CS7015 : DEEP LEARNING

PROGRAMMING ASSIGNMENT 5

- RBM -

April 30, 2018

Student ID:
Namida M - EE15B123
Ganga Meghanath - EE15B025

Contents

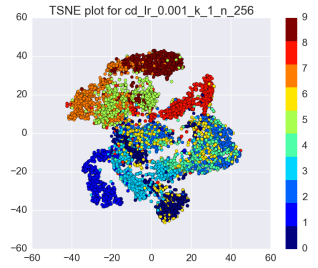
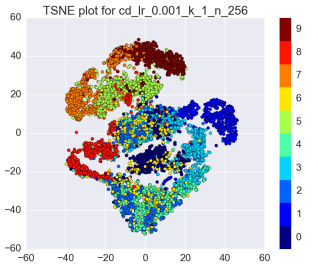
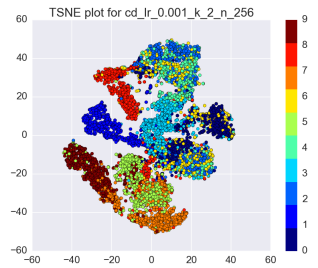
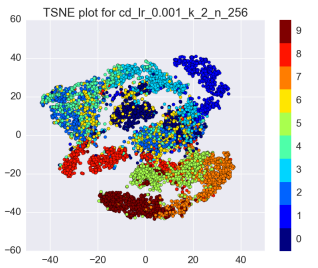
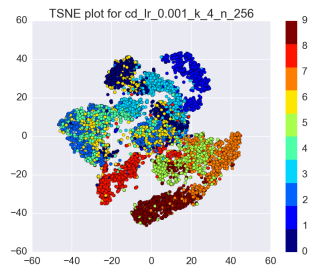
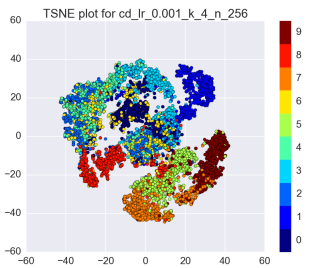
1 INTRODUCTION

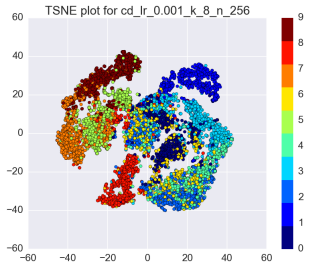
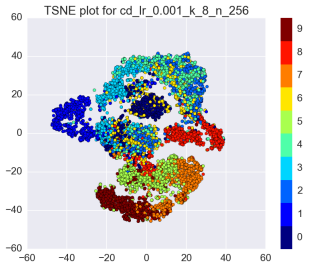
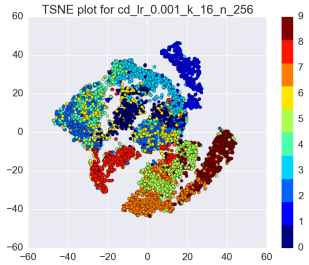
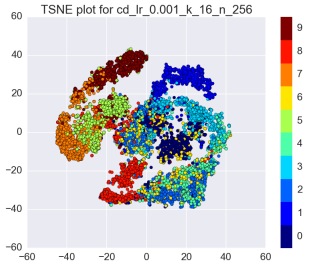
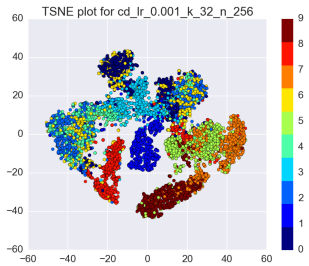
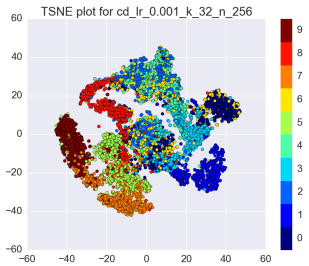
RBMs can be used to learn hidden representations (h) from the raw features (V). The objective of the assignment is to train RBMs using the Contrastive Divergence (CD) algorithm. The training data used is Fashion MNIST. The 784 dimensional (V) dataset is thresholded into binary data (using a threshold of 127) to learn a n-dimensional hidden representation (h).

2 T-SNE PLOTS

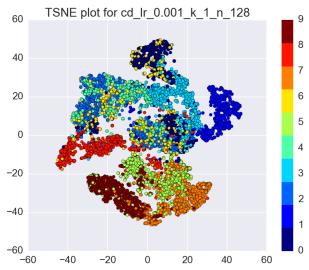
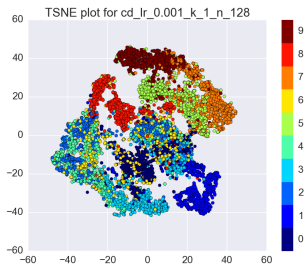
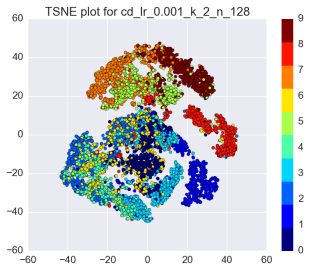
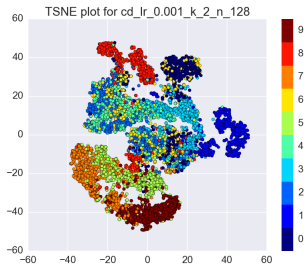
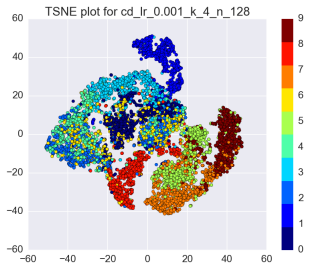
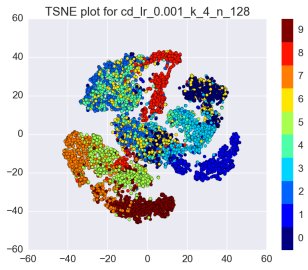
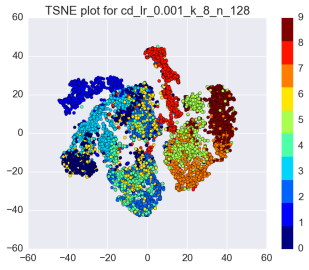
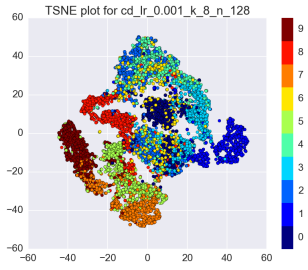
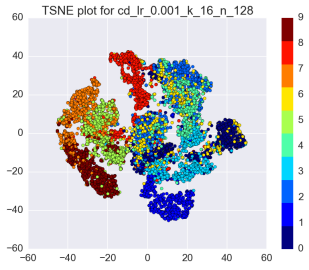
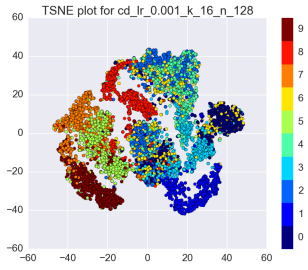
2.1 lr = 1e-3, n = 256

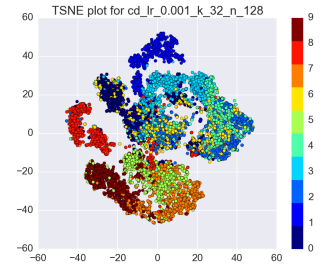
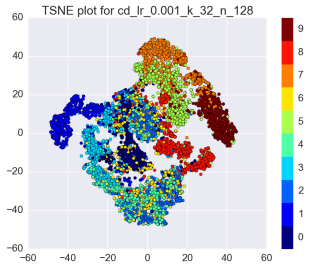
The t-SNE plots for train and test data has been depicted below :

lr	n	k	Train t-SNE	Test t-SNE
1e-3	256	1		
1e-3	256	2		
1e-3	256	4		

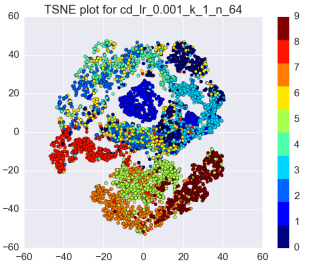
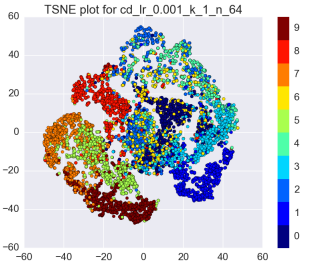
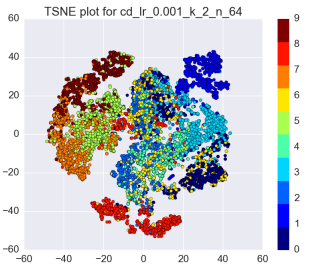
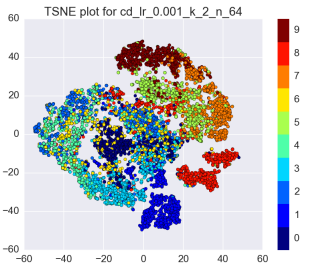
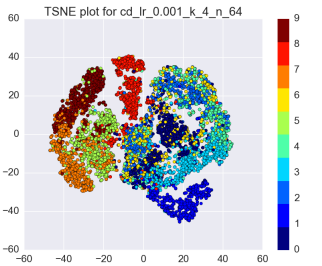
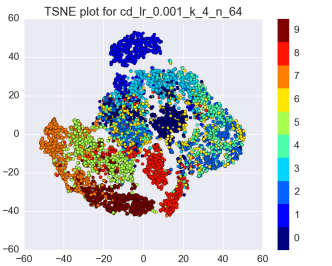
lr	n	k	Train t-SNE	Test t-SNE
1e-3	256	8		
1e-3	256	16		
1e-3	256	32		

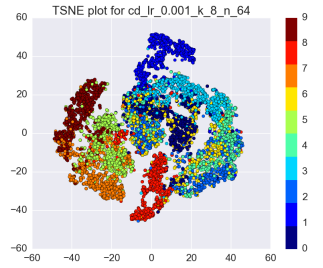
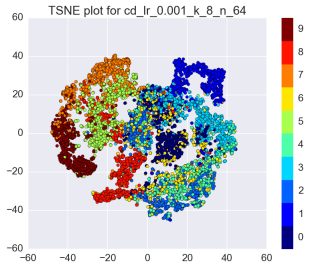
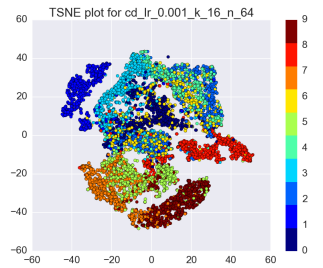
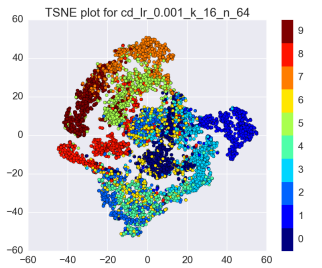
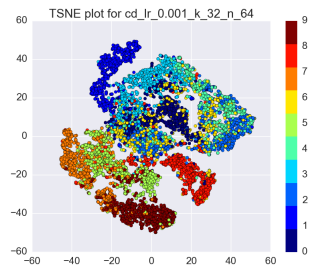
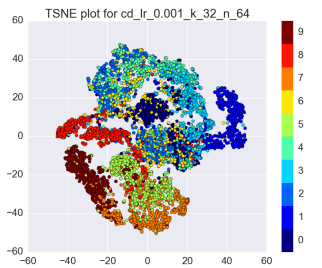
2.2 lr = 1e-3, n = 128

lr	n	k	Train t-SNE	Test t-SNE
1e-3	128	1		
1e-3	128	2		
1e-3	128	4		
1e-3	128	8		
1e-3	128	16		

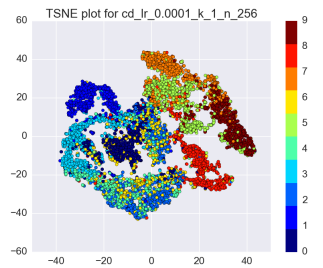
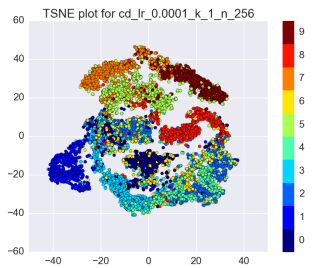
lr	n	k	Train t-SNE	Test t-SNE
1e-3	128	32		

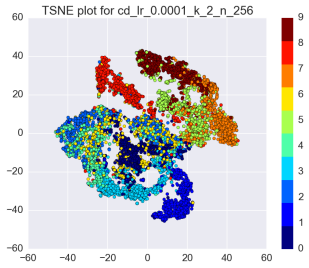
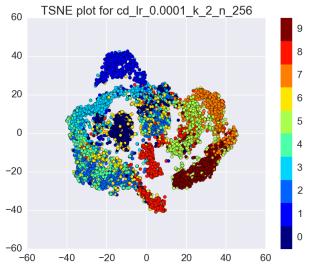
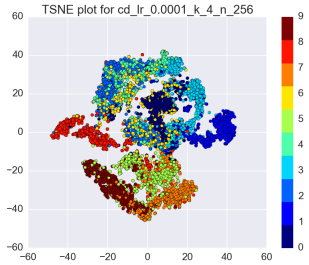
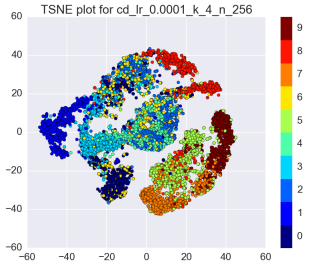
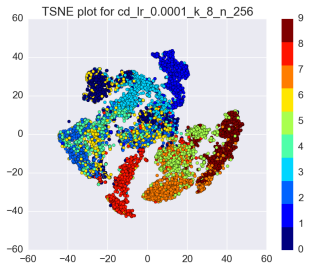
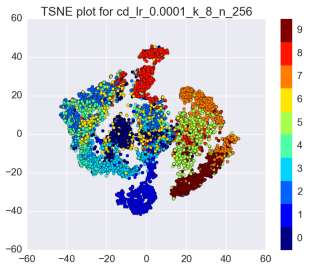
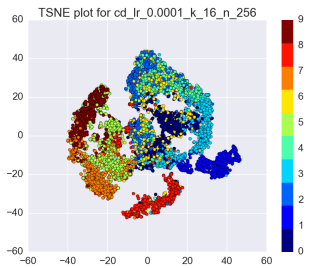
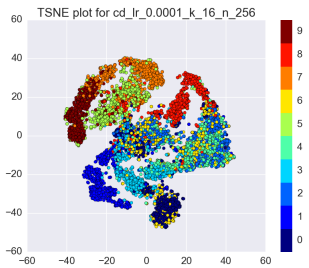
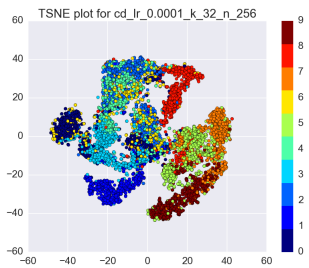
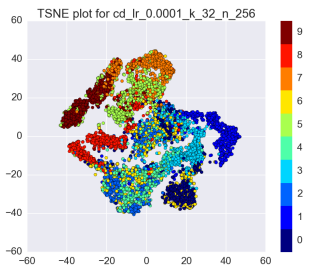
2.3 lr = 1e-3, n = 64

lr	n	k	Train t-SNE	Test t-SNE
1e-3	64	1		
1e-3	64	2		
1e-3	64	4		

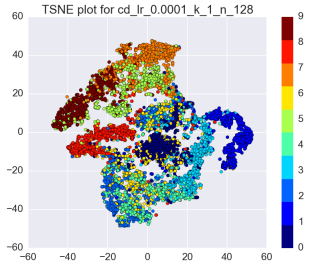
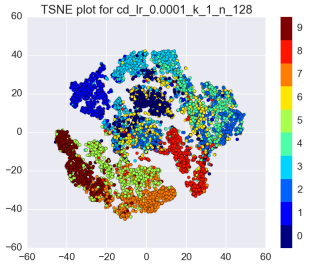
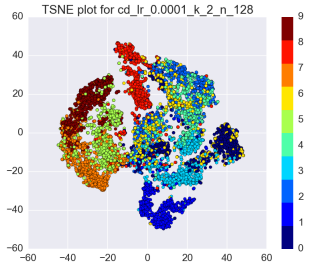
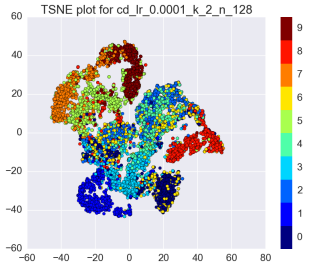
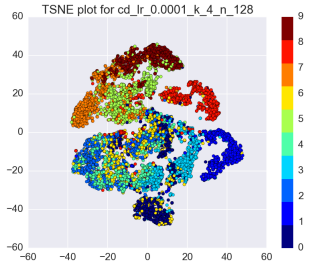
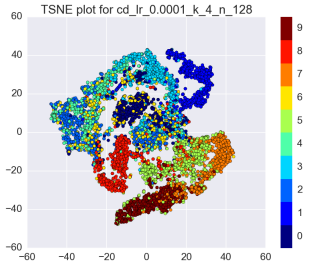
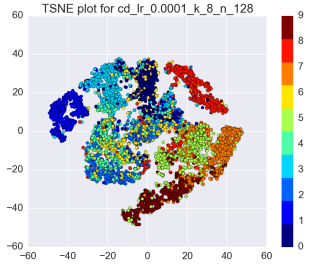
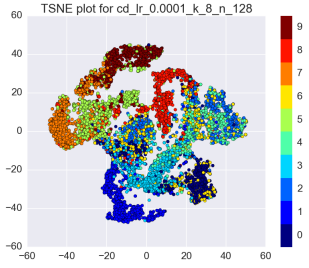
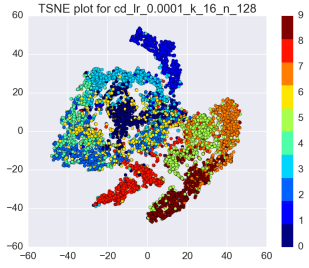
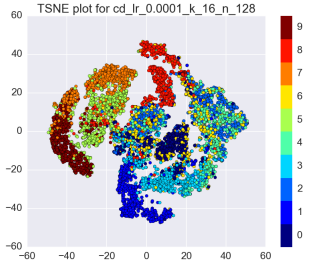
lr	n	k	Train t-SNE	Test t-SNE
1e-3	64	8		
1e-3	64	16		
1e-3	64	32		

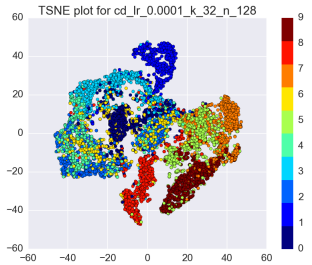
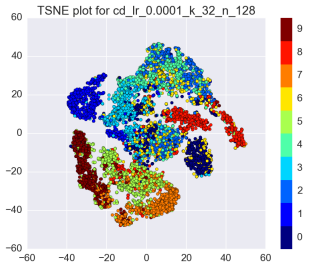
2.4 lr = 1e-4, n = 256

lr	n	k	Train t-SNE	Test t-SNE
1e-4	256	1		

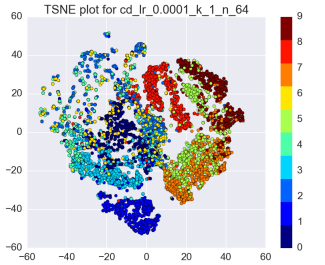
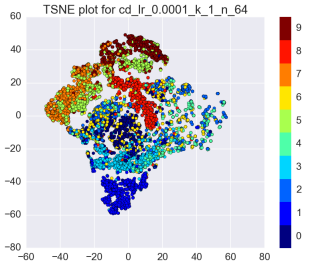
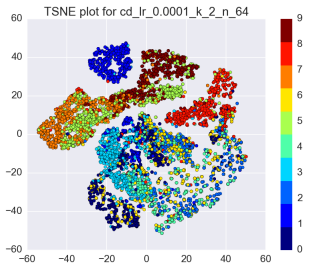
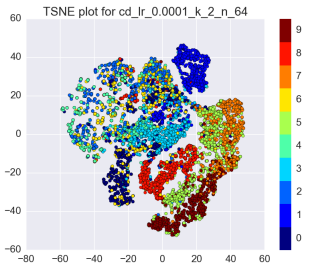
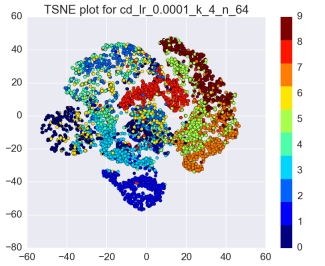
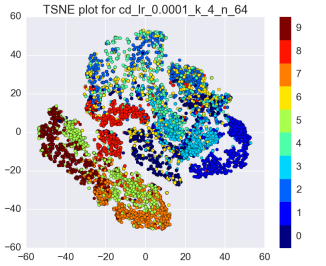
lr	n	k	Train t-SNE	Test t-SNE
1e-4	256	2		
1e-4	256	4		
1e-4	256	8		
1e-4	256	16		
1e-4	256	32		

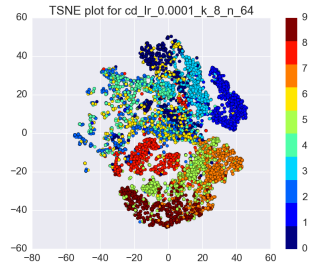
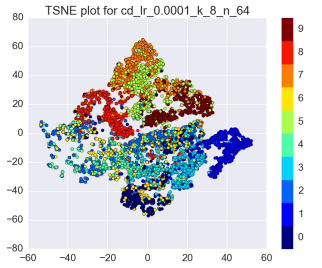
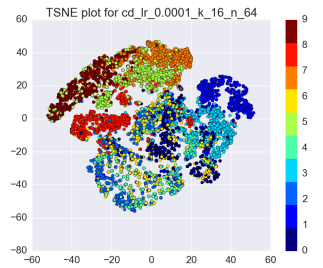
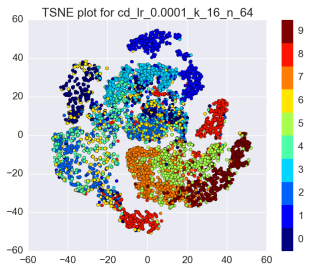
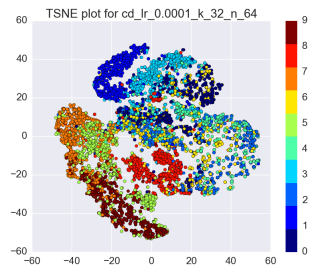
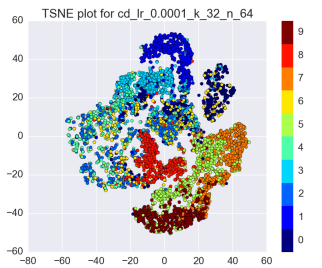
2.5 lr = 1e-4, n = 128

lr	n	k	Train t-SNE	Test t-SNE
1e-4	128	1		
1e-4	128	2		
1e-4	128	4		
1e-4	128	8		
1e-4	128	16		

lr	n	k	Train t-SNE	Test t-SNE
1e-4	128	32		

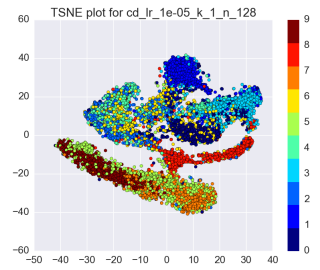
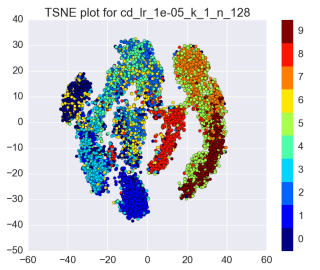
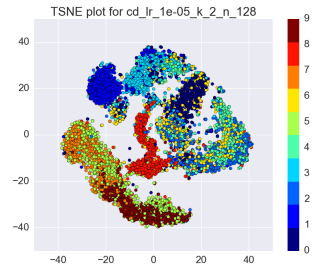
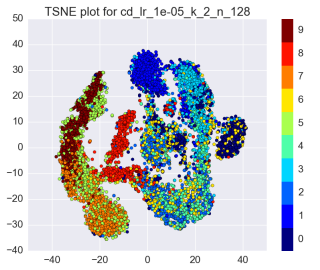
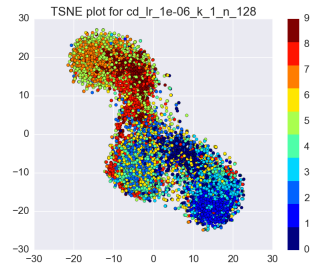
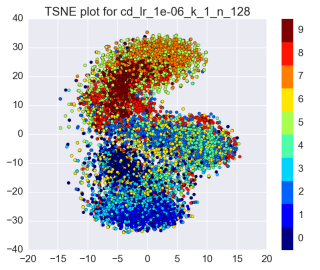
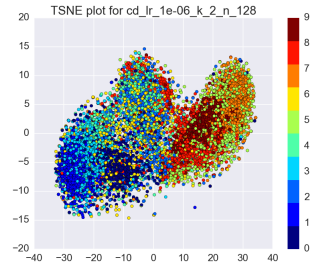
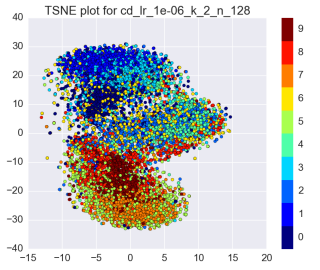
2.6 lr = 1e-4, n = 64

lr	n	k	Train t-SNE	Test t-SNE
1e-4	64	1		
1e-4	64	2		
1e-4	64	4		

lr	n	k	Train t-SNE	Test t-SNE
1e-4	64	8		
1e-4	64	16		
1e-4	64	32		

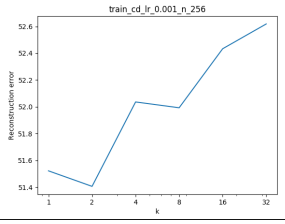
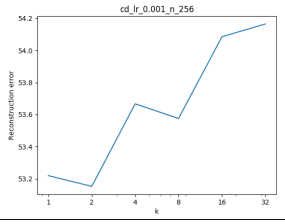
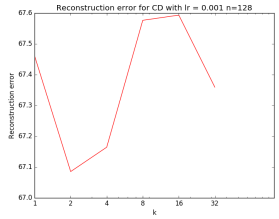
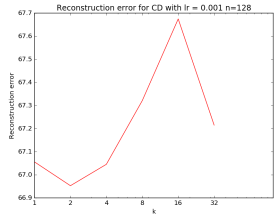
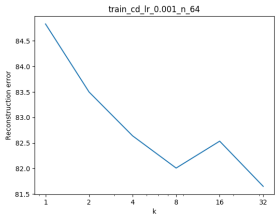
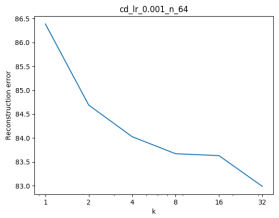
On observing the TSNE plots we note that the points mostly cluster according to their class. Similar classes tend to overlap. The tabulations are all for 20 epochs, after which, the based models were restored and trained using early stopping. On increasing the n value, we note an decrease in reconstruction error. (steeper for smaller values of n, and magnitude of decrease decreases as we increase the n value. Naturally, this makes sense, as in the limiting case of n=784, we achieve perfect reconstruction. At lower n values, we are restricting the hidden variables ability to capture information about the images. Around n=64 we observe reconstruction errors in the range of 80s, at n=128 around 60s, and further decreases to around 50s for n=256.

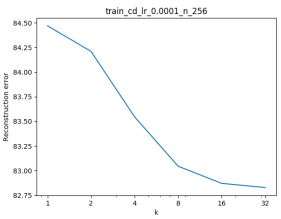
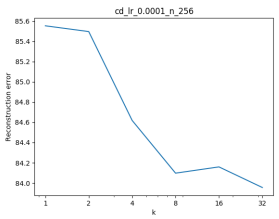
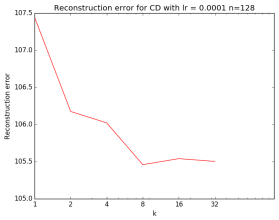
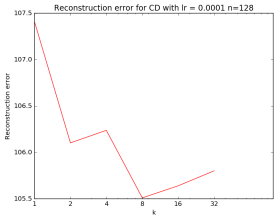
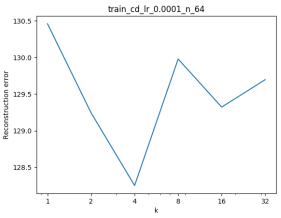
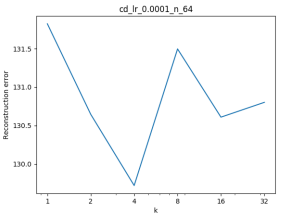
2.7 lr = (1e-5, 1e-6), n = 128

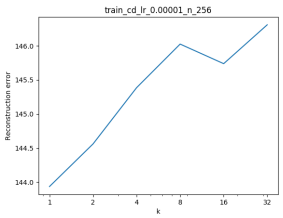
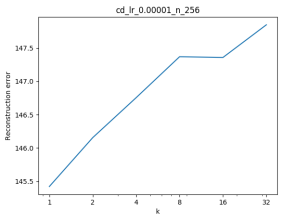
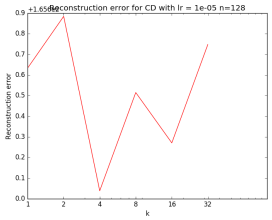
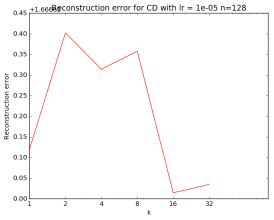
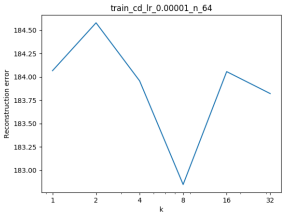
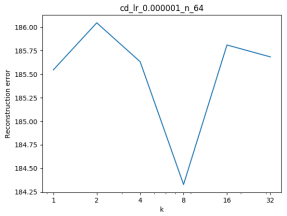
lr	n	k	Train t-SNE	Test t-SNE
1e-5	128	1		
1e-5	128	2		
1e-6	128	1		
1e-6	128	2		

As the t-SNE plots were not very well separated for the cases were learning rate was less that 1e-4, such cases were avoided further on while plotting.

3 EFFECTS OF K ON GIBBS CHAIN

lr	n	Train Plot	Test Plot
1e-3	256		
1e-3	128		
1e-3	64		

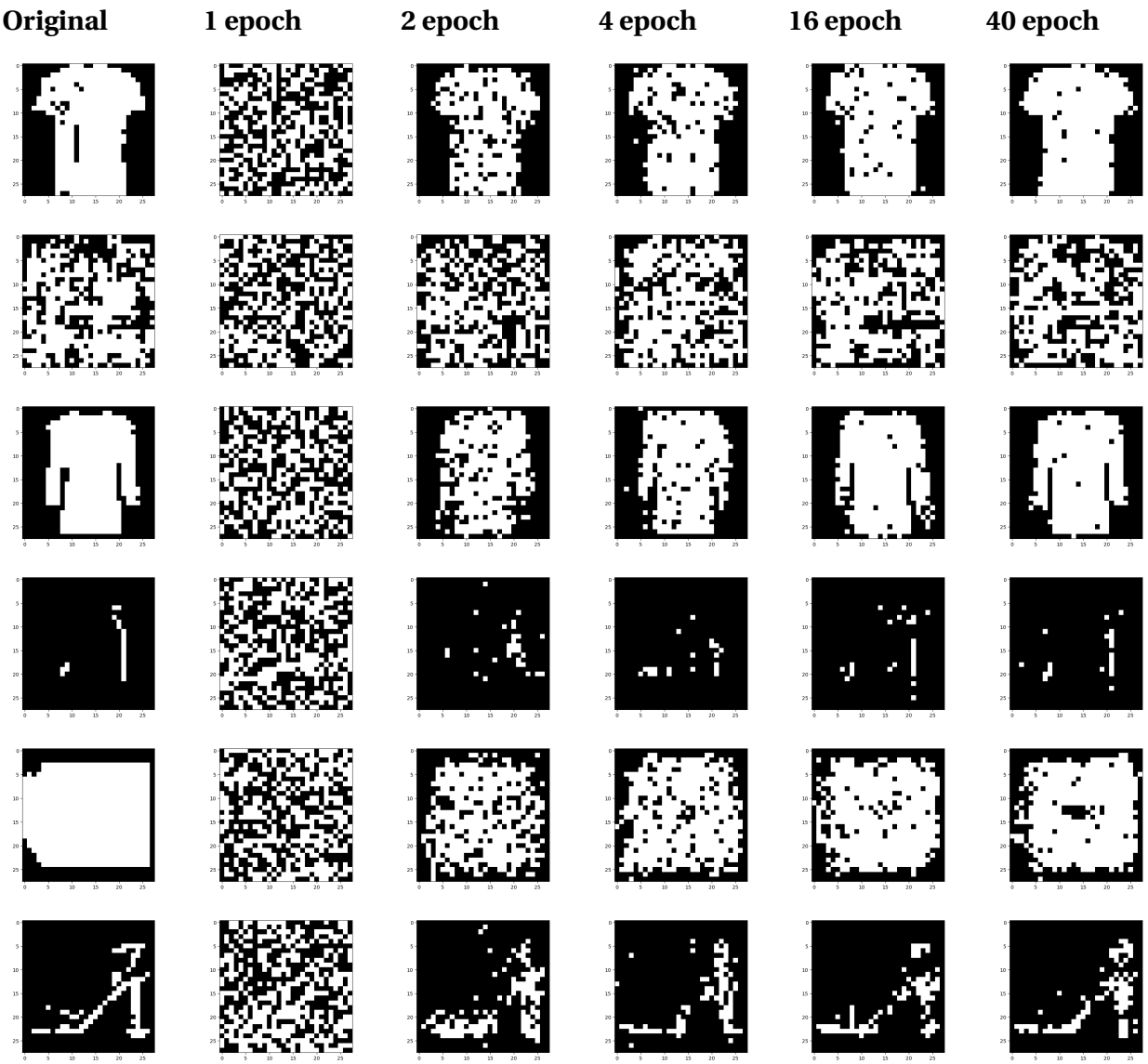
lr	n	Train Plot	Test Plot
1e-4	256		
1e-4	128		
1e-4	64		

lr	n	Train Plot	Test Plot
1e-5	256		
1e-5	128		
1e-5	64		

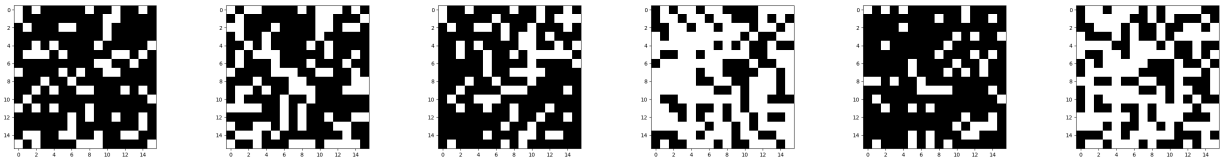
We note a non-linear trend on using different k values (note this could be because we trained it to a certain fixed point for all models) . Ideally as number of iterations tends to infinity, we should expect them all to arrive at similar models. The rate of convergence depends on k. We notice faster convergence for moderate k values (in terms of time and iterations) . If we fix the number of iterations , larger k values perform better, however, there is a tradeoff, as the amount of time taken for each step increases with larger k.

4 PLOTS OF SAMPLES GENERATED BY GIBBS CHAIN

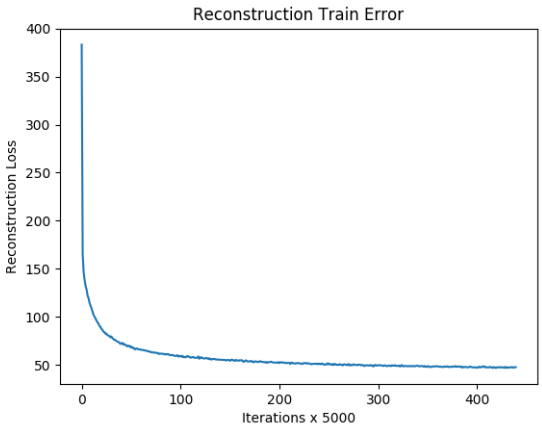
The best model that was obtained using a learning rate of 0.001 and hidden representation of size 256 and k value as 2 (as can be seen from the plot above), was ran for 40 epochs and the reconstructed images of certain validation samples were saved as examined as follows :

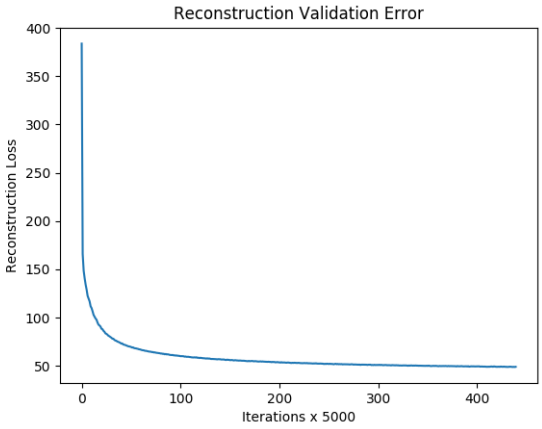


The corresponding hidden representations of the images is as shown below :



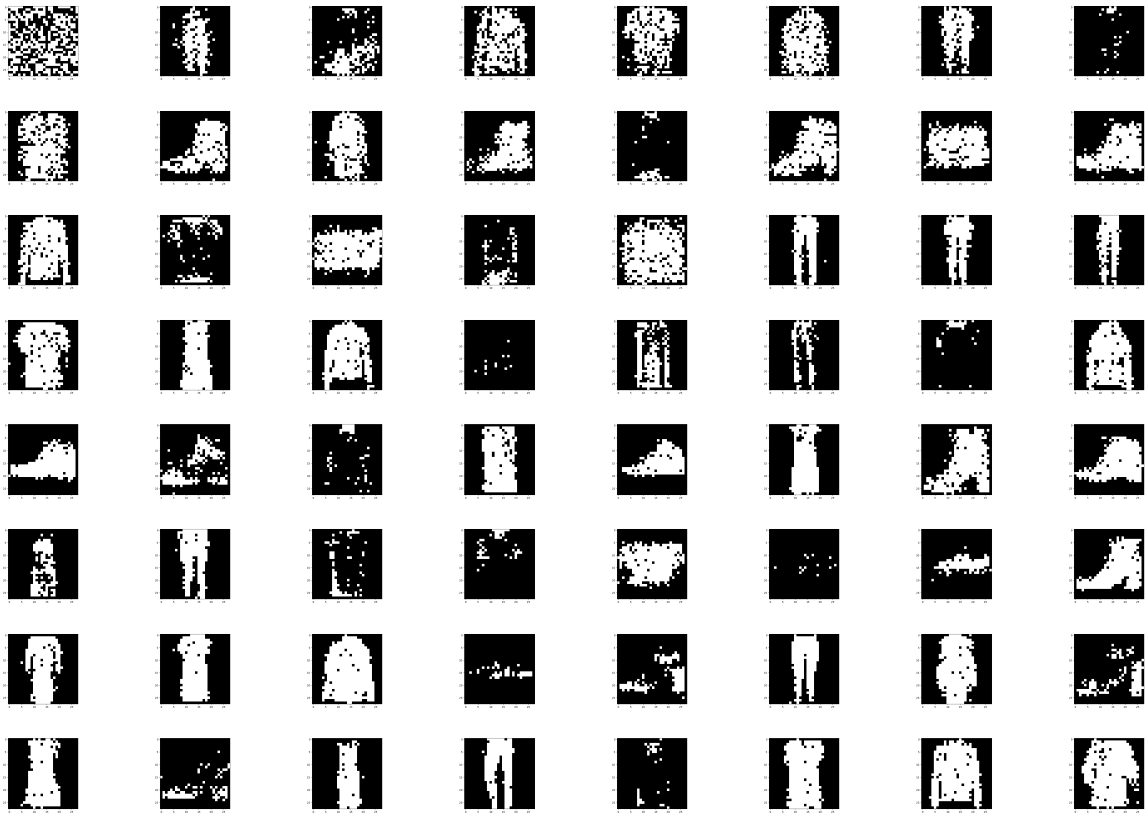
The average reconstruction error for train and validation datasets were recorded after every 5000 steps to measure convergence.





Initially the sampled images are meaningless, but over time/iterations, they start resembling blurry versions of the images from the training data, and around convergence they look like very convincing samples from the provided data.

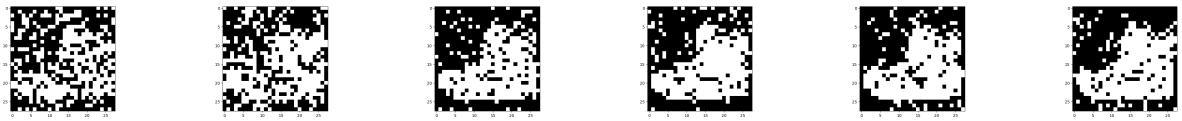
As can be seen from the above graph, the reduction in reconstruction error is quite small beyond 100×5000 iterations. Hence, the m value was fixed to be that and the reconstructed images used while training was plotted on an 8×8 grid.



5 GIBBS SAMPLING

Withing the first 100 iterations ,(when sampled to large k values around 1000) we started getting meaningful images. As we approach convergence, the number of time steps required to get meaningful images decreases .(this is because our weights and biases are accurate, and we are quickly able to converge to the true distribution even after starting form a random point.

Despite starting from a random v value, we were able to achieve the following reconstructions for it during the training process :



Hence, we decided to go with the same random v and analyse the reconstruction with varying values of k and number of iterations.

From the above observations, it is noted that the sample seems to generate a boot each time. This is understandable as the random seen for initialising the vector in every run remains the same over time. Also, convergence is probably quicker for the same reason.

The number of steps required for convergence of Gibbs Sampling was found to be much more than RBM and the time taken to run the code was also found to be larger.

Traversing the matrix shown above row-wise gives increasing iterations (500 to 20000) and decreasing k (1024×16 to 2). We can see that the images generated are almost the same. Hence we conclude that the value of k required reduces as SGD reaches convergence.

Bibliography

[1] Mitesh M Khapra. *CS7015 Deep Learning: Lecture 18-20*, Indian Institute of Technology Madras, 2018