Optimizing Space Complexity in Tensor Operations

Algorithmic Optimization Report

---

1. Problem Statement

Given a matrix $X \in \mathbb{R}^{n \times c}$ and a 3D tensor $Y \in \mathbb{R}^{n \times n \times g}$, we want to perform a sequence of operations that traditionally requires $O(n^2 c)$ memory due to large intermediate tensors. The operations are:

1. **Projection of Y:** Use a weight matrix $W \in \mathbb{R}^{g \times c}$ to project Y into a tensor $Z \in \mathbb{R}^{n \times n \times c}$.

2. **Projection and Broadcasting of X:** Use matrices $P, Q \in \mathbb{R}^{c \times c}$ to project X into matrices $X_1, X_2 \in \mathbb{R}^{n \times c}$. Broadcast X_1 along the first axis to form $A_1 \in \mathbb{R}^{n \times n \times c}$, and broadcast X_2 along the second axis to form $A_2 \in \mathbb{R}^{n \times n \times c}$. Add A_1 and A_2 to form a tensor $B \in \mathbb{R}^{n \times n \times c}$.

3. **Element-wise Product and Reduction:** Compute the element-wise product of B and Z, then sum the result along the first axis to obtain a final output matrix of shape $n \times c$.

2. Mathematical Breakdown

The core issue is the materialization of the $n \times n \times c$ intermediate tensors (Z, A_1, A_2, and B). When n and c are large, this $O(n^2 c)$ space complexity quickly becomes a memory bottleneck. We can avoid this by algebraically distributing the operations and changing the order of summation.