CSE343/ECE343: Machine Learning
Assignment-4 CNN, PCA, K-means clustering
Section A

Question 1:

A.

A.

Conv2d Layer:

Output Size = (Input size - Kernel size + 2 * Padding) / Stride + 1

MaxPooling Layer:

Output Size = (Input Size - Kernel Size) / Stride + 1

Given the input image size of 15×15×4:

I. Output size = (15-5+2*1)/1 + 1 = 13 x 13
II. Output size = (13-3)/2 + 1 = 6 x 6
III. Output size = 3.5 x 4.5

Therefore, the final output of the last conv layer: 3x4 and the output of image is 3 x 4 x 1

B. Pooling in CNN helps reduce the spatial dimensions of the input data, reducing computation and memory requirements while retaining important features for effective feature learning.

C. Number of Learning parameters:

Conv2D layer with a kernel of 5×5×4×1:
Number of parameters=5×5×4×1=100
Conv2D layer with a kernel of 5×3×4×1:
Number of parameters=5×3×4×1=60

Total = 100+60=160

B. Yes, it is possible for the k-means algorithm to revisit a configuration during its iterations. However, the algorithm is guaranteed to converge in a finite number of steps. This is because at each iteration, the algorithm minimizes the sum of squared distances between data points and their assigned cluster centroids. As the algorithm progresses, the total distortion decreases with each iteration until convergence, where the centroids no longer change. Since there is a finite number of possible configurations, and the distortion decreases at each step, the algorithm must eventually reach a configuration that it has encountered before, signaling convergence. This convergence ensures that the k-means algorithm terminates after a finite number of iterations.

C. Yes, a neural network can emulate aspects of the K-Nearest Neighbors (KNN) algorithm by learning a similarity metric between data points. The neural network structure typically involves an input layer, one or more hidden layers for capturing complex relationships, and an output layer producing an embedding representing the data point. Activation functions like ReLU are commonly used in hidden layers, while the output layer may utilize linear activation or a similarity-based function. However, this approach transforms the KNN problem into a supervised learning task, introducing computational complexity and data requirements. Traditional KNN might be more suitable for straightforward instance-based tasks, while neural networks can excel in tasks requiring intricate feature learning and representation.

D. Linear filters in CNN capture simple patterns, while non-linear filters capture complex, hierarchical features; for instance, a linear filter might detect edges, while a non-linear filter (ReLU activation) captures more abstract structures like textures or object parts.