ISTA-NAS: Efficient and Consistent Neural Architecture Search by Sparse Coding

Presenter: Yibo Yang

Authors: Yibo Yang[1], Hongyang Li[1], Shan You[2], Fei Wang[2], Chen Qian[2], Zhouchen Lin[1]

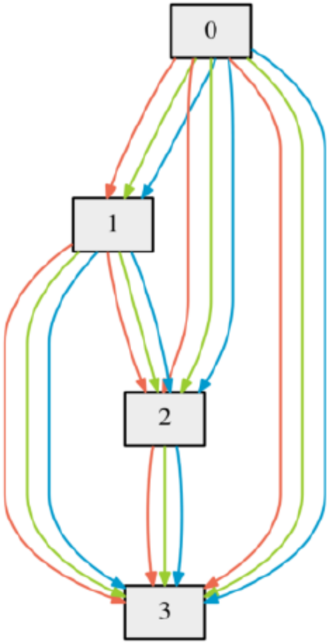
1: Peking University; 2: SenseTime

ISTA-NAS (NeurIPS 2020)

- Introduction
  - A DAG (directed acyclic graph):

$$x_j = \sum_{i=1}^{j-1} \sum_{k=1}^{K} z_k^{(i,j)} o_k(x_i) = \mathbf{z}_j^T \mathbf{o}_j$$

where $z_k^{(i,j)} \in \{0,1\}$ indicates whether the connection is active, o_k is the k-th operation from $\mathcal{O} = \{o_1, o_2, \dots, o_K\}$, $\mathbf{z}_j \in \{0,1\}^{(j-1)K}$, $\mathbf{o}_j \in \mathbb{R}^{(j-1)K}$ are vectors formed by $z_k^{(i,j)}$ and $o_k(x_i)$, respectively.



A DAG

  - Continuous relaxation:

$$z_k^{(i,j)} = \frac{\exp\left(\alpha_k^{(i,j)}\right)}{\sum_k \exp\left(\alpha_k^{(i,j)}\right)}$$

where $\alpha_k^{(i,j)}$ is the trainable variables

ISTA-NAS (NeurIPS 2020)

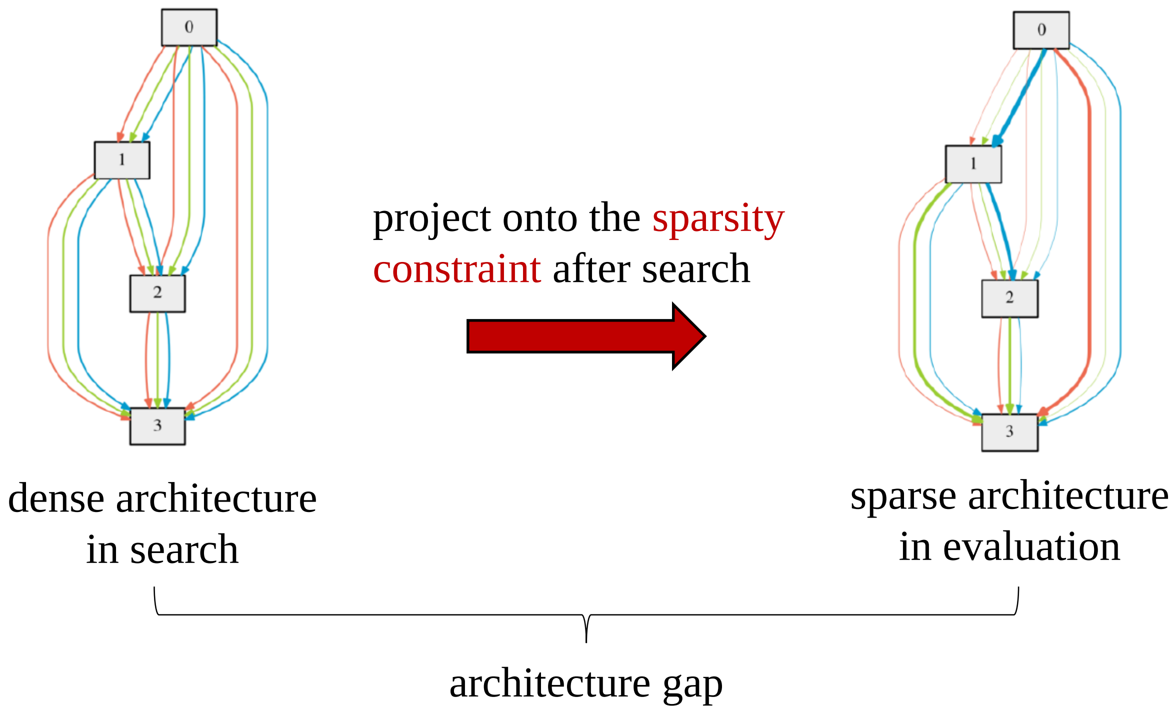
- Introduction
  - The objectives of current differentiable NAS (Liu et al., 2019):

$$Z^* = \underset{Z}{\operatorname{argmin}} \mathcal{L}_{val}\big(\mathcal{N}(W^*, Z)\big),$$
$$W^* = \underset{W}{\operatorname{argmin}} \mathcal{L}_{train}\big(\mathcal{N}(W, Z)\big)$$
$$\textbf{s.t.} \quad \left\|\mathbf{z}_j\right\|_0 = s_j, 1 < j \leq n,$$ (sparsity constraint, ignored during search)

where $Z = \{\mathbf{z}_j\}_{j=2}^{n}$, W is the weights of super-net $\mathcal{N}$, and s_j denotes the sparseness for node j.

ISTA-NAS (NeurIPS 2020)

- Introduction

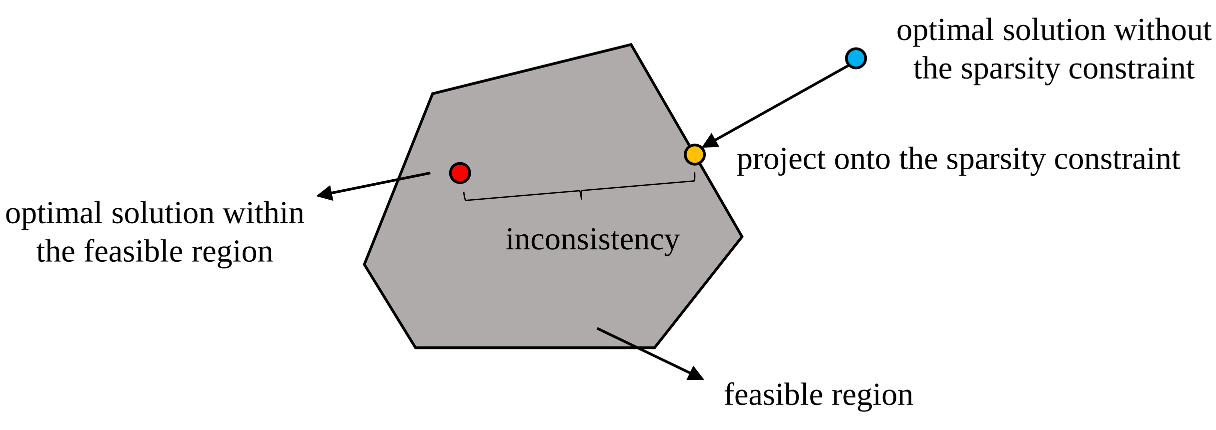


ISTA-NAS (NeurIPS 2020)

- Introduction
  - Problems:

    1) There is a poor correlation between the performances of the super-net in search and the target-net in evaluation.

    2) Besides, the dense super-net covering all candidate connections is inefficient to train due to its huge computational and memory cost.



ISTA-NAS (NeurIPS 2020)

- Introduction
  - Motivations:

1) Architecture variables have a sparse structure so can be well-represented by a compact space

2) We can perform the gradient-based search in an equivalent network defined on a compressed search space where each point corresponds to a sparse solution in the original high-dimensional space, and recover the architecture by sparse coding.

ISTA-NAS (NeurIPS 2020)

- Methods

  - An equivalent network defined on a compressed space $\Omega(\mathbf{b}_j) = \mathbb{R}^{m_j}$:

$$N(W,Z):\rightarrow\ x_j = \mathbf{z}_j^T \mathbf{o}_j = \mathbf{z}_j^T (\mathbf{A}_j^T \mathbf{A}_j - \mathbf{E}_j)\mathbf{o}_j = (\mathbf{A}_j \mathbf{z}_j)^T (\mathbf{A}_j \mathbf{z}_j) - \mathbf{z}_j^T \mathbf{E}_j \mathbf{o}_j$$

$$= \left(\mathbf{b}_j^T \mathbf{A}_j - [\mathbf{z}_j(\mathbf{b}_j)]^T \mathbf{E}_j\right) \mathbf{o}_j :\rightarrow N(W,B)$$

    where $\mathbf{A}_j$ is the measurement matrix, $\mathbf{E}_j$ is the residual matrix of $\mathbf{A}_j$ such that $\mathbf{A}_j^T \mathbf{A}_j - \mathbf{E}_j = \mathbf{I}$.

  - The optimal solution in $\Omega(\mathbf{z})$ can be searched by optimization in $\Omega(\mathbf{b})$:

Proposition 1. *Assume that* $\mathbf{A}$ *satisfies the RIP with its constant* δ_{2s} *and the exact* s*-sparse solution* $\mathbf{z}^*$ *can be recovered by* $\mathrm{argmin}_{\mathbf{z}} \frac{1}{2}\|\mathbf{A}\mathbf{z} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{z}\|_1$ *and satisfies* $\mathbf{A}\mathbf{z}^* = \mathbf{b}$. *Then we have that* $\mathbf{z}^*$ *is the optimal solution of the network* $\mathcal{N}(W, \mathbf{z})$ *if and only if* $\mathbf{b}^* = \mathbf{A}\mathbf{z}^*$ *is the optimal solution of the network* $\mathcal{N}(W, \mathbf{b})$.

ISTA-NAS (NeurIPS 2020)

- Methods
  - Formulate differentiable NAS as sparse coding:

$$\mathbf{z}_j = \underset{\mathbf{z}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{A}_j\mathbf{z} - \mathbf{b}_j\|_2^2 + \lambda\|\mathbf{z}\|_1, \quad 1 < j \leq n, \tag{9}$$

$$\begin{cases} B^* = \underset{B}{\operatorname{argmin}}\, \mathcal{L}_{val}(\mathcal{N}(W^*, B)), \\ W^* = \underset{W}{\operatorname{argmin}}\, \mathcal{L}_{train}(\mathcal{N}(W, B)), \end{cases} \tag{10}$$

where $B = \{\mathbf{b}_j\}_{j=2}^{n}$ is the trainable architecture variables in the network $N(W, B)$.

  - Sparsity:

$$x = \mathbf{z}^T\mathbf{o} = \mathbf{z}_{(\mathcal{S})}^T\mathbf{o}_{(\mathcal{S})} = \mathbf{z}_{(\mathcal{S})}^T\left(\mathbf{A}_{(\mathcal{S})}^T\mathbf{A}_{(\mathcal{S})} - \mathbf{E}_{(\mathcal{S},\mathcal{S})}\right)\mathbf{o}_{(\mathcal{S})} = \left(\mathbf{b}^T\mathbf{A}_{(\mathcal{S})} - \mathbf{z}_{(\mathcal{S})}^T\mathbf{E}_{(\mathcal{S},\mathcal{S})}\right)\mathbf{o}_{(\mathcal{S})}, \tag{11}$$

where $\mathbf{z}_{(\mathcal{S})}$ denote the elements of $\mathbf{z}$ indexed by $\mathcal{S}$, $\mathbf{A}_{(\mathcal{S})}$ denotes the columns of $\mathbf{A}$ indexed by $\mathcal{S}$, $\mathbf{E}_{(\mathcal{S})}$ denotes the rows and columns of $\mathbf{E}$ indexed by $\mathcal{S}$.

ISTA-NAS (NeurIPS 2020)

- Methods
  - Two-stage ISTA-NAS

Algorithm 1 Two-stage ISTA-NAS (for search only)

Input: Initialize the network weights W of the whole super-net $\mathcal{N}(W, B)$ and architecture variables $\mathrm{b}_j \in \mathbb{R}^{m_j}$ for each intermediate node $1 < j \leq n$. Sample $\mathbf{A}_j \in \mathbb{R}^{m_j \times (j-1)K}, \forall 1 < j \leq n$.

while *not converged* **do**

  Recover $\mathbf{z}$ by solving Eq. (9) with ISTA. Keep the top-s strongest magnitudes and set other dimensions as zeros. The support set $\mathcal{S}(\mathbf{z}) = \{i|\mathbf{z}(i) \neq 0\}$;

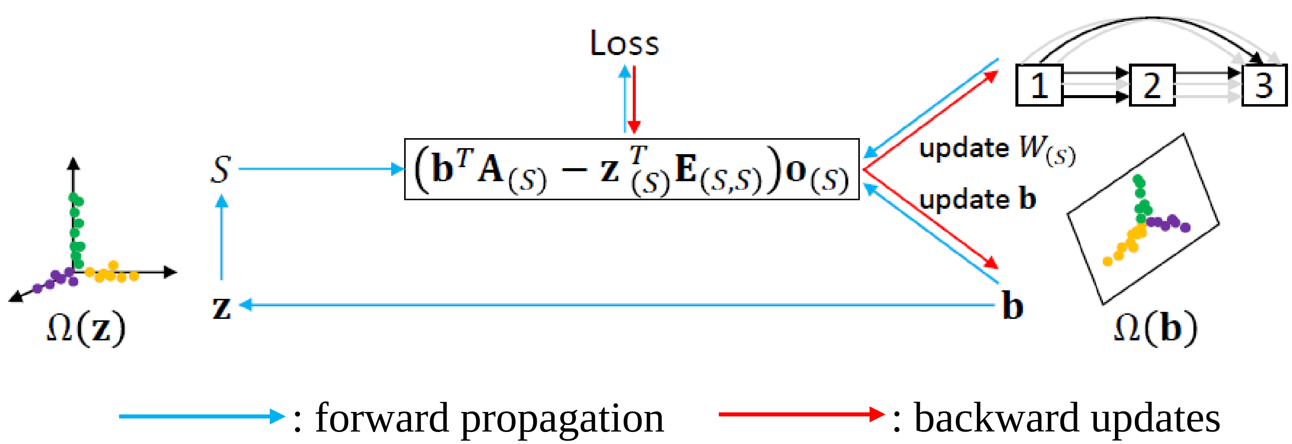
  Derive a sub-graph $\mathcal{N}(W_{(\mathcal{S})}, B)$ of the super-net by only propagating the dimensions in $\mathcal{S}$;

  Update network weights $W_{(\mathcal{S})}$ by descending $\nabla_{W_{(\mathcal{S})}} \mathcal{L}_{train}(\mathcal{N}(W_{(\mathcal{S})}, B))$;

  Update architecture variables b by descending $\nabla_{\mathrm{b}} \mathcal{L}_{val}(\mathcal{N}(W_{(\mathcal{S})}, B))$;

end while

Output: Produce a sparse architecture for evaluation according to the final $\mathcal{S}(\mathbf{z})$.



ISTA-NAS (NeurIPS 2020)

- Methods
  - One-stage ISTA-NAS

Algorithm 2 One-stage ISTA-NAS (for both search and evaluation)

Input: Initialize $\mathcal{N}(W, B)$ with depth, width, and batch size in the target-net setting. γ and β of BN layers in all candidate operations are frozen and initialized as 1 and 0. $search\_flag := True$.

while *not converged* **do**
    if $search\_flag$ **then**
        Perform the Line 2 and Line 3 of Algorithm 1; $\mathbf{z}^{new} := \mathbf{z}$;
    end if
    if $search\_flag$ **and** $\|\mathbf{z}^{new} - \mathbf{z}^{old}\| \leq \epsilon$ **then**
        $\gamma.requires\_grad := True$; $\beta.requires\_grad := True$; $search\_flag := False$;
    end if
    Update network weights $W_{(\mathcal{S})}$ by descending $\nabla_{W_{(\mathcal{S})}} \mathcal{L}_{train}(\mathcal{N}(W_{(\mathcal{S})}, B))$;
    if $search\_flag$ **then**
        Update architecture variables b by descending $\nabla_{\mathrm{b}} \mathcal{L}_{train}(\mathcal{N}(W_{(\mathcal{S})}, B))$; $\mathbf{z}^{old} := \mathbf{z}^{new}$;
    end if
end while
Update the parameters of BN layers by Eq. (12);

Output: Produce a sparse architecture and its optimized parameters.

$$\hat{\gamma} = \left(\mathrm{b}^T \mathbf{A}_{(\mathcal{S})} - \mathbf{z}_{(\mathcal{S})}^T \mathbf{E}_{(\mathcal{S},\mathcal{S})}\right) \circ \gamma; \quad \hat{\beta} = \left(\mathrm{b}^T \mathbf{A}_{(\mathcal{S})} - \mathbf{z}_{(\mathcal{S})}^T \mathbf{E}_{(\mathcal{S},\mathcal{S})}\right) \circ \beta; \tag{12}$$

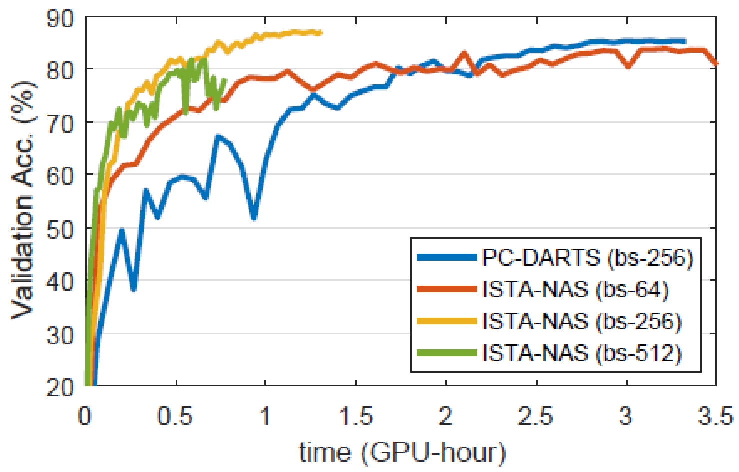
where γ and β are weight and bias of BN layers and are viewed as vectors in $\mathbb{R}^s$ formed by s active connections to the same node, $\circ$ is the element-wise multiplication, and $\hat{\gamma}, \hat{\beta}$ are updated parameters that keep the trained network accuracy unchanged.

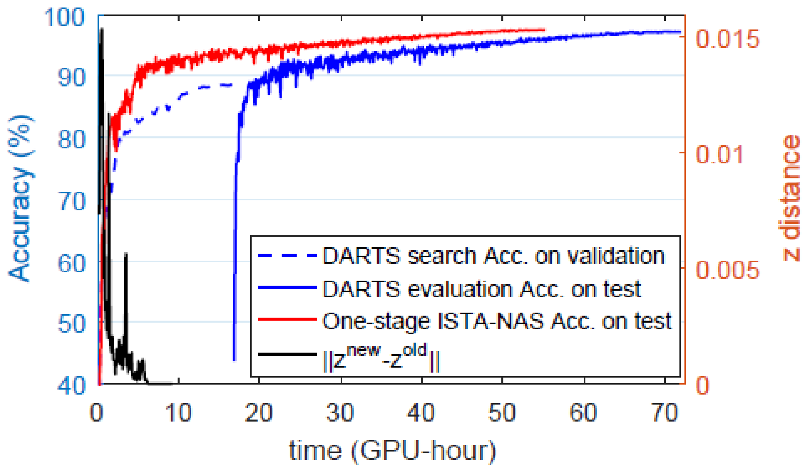
ISTA-NAS (NeurIPS 2020)

- Results
  - Improved efficiency and consistency

	Bs.	Mem.	Search Cost
DARTS (1st order)	64	9.1 G	0.70 day
PC-DARTS	256	11.6 G	0.14 day
ISTA-NAS	64	1.9 G	0.15 day
ISTA-NAS	256	5.5 G	0.05 day
ISTA-NAS	512	10.5 G	0.03 day

	Kendall τ
DARTS (1st order)	-0.36
PC-DARTS	-0.21
Two-stage ISTA-NAS	0.43
One-stage ISTA-NAS	0.57





ISTA-NAS (NeurIPS 2020)

- Results
  - On CIFAR-10

Methods	Test Error (%)	Params (M)	Cost (GPU-day) search	Cost (GPU-day) eval.	Search Method
DenseNet-BC [22]	3.46	25.6	-	-	manual
NASNet-A + cutout [61]	2.65	3.3	1800	3.2	RL
ENAS + cutout [39]	2.89	4.6	0.5	3.2	RL
AmoebaNet-B +cutout [40]	2.55±0.05	2.8	3150	-	evolution
NAONet-WS [31]	3.53	3.1	0.4	-	NAO
DARTS (2nd order) + cutout [30]	2.76±0.09	3.3	4.0	2.3	gradient
SNAS (moderate) + cutout [48]	2.85±0.02	2.8	1.5	2.2	gradient
P-DARTS+cutout [9]	2.50	3.4	0.3	2.9	gradient
NASP + cutout [53]	2.83±0.09	3.3	0.1	-	gradient
PC-DARTS + cutout [49]	2.57±0.07	3.6	0.1	3.1	gradient
Two-stage ISTA-NAS + cutout	2.54±0.05	3.32	**0.05**	2.0	gradient
One-stage ISTA-NAS + cutout	**2.36**±0.06	3.37	**2.3**		gradient

Table 3: Search results on CIFAR-10 and comparison with state-of-the-art methods. Cost is tested on a GTX 1080Ti GPU. The evaluation cost is calculated by us with their searched architectures in the same experimental settings. The cost of one-stage ISTA-NAS is the time spent in a single run.

  - On ImageNet

Methods	Test Err. (%) top-1	Test Err. (%) top-5	Params (M)	Flops (M)	Cost (GPU-day) search	Cost (GPU-day) eval.	Search Method
Inception-v1 [43]	30.2	10.1	6.6	1448	-	-	manual
MobileNet [20]	29.4	10.5	4.2	569	-	-	manual
ShuffleNet 2× (v2) [32]	25.1	-	∼5	591	-	-	manual
NASNet-A [61]	26.0	8.4	5.3	564	1800	-	RL
MnasNet-92 [44]	25.2	8.0	4.4	388	-	-	RL
AmoebaNet-C [40]	24.3	7.6	6.4	570	3150	-	evolution
DARTS (2nd order) [30]	26.7	8.7	4.7	574	4.0	3.6×8	gradient
SNAS [48]	27.3	9.2	4.3	522	1.5	3.3×8	gradient
P-DARTS [9]	24.4	7.4	4.9	557	0.3	3.6×8	gradient
ProxylessNAS (ImgNet) [5]	24.9	7.5	7.1	465	8.3	-	gradient
PC-DARTS (ImgNet) [49]	24.2	7.3	5.3	597	3.8	3.9×8	gradient
One-stage ISTA-NAS (C-10)	25.1	7.7	4.78	550	2.3	3.4×8	gradient
One-stage ISTA-NAS (ImgNet)	**24.0**	**7.1**	5.65	638	4.2×8		gradient

Table 4: Search results on ImageNet and comparison with state-of-the-art methods. Cost is tested on eight GTX 1080Ti GPUs. "ImgNet" denotes it is directly searched on ImageNet. Otherwise, it is searched on CIFAR-10 and then transfered to ImageNet for evaluation.

Thank You !

For any question, please contact ibo@pku.edu.cn

QR code for paper:

QR code for code: