Optimally Partitioning a Deep Learning Model Across Multiple Devices

Problem Overview

The goal is to optimally partition a deep learning model across multiple devices to minimize computation time and transfer time while adhering to memory and accuracy constraints.

Matrices Involved

1. Layer-Device Assignment Matrix (X)

- Represents which device each layer is assigned to.
- Rows correspond to layers, and columns correspond to devices.
- A `1` in the matrix indicates the layer is assigned to that device.

Example:

$X = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ \end{bmatrix}$

This matrix indicates:

- Layers 1, 2, and 3 are assigned to Device 1.
- Layer 4 is assigned to Device 2.
- Layers 5 and 6 are assigned to Device 3.

2. Transfer Time Matrix (T)

- Represents the time required to transfer data between devices.
- Rows and columns correspond to devices.
- An element $T[i, j]$ indicates the time to transfer data from Device i to Device j.

Example:

$T = \begin{bmatrix} 0 & 2 & 4 \\ 2 & 0 & 3 \\ 4 & 3 & 0 \\ \end{bmatrix}$

This matrix indicates:

- It takes 2 units of time to transfer data from Device 1 to Device 2.
- It takes 4 units of time to transfer data from Device 1 to Device 3.
- It takes 3 units of time to transfer data from Device 2 to Device 3.

Steps to Calculate Transfer Time

Step 1: Create the Shifted Assignment Matrix (X_{shifted})

- Shift the X matrix **up** by one row to compare consecutive layers.
- The first row is set to zeros because there's no layer before the first one.

Example:

$X_{\text{shifted}} = \begin{bmatrix} 1 & 0 & 0 \ 1 & 0 & 0 \ 0 & 1 & 0 \ 0 & 0 & 1 \ 0 & 0 & 1 \ 0 & 0 & 0 \ \end{bmatrix}$

Step 2: Compute the Transition Matrix (M)

- Calculate the transition matrix M by taking the dot product of X^T (transpose of X) and X_{shifted}.
- The matrix M counts how often a layer on one device is followed by a layer on the same or a different device.

Example:

$M = X^T \times X_{\text{shifted}}$

This gives us:

$M = \begin{bmatrix} 2 & 1 & 0 \ 0 & 0 & 1 \ 0 & 0 & 1 \ \end{bmatrix}$

Step 3: Calculate the Transfer Time

- Perform element-wise multiplication of the transition matrix M with the transfer time matrix T.
- The result gives the transfer time required for each transition between devices.

Example:

$\text{transfer_matrix} = M \odot T$

This gives us:

$\text{transfer_matrix} = \begin{bmatrix} 0 & 2 & 0 \ 0 & 0 & 3 \ 0 & 0 & 0 \ \end{bmatrix}$

Step 4: Sum the Transfer Times

- Finally, sum all the elements in the transfer_matrix to get the total transfer time.

Example:

$T_{\text{transfer}} = \sum \text{transfer_matrix}$

This results in:

$T_{\text{transfer}} = 5 \text{ units}$

Steps to Calculate Computation Time

Step 1: Define Computation Time for Each Layer

- Assume we have a matrix C that represents the computation time for each layer on each device.

Example:

$C = \begin{bmatrix} 3 & 0 & 0 \ 3 & 0 & 0 \ 3 & 0 & 0 \ 0 & 2 & 0 \ 0 & 0 & 4 \ 0 & 0 & 4 \ \end{bmatrix}$

Step 2: Calculate Computation Time for Each Device

- The computation time for each device is the sum of the computation times for the layers assigned to that device.

Example:

$T_{\text{comp}} = \sum C, \text{ axis=0}$

This results in:

$T_{\text{comp}} = \begin{bmatrix} 9 & 2 & 8 \end{bmatrix}$

Step 3: Determine the Total Computation Time

- The total computation time is determined by the device with the maximum computation time, as the devices work in parallel.

Example:

$T_{\text{total_comp}} = \max(T_{\text{comp}})$

This results in:

$T_{\text{total_comp}} = 9 \text{ units}$

Final Objective

Minimize the following:

1. **Computation Time** ($T_{\text{total_comp}}$):
    - The maximum computation time across all devices.
    - Example: Minimize $T_{\text{total_comp}}$ to reduce processing delays.
2. **Transfer Time** (T_{transfer}):
    - The total transfer time between devices.
    - Example: Minimize T_{transfer} to reduce latency.