Report for Programming Assignment-4
CS7015 - Deep Learning

April 26, 2019

Team Mate 1 : Raghul R
Roll No : CS16B021
Team Mate 2 : E Santhosh Kumar
Roll No : CS16B107
Team Name : PRML pt. 2

1 Question 1

Figures 1 to 4 shows the plots of the learned representations (plotted in a 2-D space using t-SNE) for three different values of number of latent variables (n) - 50, 100, 200, 500 respectively. We use $k = 1$ in all four cases.

The different classes are:

- Class 0: T-shirt/top
- Class 1: Trouser/pants
- Class 2: Pullover shirt
- Class 3: Dress
- Class 4: Coat
- Class 5: Sandal
- Class 6: Shirt
- Class 7: Sneaker
- Class 8: Bag
- Class 9: Ankle boot

As the number of latent variables is increased, the variance within the cluster corresponding to each class decreases and the clusters become more condensed (this can clearly be observed in the transition from n = 50 to n = 100). Thus,

more information about the classes is captured, resulting in lower intra-cluster variance and higher inter-cluster variance. There is an overlap between classes with similar-looking objects (for example, "shirt" and "t-shirt/top", "dress" and "coat", "sandal" and "sneaker"). While some of these overlaps are resolved as the number of latent variables is increased (for example, "sandal" and "sneaker" get separated as n is increased), some classes are too close together for proper differentiation even with large n ("shirt" and "t-shirt/top", "dress" and "coat" still overlap in the case of n = 500).

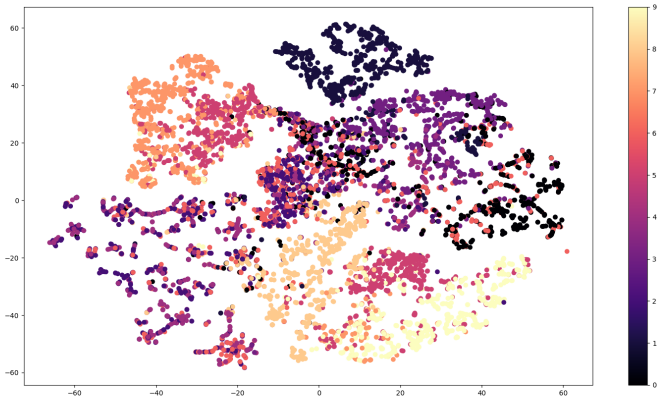

Figure 1: n = 50

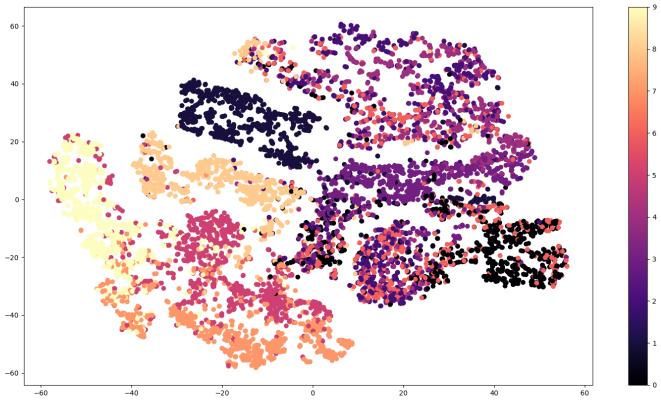

Figure 2: n = 100

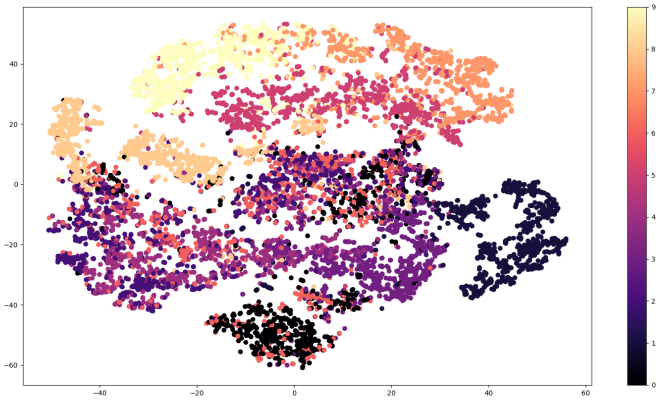

Figure 3: n = 200

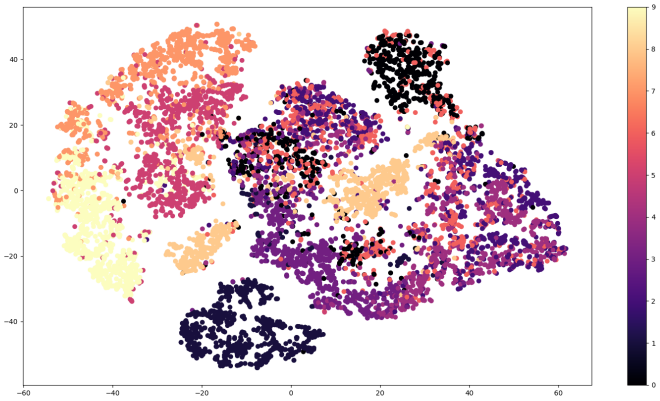

Figure 4: n = 500

2 Question 2

- Figure 5 shows the average reconstruction loss (Sum of square error between V_d and $\tilde{V}$) vs iterations for various values of k.
- We plot the samples generated by the Gibbs chain (after intervals of 100 iterations) for different values of k. Figure 6 shows the input samples and Figures 7 to 11 shows the samples from the Gibbs chains.

We use $n = 512$ in all five cases.

We find that as k increases, $\tilde{V}$ (which was initialized to V_d) undergoes an increasing number of transformations, and is hence more different compared to V_d. Hence the reconstruction loss increases with k. We can also qualitatively see that the reconstructed images are better for smaller values of k.

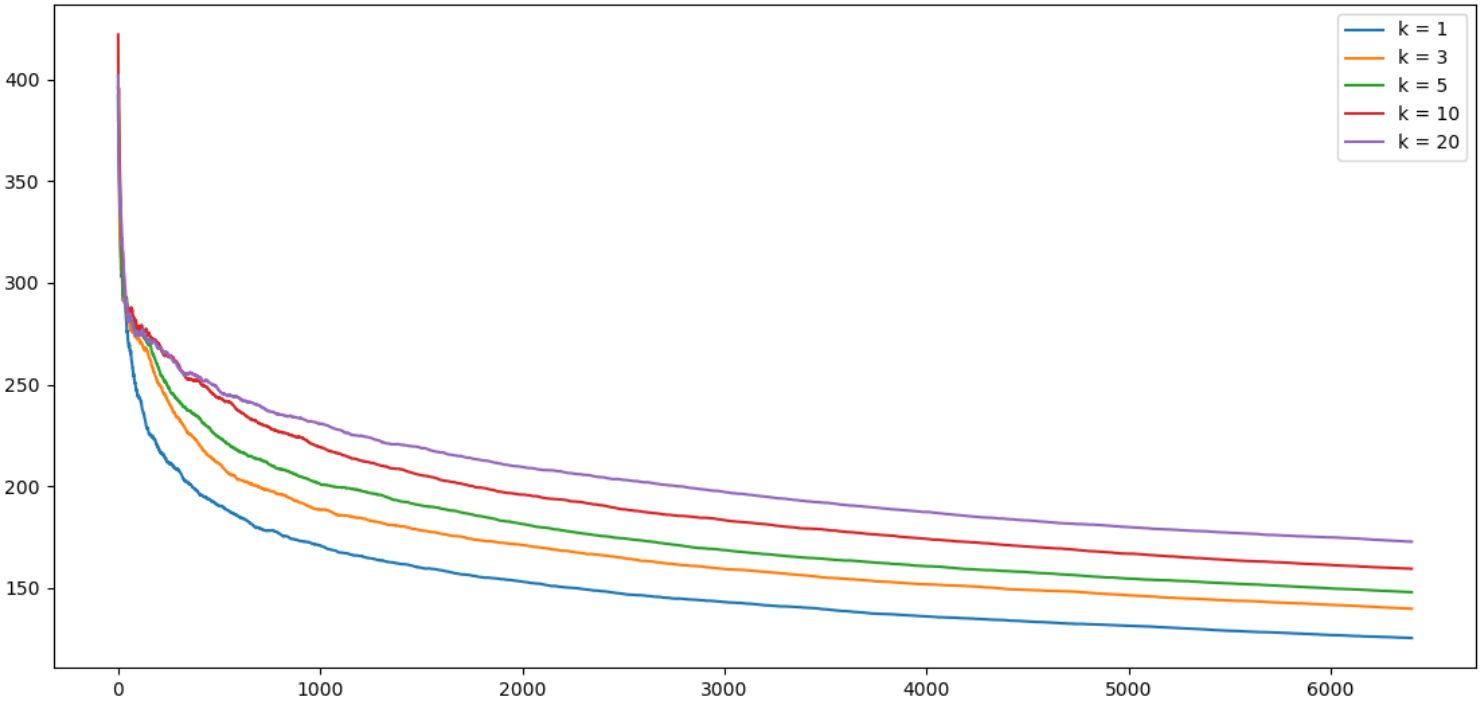

Figure 5: Average Reconstruction vs iterations

3 Question 3

Figure 6 shows the input samples used after intervals of 100 iterations. Figures 7 shows the corresponding samples from the Gibbs chain for $k = 1$, $n = 512$. We find that the quality of reconstructed images get better with the number of iterations. Qualitatively, they appear more and more similar to the corresponding sample input at that iteration.

Figure 6: Input samples

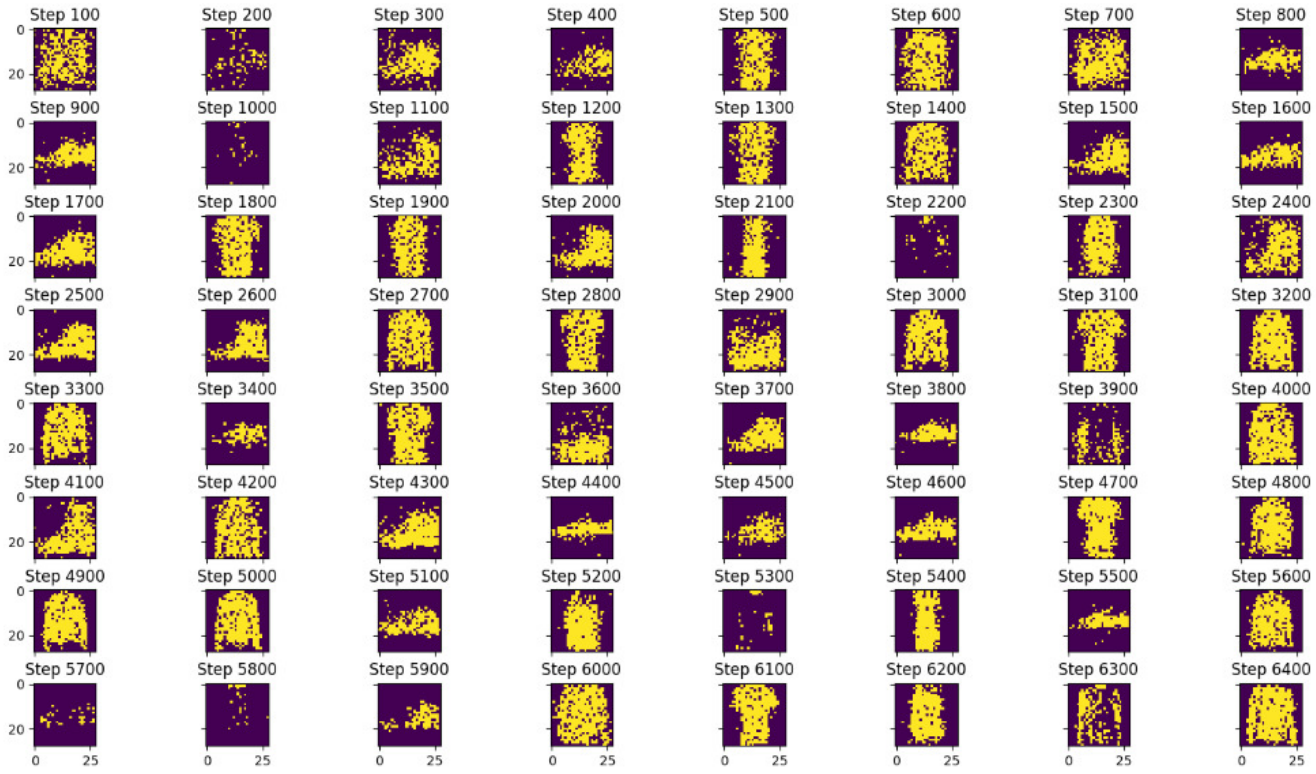
Step 100
Step 200
Step 300
Step 400
Step 500
Step 600
Step 700
Step 800
Step 900
Step 1000
Step 1100
Step 1200
Step 1300
Step 1400
Step 1500
Step 1600
Step 1700
Step 1800
Step 1900
Step 2000
Step 2100
Step 2200
Step 2300
Step 2400
Step 2500
Step 2600
Step 2700
Step 2800
Step 2900
Step 3000
Step 3100
Step 3200
Step 3300
Step 3400
Step 3500
Step 3600
Step 3700
Step 3800
Step 3900
Step 4000
Step 4100
Step 4200
Step 4300
Step 4400
Step 4500
Step 4600
Step 4700
Step 4800
Step 4900
Step 5000
Step 5100
Step 5200
Step 5300
Step 5400
Step 5500
Step 5600
Step 5700
Step 5800
Step 5900
Step 6000
Step 6100
Step 6200
Step 6300
Step 6400
0
20
0
25

Figure 7: k = 1


Step 100
Step 200
Step 300
Step 400
Step 500
Step 600
Step 700
Step 800
Step 900
Step 1000
Step 1100
Step 1200
Step 1300
Step 1400
Step 1500
Step 1600
Step 1700
Step 1800
Step 1900
Step 2000
Step 2100
Step 2200
Step 2300
Step 2400
Step 2500
Step 2600
Step 2700
Step 2800
Step 2900
Step 3000
Step 3100
Step 3200
Step 3300
Step 3400
Step 3500
Step 3600
Step 3700
Step 3800
Step 3900
Step 4000
Step 4100
Step 4200
Step 4300
Step 4400
Step 4500
Step 4600
Step 4700
Step 4800
Step 4900
Step 5000
Step 5100
Step 5200
Step 5300
Step 5400
Step 5500
Step 5600
Step 5700
Step 5800
Step 5900
Step 6000
Step 6100
Step 6200
Step 6300
Step 6400
0
20
0
25

Figure 8: k = 3

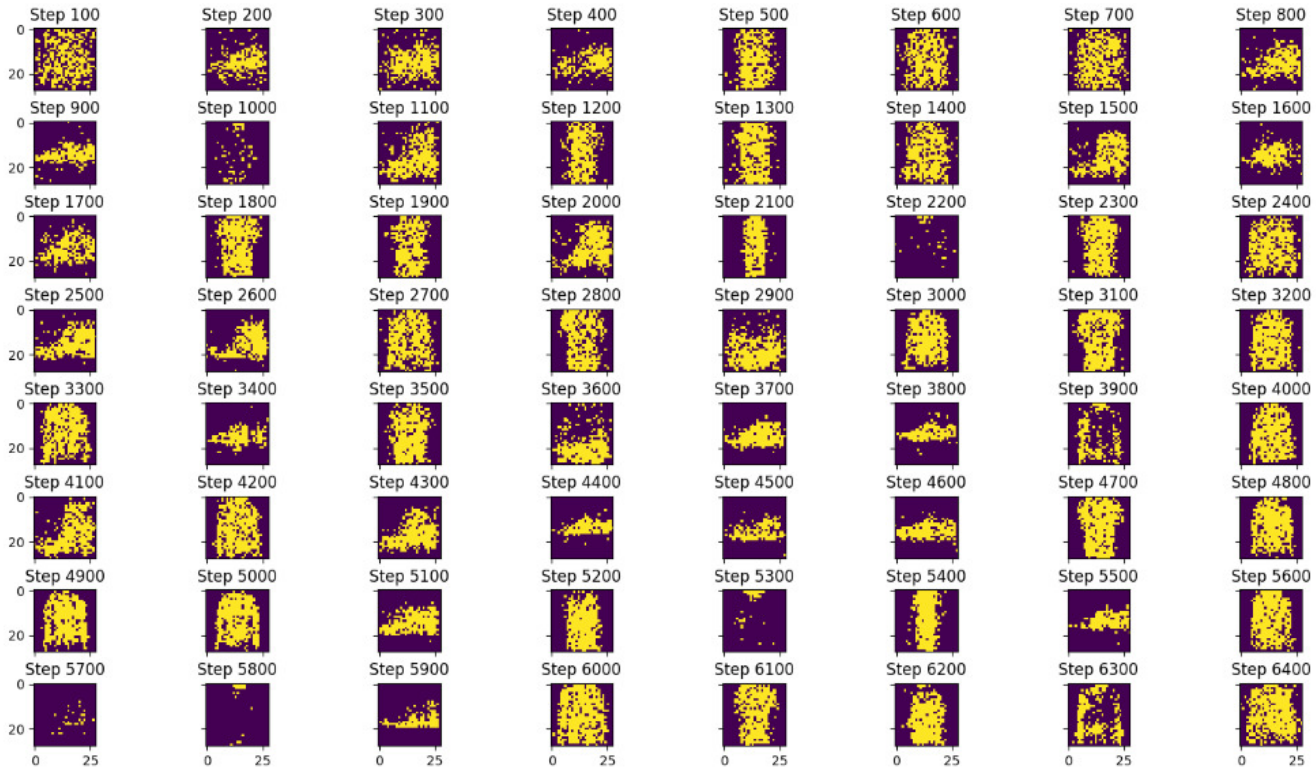
Step 100
Step 200
Step 300
Step 400
Step 500
Step 600
Step 700
Step 800
Step 900
Step 1000
Step 1100
Step 1200
Step 1300
Step 1400
Step 1500
Step 1600
Step 1700
Step 1800
Step 1900
Step 2000
Step 2100
Step 2200
Step 2300
Step 2400
Step 2500
Step 2600
Step 2700
Step 2800
Step 2900
Step 3000
Step 3100
Step 3200
Step 3300
Step 3400
Step 3500
Step 3600
Step 3700
Step 3800
Step 3900
Step 4000
Step 4100
Step 4200
Step 4300
Step 4400
Step 4500
Step 4600
Step 4700
Step 4800
Step 4900
Step 5000
Step 5100
Step 5200
Step 5300
Step 5400
Step 5500
Step 5600
Step 5700
Step 5800
Step 5900
Step 6000
Step 6100
Step 6200
Step 6300
Step 6400
0
20
0
25

Figure 9: k = 5

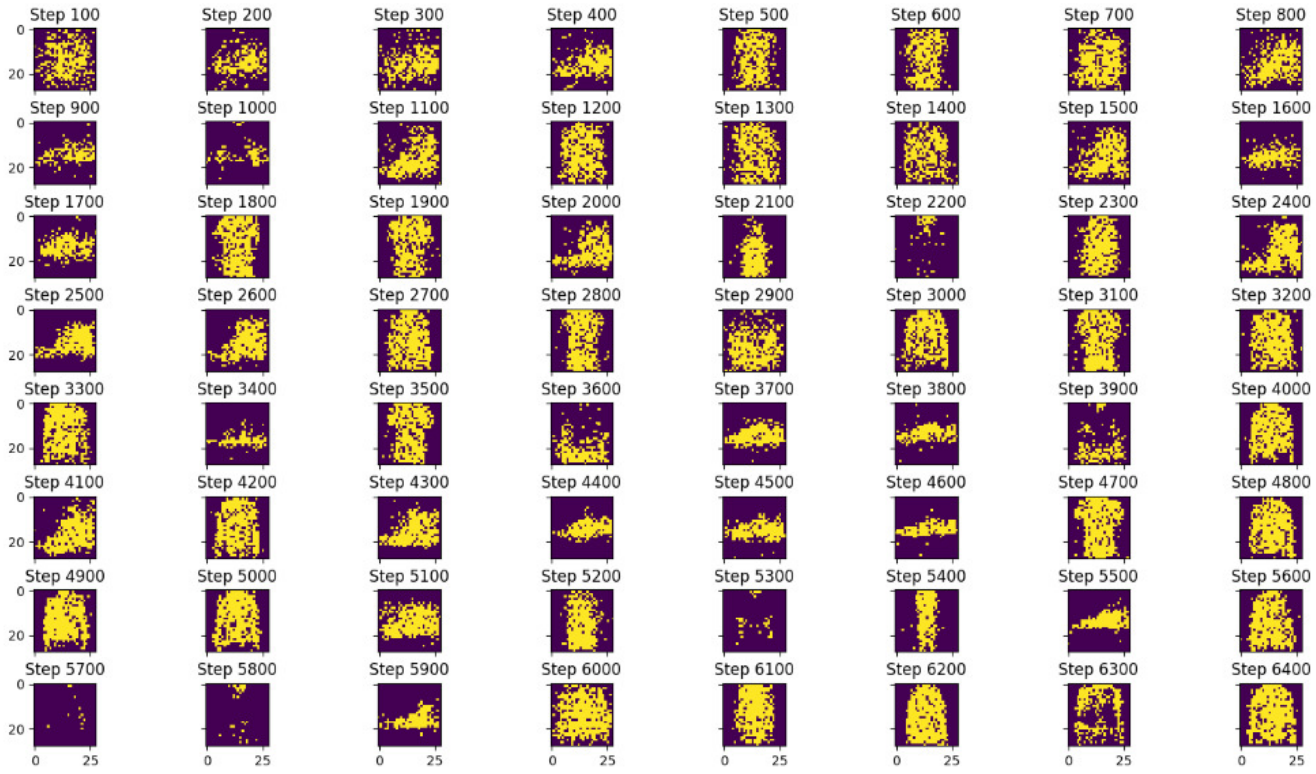

Figure 10: k = 10

Figure 11: k = 20