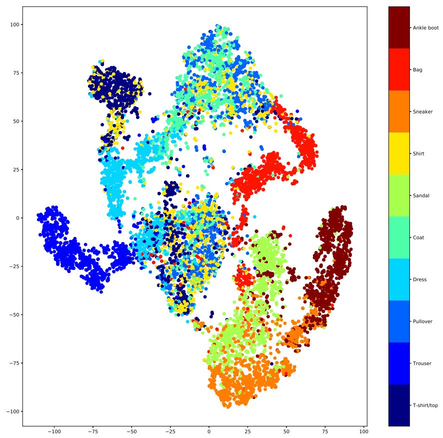
CS7015-Deep Learning
Programming Assignment 5

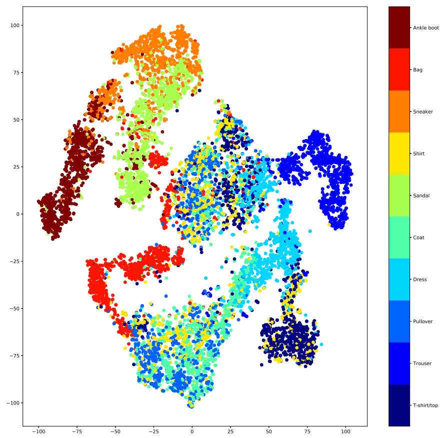
Jeshuren Chelladurai (CS17M017)
Ajith Kumar M (CS17M009)

April 30, 2018

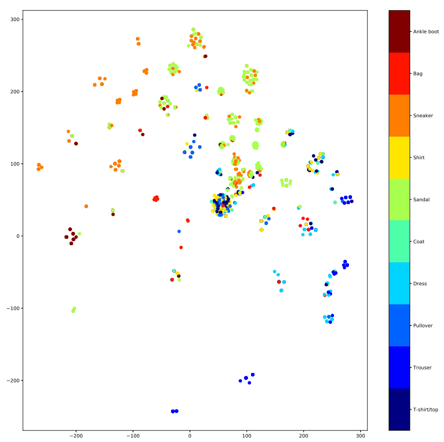
1. **Effect of the number of hidden layer units:** As could be seen from the t-SNE plots in Fig. 7, the hidden representations of the images belonging to one class appear to be clustered accordingly. Also, there is overlap in the clusters of the images belonging to classes whose images are similar. These images represent the set of images that cause confusion to a classifier while classifying.



(a) H = 200



(b) H = 100



(c) H = 10

Figure 1: t-SNE plots for the hidden representations learned by the model for various values of the hidden layer units(H)

Inferences

(a) The hidden representations of t-shirts/shirts/tops are close to each other and overlapping in some cases, which indicates that a lot of common shared features among them being efficiently captured by the hidden variables.

(b) The overlapping images represent the set of images that cause confusion to a classifier while classifying. During classification experiments, the classes that were confused to the maximum are overlapped in the hidden representations, which provides insight on to the learning of the hidden layer features.

(c) In the case when H = 10, even though the test data size is 10,000 it may appear that only a few points are being plotted, this is due to the same point being projected to the same region due to the limited representational capacity of the hidden layer.

Hidden Layer Units v/s Reconstruction Error: The effect of the number of hidden layer units was also tested against the reconstruction error of the images generated by RBM. As expected, when the number of hidden layer units is less, the reconstruction error is large and as the hidden layer units increase, the error goes on decreasing. So, from this we can conclude that the number of hidden layers play a crucial role in the model underfitting or overfitting with the data presented during the training time.

$$\text{Reconstruction Error} = \frac{1}{N}\sum_{i=1}^{N}(v^i_{original} - v^i_{reconstructed})^2$$

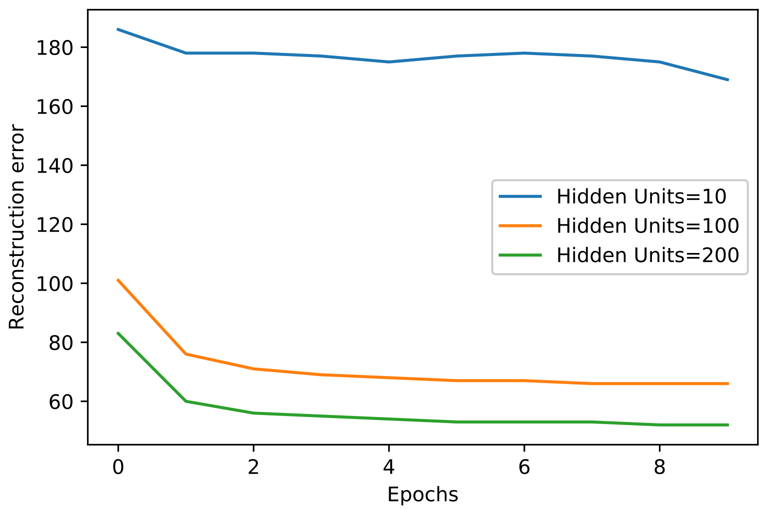


Figure 2: Effect of Hidden units

2. **Effect of the number of runs of the Gibbs Chain in Contrastive Divergence**

   To find out the effect of the number of runs of the Gibbs Chain(K), we used the reconstruction error to analyze its trend across various values of K. From the plot, it can be noticed that, as the value of K increases, the reconstruction error is increasing as the value of K increases. Hence, as mentioned during the lectures, it is good to use K-value as 1.

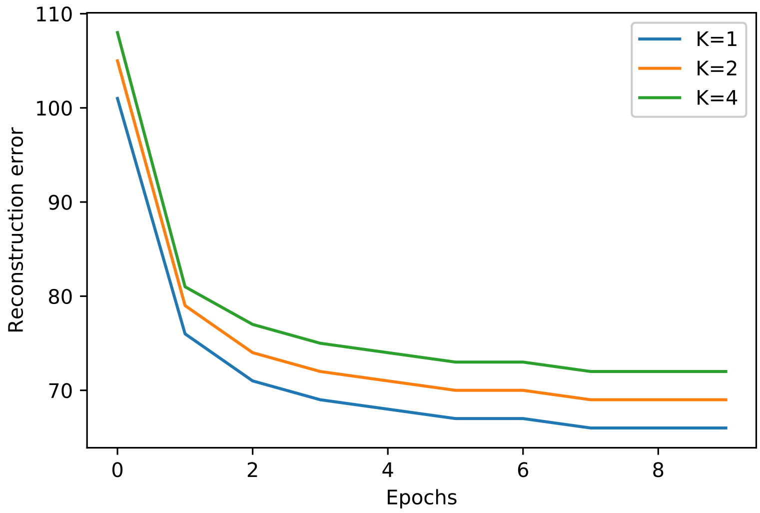


Figure 3: Effect of the number of runs of the Gibbs Chain in Contrastive Divergence with 100 hidden units.

3. **Images generated during the learning of the RBM at different phases:** Once the RBM model reached convergence, we split the number of updates to the weights happened into 64 parts and plotted the image generated through the learning. As could be seen from the figures, initially the image was being reconstructed as noise during the first few stages. But after the model has learnt through the updates, several features are being progressively added to the images and the final images are a near reconstruction of the original image. As could be noted from the learning curve, the final generated images had a reconstruction error of less than 50, meaning that they differed from the original image in less than 50 pixels. Expressing this in percentage values, the images were reconstructed with an accuracy of **92 percent.**

Figure 4: Images generated during the learning of the RBM at different phases(top) with 100 hidden variable and the original image(bottom).

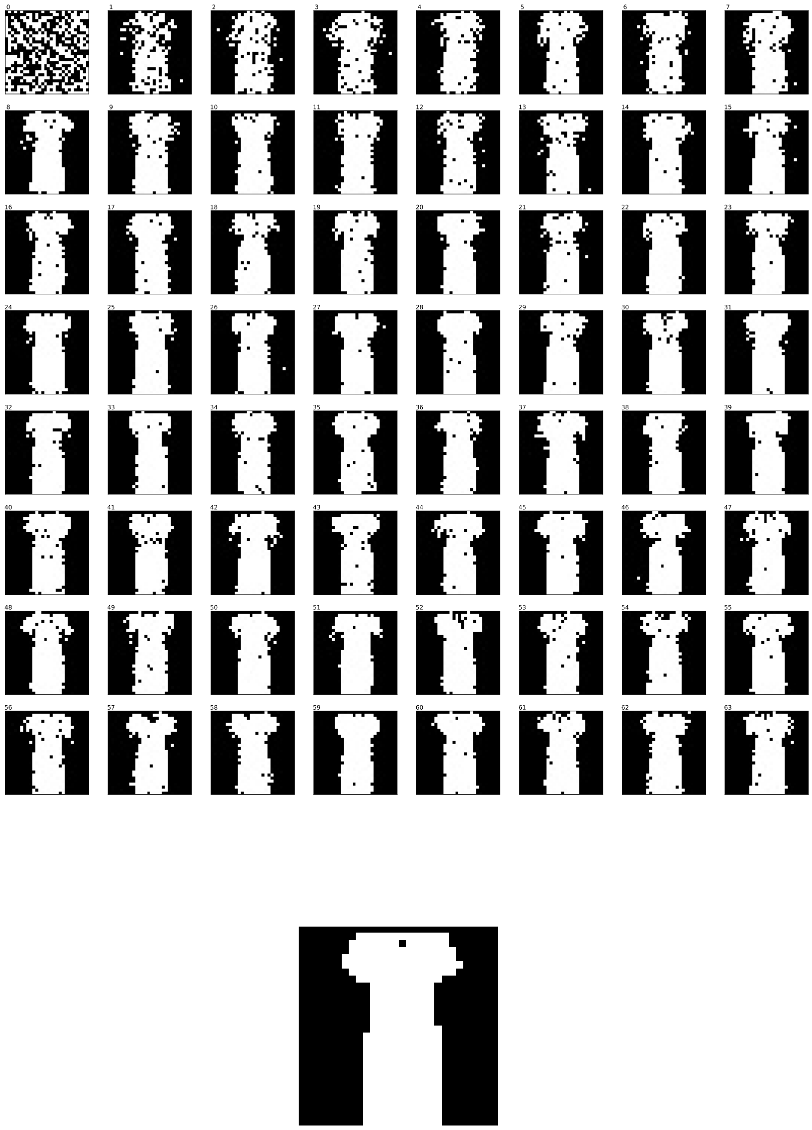


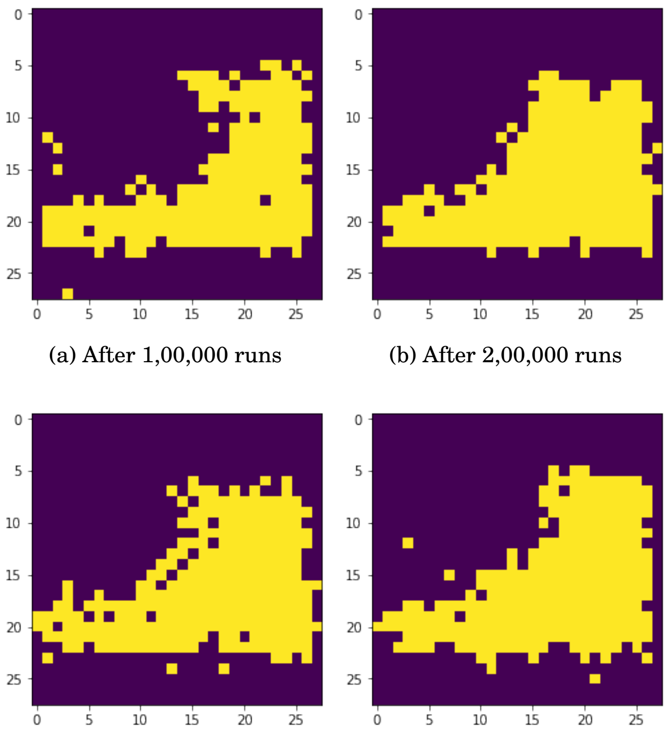
Figure 5: Images generated during the learning of the RBM at different phases(top) with 100 hidden variable and the original image(bottom).

4. **Gibbs Sampling - RBM**

   We implemented Gibbs Sampling for RBM and used it for generating hidden representations at various levels of the training. Since, we ran the Gibbs chain with a random vector for 10000 steps, before calculating the expectation of the estimates to be used for the backpropogation, only a partial dataset was used. And 1000, 10000 images were used as test dataset to check on the quality of the hidden representations.

   Initially after running the training for 10 epochs, and seeing signs of convergence, we decided to anneal the number of runs of gibbs chain by a discount factor of 0.5 after every two epochs. As can be seen from the plots, once the values have converged, there is no big change in the clusters formed by the hidden representations.

   Also, to see samples from the P(V,H) distribution, it is required to run the Gibbs chain for more than 1,00,000 times.



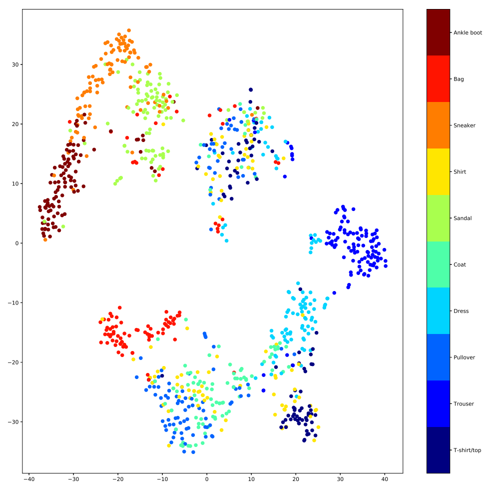
(a) After 1,00,000 runs

(b) After 2,00,000 runs

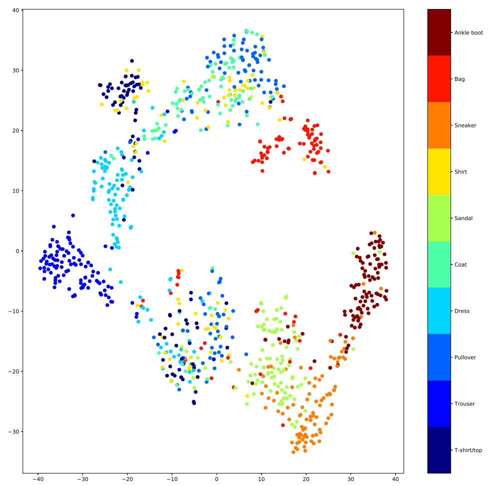
(c) After 10,00,000 runs

(d) After 20,00,000 runs

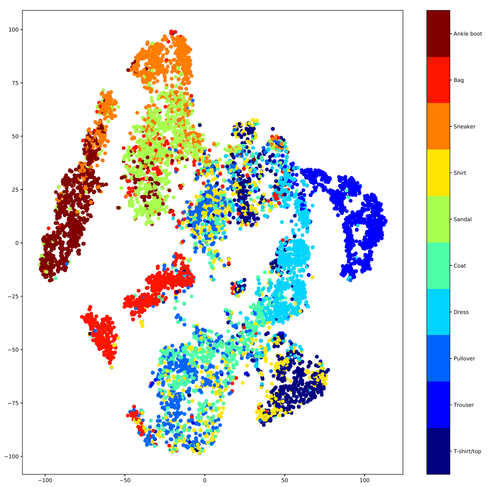
Figure 6: Image generated from random noise at different phases of the gibbs run



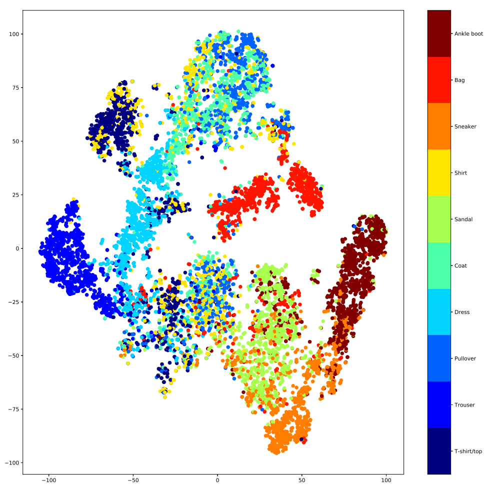
(a) 1000 Test data instances after 10 epochs



(b) 1000 Test data instances after 20 epochs



(c) 10000 Test data instances after 10 epochs



(d) 10000 Test data instances after 20 epochs

Figure 7: t-SNE plots for the hidden representations learned by the model before and after annealing of the number of runs of the Gibbs chain