings respectively.
By integrating the clustering loss(L_c) and auto-encoders reconstruction loss(L_r), IDEC can jointly optimize cluster labels assignment and learn features that are suitable for clustering with local structure preservation.

3. Model description

3.1. DEC : Deep Embedded Clustering(Xie et al., 2016)

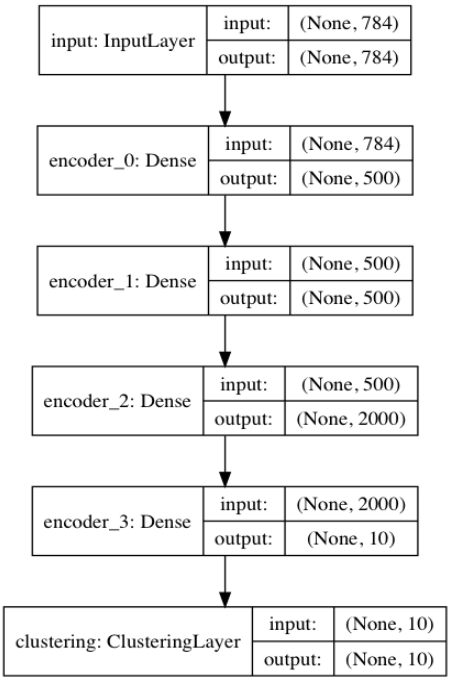


Figure 1: Implemented DEC model over MNIST data-set

The input in the given model are data-points from MNIST data-set, the encoder takes in data-points with 784 dimensions and outputs clustering friendly data-points with 10 dimensions. There is no reconstruction of original data.

3.2. Towards K-means-friendly Spaces(Yang et al., 2016)

The model of this technique resembles to that of IDEC expained in next section.

3.3. IDEC : Improved Deep Embedding Clustering(Xifeng Guo, 2017)

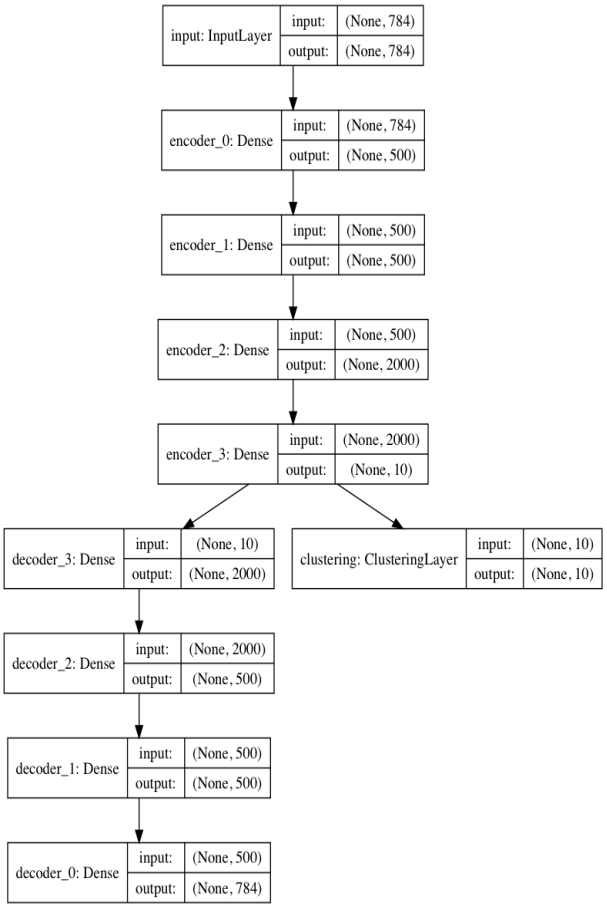


Figure 2: Implemented IDEC model over MNIST data-set

The model given above is for MNIST data-set .The first part resembles to the model of DEC where input is data-point with 784 dimensions and output is clustering friendly data-point with 10 dimensions. This output is also sent for reconstruction as shown. So the model takes into account clustering as well as reconstruction losses.

4. Data-set description

4.1. MNIST

The MNIST data-set consists of 70000 handwritten digits. The images are centered and of size 28 by 28 pixels. We reshaped each image to a 784 dimensional vector.

4.2. HHAR

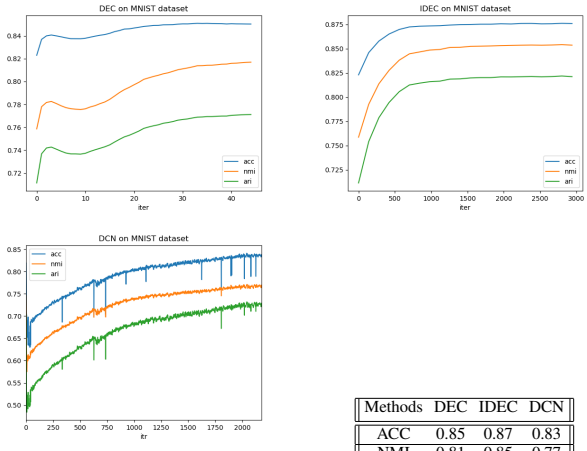
The Heterogeneity Human Activity Recognition (HHAR) data-set contains 10299 sensor records from smart phones and smart watches. All samples are partitioned into 6 categories of human activities and each sample is of 561 dimensions.

4.3. Synthetic Data

We have generated four clusters, each of which has 2,500 samples in 2 dimension. This two dimensional domain is a latent domain which we do not observe, what we observe is $x_i \in R^{100}$ that is obtained via the following transformation: $x_i = \sigma(U\sigma(Wh_i))$
where $W \in R^{10x2}$ and $U \in R^{100x10}$ are matrices whose entries follow the zero-mean unit-variance i.i.d. Gaussian distribution and $h_i \in R^2$ which is our original data point in 2 dimension.

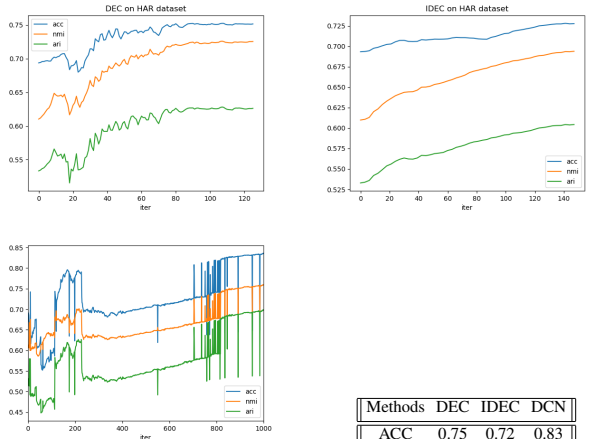
5. Experiments

We have evaluated the methods DEC, IDEC and Towards K-means-friendly Spaces(DCN) on the three data sets:- MNIST, HHAR and Synthetic Data. The metrics used are ACC(unsupervised clustering accuracy (ACC), NMI(Normalized Mutual Information) and ARI(Adjusted Random Index). All the methods have been implemented using KERAS framework. The results of comparative analysis are shown in form of graph and tables.



Methods	DEC	IDEC	DCN
ACC	0.85	0.87	0.83
NMI	0.81	0.85	0.77
ARI	0.77	0.82	0.73

Figure 3: DEC,IDEC and Towards K-means friendly spaces performances(DCN),respectively from left to right, over MNIST DATA-set using evaluation metrics ACC, NMI and ARI



Methods	DEC	IDEC	DCN
ACC	0.75	0.72	0.83
NMI	0.72	0.69	0.76
ARI	0.62	0.60	0.70

Figure 4: DEC,IDEC and Towards K-means friendly spaces performances(DCN),respectively from left to right, over HHAR DATA-set

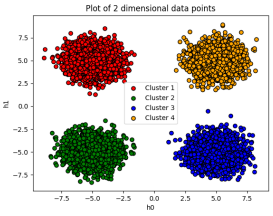
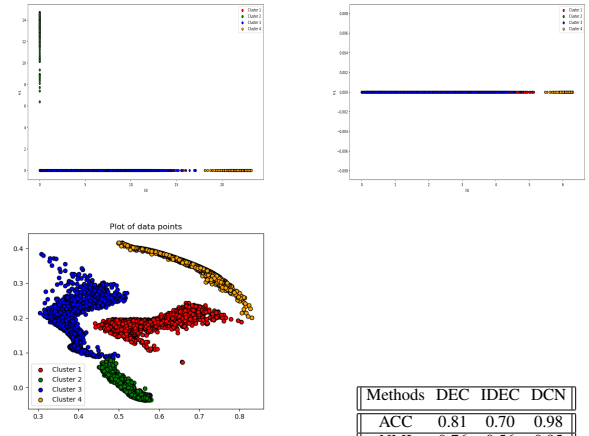


Figure 5: Synthetic data as viewed in two dimensional, non linear transformation is applied over it to get high dimensional data



Methods	DEC	IDEC	DCN
ACC	0.81	0.70	0.98
NMI	0.76	0.56	0.95
ARI	0.69	0.50	0.97

Figure 6: Reconstructed synthetic two dimensional clusters using DEC, IDEC and DCN

6. Conclusion

IDEC(Xifeng Guo, 2017) and DCN(Yang et al., 2016) take into account reconstruction losses and hence perform better than DEC.(Xie et al., 2016). For the same reason DEC(Xie et al., 2016) takes less computational time as compared to the other two.
DCN(Yang et al., 2016) outputs far more visibly differential clusters in two dimensions than the other two for synthetic data-set(refer to figure 5).

7. Work Distribution

Mayank Singh and Tapan Bhardwaj implemented DEC(Xie et al., 2016) and IDEC(Xifeng Guo, 2017) collaboratively. Shivank Gupta implemented DCN(Yang et al., 2016).

References

Bengio, Yoshua, Lamblin, Pascal, Popovici, Dan, and Larochelle, Hugo. Greedy layer-wise training of deep networks. In Schölkopf, B., Platt, J. C., and Hoffman, T. (eds.), *Advances in Neural Information Processing Systems 19*, pp. 153–160. MIT Press, 2007.

Xie, Junyuan, Girshick, Ross B., and Farhadi, Ali. Unsupervised deep embedding for clustering analysis. *CoRR*, abs/1511.06335, 2016. URL http://arxiv.org/abs/1511.06335.

Xifeng Guo, Long Gao, Xinwang Liu Jianping Yin. Improved deep embedded clustering with local structure preservation. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 1753–1759, 2017. doi: 10.24963/ijcai.2017/243. URL https://doi.org/10.24963/ijcai.2017/243.

Yang, Bo, Fu, Xiao, Sidiropoulos, Nicholas D., and Hong, Mingyi. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. *CoRR*, abs/1610.04794, 2016. URL http://arxiv.org/abs/1610.04794.