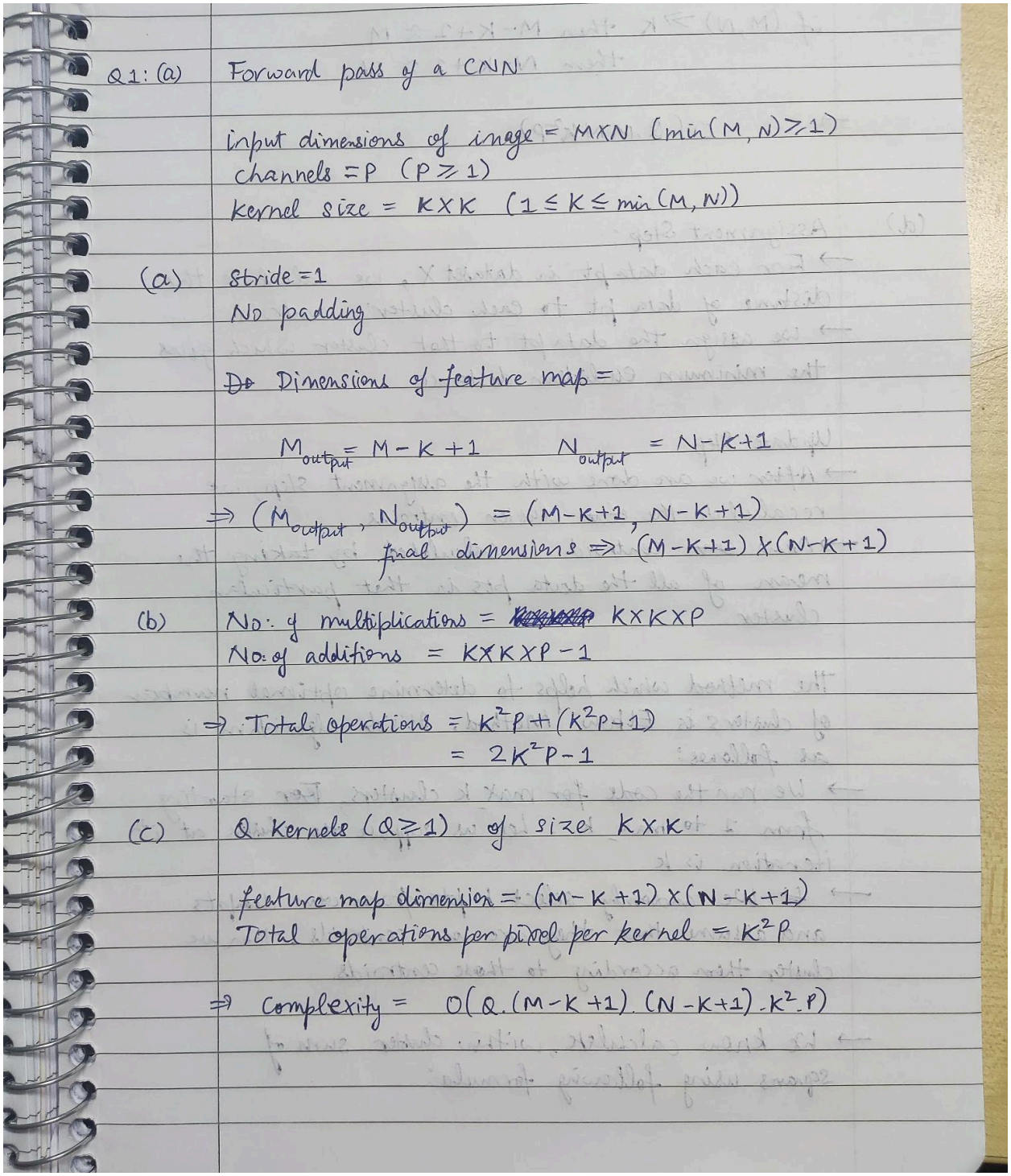
ML Assignment 4

Saumil Lakra, 2021097

Section A



Q1: (a) Forward pass of a CNN

input dimensions of image = $M \times N$ $(\min(M, N) \geq 1)$
channels = P $(P \geq 1)$
kernel size = $K \times K$ $(1 \leq K \leq \min(M, N))$

(a) Stride = 1
No padding

Dimensions of feature map =

$$M_{output} = M - K + 1 \qquad N_{output} = N - K + 1$$

$$\Rightarrow (M_{output}, N_{output}) = (M - K + 1, N - K + 1)$$

final dimensions $\Rightarrow (M - K + 1) \times (N - K + 1)$

(b) No. of multiplications = $K \times K \times P$
No. of additions = $K \times K \times P - 1$

$\Rightarrow$ Total operations = $K^2P + (K^2P - 1)$
$= 2K^2P - 1$

(c) Q Kernels $(Q \geq 1)$ of size $K \times K$

feature map dimension = $(M - K + 1) \times (N - K + 1)$
Total operations per pixel per kernel = K^2P

$\Rightarrow$ Complexity = $O(Q \cdot (M - K + 1) \cdot (N - K + 1) \cdot K^2 \cdot P)$

__/__/____

if $(M, N) \gg K$ then $M - K + 1 \approx M$
then $N - K + 1 \approx N$

$\Rightarrow O(Q \cdot M N K^2 P)$

(d) Assignment Step:
→ For each data pt in dataset X, we calculate the distance of data pt to each cluster's centroid.
→ we assign the data pt to that cluster which gives the minimum euclidean distance

Update Step:
→ After we are done with the assignment step, we recalculate the ~~clus~~ cluster centroids
→ New ~~cen~~ centroid is calculated by taking the mean of all the data pts in that particular cluster.

The method which helps to determine optimal number of clusters is Elbow Method. It's ~~th~~ algorithm is as follows:
→ We run the code for max_k clusters, ~~for~~ starting from 1 to max_k. Let us suppose the clusters at i^{th} iteration is k.
→ ~~t~~ We randomly select k pts from our data pts and assume that they are our centroids. Then we cluster them according to those centroids.
→ We know calculate, within cluster sum of squares using following formula:

__/__/____

$$WCSS = \sum_{i=1}^{K} \sum_{x \in C_i} ||x - \mu_i||^2$$

C_i are the pts in cluster i & ~~x_i~~ μ_i is the centroid of that cluster.

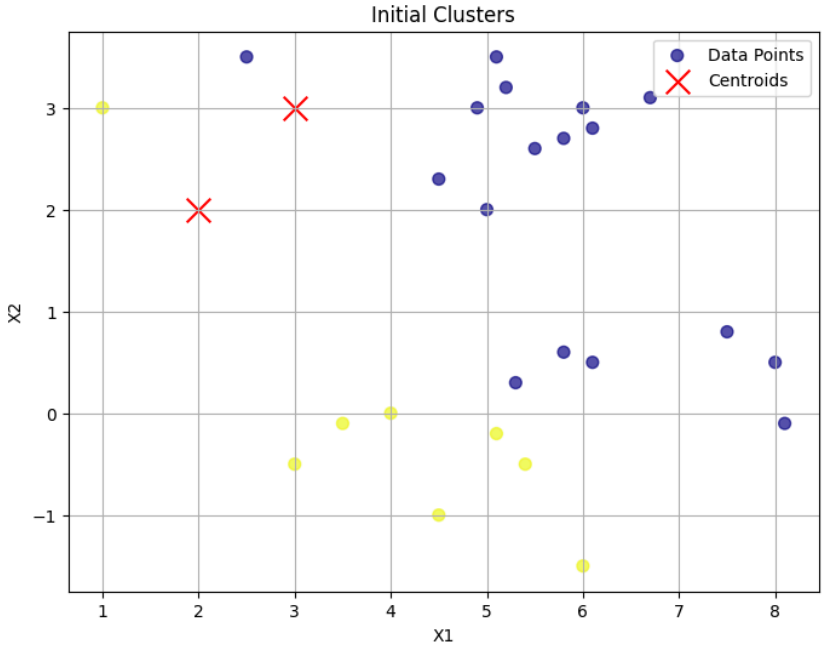
→ Then we plot ~~WCSS~~ WCSS v/s k plot and ~~find~~ find the elbow pt. It is the pt on the graph where rate decreases in WCSS significantly.

No, we cannot assign clusters centroids & arrive at global minima because it may lead to slow ~~conver~~ convergence. K-~~me~~ means may converge to a local minimum based on the initial random assignment of centroids.

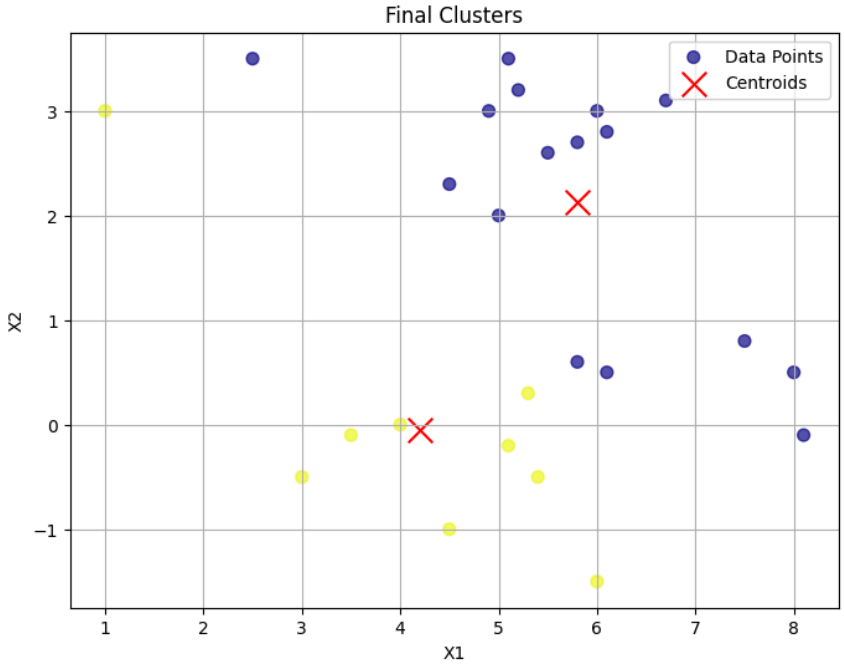
Section B

(b).

Initial Clusters:



Final Clusters:



(c).

Final Centroids:

```
Converged after 3 iterations
array([[ 5.8       ,  2.125     ],
       [ 4.2       , -0.05555556]])
```

Random Centroids:

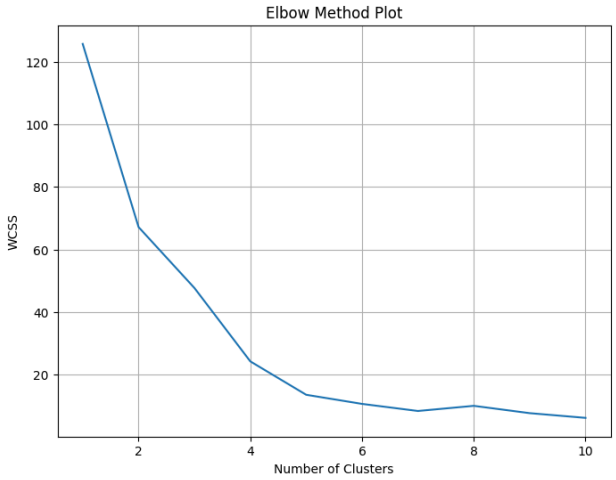
```
Converged after 7 iterations
array([[ 4.85833333,  2.89166667],
       [ 5.56153846, -0.09230769]])
```

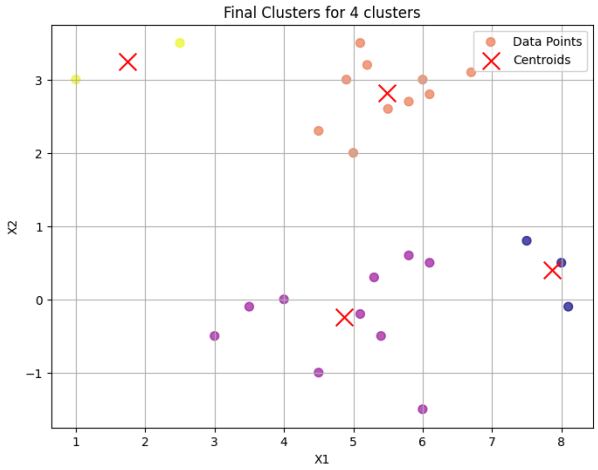
The algorithm is able to find the clusters but it is taking a lot more iterations. Therefore random centroid assignment takes a longer time to converge. Convergence may take a longer time if the dataset is huge.

(d).

Maximum number of clusters takes = 10

Optimal number of clusters, M = 4 (In the plot after running the algorithm again and again, I noticed that the value of M (optimal number of clusters) is 4)





Section C

2.

```
Class: plane
```

```
Class: car
```

```
Class: bird
```

References:

- https://pytorch.org/tutorials/beginner/basics/data_tutorial.html
- https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.htm