Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 4

Student Name: Bhavy Khatri
Roll Number: 150186
Date: November 17, 2018

QUESTION 1

We have eigenvectors $\mathbf{v}$ of the matrix $\frac{1}{N}\mathbf{X}\mathbf{X}^T$. So,

$$\frac{1}{N}\mathbf{X}\mathbf{X}^T\mathbf{v} = \lambda\mathbf{v}$$

Multiply both sides with $\mathbf{X}^T$ we get,

$$\frac{1}{N}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{v}) = \lambda(\mathbf{X}^T\mathbf{v})$$

Which clearly means that $\mathbf{X}^T\mathbf{v}$ will be the eigenvectors of the matrix $\frac{1}{N}\mathbf{X}^T\mathbf{X}$.
Advantage: Using this method we can have much better time complexity. Suppose we want to find the first K eigenvectors then if we directly use power method on $\frac{1}{N}\mathbf{X}^T\mathbf{X}$ then it will cost $O(KD^2)$ but if we go by the above procedure then it will be $O(KN^2 + KND)$ which is clearly an improvement given $D > N$.

Student Name: Bhavy Khatri
Roll Number: 150186
Date: November 17, 2018

QUESTION 2

Note that,

$$h(x) = x\sigma(\beta x) = x\frac{1}{(1+e^{-\beta x})}$$

Linear Activation Function

As $\beta \to 0$, $h(x) = \frac{x}{2}$ which is a linear activation function.

RELU Activation Function

As $\beta \to \infty$, $h(x) = max(0, x)$ because when $x > 0$ then $h(x) = x$ and when $x < 0$ then $h(x) = 0$.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 4

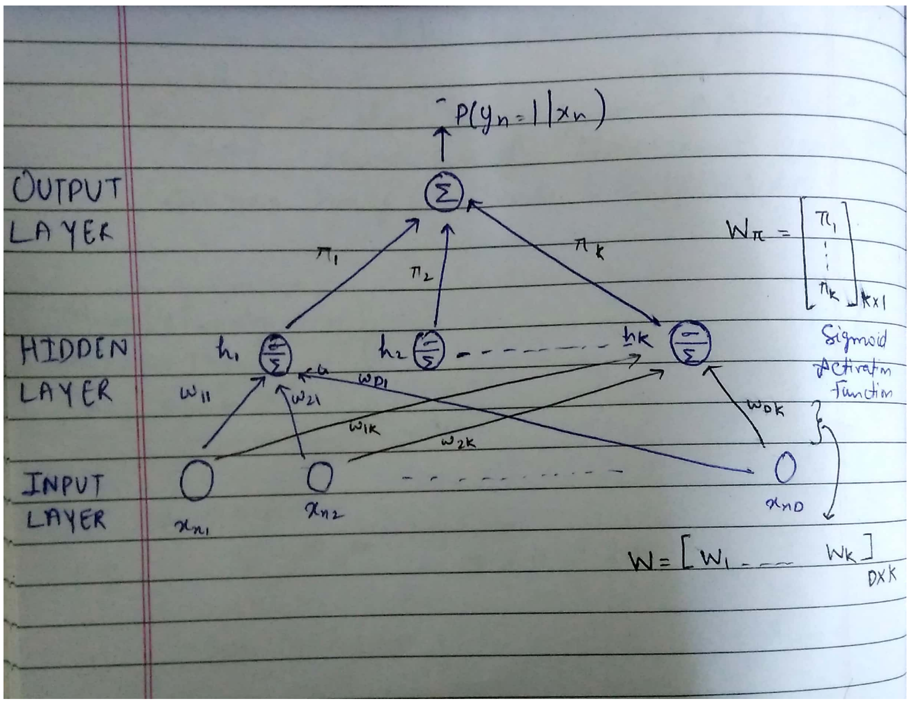
Student Name: Bhavy Khatri
Roll Number: 150186
Date: November 17, 2018

QUESTION 3

Note that,

$$\begin{aligned} P(y_n = 1|\mathbf{x}_n) &= \sum_{k=1}^{K} P(y_n = 1|z_n = k, \mathbf{x}_n) P(z_n = k) \\ &= \sum_{k=1}^{K} \pi_k \sigma(\mathbf{w}_k^T \mathbf{x}_n) \end{aligned}$$

The following picture explains the design of neural network whose output is same as the marginal probability of $y_n = 1$.



Input Layer: In this layer there will be D many nodes $[x_{n1}, \ldots, x_{nD}]$ where x_{nk} represents kth feature of nth example.
Hidden Layers: There will only one hidden layer and K many hidden nodes in that layer represented by $[h_1, \ldots, h_K]$. Every node h_i will be connected through the weight vector $\mathbf{w}_i, \forall i = 1 \ldots, K$ with the input layer. Activation function used is sigmoid function.
Output: Single node with connections to the hidden layers are $[\pi_1, \ldots, \pi_K]$.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 4

Student Name: Bhavy Khatri
Roll Number: 150186
Date: November 17, 2018

QUESTION

4

Let, $\mathbf{X}$ denotes user-movie rating matrix and parameters be denoted by $\Theta = \{\{\mathbf{u}_n, \theta_n\}_{n=1}^N \{\mathbf{v}_m, \phi_m\}_{m=1}^M, \mathbf{W}_u,$
Then posterior probabiliy is given by,

$$P(\Theta|\mathbf{X}) = \frac{P(\mathbf{X}|\Theta)P(\Theta)}{P(\mathbf{X})}$$

The MAP estimate is given by,

$$\begin{aligned}\hat{\Theta_{MAP}} &= \arg \underset{\Theta}{max} \left[\log P(\mathbf{X}|\Theta) + \log P(\Theta)\right] \\ &= \arg \underset{\Theta}{max} \sum_{n=1}^N \log P(\mathbf{u}_n) + \sum_{m=1}^M \log P(\mathbf{v}_m) + \sum_{(n,m)\in\Omega} p(X_{nm}|\theta_n + \phi_m + \mathbf{u}_n^T\mathbf{v}_m, \lambda_x^{-1})\end{aligned}$$

Use:

$$\mathcal{N}(\mathbf{x}|\mu, \Sigma) \propto \exp\left[\frac{-1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right]$$

We will get estimate equation as follows:

$$\begin{aligned}\hat{\Theta_{MAP}} &= \arg \underset{\Theta}{max} \sum_{n=1}^N \frac{-\lambda_u}{2}(\mathbf{u}_n - \mathbf{W}_u\mathbf{a}_n)^T(\mathbf{u}_n - \mathbf{W}_u\mathbf{a}_n) + \sum_{m=1}^M \frac{-\lambda_v}{2}(\mathbf{v}_m - \mathbf{W}_v\mathbf{b}_m)^T(\mathbf{v}_m - \mathbf{W}_v\mathbf{b}_m) \\ &+ \sum_{(n,m)\in\Omega} \frac{-\lambda_x}{2}[X_{nm} - (\theta_n + \phi_m + \mathbf{u}_n^T\mathbf{v}^m)]^2 \\ &= \arg \underset{\Theta}{min} \frac{\lambda_u}{2} \sum_{n=1}^N (\mathbf{u}_n - \mathbf{W}_u\mathbf{a}_n)^T(\mathbf{u}_n - \mathbf{W}_u\mathbf{a}_n) + \frac{\lambda_v}{2}\sum_{m=1}^M (\mathbf{v}_m - \mathbf{W}_v\mathbf{b}_m)^T(\mathbf{v}_m - \mathbf{W}_v\mathbf{b}_m) \\ &+ \frac{\lambda_x}{2}\sum_{(n,m)\in\Omega} [X_{nm} - (\theta_n + \phi_m + \mathbf{u}_n^T\mathbf{v}^m)]^2\end{aligned}$$

So the loss function is given by:

$$\begin{aligned}\mathcal{L}(\Theta) &= \frac{\lambda_u}{2} \sum_{n=1}^N (\mathbf{u}_n - \mathbf{W}_u\mathbf{a}_n)^T(\mathbf{u}_n - \mathbf{W}_u\mathbf{a}_n) + \frac{\lambda_v}{2}\sum_{m=1}^M (\mathbf{v}_m - \mathbf{W}_v\mathbf{b}_m)^T(\mathbf{v}_m - \mathbf{W}_v\mathbf{b}_m) \\ &+ \frac{\lambda_x}{2}\sum_{(n,m)\in\Omega} [X_{nm} - (\theta_n + \phi_m + \mathbf{u}_n^T\mathbf{v}^m)]^2\end{aligned}$$

We will use alternating optimization scheme to find the Θ.

$$\frac{\partial\mathcal{L}}{\partial\mathbf{u}_n} = \lambda_u[\mathbf{u}_n - \mathbf{W}_u\mathbf{a}_n] - \lambda_x \sum_{\Omega_{r_n}} [X_{nm} - (\theta_n + \phi_m + \mathbf{u}_n^T\mathbf{v}_m)]\mathbf{v}_m = 0$$

Solving the above equation we get the folllowing closed form expression for $\mathbf{u}_n$,

$$\hat{\mathbf{u}}_n = \left(\lambda_u \mathbf{I}_K + \lambda_x \sum_{(n,m)\in\Omega_{r_n}} \mathbf{v}_m \mathbf{v}_m^T\right)^{-1} \left[\lambda_u \mathbf{W}_u \mathbf{a}_n + \lambda_x \sum_{(n,m)\in\Omega_{r_n}} [X_{nm} - (\theta_n + \phi_m)]\mathbf{v}_m\right]$$

Similarly,

$$\hat{\mathbf{v}}_m = \left(\lambda_v \mathbf{I}_K + \lambda_x \sum_{(n,m)\in\Omega_{c_m}} \mathbf{u}_n \mathbf{u}_n^T\right)^{-1} \left[\lambda_v \mathbf{W}_v \mathbf{b}_m + \lambda_x \sum_{(n,m)\in\Omega_{c_m}} [X_{nm} - (\theta_n + \phi_m)]\mathbf{u}_n\right]$$

$$\frac{\partial \mathcal{L}}{\partial \theta_n} = \frac{\lambda_x}{2} \sum_{\Omega_{r_n}} [X_{nm} - (\theta_n + \phi_m + \mathbf{u}_n^T \mathbf{v}_m)]$$

$$\hat{\theta}_n = \frac{1}{|\Omega_{r_n}|} \sum_{\Omega_{r_n}} (X_{nm} - \mathbf{u}_n^T \mathbf{v}_m) - \phi_m$$

$$\hat{\phi}_m = \frac{1}{|\Omega_{c_m}|} \sum_{\Omega_{c_m}} (X_{nm} - \mathbf{u}_n^T \mathbf{v}_m - \theta_n)$$

$$\mathbf{W}_u = \left(\sum_{n=1}^{N} \mathbf{a}_n \mathbf{a}_n^T\right)^{-1} \sum_{n=1}^{N} \mathbf{u}_n \mathbf{a}_n^T$$

$$\mathbf{W}_v = \left(\sum_{m=1}^{M} \mathbf{b}_m \mathbf{b}_m^T\right)^{-1} \sum_{m=1}^{M} \mathbf{v}_m \mathbf{b}_m^T$$

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 4

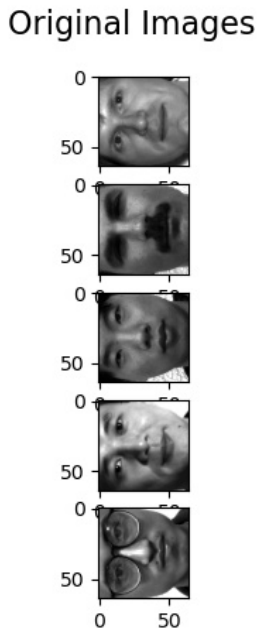
Student Name: Bhavy Khatri
Roll Number: 150186
Date: November 17, 2018

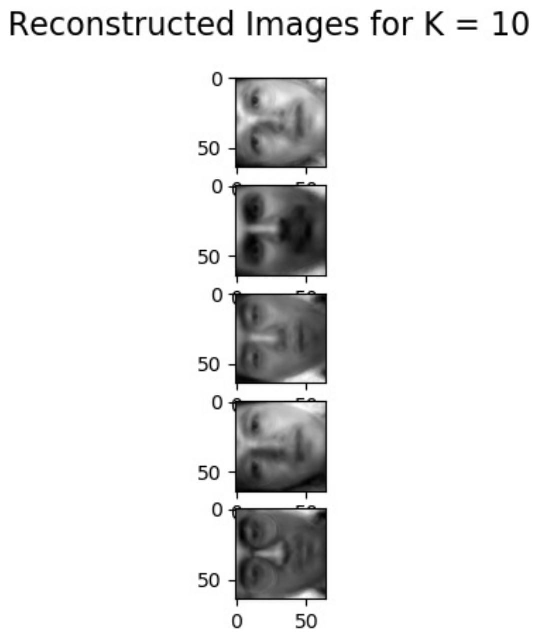
QUESTION

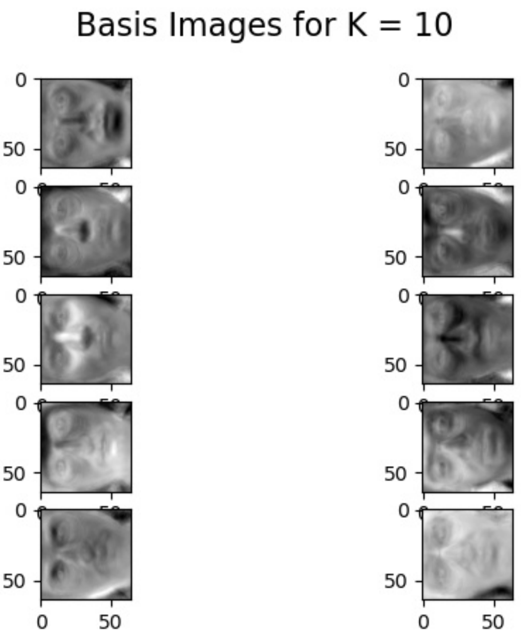
5

Probabilistic PCA

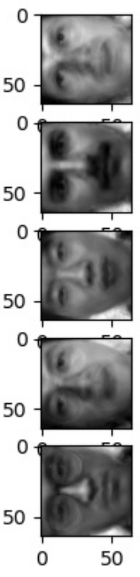
Reconstructed images are getting better and better for increasing value of K which also makes sense as we the larger lower dimensional representation of images the lesser will be reconstruction error. The most funny part was of basis images which looks like the picture of some horror movie. Although for increasing K the features in the basis images become more subtle. The original, reconstructed and basis images are as follows:



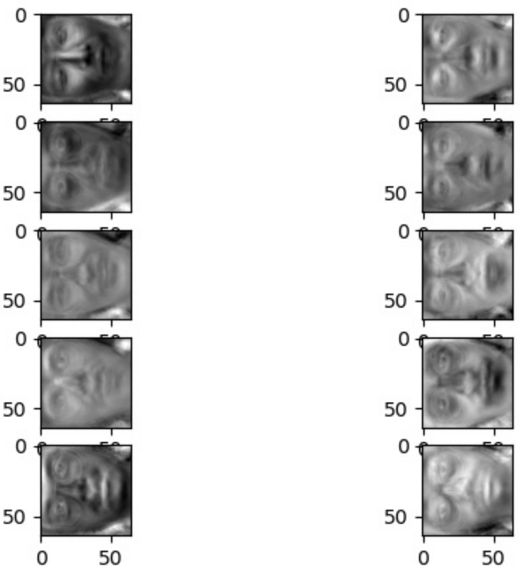

Reconstructed Images for K = 10
0
50
0
50


Basis Images for K = 10
0
50
0
50

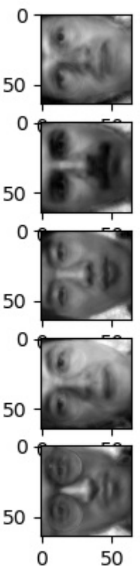
Reconstructed Images for K = 20



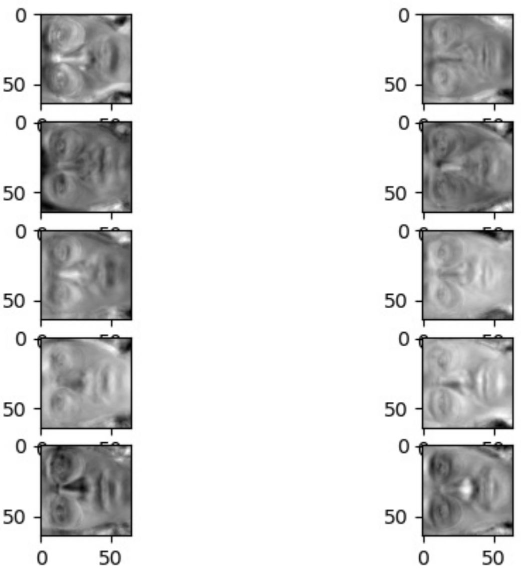
Basis Images for K = 20



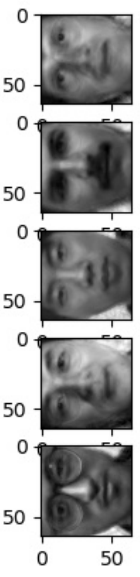
Reconstructed Images for K = 30



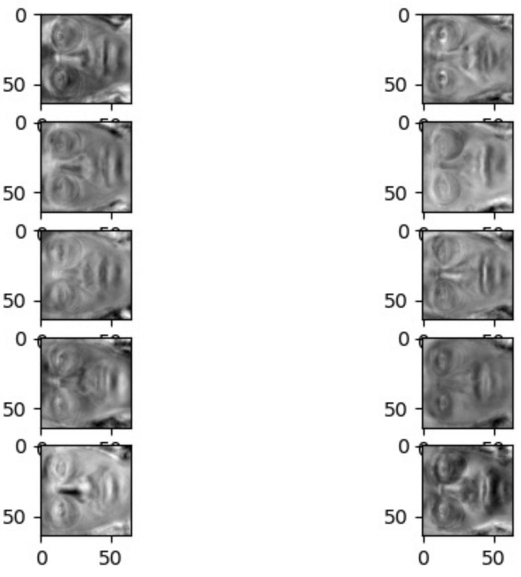
Basis Images for K = 30



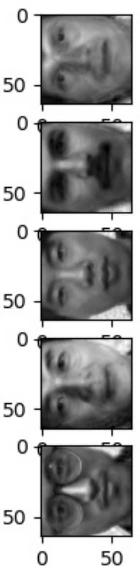
Reconstructed Images for K = 40



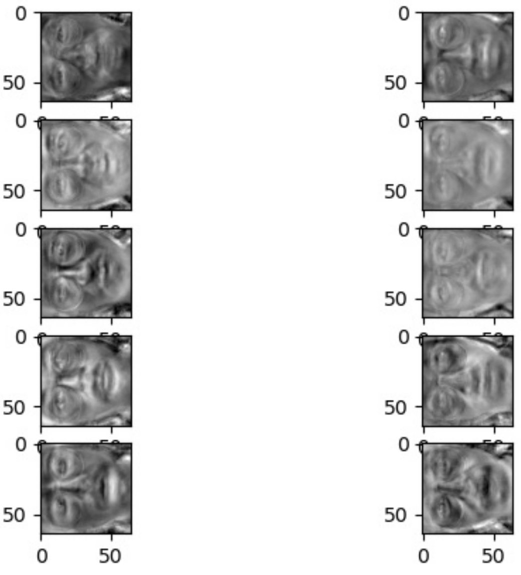
Basis Images for K = 40



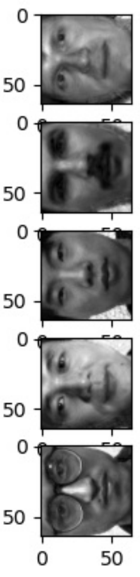
Reconstructed Images for K = 50



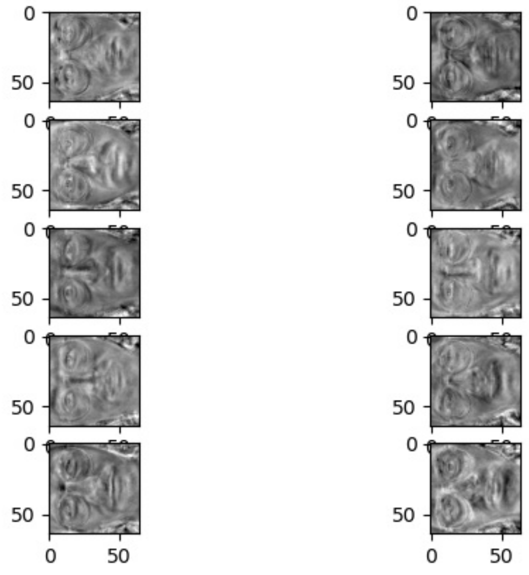
Basis Images for K = 50



Reconstructed Images for K = 100



Basis Images for K = 100

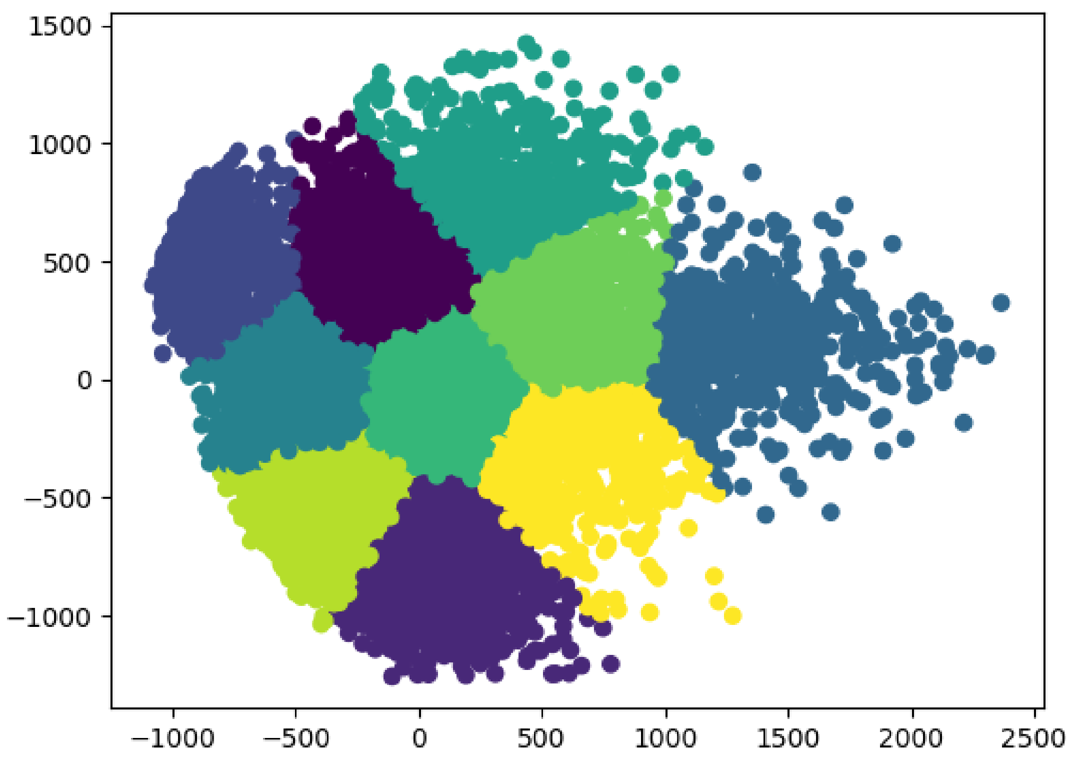


Classical PCA and t-SNE

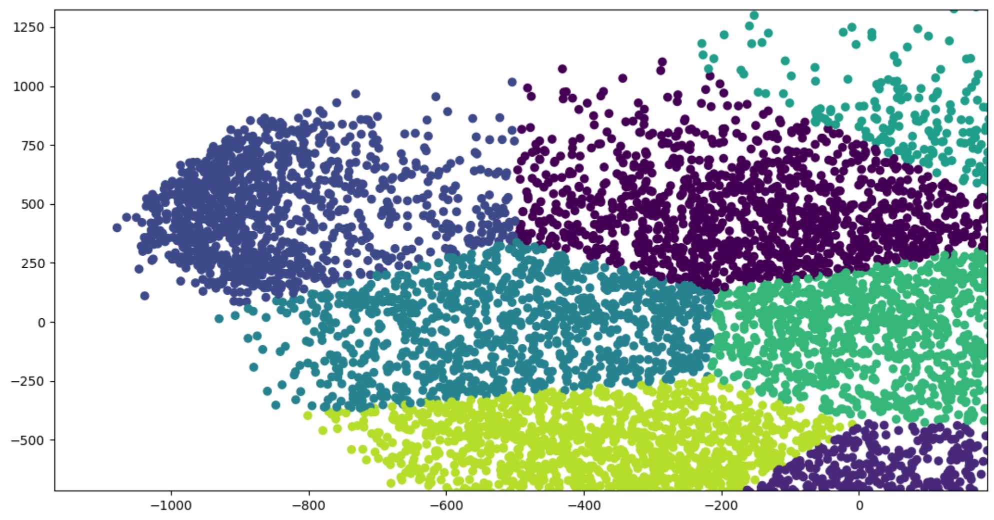
Dimensionality reduction using PCA and t-SNE was done on the MNIST dataset. The results were interesting and t-SNE outperformed PCA algorithm in terms of visualisation of data as the clusters were clearly visible in the embedded space. The embedded data using PCA was very uniform and not suitable for visualisation while in case of t-SNE there was a clear margin between clusters i.e. clusters were clearly visible. This experiment also shows that t-SNE is suitable for supervised dimensionality reduction.

PCA

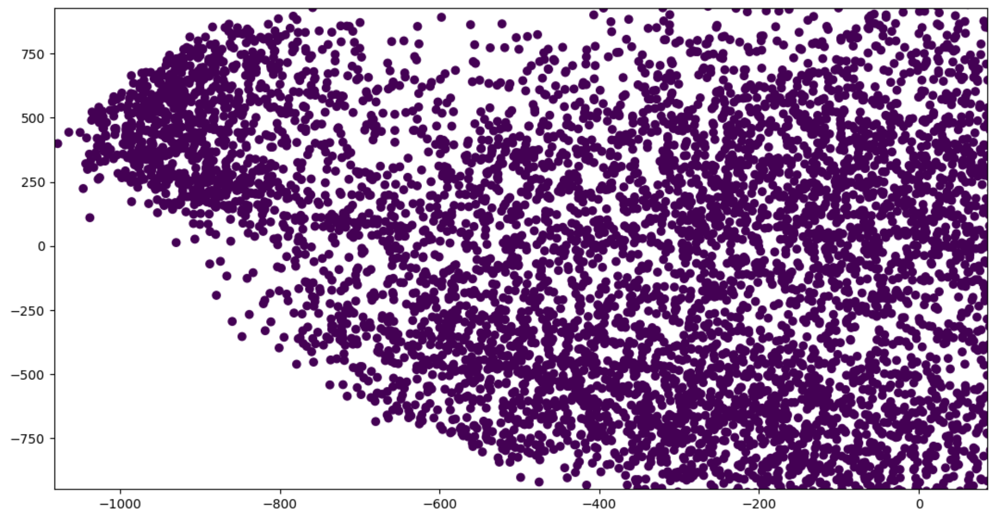
First, let's have a look at the data in embedded space and applying k-means for a single initialization with K =10.



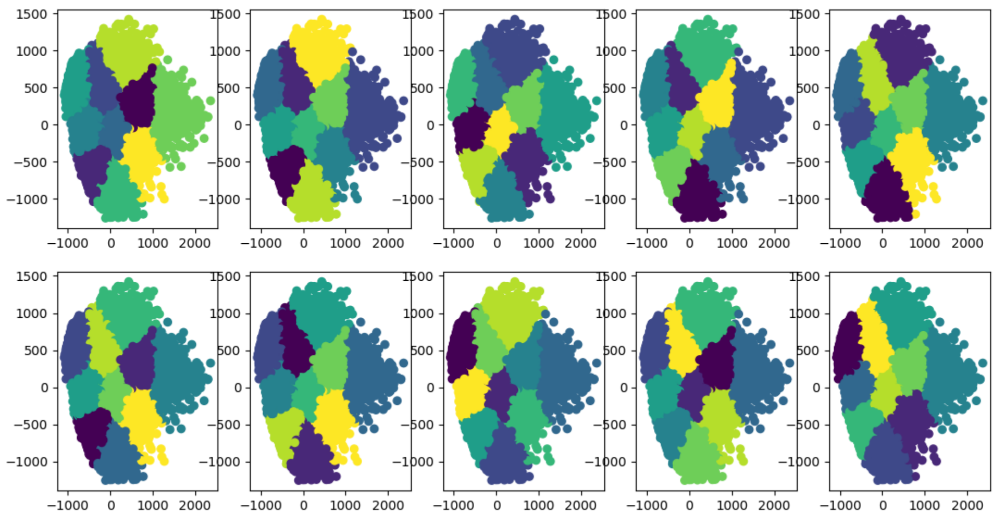
Let's look at the maginified version:



You can easily see that data is uniformly distributed and clusters are not clearly visible. It becomes more clear when we remove different colors.

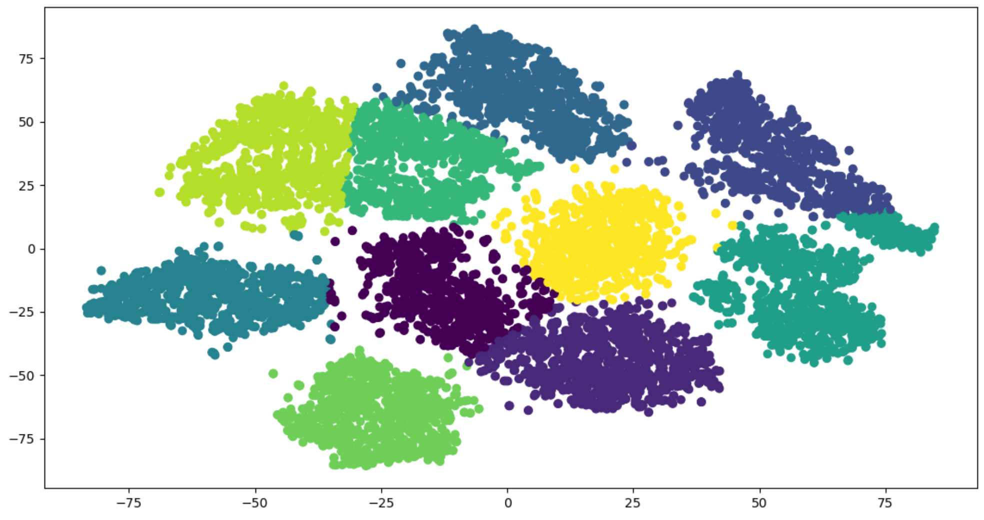


After applying K-means with 10 different initialization following plot was obtained.

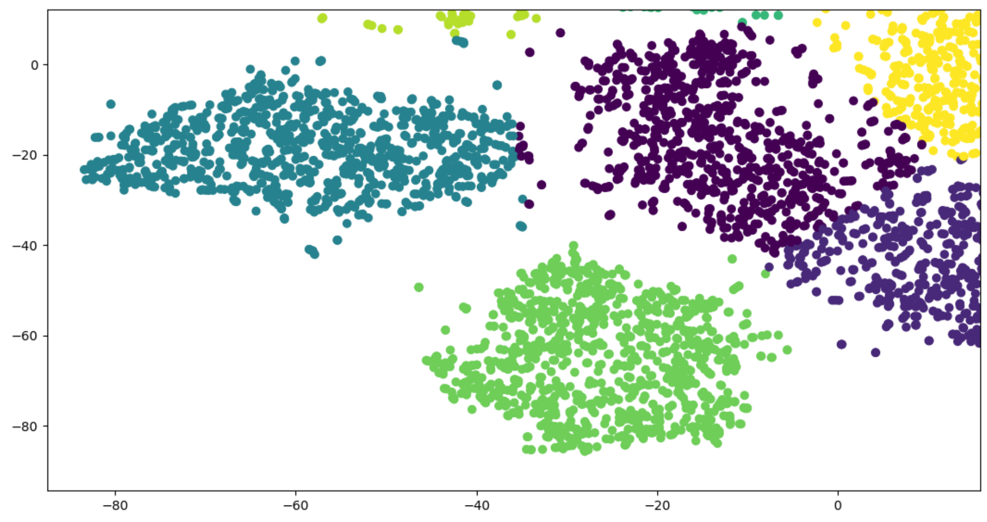


t-SNE

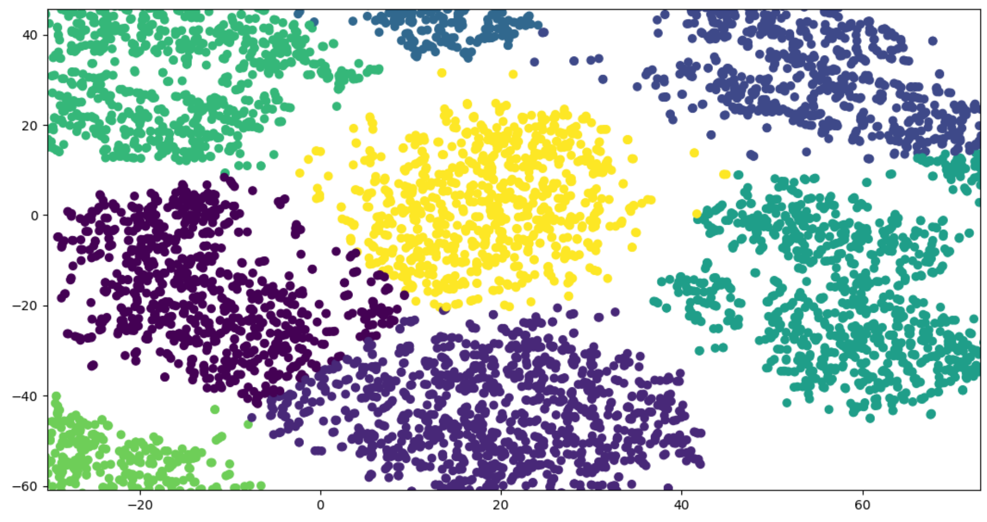
Again, let's have a look at the data in embedded space and applying k-means for a single initialization with K =10.



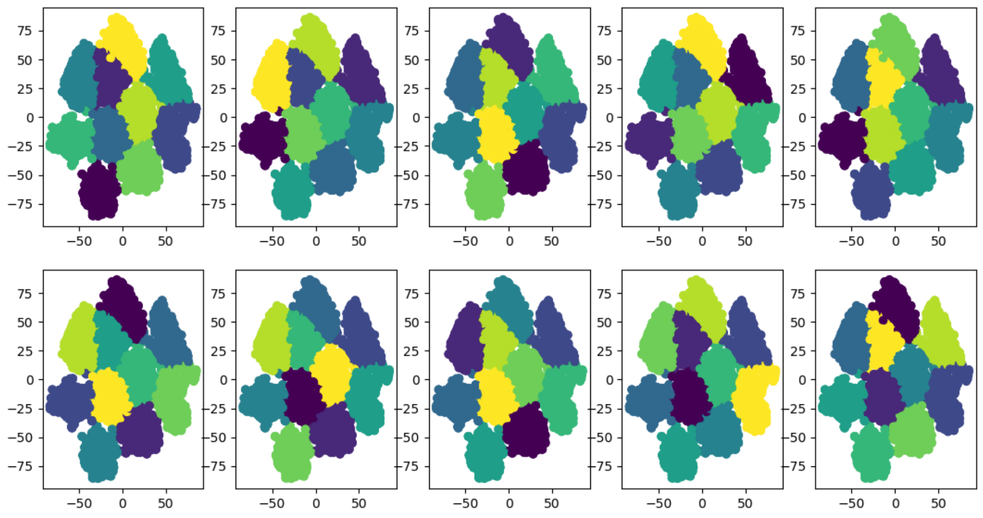
Let's look at the maginified version:



Magnification from other portion:



Its interesting that clusters are clearly visible. After applying K-means with 10 different initialization following plot was obtained.


75
50
25
0
−25
−50
−75
−50
0
50