ECE421 – Introduction to Machine Learning

Assignment 3

Neural Networks

Shiyi Zhang	Jiani Li	Tianyi Wang
Contribution: 33%	Contribution: 33%	Contribution: 33%

Part I. K-means-Learning K-means

1.1

$$\mathcal{L}(\boldsymbol{\mu}) = \sum_{n=1}^{N} \min_{k=1}^{K} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|_2^2$$

- **code for distanceFunc:**

```
# Distance function for K-means
def distanceFunc(X, MU):
    # Inputs
    # X: is an NxD matrix (N observations and D dimensions)
    # MU: is an KxD matrix (K means and D dimensions)
    # Outputs
    # pair_dist: is the squared pairwise distance matrix (NxK)
    # TODO
    X_X = tf.reshape(tf.reduce_sum(tf.square(X), axis=1), [-1, 1])
    MU_MU = tf.reshape(tf.reduce_sum(tf.square(MU), axis=1), [1, -1])
    X_MU = (-2) * tf.matmul(X, tf.transpose(MU))
    return X_X + MU_MU + X_MU
```

- **Derivation:**

  according to the squared loss function,

  $Loss = \|x_n - \mu_k\|^2 = \|x_n\|^2 + \|\mu_n\|^2 - 2 \cdot x_n \mu_n^T$ (with proper reshaping)

- **Results:**

  Final Training Loss is: 5110.9482421875

  Final Mean μ = [[0.12183347 -1.5230418]
  [-1.0559268 -3.2431977]
  [1.251753 0.24656858]]

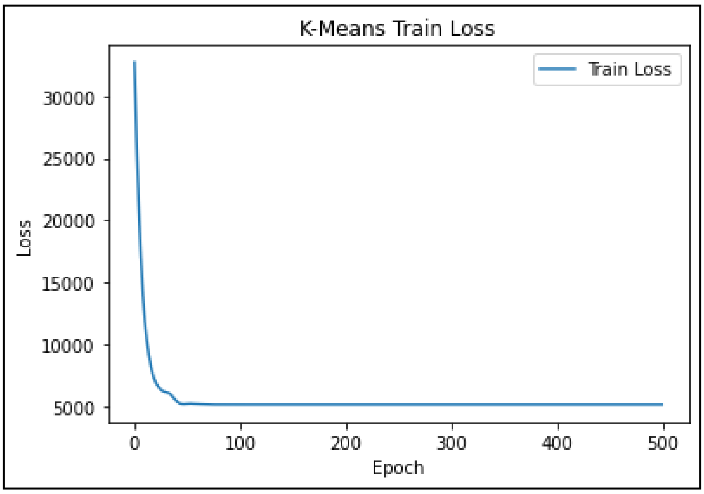


Figure1. Training Loss Function vs. Epochs of K-Means for K = 3

1.2

- **Percentage of the data points in each cluster:**

	Z = 1	Z = 2	Z = 3	Z = 4	Z = 5
K = 1	100%	NA	NA	NA	NA
K = 2	49.54%	50.46%	NA	NA	NA
K = 3	23.81%	38.13%	38.06%	NA	NA
K = 4	13.52%	37.13%	37.31%	12.04%	NA
K = 5	8.41%	35.76%	36.24%	10.79%	8.8%

- **2D Scatter Plots:**

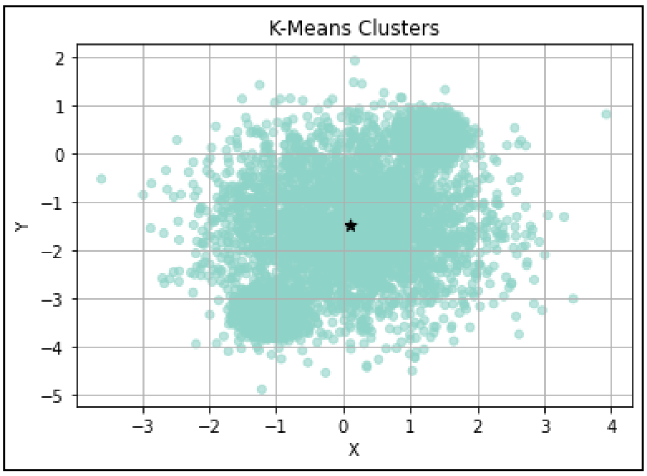


Figure 2.a. Clusters of K = 1

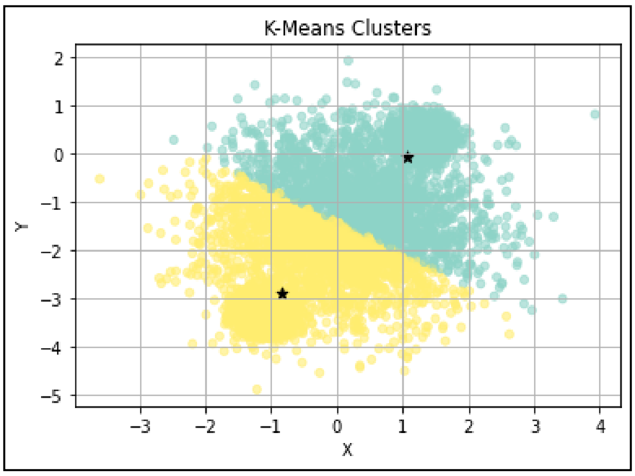
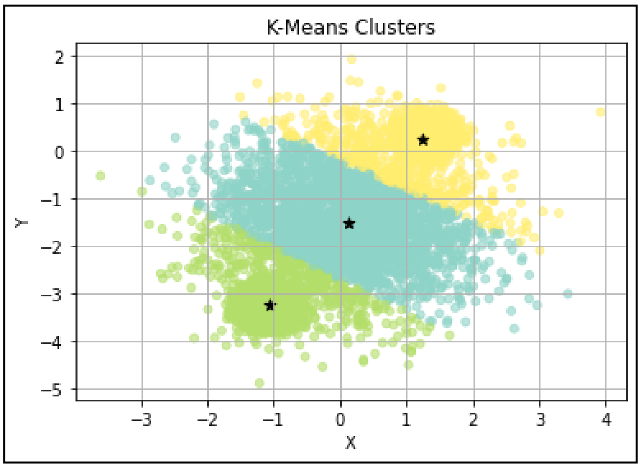


Figure 2.b. Clusters of K = 2



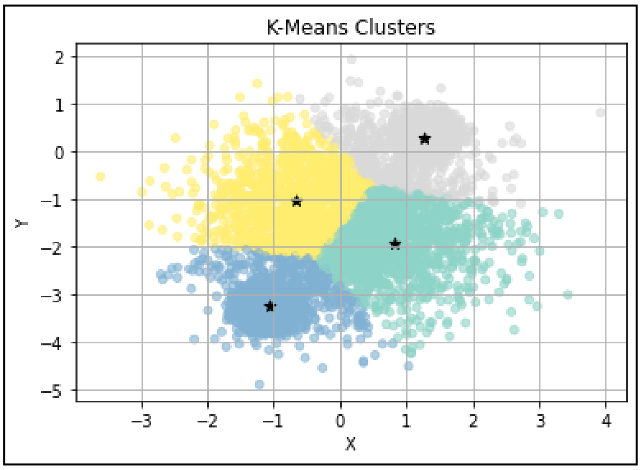


Figure 2.c. Clusters of K = 3 **Figure 2.d. Clusters of K = 4**

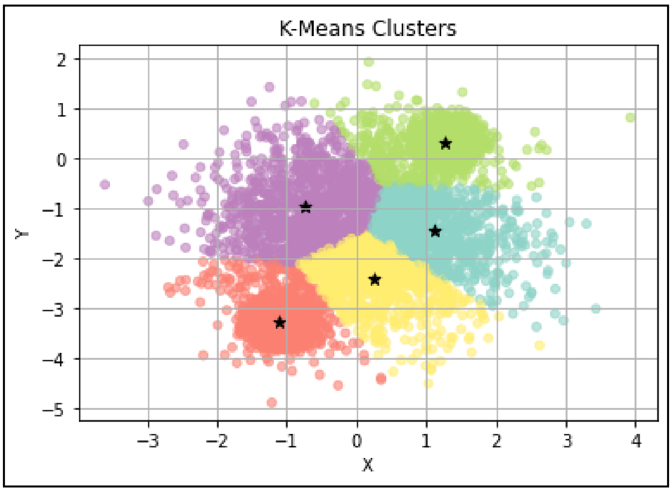


Figure 2.e. Clusters of K = 5

- **Comment:**

According to the percentage distributions and the scattering plots above, the best choice for the number of clusters would be K = 3. Because firstly, in terms of the loss functions, larger K value provides higher accuracy; on the other hand, the data are separated more and more unevenly with larger K, and at K = 3, the distribution is balanced while maintaining a relatively low loss.

1.3

- **Validation loss with different K:**

	K = 1	**K = 2**	**K = 3**	**K = 4**	**K = 5**
Validation Loss	**12862.3896**	**2960.20093**	**1617.58484**	**1053.92322**	**886.76892**

- **Plot for validation loss:**

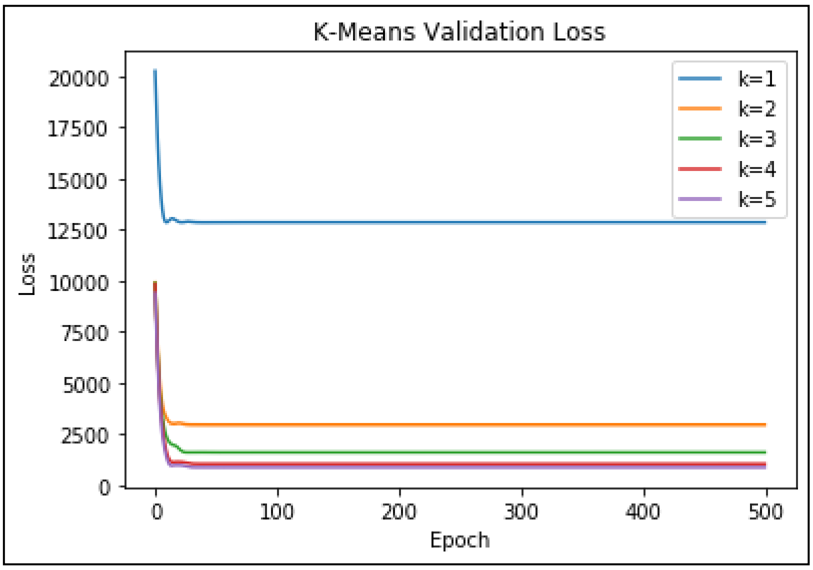


Figure3. K-Means Validation Loss vs. Epochs with K = 1→ 5

- **Conclusion:**

Based on the trend of the validation loss, it is evident that larger K leads to lower loss, thus in this sense, K = 5 is the best choice. We can increase the value of K to infinitely large, but the scattering may be overfitting, so the level of balance among the distribution should also be taken into consideration.

Part II. Mixtures of Gaussians

2.1 The Gaussian Cluster Mode

2.1.1

- **Code for the log Gaussian pdf function**

```
def log_GaussPDF(X, mu, sigma):
        # Inputs
        # X: N X D
        # mu: K X D
        # sigma: K X 1, passed in as sigma^2

        # Outputs:
        # log Gaussian PDF N X K

        # TODO
        dim = tf.to_float(tf.rank(X)) #convert to correct data type
        sigma = tf.squeeze(sigma)
        dist = distanceFunc(X, mu)
        exp = (-1) * dist / (2 * sigma)
        coeff = (-1) * (dim/2) * tf.log(2*np.pi * sigma)
        return coeff + exp
```

- **Derivation:**

$$logN(x;\mu_k^2,\sigma_k^2) = log(P(x|\mu_k,\sigma_k))$$

$$= log\left(\prod_{d=1}^{Dim} \frac{1}{\sqrt{2\pi\sigma_k^2}} \times exp\left(\frac{-(x-\mu_k)^2}{2\sigma_k^2}\right)\right)$$

$$= \frac{Dim}{2} \times log\ \frac{1}{2\pi\sigma_k^2} + exp\left(\frac{-(x-\mu_k)^2}{2\sigma_k^2}\right)$$

2.1.2 Code for the log posterior function

- **Derivation:**

  According the Baye's Rule:

  $P(z=k|x) = \frac{P(x|z=k)P(z=k)}{\sum_{k=1}^{K} P(x|z=k)P(z=l)}$

  $$=> logP(z=k|x) = log(P(x|z=k)) + log(P(z=k)) - log\sum_{k=1}^{K} P(x|z=k)P(z=l)$$

  $$= logPDF + logPi - log\sum_{k=1}^{K} exp(logPDF + logPi)$$

- **Comment:**

  We are using the reduce_logsumexp function instead of reduce_sum because when we are dealing with the denominator of the posterior probability, we want to get 'the log of sum'. However, we can only achieve 'the sum of logs' if we use reduce_sum (which is not equal to 'the log of sum'), since we are provided with the log terms of pdf and Pi. Therefore, we should use the reduce_sum function to take back the log terms and then get the desired 'the log of sum'.

  In addition, we should pay attention to the exponential term, as it is possible to grow to infinity, so we need the subtraction of the maximum term to prevent overflow and achieve numerical stability in the reduce_logsumexp function.

2.2 Learning the MoG

2.2.1 K=3, no validation

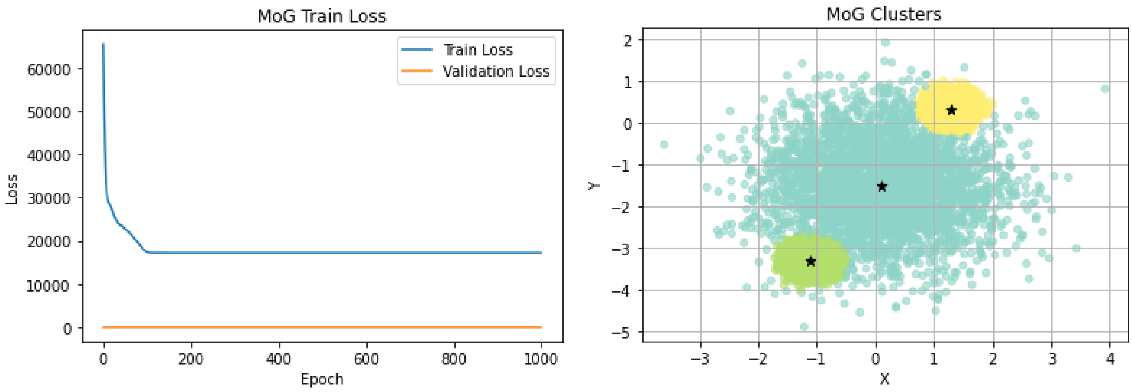


Figure 4(a) & 4(b). Loss plot of K=3 & MoG Clusters plot

Final Model Parameters are:

u = [[0.10382269 -1.5265145] [-1.1033583 -3.3031693] [1.2977628 0.31047803]]

sigma = [[0.9849751] [0.03908092] [0.03892423]]

pi = [[-0.43759137] [-0.44669396] [-0.44119304]]

2.2.2 With validation

- **K = 1**

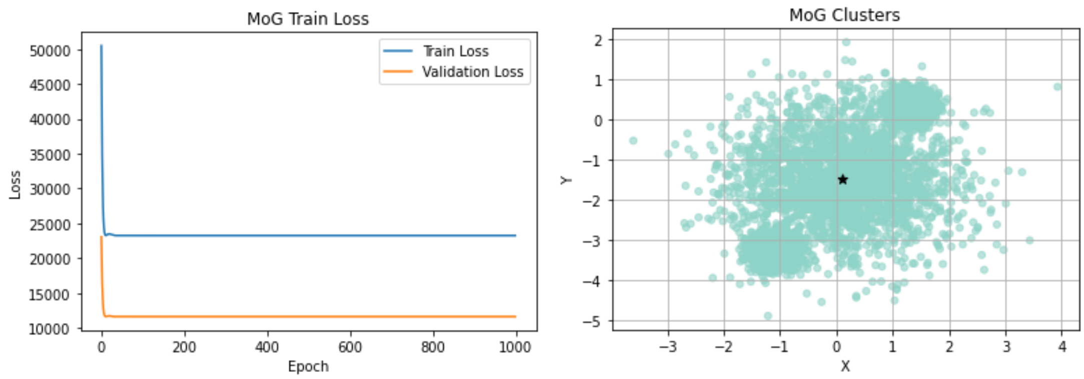


Figure 5(a) & 5(b). Loss plot of K=1 & MoG Clusters plot

- **K = 2**

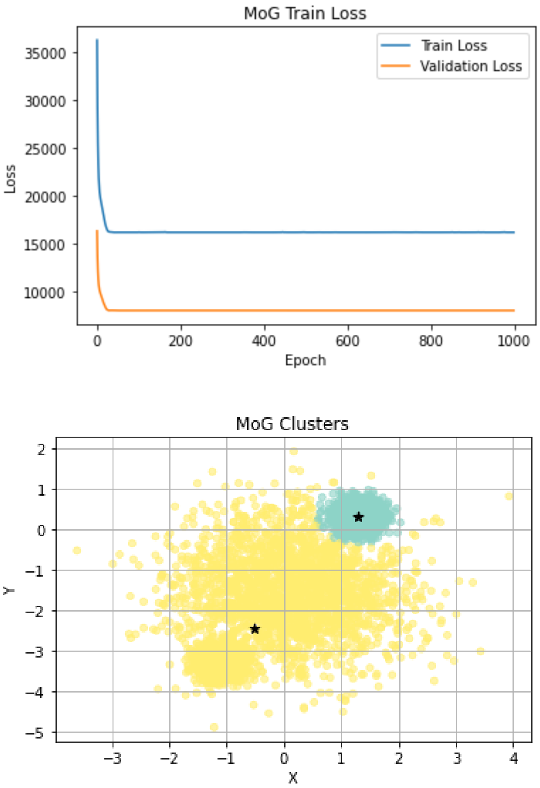


Figure 6(a) & 6(b). Loss plot of K=2 & MoG Clusters plot

- **K = 3**

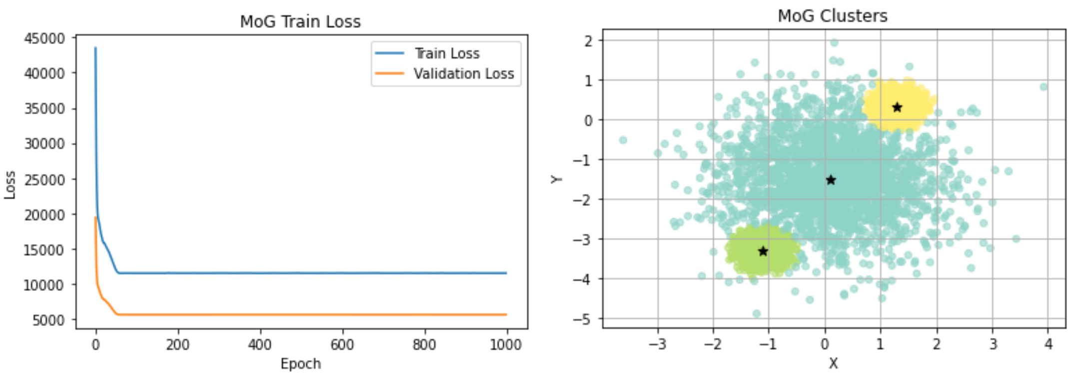


Figure 7(a) & 7(b). Loss plot of K=3 & MoG Clusters plot

- **K = 4**

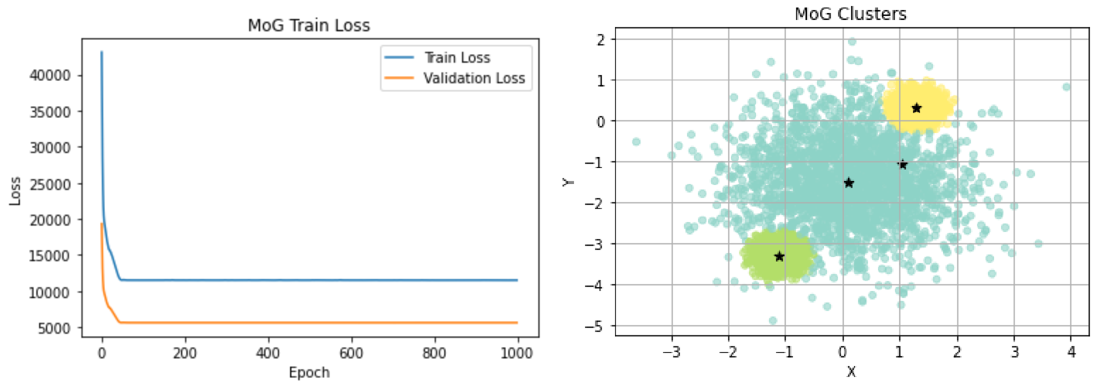


Figure 8(a) & 8(b). Loss plot of K=4 & MoG Clusters plot

- **K = 5**

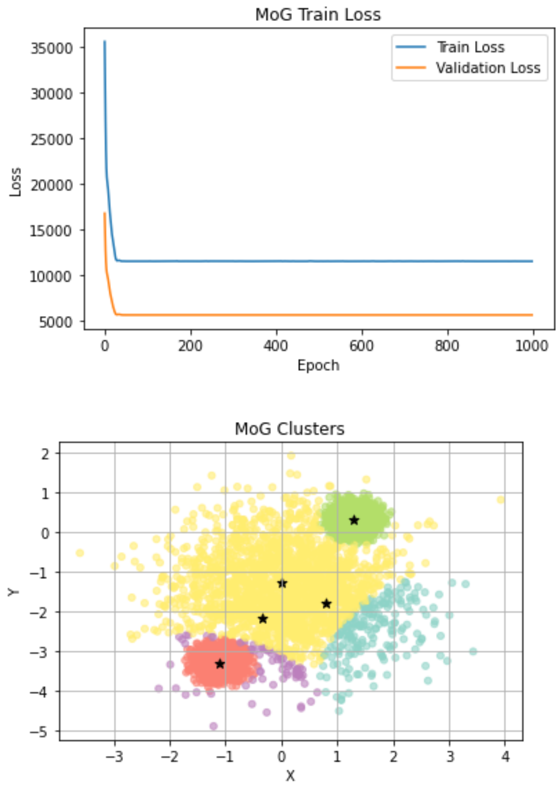


Figure 8(a) & 8(b). Loss plot of K=5 & MoG Clusters plot

K	**Final validation loss**
1	11649.4345703125
2	7981.54541015625
3	5625.50634765625

4	5625.80322265625
5	5624.48828125

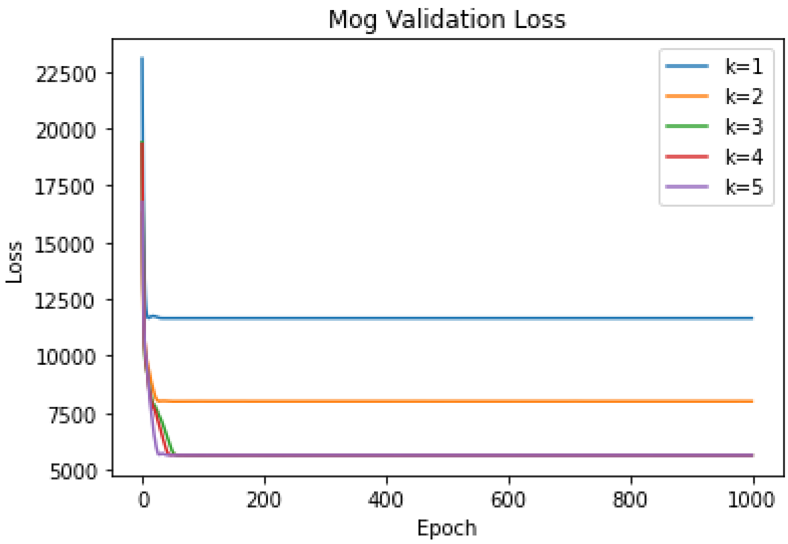


Figure 9. Validation loss plot of K=1 to 5

- **Comment:**

  K = 3 is the best because when k reaches 3, the final validation loss hardly changes. Therefore, 3 is good enough looking at validation loss, and also avoids having too many clusters and remains a relatively simple model for this problem.

2.3 Comparison between K-means and MoG

K	**K-means**	**MoG**
5	153941.703125	22485.708984375
10	120885.7421875	22485.703125
15	69379.9765625	22371.765625
20	69011.625	21766.65625
30	68173.9140625	21766.609375

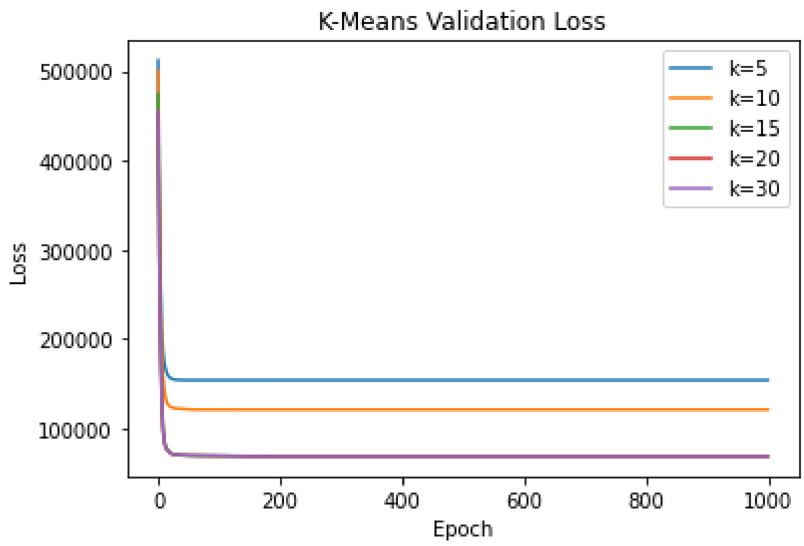


Figure 10. K-means validation loss plot of K=5, 10, 15, 20, 30

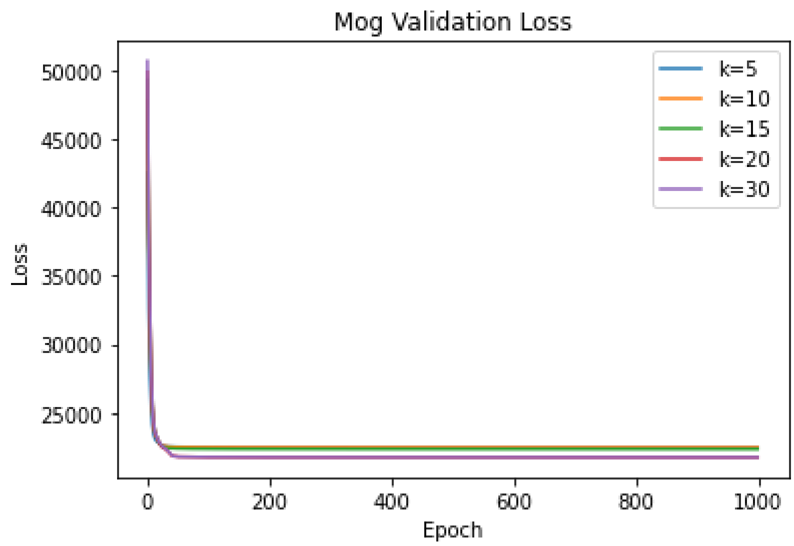


Figure 10. MoG validation loss plot of K=5, 10, 15, 20, 30

- **Comment**

  For the K-means algorithm, when K reaches 15, the total loss is almost unchanged. Therefore, K = 15 is the best choice of number of clusters. And for the MoG algorithm, the differences between all the five Ks are all slight, therefore, K = 5 is the best choice.

  MoG algorithm achieves much lower loss than the K-means algorithm, which indicates MoG works better for the dataset.