Assignment 4 - CNN, K-means

by Kunal Sharma

Machine Learning (CSE343/ECE343)

Prof. Jainendra Shukla

Indraprastha Institute of Information Technology

Delhi

Nov 27, 2024

Section A (Theoretical)

Question 1:

— A —

a)

We are given that:

M = Input Dimension for Height

N = Input Dimension for Width

K = Kernel Size

Formula that can be used to calculate the final feature map's dimensions is as follow:

$$Output\ Dimension\ =\ \frac{Input\ Dimension - Kernel\ Size + 2 \times Padding}{Stride}\ +\ 1$$

Now, as given we have stride = 1 & padding = 0, Putting this in above formula we get,

$$Output\ Height\ =\ M\ -\ K\ +\ 1$$

$$Output\ Width\ =\ N\ -\ K\ +\ 1$$

The dimensions of the resultant feature map are:

$$Output\ Height \times Output\ Width$$

$$(M\ -\ K\ +\ 1) \times (N\ -\ K\ +\ 1)$$

b)

To compute a single output pixel in the resulting feature map, the kernel, which has dimensions $K \times K\ and\ P$ channels, contains a total of $K \times K \times P$ weights. For each pixel, the computation involves $K \times K \times P$ multiplications, as every weight in the kernel is paired with a corresponding input pixel. Following the multiplications, the results are summed together, requiring $K \times K \times P - 1$ addition operations.

Now calculating total number of operations:

$$Total\ Operations = (Multiplications) + (Additions)$$

$$Total\ Operations = (K \times K \times P) + (K \times K \times P - 1)$$

$$Total\ Operations = 2 \times (K \times K \times P) - 1$$

Hence, $2 \times (K \times K \times P) - 1$ operations are required to compute a single output pixel.

c)

To derive the computational time complexity of the forward pass with Q kernels of size$K \times K$, we have to consider the following:

1. Dimensions of the Feature Map

For each kernel, the resulting feature map has dimensions:

$$(M - K + 1) \times (N - K + 1)$$

2. Operations per Pixel

For a single kernel, the operations to compute one pixel are:

$$2 \times (K \times K \times P) - 1 \approx 2 \times (K \times K \times P)$$

3. Total Pixels in the Output

The total number of output pixels for one feature map is:

$$(M - K + 1) \times (N - K + 1)$$

4. Total Operations for One Kernel

The total operations for one kernel are:

$$Operations\ per\ Kernel = 2 \times (K \times K \times P) \times (M - K + 1) \times (N - K + 1)$$

5. Total Operations for All Kernels

With Q kernels, the total operations are:

$$Total\ Operations = Q \times 2 \times (K \times K \times P) \times (M - K + 1) \times (N - K + 1)$$

Hence, Big-O notation:

$$O(Q \times K^2 \times P \times (M - K + 1) \times (N - K + 1))$$

Now, Assuming $min(M, N) \gg K$:

So we will have,

$$M - K + 1 \approx M$$

$$N - K + 1 \approx N$$

So, Now the Big-O notation became,

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

— B —

Assignment Step and Update Step in the K-Means algorithm:

In the K-Means algorithm, the assignment step involves assigning each data point to the nearest cluster centroid by minimizing the distance, usually Euclidean, between the data point and the centroid. For each data point x_i, it is assigned to the cluster C_k such that:

$$C_k = arg\,min_k \left|\left| x_i - \mu_k \right|\right|$$

where μ_k is the centroid of cluster k.

In the update step, the centroids of the clusters are recalculated based on the mean of all points currently assigned to each cluster. For each cluster k, the new centroid μ_k is calculated as:

$$\mu_k = \frac{1}{\left|C_k\right|} \sum_{x_i \in C_k} x_i$$

where $\left|C_k\right|$ is the number of points in cluster k.

These both steps are repeated iteratively until convergence, defined by minimal changes in centroids or reaching a set maximum number of iterations.

Method that helps in determining the optimal number of clusters:

The Elbow Method is a common approach to determine the optimal number of clusters in K-Means. It involves plotting the within-cluster sum of squares (WCSS) against the number of clusters (k). WCSS is calculated as:

$$WCSS = \sum_{k=1}^{K} \sum_{x_i \in C_k} \left|\left| x_i - \mu_k \right|\right|^2$$

As k increases, WCSS decreases, but the rate of decrease slows after a certain point. The "elbow point" on the curve, where the rate of decrease changes significantly, suggests the optimal number of clusters, balancing the trade-off between increasing model complexity and improving fit.

Can we randomly assign cluster centroids and arrive at global minima?

Randomly assigning cluster centroids in the K-Means algorithm does not ensure reaching the global minimum of the objective function. The algorithm is sensitive to initial centroid positions and often converges to local minima. Different initializations can lead to varying results, making the outcome unpredictable. To address this, methods like K-Means++ are used, which select initial centroids in a way that spreads them out across the data space. This improves the chances of finding a better local or even the global minimum and enhances the robustness of the algorithm.