Automated Neural Architecture Search for Resource-Constrained Environments

Sahajpreet Singh Khasria, Computer Science
Mentor: Rong Pan, Professor
School of Computing and Augmented Intelligence

Introduction

Neural Architecture Search (NAS) is an emerging field in machine learning that aims to automate the design of neural network architectures. This study focuses on implementing NAS in resource-constrained environments, addressing the challenges of limited computational power and energy consumption. Our work is particularly relevant in an era where edge computing and mobile AI applications are becoming increasingly important, necessitating efficient model architectures that can perform well under hardware limitations.

We explore a customized evolutionary search algorithm for NAS, tailored to work effectively in environments with constrained computational resources. Our approach balances the trade-off between model performance and resource utilization, aiming to discover architectures that are both accurate and efficient.

Impact

- Improved efficiency in model design process:
  - Reduction in manual effort required for architecture design
  - Faster iteration and experimentation cycles
  - Potential for discovering novel architectural patterns
- Reduction in overall energy usage during architecture search:
  - Lower carbon footprint compared to traditional NAS approaches
  - Enables NAS on commodity hardware, reducing reliance on large-scale compute clusters
- Potential for creating more compact and efficient neural network models:
  - Improved inference speed on resource-constrained devices
  - Reduced memory footprint, enabling deployment on devices with limited RAM
- Democratization of AI by making advanced model design accessible with limited resources:
  - Enables researchers and developers with limited computational resources to participate in cutting-edge AI research
  - Promotes innovation in fields where large-scale computing infrastructure is not readily available

Methodology

We implemented a customized evolutionary search algorithm for NAS, leveraging the following technologies:

- PyTorch: Primary deep learning framework for model implementation and training
- torchvision: For dataset loading and image transformations
- NumPy: For numerical computations and data manipulation
- torchsummary: For model visualization and parameter counting

Our NAS approach consists of the following key components:

1. Search space definition:
   - Convolutional Neural Networks with varying layers (1 to 5)
   - Number of filters per layer: 16 to 128
   - Kernel sizes: 3x3, 5x5, 7x7
   - Optional batch normalization
2. Population initialization: Random generation of architectures within the defined search space
3. Evaluation:
   - Dataset: CIFAR-10
   - Data augmentation: Random cropping, horizontal flipping
   - Training: SGD optimizer with momentum, learning rate scheduling
   - Metrics: Classification accuracy, model size (number of parameters)
4. Selection: Top-performing models based on accuracy are chosen for the next generation
5. Crossover: Architectural features of parent models are combined to create offspring
6. Mutation: Random alterations to model parameters to introduce diversity

The search process is iterated over multiple generations, with each generation refining the population of model architectures.

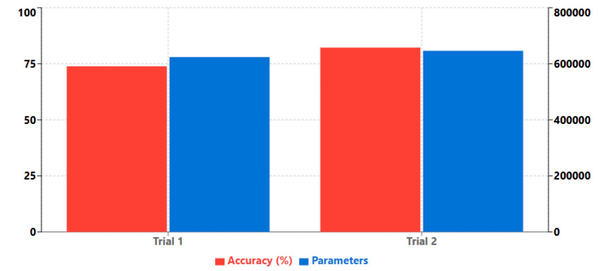
Observations

Our experiments yielded the following key observations:

- Significant improvement in accuracy:
  - Trial 1: 73.86% accuracy with ~624K parameters
  - Trial 2: 82.23% accuracy with ~646K parameters
  - 8.37 percentage point increase in accuracy with only a 3.5% increase in model size
- Effectiveness of data augmentation and learning rate scheduling:
  - Introduction of data augmentation techniques in Trial 2 contributed to improved generalization
  - Learning rate scheduling helped in fine-tuning the model, leading to better convergence
- Trade-off between model size and accuracy:
  - Slight increase in model complexity led to substantial accuracy gains
  - Demonstrates the importance of balancing model size and performance in resource-constrained settings
- Evolutionary approach effectiveness:
  - The evolutionary algorithm successfully navigated the search space to find improved architectures
  - Crossover and mutation operations contributed to the discovery of better-performing models

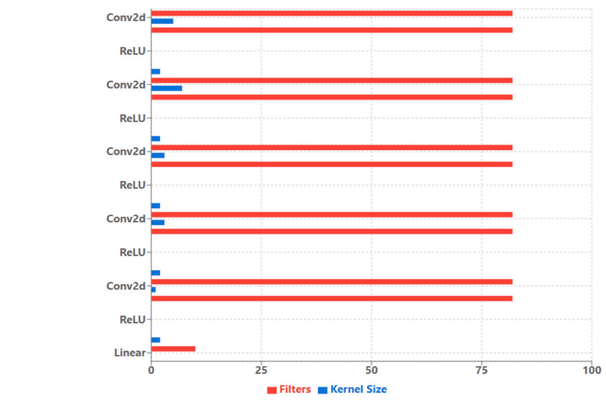
NAS Observation Charts

Trial Comparison



Legend: This chart compares the accuracy and number of parameters between Trial 1 and Trial 2. The red bars represent the accuracy percentage (left Y-axis), while the blue bars represent the number of parameters (right Y-axis).

Best Model Architecture (Trial 2)



Legend: This chart shows the architecture of the best-performing model from Trial 2. Each row represents a layer in the model. The red bars show the number of filters in each layer, while the blue bars represent the kernel size (where applicable).

Future Work

We propose the exploration of Recnas (Recursive Network Architecture Search) as a promising direction for future research:

- Implement recursive search strategies for more efficient exploration of the architecture space:
  - Develop hierarchical search methods to decompose the architecture search problem
  - Investigate the use of meta-learning techniques to guide the recursive search process

FURI

Ira A. Fulton Schools of Engineering
Arizona State University