Understanding the wiring evolution in differentiable neural architecture search

Sirui Xie*[1], Shoukang Hu*[2], Xinjiang Wang[3], Chunxiao Liu[3], Jianping Shi[3], Xunying Liu[2], Dahua Lin[2]

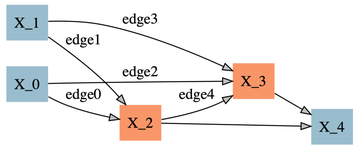
1. University of California, Los Angeles 2. The Chinese University of Hong Kong 3. SenseTime Research

Abstract

Motivated by three searching patterns of differentiable NAS, we

- Propose a unified view on searching algorithms of existing frameworks, transferring the global optimization to local cost minimization.
- Conduct empirical and theoretical analyses, revealing implicit biases in the cost's assignment mechanism and evolution dynamics that cause the phenomena.
- These biases indicate strong discrimination towards certain topologies.

Search Space and Operation Selection

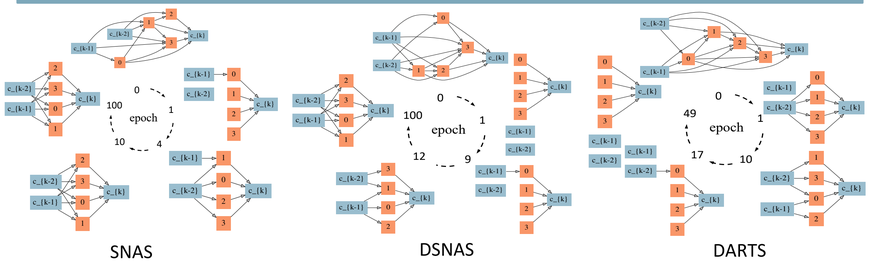


- Nodes X_i represent feature maps
- *input nodes* and *output nodes* (blue), *intermediate nodes* (orange)
- Operation candidates: None, skip connect, max_pool_3x3, avg_pool_3x3, sep_conv_3x3, dil_conv_3x3, dil_conv_5x5, sep_conv_5x5

- Each edge (i, j) multiply a one-hot variable $\boldsymbol{Z}_{i,j}$ with operation candidates $\boldsymbol{O}_{i,j}$:

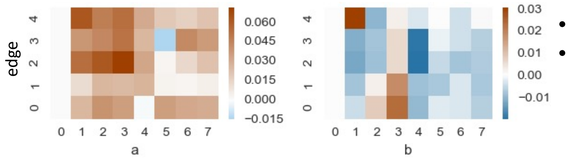
$$\widetilde{\boldsymbol{O}}_{i,j}(\cdot) = \boldsymbol{Z}_{i,j}^T \boldsymbol{O}_{i,j}(\cdot)$$

Evolution Pattern



- (**P1**) Growing tendency: all edges are dropped in the beginning, some of them gradually recover
- (**P2**) Width preference: intermediate edges barely recover from the initial drop
- (**P3**) Catastrophic failure: in bi-level setting, no edge can recover from the initial drop

Probing into the Evolution with Local Cost Mean



- the cost of None is 0
- the network topology is determined by signs of cost of other operations

Hypothesis: Cost of operations except *None* are positive *near* initialization and negative near convergence. Cell topology thus exhibits a tendency of growing.

A close Look at Cost Assignment

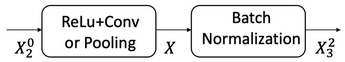
- Type1: edge 0, 1. E.g., $edge$ 0 involves two paths of back-propagation,*i.e.* 1^{st} path (4-2-0) and 2^{nd} path (4-3-2-0):

$$C(Z^s_{0,2}) = \frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_2^2}\boldsymbol{X}_2^0 + \frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_3^2}\frac{\partial \boldsymbol{X}_3^2}{\partial \boldsymbol{X}_2^0}\boldsymbol{X}_2^0.$$

- Type2: edge 2, 3, 4. E.g., $edge$ 2 only involves one paths of back-propagation, *i.e.* path (4-3-0):

$$C(Z^s_{0,3}) = \frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_4^3}\frac{\partial \boldsymbol{X}_4^3}{\partial \boldsymbol{X}_3^0}\boldsymbol{X}_3^0 = \frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_4^3}\boldsymbol{X}_3^0.$$

Theorem 1. A path does not distribute cost from the output edge after passing one intermediate edge.



Let $X \in \mathbb{R}^{B\times C_{out}\times W_{out}\times H_{out}}$ denote the Conv output on edge 4, we expand $C(Z^s_{0,2})$ at path (4-3-2-0):

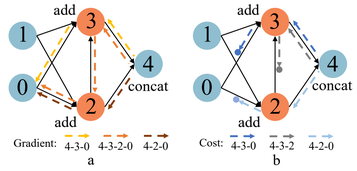
$$\frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_3^2}\frac{\partial \boldsymbol{X}_3^2}{\partial \boldsymbol{X}_2^0}\boldsymbol{X}_2^0 = \frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_3^2}\frac{\partial \boldsymbol{X}_3^2}{\partial \boldsymbol{X}}\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{X}_2^0}\boldsymbol{X}_2^0.$$

Expanding to the element-wise calculation:

$$\frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_3^2}\frac{\partial \boldsymbol{X}_3^2}{\partial \boldsymbol{X}}\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{X}_2^0}\boldsymbol{X}_2^0 = \frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_3^2}\frac{\partial \boldsymbol{X}_3^2}{\partial \boldsymbol{X}}\boldsymbol{X}.$$

Exploiting the property of normalization:

$$\frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_3^2}\frac{\partial \boldsymbol{X}_3^2}{\partial \boldsymbol{X}}\boldsymbol{X} = 0.$$



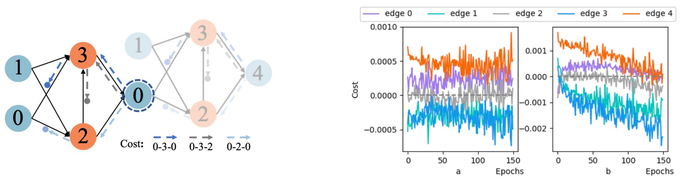
Total cost distributes to edges in the same cell as

$$\frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_{output}}\boldsymbol{X}_{output} = \sum_j \frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}^j_{output}}\left(\sum_{i<j}\boldsymbol{X}^i_j\right),$$

where $\boldsymbol{X}_{output}$ is the output node, $j \in \{intermediate\ nodes\}$, $i \in \{nodes: i < j\}$

e.g., in the minimal cell, we have $\frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_4}\boldsymbol{X}_4 = \frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_4^0}(\boldsymbol{X}_3^0 + \boldsymbol{X}_3^1 + \boldsymbol{X}_3^2) + \frac{\partial L_{\boldsymbol{\theta}}}{\partial \boldsymbol{X}_4^2}(\boldsymbol{X}_2^0 + \boldsymbol{X}_2^1).$

Corollary. *In cells except the last, for intermediate nodes that are connected the same output node, cost of edges pointing to them sums up to 0.*



Cost assignment in the last two cells. Cost of the left cell originates from node 0 of the right cell and should sum up to zero.

Cost Dynamics

Theorem 2. *Cost at output edges of the last cell has the form $C_{\boldsymbol{Z}} = L_{\boldsymbol{\theta}} - H_{\boldsymbol{\theta}}$. It is **negatively related** to classification accuracy. It tends to be positive at low accuracy, negative at high accuracy.*

- **Negatively related:** for the last cell's output edges, cost of one batch M with sampled architecture $\boldsymbol{Z}$ has an equivalent form:

$$C_{\boldsymbol{Z}} = \sum_{b,c,d}\frac{\partial L_{\boldsymbol{\theta}}}{\partial [\boldsymbol{X}_4]_{b,c,d}}[\boldsymbol{X}_4]_{b,c,d} = L_{\boldsymbol{\theta}} - H_{\boldsymbol{\theta}},$$

where $L_{\boldsymbol{\theta}}$ is the loss function, $H_{\boldsymbol{\theta}}$ is the entropy of network output.

- **Positive at low accuracy:** Exploiting normalization and weight initialization,

$$\mathbb{E}_{\boldsymbol{\theta}_0}[C_{\boldsymbol{Z}} > 0],$$

$$\text{since } \mathbb{E}_{y_1,y_2\sim\mathcal{N}(0,\cdot)}[y_1 \exp(y_1 + y_2)] > 0.$$

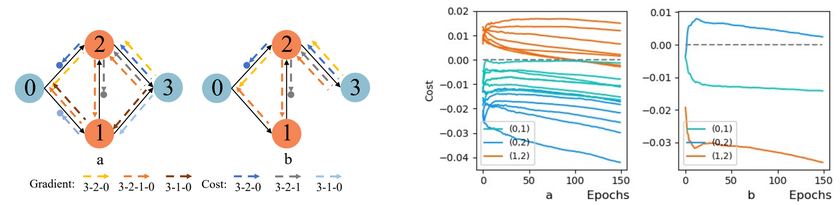
- **Negative at high accuracy:** With operation parameters updated to convergence, the probability of b-th image being classified to the correct label n_b increases towards 1. Since $Y_{bn_b} = \max\{Y_{bn}\}$, we have

$$C_{\boldsymbol{Z}} \propto \sum_n [-Y_{bn_b} + \frac{\exp(Y_{bn})}{\sum_q \exp(Y_{bq})}Y_{bn}] \leq \sum_n [-Y_{bn_b} + \frac{\exp(Y_{bn})}{\sum_q \exp(Y_{bq})}Y_{bn_b}] = 0.$$

Remark: *If the trends of cost at all edges are consistent with Thm.2, eventually the growing tendency (**P1**) occurs.*

Distinction of Intermediate Edges

By the cost assignment mechanism, all edges are born equally. Can not explain the distinction of intermediate edges. Conduct analysis on simplified minimal cell:



Cost at intermediate edge(1,2) is higher than cost of the same operation at edge(0, 2).

- $\theta_{0,1}$ and $\theta_{0,2}$ are always trained as None is not sampled.
- $\theta_{0,1}$ is updated with gradients from two paths (3-2-1-0) and (3-1-0).
  - *None* is sampled on edge(1, 2), $\theta_{0,1}$ can be updated.
  - *None* is sampled on edge (0, 1), $\theta_{1,2}$ cannot be updated.
- Even if *None* is not included in edge (0, 1), there are more model instances on path (3-2-1-0) than path (3-2-0) and (3-1-0).

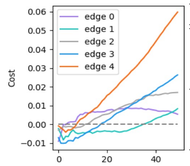
Design controlled experiment by deleting the output edge(1, 3) and fixing operation at edge(0, 1)

- path (3-2-1-0) is deeper than (3-2-0), but the cost on edge(0, 2) becomes positive
- the distinction of intermediate edges is likely due to **unequal** training

Remark: intermediate edge tends to have higher cost than input edges (if one with positive cost, it must be the intermediate edge.)

Effect of Bi-level Optimization

Cost minimization formulation satisfies to single-level and bi-level ones. **Exception**: every edge drops and almost none of them finally recovers in DARTS's bi-level version



	train	search
$C_{correct}$	-0.11	-0.11
$L_{correct}$	0.06	0.08
$H_{correct}$	0.17	0.19
C_{wrong}	0.80	**2.18**
L_{wrong}	1.70	**2.92**
H_{wrong}	0.90	**0.74**

scrutinize its classification loss $L_{\boldsymbol{\theta}}$ and output entropy $H_{\boldsymbol{\theta}}$

- For correct classification, $L_{\boldsymbol{\theta}}$ and $H_{\boldsymbol{\theta}}$ are comparable in the training and search set.

- But for data classified incorrectly, the classification loss $L_{\boldsymbol{\theta}}$ is much larger in the search set. That is, data in the search set are classified poorly.
- $H_{\boldsymbol{\theta}}$ in the search set is much lower than its counterpart in the training set.

Remark: explains catastrophic failure (**P3**) in bi-level DARTS

Conclusion

- The cost assignment for architecture α differs from that in gradient back-prop for parameters θ. Exaggerating discrepancy leads to distinction of intermediate edges.
- During training, cost decreases from **positive** and eventually turns **negative**, promoting the tendency of topology growth. If this cost-decreasing process is hindered, as in bi-level optimization, there will be a catastrophic failure.

Reference:

Liu et al. DARTS: Differentiable architecture search. ICLR 2019.
Xie et al. Snas: stochastic neural architecture search. ICLR 2019.
Hu* and Xie* et al. Dsnas: Direct neural architecture search without parameter retraining. CVPR 2020.