Internship Report

Rahul Chakwate
IIT Madras, India

Work done at ESIEE Paris
Advisor: Prof. Laurent NAJMAN

13th May 2019 - 19th July 2019

1 Overview of Internship

”Watershed” is a technique from mathematical morphology used mainly for image segmentation applied on an undiracted graph. During my internship at ESIEE Paris, an attempt was made to integrate this widely applicable technique with the latest machine learning techniques to obtain state-of-the-art results.

2 Literature Survey

- ”Watersheds for Semi-supervised Classification[2]”: The main concept of this paper is the ”MorphMedian” operator. The paper describes the Minimum Spanning Forest (MSF) Watershed with arbitrary seeds. The input is a weighted graph with some labelled seeds and outputs a partition of V into appropriate segments. The notion of Maximum Margin Partition used in SVMs is used in this paper to define the MorphMedian partition. The MSF-Watershed returns a MorphMedian partition and hence a Maximum Margin Partition. DIfference with 1 NN method is that 1NN considers a distance, while MORPHMEDIAN generalizes this to any dissimilarity measure.
- ”Tour on Watersheds[4]”: Learned the fundamentals of graph, MST, erosion and dilution, Watershed cuts based on drop of water principle and rising water principle. Dilation and erosion operations are carried out on graph. Learned about half opening and half closing. The step by step process to find the watershed cut is explained in this paper. Applications such as surface segmentation are illustrated. MSF hierarchical watersheds are discussed.
- ”A graph-based mathematical morphology reader[3]”: This is a survey paper on morphological operations. Graphs, adjunctions, basic morphological operations, connected filters, watersheds and hierarchies are discussed in details in this paper. Applications beyond the graph like the point clouds are also discussed in this paper.
- ”Convolutional Oriented Boundaries[5]”: End to end learning of CNN for object boundary or contour detection and boundary orientation and heirarchical segmentation. First, contour detection at different scales. Feature maps are extracted from every last layer of same scale. These maps are aggregated using trainable weights. They are compared with the ground truth using the given loss function. For orientation, a sub network is attached to every layer output and it is divided into K classes. Boundaries are stored using sparse representation.
- ”Metric Learning with Adaptive Density Discrimination[1]”: Discussed in the next section.

3 Magnet Loss discussion

”METRIC LEARNING WITH ADAPTIVE DENSITY DISCRIMINATION[1]” introduces a new loss function called the ”magnet loss” which claim to achieve state-of-the-art classification results on visual recognition datasets outperforming the triplet loss with a 30-40% margin. It also shows good performance on hierarchical recovery properties. Motivation behind the paper is to capture the intraclass variation and interclass similarities in the data. Distance Metric Learning (DML) approaches transform data to a representation space according to a similarity measure. However earlier DML methods are incompetent with the modern classification algorithms. The magnet loss method described in the paper claim to outperform these modern algorithms. Some of the benifits of DML are zero-shot learning, visualizing high dimensional data, learning invariant maps, scaling of instances to millions of classes.

Magnet loss is a DML approach which overcomes the issue of predefined target neighbourhood structure and the issue of target formulation.

Triplet loss, which is a special case of magnet loss, is formulated as follows:

$$\mathscr{L}_{\text{triplet}}(\mathbf{\Theta}) = \frac{1}{M}\sum_{m=1}^{M}\left\{\left\|\mathbf{r}_m - \mathbf{r}_m^-\right\|_2^2 - \left\|\mathbf{r}_m - \mathbf{r}_m^+\right\|_2^2 + \alpha\right\}_+$$

where $\{.\}_+$ is the hinge function and rm, rm_+ and rm_- are the representation of seed example, positive example and negative example respectively.

Magnet loss adapts clustering techniques to capture the distributions in the representation space. For each class index of clusters is maintained which is updated continuously throughout training. Objective function jointly manipulates the entire cluster as opposed to individual examples. Clusters attract and repel each other and hence the name ”Magnet Loss”.

The k clusters are obtained via K-means algorithm. The loss function is formulated as below:

$$\begin{aligned}\mathcal{I}_1^c, \ldots, \mathcal{I}_K^c &= \arg\min_{I_1^c,\ldots,I_K^c} \sum_{k=1}^{K} \sum_{\mathbf{r}\in I_k^c} \|\mathbf{r} - \boldsymbol{\mu}_k^c\|_2^2 \\ \boldsymbol{\mu}_k^c &= \frac{1}{|I_k^c|} \sum_{\mathbf{r}\in I_k^c} \mathbf{r}\ .\end{aligned}$$

Further the loss function is modelled as follows:

$$\mathscr{L}\left(\boldsymbol{\Theta}\right) = \frac{1}{N}\sum_{n=1}^{N}\left\{-\log \frac{e^{-\frac{1}{2\sigma^2}\|\mathbf{r}_n-\boldsymbol{\mu}(\mathbf{r}_n)\|_2^2-\alpha}}{\sum_{c\neq C(\mathbf{r}_n)}\sum_{k=1}^{K} e^{-\frac{1}{2\sigma^2}\left\|\mathbf{r}_n-\boldsymbol{\mu}_k^c\right\|_2^2}}\right\}_+$$

where C(r) is the class of the representation r, $\mu(r)$ are its cluster centers, $\{.\}_+$is hinge function, α is a scalar and σ^2 is the variance of all examples away from their respective cluster centers.

While evaluating, k nearest cluster is used which is a variant of soft KNN.

4 Limitations of magnet loss

In the future work of the DML paper, it is discussed that varying K adaptively while training rather than keeping it fixed can improve its performance.The author asks to try out a more sophisticated approach than K-means like the tree based method.

In reality, almost no dataset will have fixed number of sub clusters within a class. In any dataset, there can be a mismatch in the number of clusters within each class. Some classes are broad while some are small. In such cases, fixing the same k performs poorly as illustrated in the experiments below.

Watersheds have the ability to adaptively select the number of clusters by keeping some other criterion like the area fixed. Hence, we aim at using complete watershed to select the number of clusters on the go of the training process.

5 Theory of Improvement using Complete Watershed

Watershed are known to represent the purest sub-clusters. First use the Minimum Spanning Tree on the representation. Then apply watershed hierarchy by area filtering. This operation carries hierarchical clustering of the MST and removes the small nodes by area thresholding.

Since the criterion is the area thresholding, the number of clusters within a class is determined my the number of clusters above the threshold area. "K" is no longer fixed as in the case of original magnet loss. This gives better results especially when the number of sub clusters in each class are unbalanced.

6 Experiments on SSL dataset

Major python libraries used are:

- tensorflow
- sklearn (for TSNE)
- mlpack
- higra

6.1 Comparison of Magnet loss with and without iterative Watershed for 6 distinct classes

The SSL6 dataset contains data points belonging to 6 different classes. Experiment was to implement the magnet loss function on the SSL6 dataset so that it creates a baseline for our work. Further we compared the baseline with the inclusion of the watershed layer. Equation 5 of the magnet loss paper was implemented in a mini batch setting. Parameters chosen are:

m=6

d=8
k=1
embedding dimensions = 2 to avoid any changes in the representation due to TSNE.
optimizer = Adam with lr=1e-4
The results are shown in the table below.

Table 1: Results

SSL 6 class classification				
W/o WS		With WS		
Train	Val	Train	Val	
99.66	96.33	97.66	97	
99.75	96	99.83	97.33	
99.58	98.33	96.83	95	
99.83	96.66	98	96	
100	96	98.16	98	
99.91	97.66	99.41	98	
99.78833333	96.83	98.315	96.88833333	Average
0.156641842	0.9587074632	1.118190503	1.186093026	Stdev

Observations:

The iterated watershed separates cluster within a cluster. The number of clusters has to be chosen before hand similar to the original magnet loss algorithm with K means++ clustering.

On SSL 6 class classification problem, iterated watershed performs equally well on the validation set but with less overfitting on the training set. This implies that iterated watershed models the classification problem better than the Kmeans++ clustering method. However, the standard deviation is high on both train and test set suggesting that it is difficult to train a smooth model with watersheds.

6.2 Comparison of Magnet loss with and without iterative Watershed for 6 classes collapsed into 2 classes.

The 6 classes are grouped into 2 classes. That is, 3 classes in 1 class each. The parameters of the experiment are same as above. The results are given in the table below.

Table 2: Results

SSL 6 into 2 class classification				
W/o WS		With WS		
Train	Val	Train	Val	
99.83	96.33	99	97.66	
99.25	96	99.75	99.33	
99.5	95	98.66	97.33	
100	98.33	99.66	98.66	
99.58	96	99.83	97.33	
99.58	95	100	99.33	
99.62333333	96.11	99.48333333	98.27333333	Average
0.2618905624	1.22190016	0.5293266162	0.9527364099	Stdev

Observations:

The validation accuracy seems to improve by the use of iterated watershed. Clusters formed using watershed are comparatively more pure than the original magnet loss algorithm. However, more evidence is required to support this argument.

6.3 With and without L2 normalization

The features from the last layer of the network are tested with and without normalization before feeding them to the watershed layer.

Comparison results are shown below.

Table 3: Results

SSL 6 into 2 class classification				
Without L2 norm		With L2 norm		
Train	Val	Train	Val	
99.91	98.33	99	97.66	
99.75	99.33	99.75	99.33	
99.58	97	98.66	97.33	
99.08	97	99.66	98.66	
99	98	99.83	97.33	
99.91	98.66	100	99.33	
99.52875	97.91375	99.48333333	98.27333333	Average

Observations:

With L2 normalization of the final layer of the network, the validation accuracy increases by a small margin and the overfitting decreases.

6.4 Magnet loss with variable cluster complete watershed for 6 distinct classes

The number of clusters in a class is fixed in the original magnet loss algorithm as well as in the iterated watershed algorithm. But here, in the complete watershed implementation the number of clusters is set adaptively while training.

Table 4: Results

SSL 6 classes split Variable WS		
Train	Val	
99.33	92	
99.83	92.33	
99.41	92	
99.66	92.67	
99.41	92.67	
99.66	91.33	
99.5	92	
98.66	91.67	
99.08	91.67	
98.83	91.67	
99.337	92.001	Average
0.3757082201	0.4450081148	Stdev

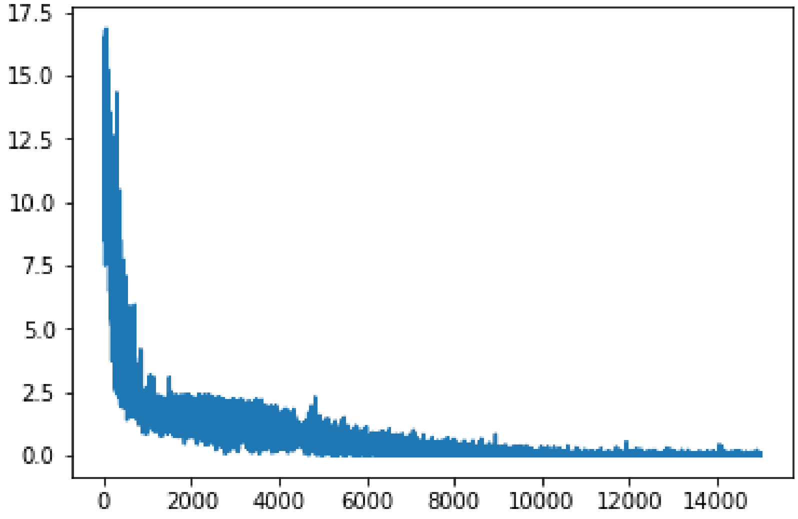


Figure 1: Error with iterations

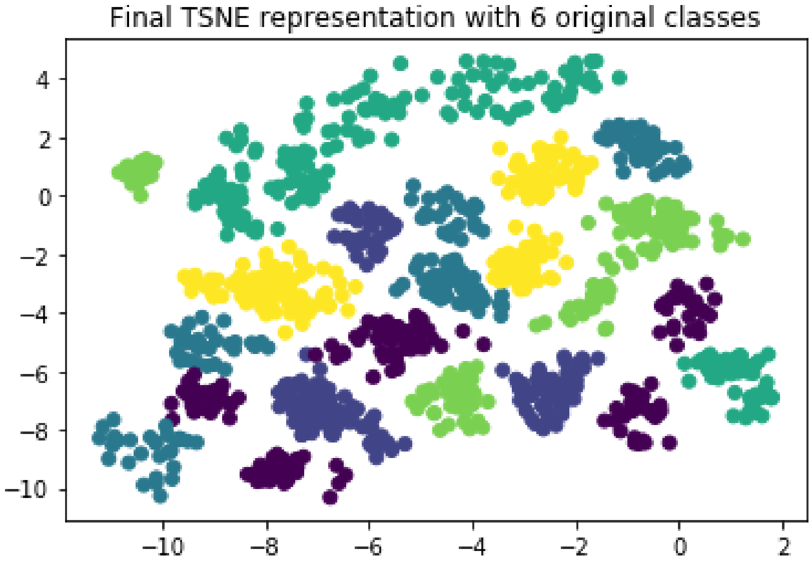


Figure 2

Observations:

The table shows that the average validation accuracy is around 92% even though the train accuracy reaches 99%. There is a large amount of overfitting taking place.

Figure 1. shows the error versus number of iterations. It becomes smooth and flat as number of iterations increases (after about 10,000 iterations).

Figure 2. shows the representation of the neural network at the end of the training process. The sub clusters appear to be pure and well separated from other class clusters. They are far apart from clusters of the same class depicting intra class variation and inter class similarity.

6.5 Magnet loss with variable cluster watershed for 6 classes collapsed into 2 classes

Table 5: Results

SSL 4-2 split Variable WS		
Train	Val	

SSL 4-2 split Variable WS		
99.416	94	
99.833	94.66	
99	92.33	
100	94.66	
99.75	93.33	
99.75	94.66	
100	93	
99.66	94.33	
98.33	93.66	
99.41	93.33	
99.51	93.796	Average
0.515679153	0.8029002982	Stdev

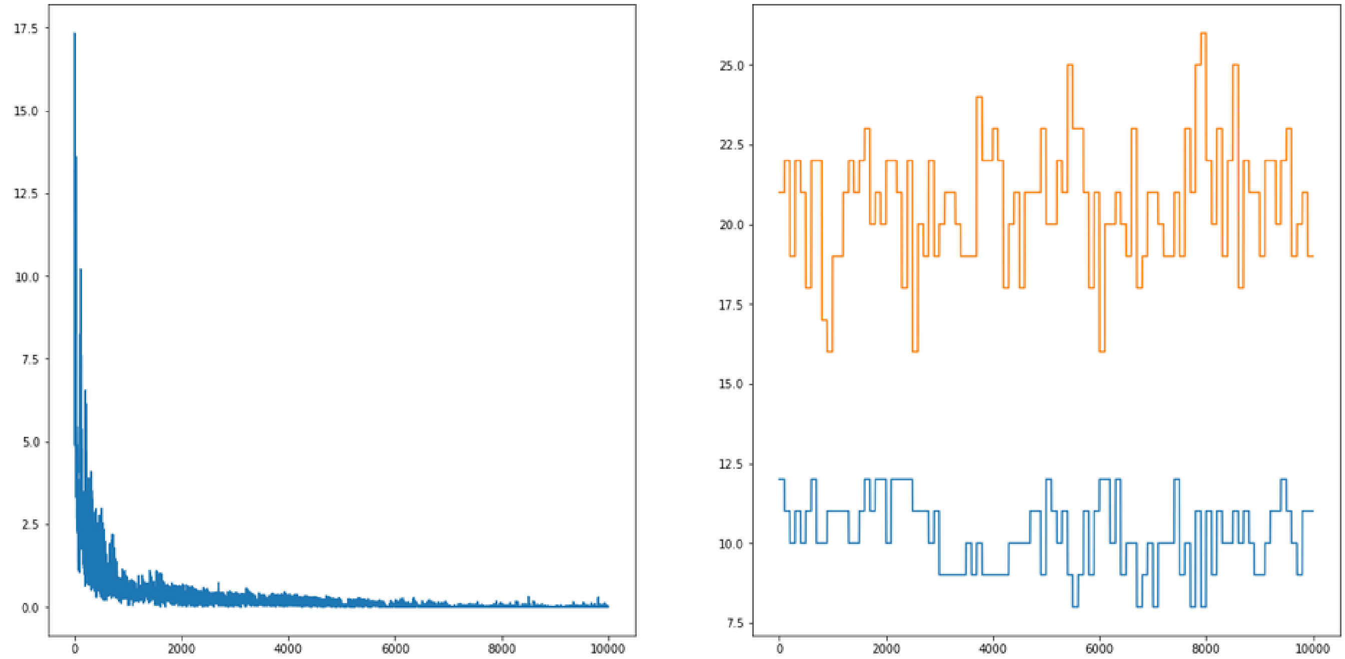
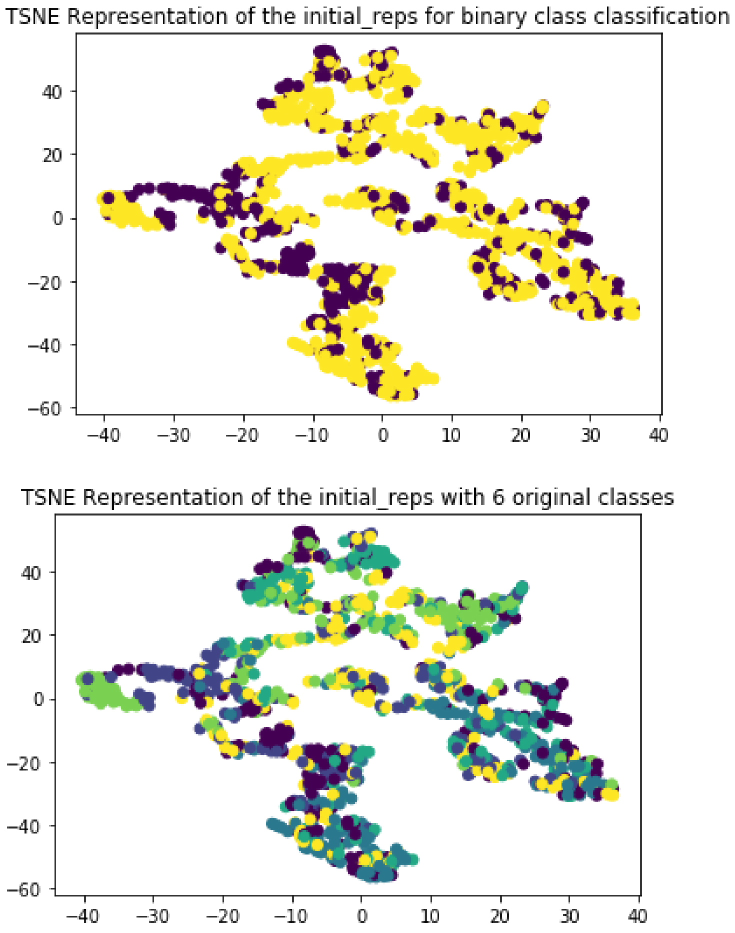
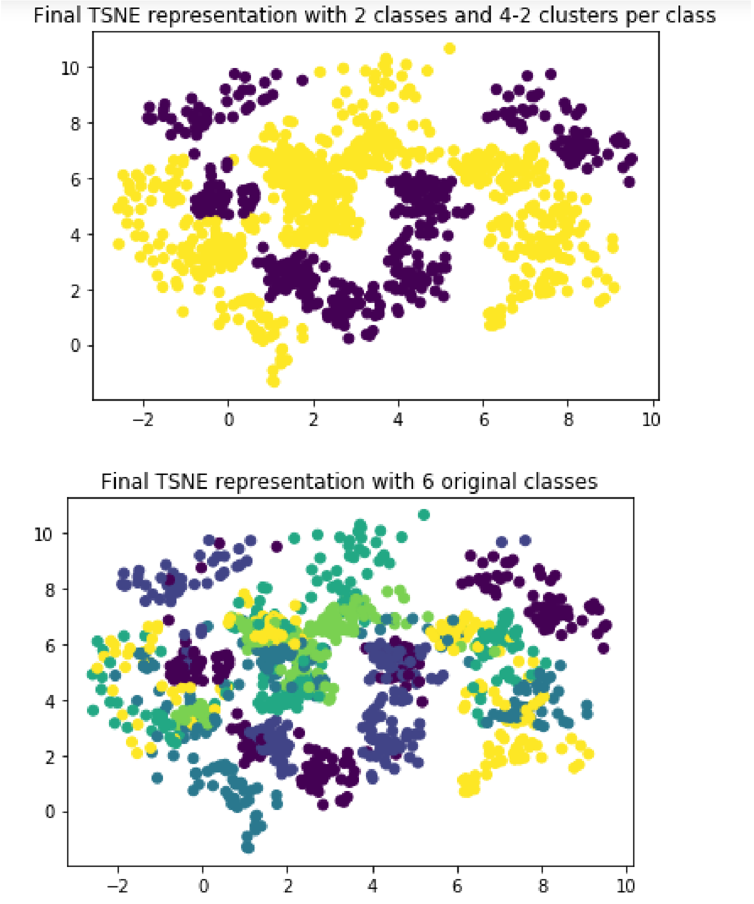


Figure 3: (a) Error v/s iterations. (b) Number of clusters with iterations.


TSNE Representation of the initial_reps for binary class classification

TSNE Representation of the initial_reps with 6 original classes

Figure 4


Final TSNE representation with 2 classes and 4-2 clusters per class

Final TSNE representation with 6 original classes

Figure 5

Observations:

From Table 5, we see that the average train accuracy is 99.5% while average validation accuracy is 93.7% which suggest subsequent amount of overfitting taking place. The standard deviation is less than one which is negligible.

figure 3(a) shows the train error versus the number of iterations graph which smoothens down after few iterations.

Since the number of sub clusters 'k' is learned adaptively, Figure 3(b) depicts the number of sub clusters within each class as a function of iterations. The sub clusters fluctuate a lot but within a certain range. The mean number of sub clusters is about 20 and 10 sub clusters in the yellow and blue classes respectively. This is also in agreement with the initially chosen split of 4:2 clusters in the two classes respectively. This means that the network is correctly able to learn the proportion in which the data is distributed without manually specifying the number of sub clusters 'k' to the network.

Figure 4. and 5. shows the initial representation and the final representation of the network respectively. 4(a) and 5(a) are color coded to representation the 2 major classes which are to be classified where as 4(b) and 5(b) codes the original 6 classes which are to be retrieved. As one can infer from Figure 5(b), Many of the sub clusters are well segregated, that is many of them are pure. Where as some of them are intermixed with one another. There are multiple sub clusters with the same color spread across the graph. This suggests the intra-class variation. Also some clusters belonging to different classes are close to one another which suggests inter-class similarity.

6.6 Experiments with slow MST update

Table 6: Results

SSL 6 classes split Variable WS slow MST update		
Train	Val	
99.75	91.66	
99.5	92.33	
99.41	92.66	
98.91	92.66	
99.08	93	
99	94.33	
98.91	94.33	
98	93.66	
99.16	92.66	
98.66	93	
99.038	93.029	Average
0.486479416	0.8535084456	Stdev

7 Experiments with Synthetic Dataset

The synthetic dataset comprises of 6 Gaussian blobs in 8 dimensions with 4 blobs belonging to one class and 2 blobs belonging to the second class. This is a simple dataset to verify if the algorithm works correctly and is able to generate pure clusters. The TSNE representation of the dataset is given in Figure 6.

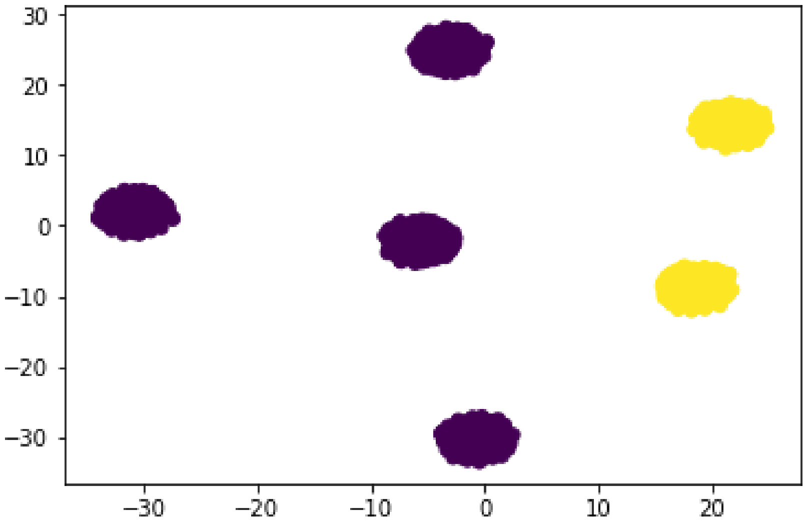


Figure 6: TSNE of Synthetic dataset.

7.1 Variable cluster watershed for 6 distinct classes

The same algorithm of magnet loss with variable cluster watershed is implemented on the synthetic dataset with 6 distinct classes.

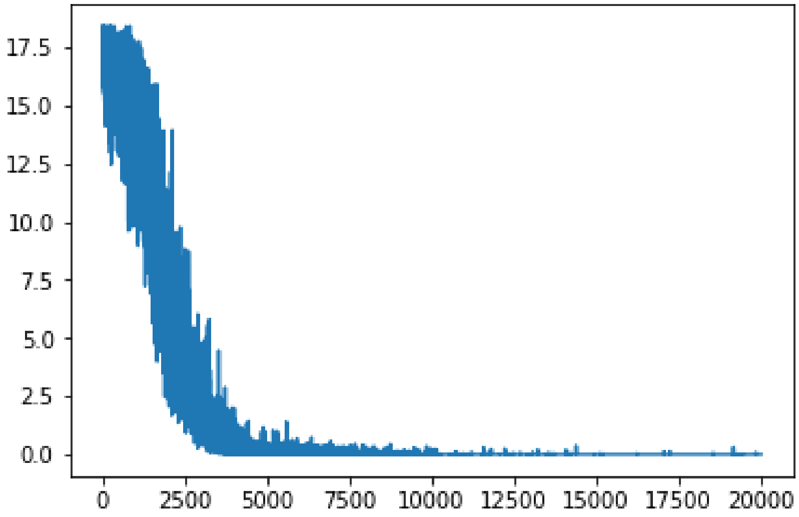


Figure 7: Error /vs iterations

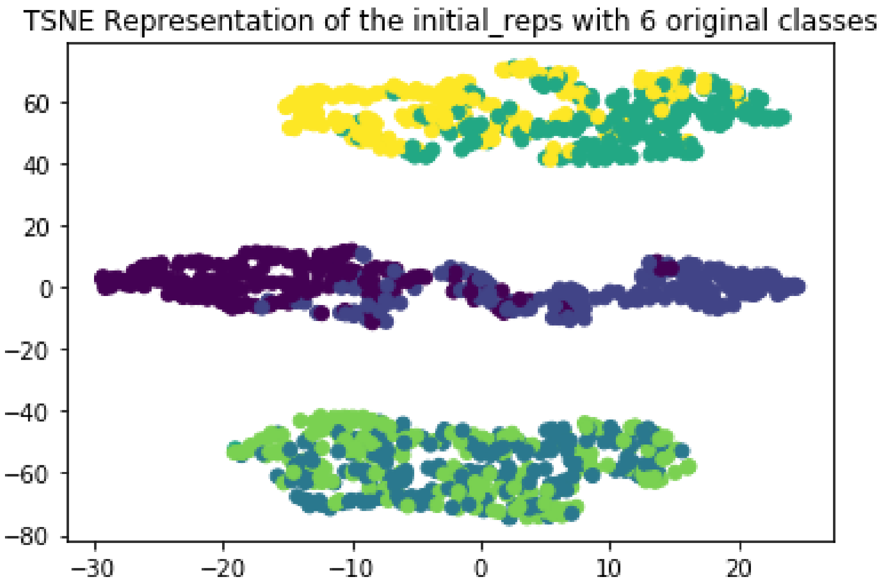


Figure 8

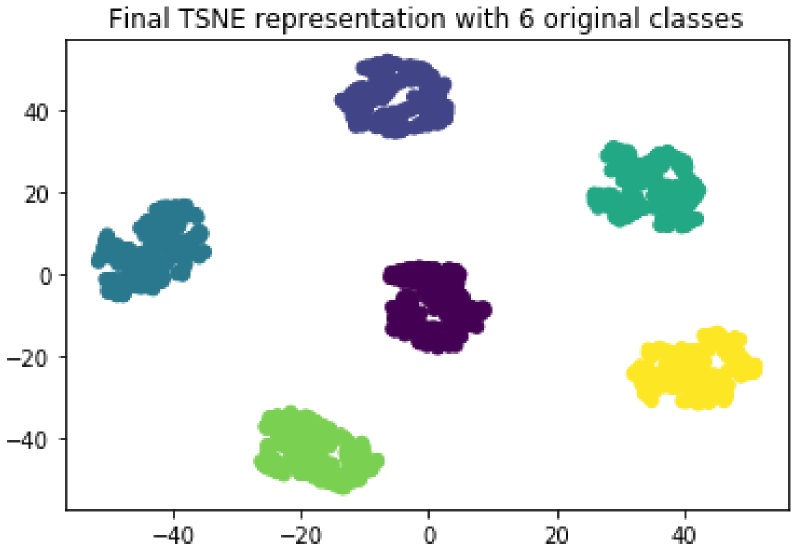


Figure 9

Observations:

Since the dataset is simple, the train and validation accuracy reaches 100% after a certain number of epochs and the train and validation error also goes to zero. The train error versus iterations is shown in Figure 7.

Figure 8. shows the initial representation generated by the network where as Figure 9. shows the final representation color coded with 6 original classes. As one can see, the 6 classes are well separated by the algorithm.

7.2 Variable cluster watershed for 6 classes collapsed into 2 classes

The algorithm is implemented on synthetic dataset with 2 classes, one having 4 sub clusters and other having 2 sub clusters respectively.

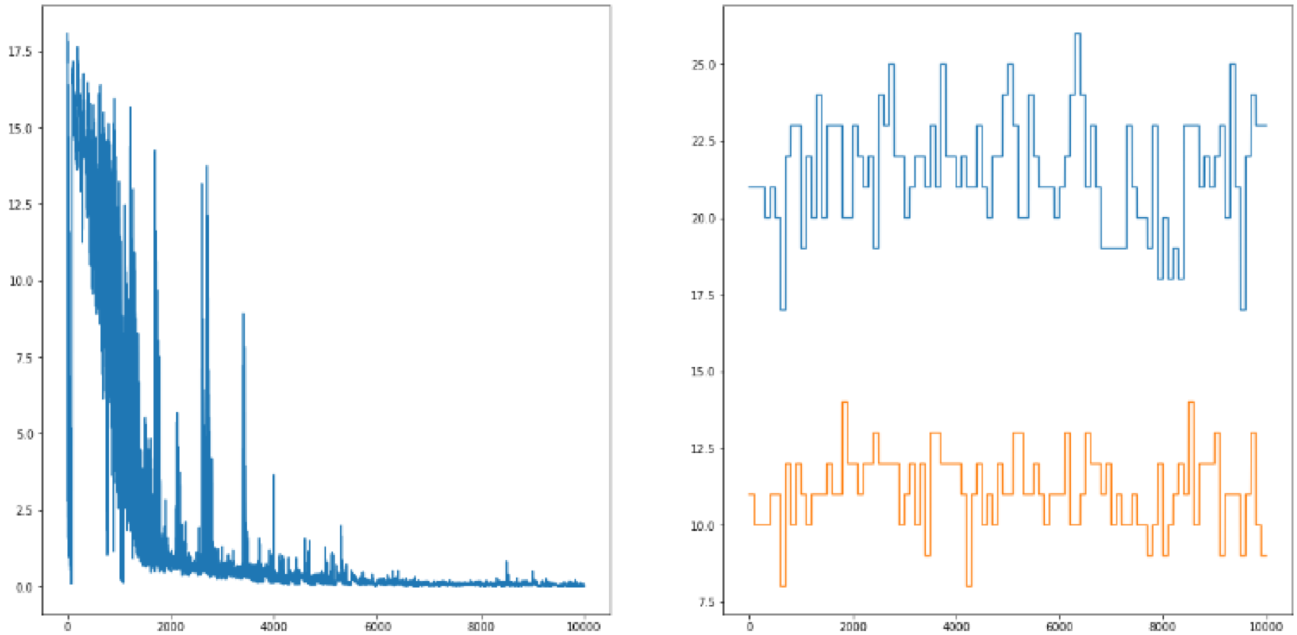


Figure 10: (a) Error v/s iterations. (b) Number of clusters with iterations.

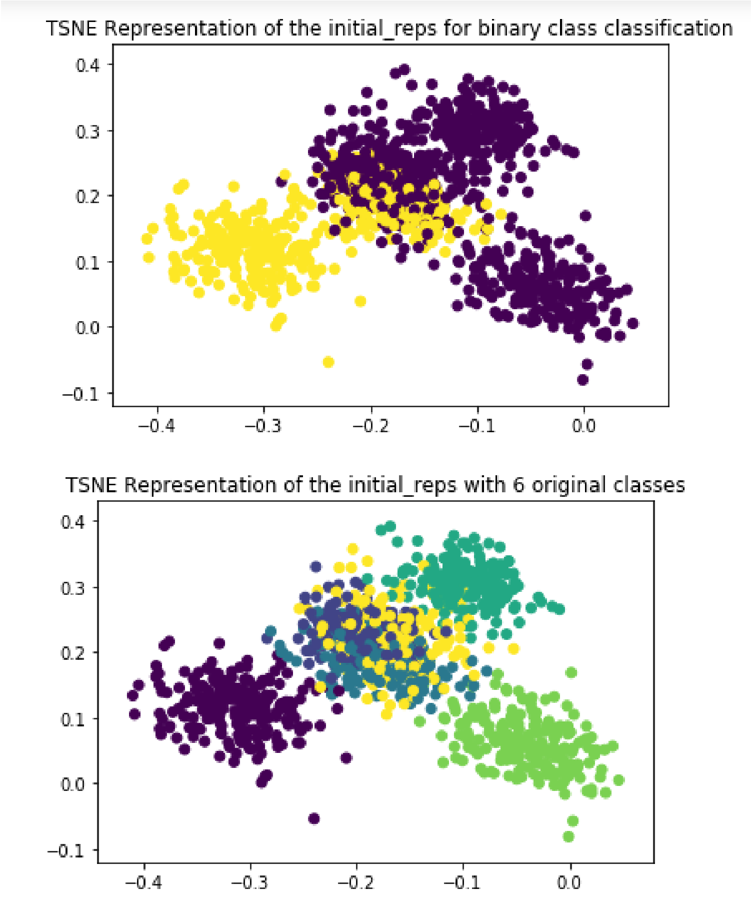


Figure 11

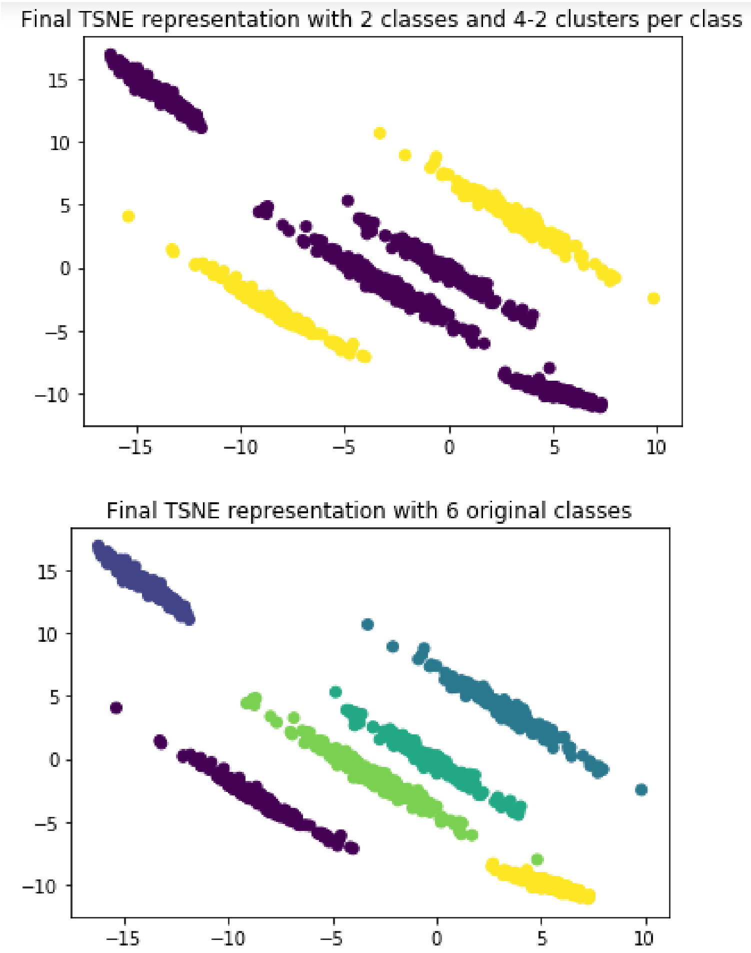


Figure 12

Observations: Since the synthetic dataset is simple, 100% train and validation accuracy is obtained.

Figure 10(a). shows error v/s iterations. The oscillations appear to dampen out after 6000 iterations. In Figure 10(b), the plots of number of sub clusters with iterations is shown for the two classes. The values fluctuate a lot but within a certain limit. The average ratio of the blue to yellow sub clusters is around 2:1 which is same as the original ratio of 4 clusters to 2 clusters in the classes respectively.

Figure 11(a) and (b) shows the initial representation of the network for 2 classes and original 6 classes respectively.

Figure 12(a) and (b) shows the final representation of the network for 2 classes and original 6 classes respectively. As one can see in Figure 12(b), the sub clusters are pure and they appear to be well separated from the opposite class clusters.

8 Conclusion

Integrating watershed with magnet loss has many advantages over the original magnet loss paper.

- The number of clusters which is assumed as a fixed parameter in the original paper is no longer a constraint in our work.
- On the synthetic dataset, the clusters obtained are very pure in terms of classes while on the SSL dataset, the clusters are considerably pure in some regions while mixed in rest of the regions.
- Some changes are required in the magnet loss with complete watershed code in order to make these clusters pure. One of the changes can be adding a regularization term in order to add restriction on the closeness of the clusters. Other suggested change is to include the clusters of the same class in the magnet loss equation and to set a lesser margin to these clusters as compared to the clusters belonging to different classes. Further experiments need to be conducted on this idea.
- Future Work: Further application of this method can be in the field of 3D point clouds. This area is untouched in the literature of 3D point clouds classification and segmentation. Point clouds can be represented as a graph. Since this is a generic algorithm on graphs, it should also work for point clouds segmentation tasks in the similar way graph CNNs work on point clouds.

References

[1] Oren Rippel, Manohar Paluri, Piotr Dollár, and Lubomir D. Bourdev. Metric learning with adaptive density discrimination. In Yoshua Bengio and Yann LeCun, editors, 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings, 2016. URL http://arxiv.org/abs/1511.05939.

[2] Aditya Challa, Sravan Danda, B Daya Sagar, Laurent Najman. Watersheds for Semi-Supervised Classification. IEEE Signal Processing Letters, Institute of Electrical and Electronics Engineers, In press. hal-01977705v2

[3] Laurent Najman, Jean Cousty. A graph-based mathematical morphology reader. Pattern Recognition Letters 2014.

[4] Laurent Najman, Fernand Meyer. A short tour of mathematical morphology on edge and vertex weighted graphs. Lezoray, O. and Grady, L. Image Processing and Analysis with Graphs: Theory and Practice, CRC Press, pp.141-174, 2012, Digital Imaging and Computer Vision, 9781439855072. hal-00741976

[5] K. Maninis, J. Pont-Tuset, P. Arbelaez, and L. V. Gool, “Convolutional oriented boundaries: From image segmentation to high-level tasks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 4, pp. 819–833, 2018.