Hardware-Constrained Neural Architecture Search

Firsov Sergey

Moscow Institute of Physics and Technology

2024

Annotation

- Neural Architecture Search (NAS) automates the design of neural networks.
- Balancing accuracy, complexity, and hardware constraints is critical for real-world applications.
- This work extends DARTS by:
  - Complexity-aware architecture representation.
  - Simplex-based complexity control.
  - Hardware-aware latency regularization.
- Result: Efficient optimization of architectures tailored to hardware constraints.

What is Neural Architecture Search (NAS)?

- **Definition:** Automated design of neural networks for specific tasks.
- **Goals:**
  - Optimize accuracy and efficiency.
  - Minimize human intervention in architecture design.
- **Challenges:**
  - High computational cost.
  - Balancing trade-offs between model performance and complexity.
  - Hardware constraints often overlooked.
- **Popular Approaches:** DARTS, ProxylessNAS, FBNet.

Upgrade 1: Complexity-Aware Architecture Representation

- Architectural parameters $\boldsymbol{\alpha}$ depend on complexity parameter λ:

$$\boldsymbol{\alpha} : \lambda \to \mathcal{A}.$$

- Generates multiple architectures with varying complexities in a single optimization process.
- Ensures flexibility in architecture design across resource constraints.

Upgrade 2: Simplex-Based Complexity Control

- Replace scalar complexity parameter (λ) with a simplex representation:
$$\boldsymbol{S} \in \Delta^{k-1}.$$
- Δ^{k-1}: $(k-1)$-dimensional simplex, where k is the number of operations.
- Sampling via Gumbel-Softmax enables differentiable selection of operations:
$$\boldsymbol{S} \sim \mathrm{GumbelSoftmax}(\boldsymbol{\theta}).$$

Upgrade 3: Hardware-Aware Latency Regularization

- Introduce a latency lookup table for operations on the target hardware.
- Compute total latency as a weighted sum of selected operations.
- Modify the objective function:

$$\mathbb{E}_{\boldsymbol{S}\sim\text{GumbelSoftmax}}\left[L + \lambda \cdot \text{Reg} + \kappa \cdot \text{Latency}\right].$$

- Trade-off between accuracy, complexity, and hardware efficiency.

Final Loss Function and Analysis

- **Objective:** Optimize across multiple factors:

  $$\mathbb{E}_{\boldsymbol{S}\sim\text{GumbelSoftmax}}\left[L(\boldsymbol{w}^*(\boldsymbol{S}),\boldsymbol{\alpha}(\boldsymbol{S}))+\lambda\cdot\text{Reg}(\boldsymbol{\alpha})+\kappa\cdot\text{Latency}(\boldsymbol{\alpha})\right].$$

- **Key Insights:**
  - Complexity control ensures efficient models.
  - Latency regularization aligns optimization with hardware constraints.

Planned Experiments

- Validate the proposed method on:
  - Image classification benchmarks (e.g.,fmnist, CIFAR-10).
  - Hardware-specific performance metrics.
- Compare against:
  - Baseline methods (DARTS, ProxylessNAS, FBNet).
  - Metrics: Accuracy, latency, and complexity.
- Evaluate trade-offs between accuracy and latency.

Results

- Placeholder for experimental results.
- Future work: Analyze results and provide insights.

References I