Neural Architecture Search: An Empirical View

Mingjie Sun

What is Neural Architecture Search?

- Start
  - (ICLR 2017) Neural Architecture Search with Reinforcement Learning
- Automatically design network architectures
  - CNN
  - RNN cell (e.g. LSTM)
- Transfer from hand-crafted design to automatically search

Ingredients

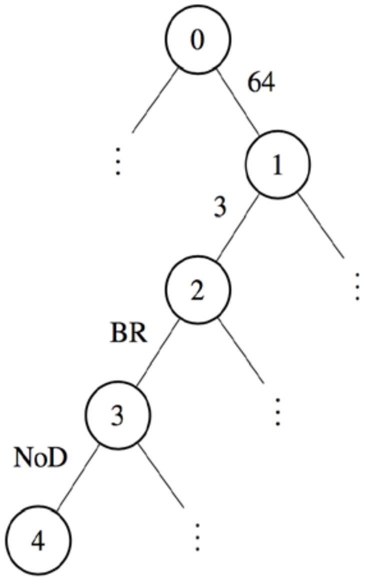
- Search Space
    - Search Tree
    - Computational Graph
    - Memory Bank
- Search Algorithm
    - REINFORCE
    - Tree Search
    - Evolutionary algorithm
- Search Algorithm and Search Space are coupled.

Framework

1. Design a search space
2. Sample models from the search space
    a. random
    b. guided by search algorithm
3. Evaluate the sampled models
    a. train the model for some epochs/time
        i. on standard datasets (e.g. CIFAR10, CIFAR100)
    b. use the validation accuracy at the end as a signal
4. Search algorithm uses the evaluation results to guide further search

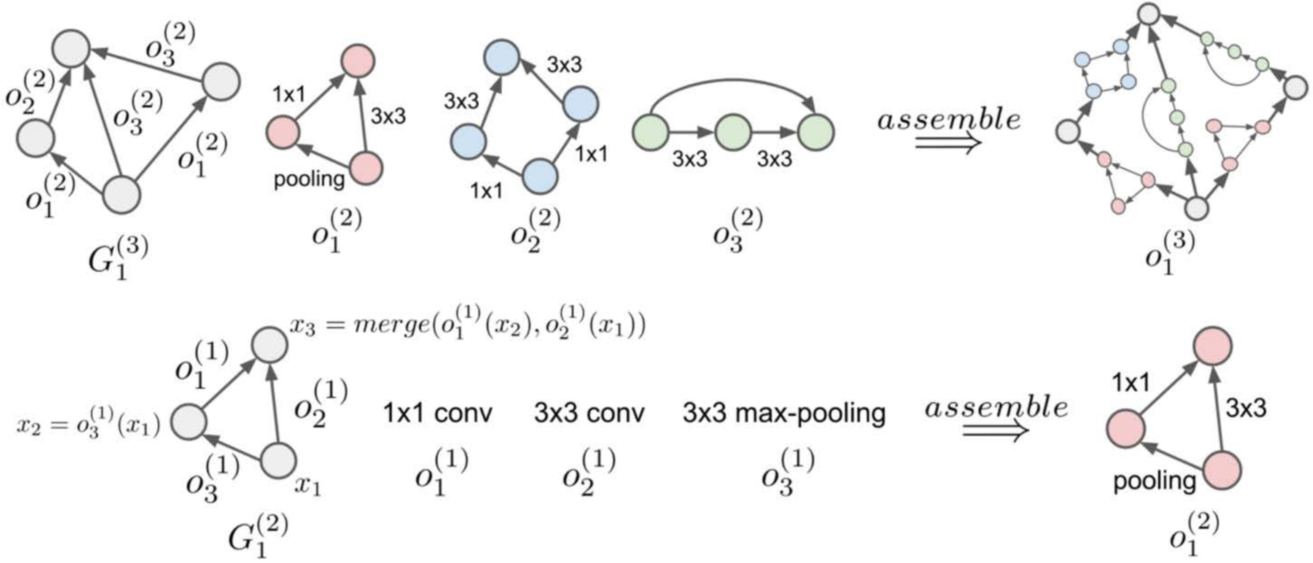
Search Tree

- Search space is represented by a rooted tree.
  - Tree Search Algorithms
- Each edge specifies an assignment.
  - depth
  - activation function
  - filter size
  - and so on ……
- Leaf node represents a specified neural network.



Computational Graph

- Use primitive computation to form complex computational graph
  - Evolutionary algorithms



Memory Bank

- Represent network architectures as memory read/write operations.

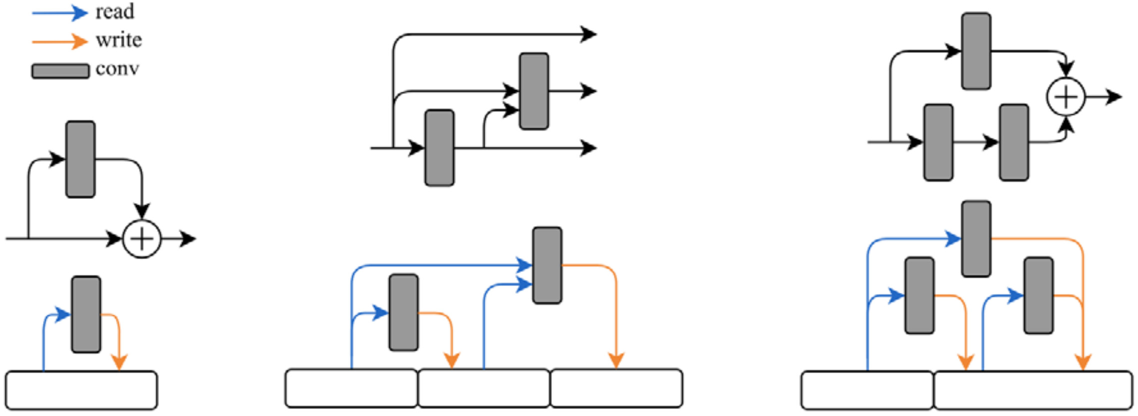


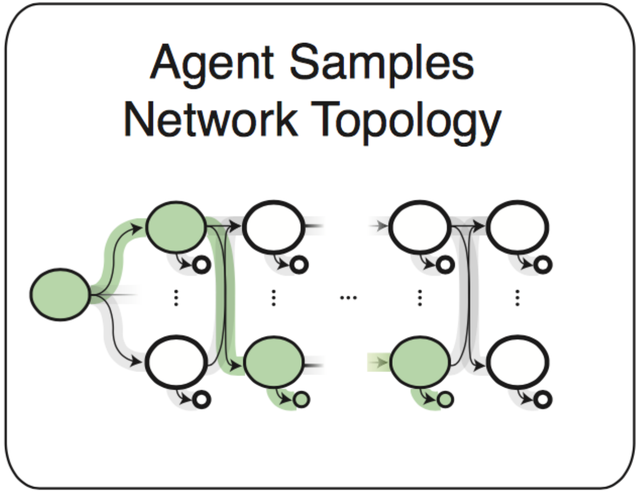
Figure: a memory bank representations for Resnet/DenseNet/FractalNet

Pitfall & Insight

- Decouple depth from the search space.
  - Going deeper, the training tends to get longer.
  - If not given enough time resource, it could add potential bias to the search process.
    - shallow model
- Encourage the algorithm to learn complex cell structure.
  - Advantages:
    - generalize to deep model easily (e.g. stacking)
    - transfer to other tasks (Imagenet)
- Different search space allows for different specialized algorithms.

Search Algorithm

- Tree search
  - Monte Carlo Tree Search
- Reinforcement Learning
  - Q-learning
  - REINFORCE
- Evolutionary Algorithm
- and so on ……



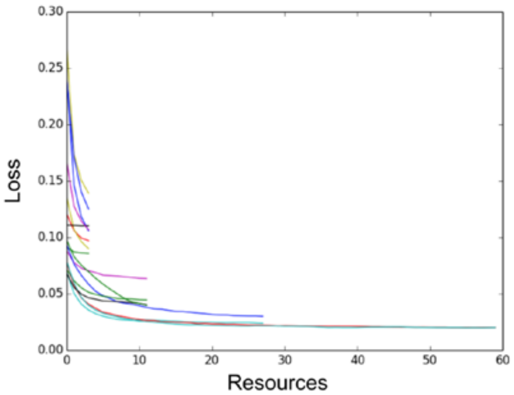
-

Acceleration

- Traditional Paradigm
  - Train each sampled model for a fixed epochs/time
  - Problem:
    - Waste time on bad models.
- Why important?
- Accelerating technique
  - Early Stopping
    - Idea : Make use of early information
  - One-shot evaluation
    - pre-trained parameters

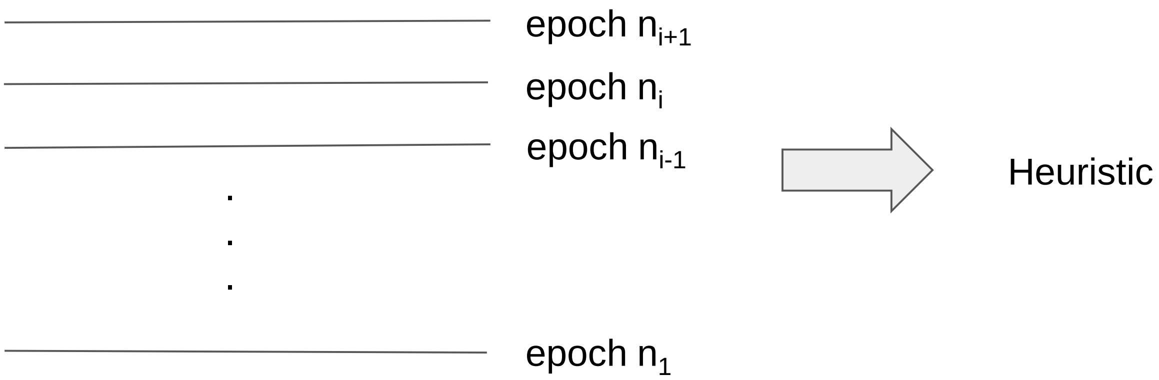
Early Stopping

- Origin
    - used as a way to prevent overfitting in neural networks
- Help accelerate search by pruning bad models
    - Hyperband (Li, et.al)
    - Population based training (DeepMind)
- Illustration



An unified view of Early Stopping

- Pre-define the “training hierarchy”
- At each epoch in the hierarchy, use heuristics to decide whether to continue training.



Heuristic

- Performance Prediction
  - Use a pre-trained model to predict the final performance.
  - Continue if the predicted performance is good.
- Comparison Based
  - Procedure:
    - Evaluate a set of models in parallel.
    - Keep training models with good performance from the set.
  - Advantages:
    - An adaptive approach: No need to design complicated heuristic

Other techniques & Discussions

- Other techniques:
  - One-shot evaluaton:
    - Use pretrained parameters
- Discussions:
  - Can we beat random search?
    - Pruning technique is the key to beat random search.
  - Challenges:
    - Design efficient search space representations
    - and so on …….

References

1. Baker, B. et.al (2017) Designing Neural Network Architecture Using Reinforcement Learning
2. Brock, A. et.al (2017) SMASH: One-Shot Model Architecture Search through HyperNetworks
3. Li, L. et.al (2017) Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization
4. Jaderberg, M. et.al (2017) Population Based Training of Neural Network
5. Liu, H. et.al (2017) Hierarchical Representations for Efficient Architecture Search
6. Zoph, B. et.al (2017) Neural architecture search with reinforcement learning.

Q&A