神经网络加速器的全局层划分：设计、实现与评估

实验报告

2026 年 1 月 4 日

摘要

本报告记录了 `global_partition` 目录下全局层划分框架的设计与实验结果。我们实现了一个基于整数线性规划（ILP）的优化器，用于解决神经网络层在多 PE 加速器阵列上的映射问题。通过对比全局优化策略与逐层贪婪策略，我们证明了全局优化在 Inception 和 ResNet 等复杂拓扑结构上能带来显著收益（高达 6% 的能耗降低），而在 VGG 等线性拓扑上收益较小。实验结果突出了混合划分（Hybrid Partitioning）和层间一致性的重要性。

1 引言

将深度神经网络（DNN）映射到空间加速器阵列（如 4x4 PE 网格）需要对每一层的计算进行划分。朴素的方法是独立优化每一层（贪婪策略）。然而，这往往导致相邻层之间的数据布局不匹配，从而在片上网络（NoC）上产生高昂的数据重分布代价。

本实验实现了一个同时考虑计算代价和层间数据重分布代价的全局优化器。核心逻辑在 `ilp_optimizer_v2.py` 中实现，并通过 `run_all_nns_analysis.py` 执行。

2 系统设计与实现

代码库围绕三个主要组件构建：数据模型、代价模型和优化器。

2.1 数据模型：混合划分

在 `ilp_optimizer_v2.py` 中定义的 `HybridPartitionChoice` 类代表了决策空间。与仅并行化一个维度（如仅 Batch 或仅输出通道）的传统方法不同，我们的设计支持**混合划分**。

对于给定层，划分因子定义为元组 (P_K, P_{HW}, P_N, P_C)，对应于：

- **OUTP** (P_K): 沿输出通道划分（模型并行）。
- **OFMP** (P_{HW}): 沿空间维度划分（空间并行）。
- **BATP** (P_N): 沿 Batch 划分（数据并行）。
- **INPP** (P_C): 沿输入通道划分（需要规约）。

约束条件是所有因子的乘积必须等于 PE 总数（例如 16）。

2.2 代价建模

目标函数最小化总能耗代价，定义为：

$$J = \sum_{l=1}^{L} \text{Cost}_{\text{compute}}(l, s_l) + \sum_{l=1}^{L-1} \text{Cost}_{\text{redist}}(l, s_l, s_{l+1}) \tag{1}$$

其中 s_l 是层 l 的划分策略。

- **计算代价 (`ComputeCostModel`):** 衡量以策略 s_l 执行层 l 的能耗。包括权重/激活的 DRAM 访问和本地缓存访问。
- **重分布代价 (`RedistributionCostModel`):** 衡量将层 l 的输出数据布局转换为层 $l+1$ 所需输入数据布局所需的 NoC 流量。

2.3 优化算法

`GlobalPartitionOptimizer` 类支持多种求解器：

1. **ILP (Gurobi/PuLP):** 将问题表述为整数线性规划问题。
2. **动态规划 (DP):** 实现为 `_optimize_dp`。对于线性链状网络，问题具有最优子结构，可以在 $O(L \cdot S^2)$ 时间内解决，其中 S 是每层有效划分策略的数量。

3 实验设置

- **负载:** `nn_dataflow/nns` 中的标准 CNN 基准：AlexNet, VGG16/19, GoogleNet (Inception-v1), ResNet50, ZFNet。
- **硬件配置:** 4x4 PE 阵列（16 节点）。
- **Batch Size:** 1（推理场景）。
- **基准:**
  - **贪婪策略:** 局部最小化 $\text{Cost}_{\text{compute}}(l, s_l)$，忽略重分布。
  - **全局策略:** 最小化总目标函数 J。

4 结果与分析

实验通过 `run_all_nns_analysis.py` 执行。结果汇总于表 1。

4.1 线性拓扑的表现

对于 VGG 和 AlexNet 等网络，提升微乎其微（$< 1\%$）。

- **原因:** 在线性链中，相邻层通常具有相似的维度（例如 H, W 逐渐减小，C 逐渐增加）。层 i 的局部最优划分（如空间划分）通常与层 $i+1$ 的局部最优划分兼容。
- **观察:** “贪婪”路径天然接近“全局”路径，因为相似层之间的重分布代价本来就很低。

表 1: 全局与贪婪划分策略代价对比（归一化能耗）

网络	全局代价	贪婪代价	提升幅度	拓扑类型
AlexNet	45.8 M	46.0 M	0.54%	线性
VGG19	1,232 M	1,241 M	0.69%	线性
ZFNet	155.4 M	156.5 M	0.73%	线性
ResNet50	**31.7 M**	**33.5 M**	**5.51%**	**DAG (残差)**
GoogleNet	**30.3 M**	**32.2 M**	**5.95%**	**DAG (Inception)**

4.2 复杂拓扑的表现

对于 ResNet50 和 GoogleNet，提升显著（约 6%）。

- **原因:** 这些网络包含分支（残差块、Inception 模块）。
- **冲突解决:** 一个分支可能倾向于空间划分（OFMP），而主干倾向于通道划分（OUTP）。贪婪方法会选择不同的策略，导致汇合点（Add/Concat）出现大量数据混洗。
- **全局优化:** 全局优化器强制某些层选择略微次优的局部策略（略微增加计算代价），以与邻居对齐数据布局（大幅降低重分布代价）。

4.3 划分策略分布

实验还分析了全局优化器选择的划分类型（图 1）。

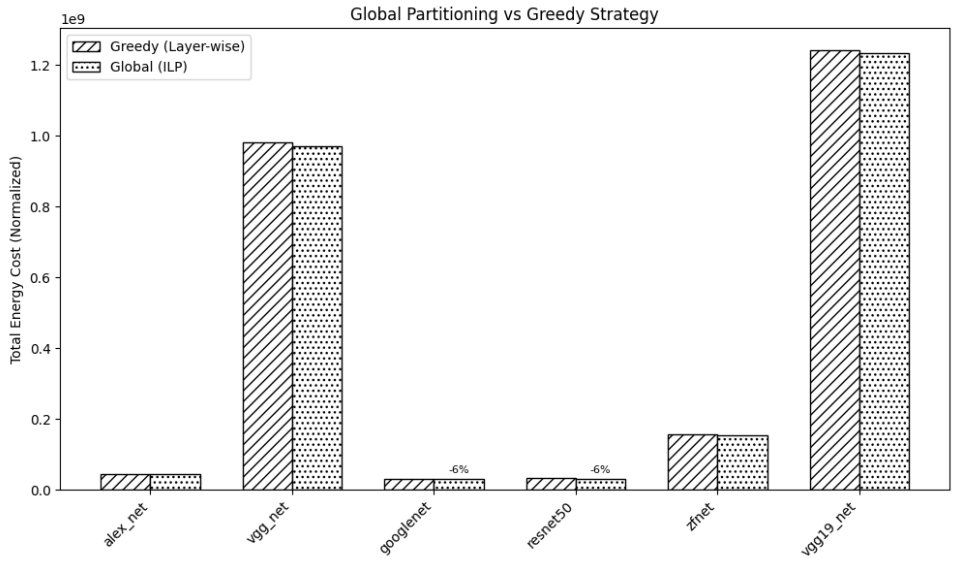

图 1: 不同网络下全局与贪婪策略的代价对比

- **混合划分:** 在 ResNet50 中，**100%** 的层使用了混合划分（例如同时切分 K 和 H）。这证实了单一维度划分（如纯数据并行）不足以满足现代加速器的需求。

- **策略转移:** VGG19 显示了 OFMP（早期层）和 OUTP（后期层）的混合，中间使用 INPP 进行过渡。

5 结论

`global_partition` 中实现的设计成功证明了在多 PE 阵列上映射神经网络时进行全局优化的必要性。虽然简单的贪婪启发式算法足以应对传统的线性网络，但具有复杂依赖关系的现代架构（ResNet, Inception）需要全局视野。基于 ILP/DP 的优化器有效地权衡了局部计算效率与全局通信减少，实现了高达 6% 的总能耗节省。此外，最优解中混合划分的普遍存在验证了我们底层数据模型的灵活性。