FGA-NAS: Fast Resource-Constrained Architecture Search by Greedy-ADMM Algorithm

Yifei Chen[*†], Junge Zhang[*†], Qiaozhe Li[†], Hao Chen[*†], Kaiqi Huang[*†]

[*]*School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China*

[†]*CRISE, Institute of Automation, Chinese Academy of Sciences, Beijing, China*

chenyifei2019@ia.ac.cn,jgzhang@nlpr.ia.ac.cn,liqiaozhe2015@ia.ac.cn,
chenhao2019@ia.ac.cn,kqhuang@nlpr.ia.ac.cn

CONTENTS

CONTENTS

1. Motivation

University of Chinese Academy of Sciences

Bi-Level Optimization：

$$\boldsymbol{\alpha}^* = \underset{\alpha}{\operatorname{argmin}}\, \mathrm{L}_{\mathrm{val}}(\mathbf{w}^*(\boldsymbol{\alpha}), \boldsymbol{\alpha})$$

$$\text{s.t.} \quad \mathbf{w}^*(\boldsymbol{\alpha}) = \underset{\mathrm{w}}{\operatorname{argmin}}\, \mathrm{L}_{\mathrm{train}}(\mathbf{w}, \boldsymbol{\alpha})$$

Hessian matrix of $\boldsymbol{w}$：

$$\frac{\partial L_{val}(\boldsymbol{w}^*(\boldsymbol{\alpha}), \boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} = \frac{\partial L_{val}(\boldsymbol{w}^*, \boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} + \left(\frac{d\boldsymbol{w}^*(\boldsymbol{\alpha})}{d\boldsymbol{\alpha}}\right)^T \frac{\partial L_{val}(\boldsymbol{w}^*, \boldsymbol{\alpha})}{\partial \boldsymbol{w}}$$

$$\frac{d\boldsymbol{w}^*(\boldsymbol{\alpha})}{d\boldsymbol{\alpha}} = \left(\frac{\partial^2 L_{val}(\boldsymbol{w}^*, \boldsymbol{\alpha})}{\partial \boldsymbol{w}^2}\right)^{-1} \frac{\partial^2 L_{val}(\boldsymbol{w}^*, \boldsymbol{\alpha})}{\partial \boldsymbol{w} \partial \boldsymbol{\alpha}}$$

1. Motivation

Bi-Level Optimization：

$$\boldsymbol{\alpha}^* = \underset{\alpha}{\operatorname{argmin}}\, \mathrm{L}_{\mathrm{val}}(\mathbf{w}^*(\boldsymbol{\alpha}), \boldsymbol{\alpha})$$

$$\mathrm{s.t.}\quad \mathbf{w}^*(\boldsymbol{\alpha}) = \underset{\mathrm{w}}{\operatorname{argmin}}\, \mathrm{L}_{\mathrm{train}}(\mathbf{w}, \boldsymbol{\alpha})$$

Heuristic algorithm:

$$\alpha_{\mathrm{t+1}} = \alpha_{\mathrm{t}} - \eta_\alpha \nabla_\alpha \mathrm{L}_{val}(w_t - \xi \nabla_w L_{train}(w_t, \alpha_{\mathrm{t}}), \alpha_{\mathrm{t}})$$

Performance Collapse

$$w_{\mathrm{t+1}} = w_{\mathrm{t}} - \eta_w \nabla_w \mathrm{L}_{train}(w_t, \alpha_{\mathrm{t+1}})$$

2. Methodology

Single-Level Optimization：

$$(\boldsymbol{w}^*, \boldsymbol{\alpha}^*) = \underset{(\boldsymbol{w},\boldsymbol{\alpha})}{\operatorname{argmin}} L_{train}(\boldsymbol{w}, \boldsymbol{\alpha})$$

$$s.t.\ \mathrm{r}_c \geq \sum_i \mathrm{h}_{c,i} \cdot \|\alpha_i\|_0, \qquad 1 \leq c \leq n_c$$

By introducing $\boldsymbol{\beta}_\mathrm{c}$ and $f_c(\boldsymbol{\beta}_\mathrm{c})$ ：

$$(\boldsymbol{w}^*, \boldsymbol{\alpha}^*) = \underset{(\boldsymbol{w},\boldsymbol{\alpha})}{\operatorname{argmin}} L_{train}(\boldsymbol{w}, \boldsymbol{\alpha}) + \sum_{c=1}^{n_c} f_c(\boldsymbol{\beta}_c)$$

$$s.t.\quad \boldsymbol{\alpha} = \boldsymbol{\beta}_c \qquad 1 \leq c \leq n_c$$

$$S_c = \{\boldsymbol{\alpha} | r_c \geq \textstyle\sum_i h_{c,i} \cdot \|\alpha_i\|_0\}, \qquad f_c(\boldsymbol{\beta}_c) = \begin{cases} 0\ , & \boldsymbol{\beta}_c \in S_c \\ +\infty, & otherwise \end{cases}$$

2. Methodology

Single-Level Optimization：

$$(\boldsymbol{w}^*, \boldsymbol{\alpha}^*) = \underset{(\boldsymbol{w},\boldsymbol{\alpha})}{\operatorname{argmin}} L_{train}(\boldsymbol{w}, \boldsymbol{\alpha}) + \sum_{c=1}^{n_c} f_c(\boldsymbol{\beta_c})$$

$$s.t. \quad \boldsymbol{\alpha} = \boldsymbol{\beta_c} \qquad 1 \leq c \leq n_c$$

Augmented Lagrangian $F_\rho(\omega, \beta, \alpha, m)$ ：

$$F_\rho(\omega, \beta, \alpha, m) = L_{train}(\omega, \alpha) - \sum_{c=1}^{n_c} \frac{\|m_c\|_F^2}{2\rho}$$

$$+ \sum_{c=1}^{n_c} \left[f_c(\beta_c) + \frac{\rho}{2} \left\| \alpha - \beta_c + \frac{m_c}{\rho} \right\|_F^2 \right]$$

2. Methodology

Augmented Lagrangian $F_\rho(\boldsymbol{\omega},\boldsymbol{\beta},\boldsymbol{\alpha},\boldsymbol{m})$ ：

$$F_\rho(\boldsymbol{\omega},\boldsymbol{\beta},\boldsymbol{\alpha},\boldsymbol{m}) = L_{train}(\boldsymbol{\omega},\boldsymbol{\alpha}) - \sum_{c=1}^{n_c}\frac{\|\boldsymbol{m_c}\|_F^2}{2\rho} + \sum_{c=1}^{n_c}\left[f_c(\boldsymbol{\beta_c}) + \frac{\rho}{2}\left\|\boldsymbol{\alpha} - \boldsymbol{\beta_c} + \frac{\boldsymbol{m_c}}{\rho}\right\|_F^2\right]$$

By introducing ADMM algorithm：

$$\begin{cases} \beta_c^{t+1} = \underset{\beta}{\operatorname{argmin}}\, F_\rho(w^t,\alpha^t,\beta,m^t) \\ (\mathrm{w}^{t+1},\alpha^{t+1}) = \underset{w,\alpha}{\operatorname{argmin}}\, F_\rho(w^t,\alpha^t,\beta,m^t) \\ m_c^{t+1} = m_c^t + \rho(\alpha^{t+1} - \beta_c^{t+1}) \end{cases}$$

2. Methodology

Augmented Lagrangian $F_\rho(\boldsymbol{\omega}, \boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{m})$:

$$F_\rho(\boldsymbol{w}, \boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{m}) = L_{train}(\boldsymbol{w}, \boldsymbol{\alpha}) - \sum_{c=1}^{n_c} \frac{\|\boldsymbol{m_c}\|_F^2}{2\rho}$$

$$+ \sum_{c=1}^{n_c} \left[f_c(\boldsymbol{\beta_c}) + \frac{\rho}{2} \left\| \boldsymbol{\alpha} - \boldsymbol{\beta_c} + \frac{\boldsymbol{m_c}}{\rho} \right\|_F^2 \right]$$

Sub problem 1 $\beta_c^{t+1} = \underset{\beta}{\operatorname{argmin}}\, F_\rho(\omega^t, \alpha^t, \beta, m^t)$ **->** 0-1 **programming :**

$$\boldsymbol{\beta_c^{t+1}} = \boldsymbol{\theta_c^{t+1}} \odot \left(\boldsymbol{\alpha^t} + \frac{\boldsymbol{m_c^t}}{\rho} \right)$$

$$\boldsymbol{\theta_c^{t+1}} = \underset{\theta_c}{\operatorname{argmin}} \sum_i \theta_{c,i} \cdot \left(\alpha_i^t + \frac{m_{c,i}^t}{\rho} \right)^2$$

$$s.t. \quad \theta_{c,i} \in \{0,1\}, \qquad r_c \geq \sum_i h_{c,i} \cdot \theta_{c,i}, \qquad 1 \leq c \leq n_c$$

2. Methodology

Augmented Lagrangian $F_\rho(\boldsymbol{\omega}, \boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{m})$ ：

$$F_\rho(\boldsymbol{w}, \boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{m}) = L_{train}(\boldsymbol{w}, \boldsymbol{\alpha}) - \sum_{c=1}^{n_c} \frac{\|\boldsymbol{m_c}\|_F^2}{2\rho}$$

$$+ \sum_{c=1}^{n_c} \left[f_c(\boldsymbol{\beta_c}) + \frac{\rho}{2} \left\| \boldsymbol{\alpha} - \boldsymbol{\beta_c} + \frac{\boldsymbol{m_c}}{\rho} \right\|_F^2 \right]$$

Sub problem 2 $(\mathrm{w}^{t+1}, \alpha^{t+1}) = \underset{w,\alpha}{\operatorname{argmin}} F_\rho(w^t, \alpha^t, \beta, m^t)$ ：

$$(\mathrm{w}^{t+1}, \alpha^{t+1}) = \underset{w,\alpha}{\operatorname{argmin}} \left(L_{train}(\boldsymbol{w}, \boldsymbol{\alpha}) + \sum_{c=1}^{n_c} \frac{\rho}{2} \left\| \boldsymbol{\alpha} - \boldsymbol{\beta_c} + \frac{\boldsymbol{m_c}}{\rho} \right\|_F^2 \right)$$

2. Methodology

中国科学院大学
University of Chinese Academy of Sciences

Greedy-ADMM Search Algorithm

$$
\begin{cases}
\beta_c^{t+1} = \underset{\beta}{\operatorname{argmin}}\, F_\rho(w^t, \alpha^t, \beta, m^t) \\
(\mathrm{w}^{t+1}, \alpha^{t+1}) = \underset{w,\alpha}{\operatorname{argmin}}\, F_\rho(w^t, \alpha^t, \beta, m^t) \\
m_c^{t+1} = m_c^t + \rho(\alpha^{t+1} - \beta_c^{t+1})
\end{cases}
$$

$$
\begin{cases}
\beta_c^{t+1} = \underset{\beta}{argmin}\, F_\rho(w^t, \alpha^t, \beta, m^t) \\
\begin{cases}
w^{t+1} = w^t - \eta_w \nabla_w F_\rho(w^t, \alpha^t, \beta, m^t) \\
\alpha^{t+1} = \alpha^t - \eta_\alpha \nabla_\alpha F_\rho(w^t, \alpha^t, \beta, m^t)
\end{cases} \\
m_c^{t+1} = m_c^t + \rho(\alpha^{t+1} - \beta_c^{t+1})
\end{cases}
$$

3 Experiments

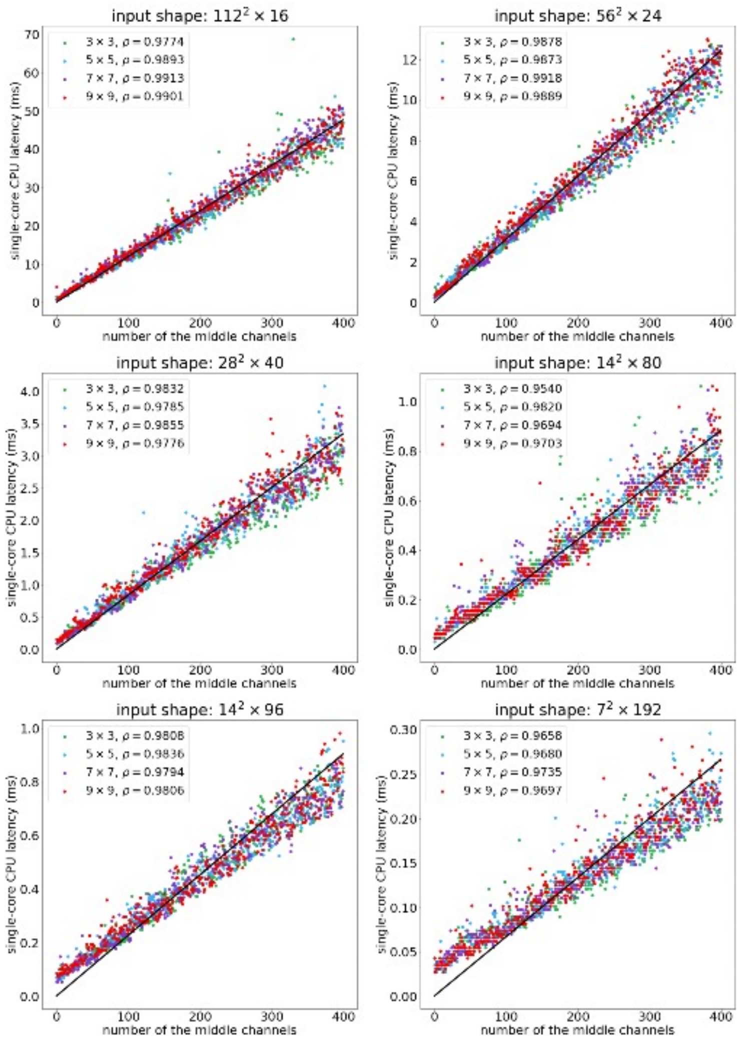

Fig. 5. **The approximation of single-core CPU latencies of inverted bottlenecks with different input shapes, different kernel sizes and different numbers of middle channels.** Each point is an average of 64 measurements.

(1) let the number of middle channels in the associated inverted bottleneck be one
(2) estimate the number of parameters and the FLOPs of the above-modified inverted bottleneck
(3) let the FLOPs and the number of parameters equal to the FLOPs and the number of parameters of the above modified block, respectively

3 Experiments

University of Chinese Academy of Sciences

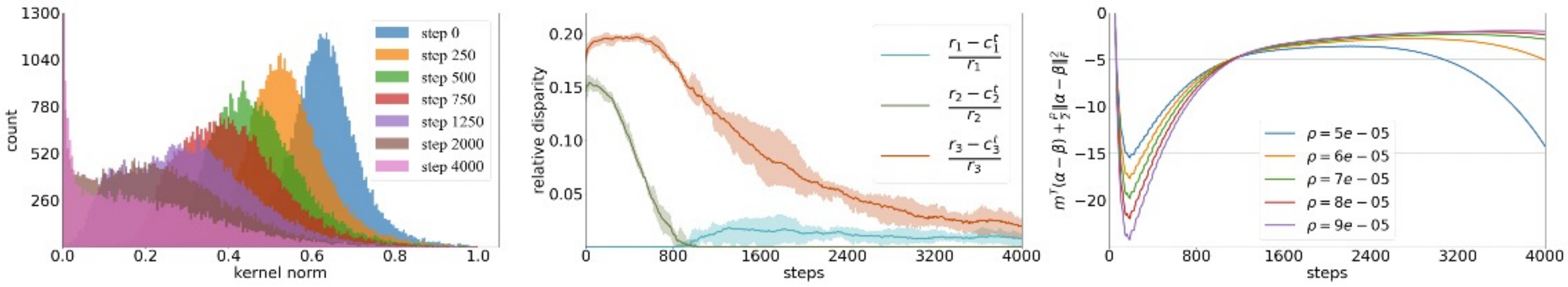

Fig. 4. **Left:** The changes in distribution of architecture parameters $\{\alpha_{i,j}\}$ during the search. **Mid:** The change of relative disparities during the search. c_1^t, c_2^t and c_3^t respectively denote the FLOPs, the number of parameters and the single-core CPU latency of the goal architecture at time step t. r_1, r_2 and r_3 respectively denote the preset maximum FLOPs, the preset maximum the number of parameters and the preset maximum single-core CPU latency. **Right:** The convergence behavior of the search algorithm under different penalties.

3 Experiments

TABLE III
COMPARISONS WITH OTHER METHODS ON IMAGENET UNDER THE MOBILE SETTING.

Model	Search			FLOPs (M)	Params (M)	Top-1 Acc (%)
	Method	Space	Cost (GPU days)			
SinglePath [15]	gradient	layer-wise	3.75	334	4.4	74.9
MobileNeXt-1.0 [32]	manual	-	-	300	3.4	74.0
MnasNet-A1 [5]	RL	stage-wise	≥ 379	312	3.9	75.2
HourNAS-E [13]	gradient	layer-wise	0.1	313	3.8	75.7
AtomNAS-B+ [23]	gradient	channel-wise	34	329	5.5	77.2
FBNetV2-L1 [9]	gradient	layer-wise	25	325	-	77.2
FGA-NAS-A (ours)	**gradient**	**channel-wise**	**0.2**	**320**	**5.7**	**77.0**
MnasNet-A2 [5]	RL	stage-wise	≥ 379	340	4.8	75.6
MobileNeXt-1.1 [32]	manual	-	-	420	4.28	76.7
HourNAS-F [13]	gradient	layer-wise	0.1	383	5.0	77.0
EfficientNet-B0 [31]	RL	stage-wise	≥ 379	390	5.3	77.3
AtomNas-C+ [23]	gradient	channel-wise	34	363	5.9	77.6
FGA-NAS-B (ours)	**gradient**	**channel-wise**	**0.2**	**358**	**6.0**	**77.4**

4 Conclusion

FGA-NAS, an efficient method for resource-constrained NAS

- A novel search space
- A novel greedy-ADMM algorithm

中国科学院大学
University of Chinese Academy of Sciences

CASIA

Thank you.