Resource Constrained Neural Network Architecture Search: Will a Submodularity Assumption Help?

Yunyang Xiong Ronak Mehta Vikas Singh
University of Wisconsin Madison
yxiong43@wisc.edu ronakrm@cs.wisc.edu vsingh@biostat.wisc.edu

In this supplement, we first show the proofs for Lemma 1 and Theorem 1 in the main paper. Then we present details for models used in experiments and provide more experimental results.

Contents

1. Technical Proofs **1**
1.1. Proof of Lemma 1 1
1.2. Proof of Theorem 1 2

2. Experimental Results **2**
2.1. Searching and Adapting Details 2
2.2. The RCNet Architecture 4
2.3. Adapting to ImageNet 7

1. Technical Proofs

1.1. Proof of Lemma 1

Lemma 1. *For any selected blocks $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$ and blocks $v \in \mathcal{V} \setminus \mathcal{B}$, it holds that*

$$F(\mathcal{A}) \geq 0 \tag{1}$$
$$F(\mathcal{A} \cup \{v\}) - F(\mathcal{A}) \geq 0 \tag{2}$$
$$F(\mathcal{A} \cup \{v\}) - F(\mathcal{A}) \geq F(\mathcal{B} \cup \{v\}) - F(\mathcal{B}) \tag{3}$$

where F reaches its convex hull point with respect to the cost.

Proof. Denote each cost-accuracy pair at global optimality as $(c_i, f_i), i = 1, \ldots, L^N$ and add three virtual points, $(0,0)$, $(c_{L^N}, 0)$, $(c_{L^N}, f^\star)$, where $c_{L^N} = \max_{\mathcal{S} \subseteq \mathcal{V}} c(\mathcal{S})$, $f^\star = \max\{f_1, \ldots, f_{L^N}\}$. This points set can be seen as a *convex hull*, where for each cost we assign its associated positive value (accuracy) on the convex hull. If $F(\mathcal{S})$ can always reach its convex hull point with respect to $c(\mathcal{S})$, denote $\tilde{f}(c(\mathcal{S})) = F(\mathcal{S})$, then $\tilde{f}$ is concave with respect to cost $c_{\mathcal{S}}$. From the convex hull, we know points, $(0,0)$, $(c_{L^N}, 0)$, $(c_{L^N}, f^\star)$ must be included.

To show $F(\mathcal{A}) \geq 0, \forall \mathcal{A} \subseteq \mathcal{V}$, assume $\exists \mathcal{A} \subseteq \mathcal{V}$, $F(\mathcal{A}) < 0$, then the convex hull contains $(0,0)$, $(c(\mathcal{A}), \tilde{f}(c(\mathcal{A})))$, $(c_{L^N}, 0)$, $(c_{L^N}, f^\star)$, $\tilde{f}(\emptyset) = 0$, $\tilde{f}(c(\mathcal{A})) < 0$, $\tilde{f}(c_{L^N}) \geq 0$. Then there are points in the triangle region by convex combination of three points, $(0,0), (c(\mathcal{A}), \tilde{f}(c(\mathcal{A})), (c_{L^N}, f^\star)$, *not* in the convex hull. This is contradictory to the convex hull definition[1]: the set of all convex combinations of points is in the convex hull. Therefore, $F(\mathcal{A}) \geq 0, \forall \mathcal{A} \subseteq \mathcal{V}$ holds.

To show $F(\mathcal{A} \cup \{v\}) - F(\mathcal{A}) \geq 0$, assume the contradiction: $\exists \mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$, $v \in \mathcal{V} \setminus \mathcal{B}$, $F(\mathcal{A} \cup \{v\}) < F(\mathcal{A})$. The triangular region by the convex combination of three points, $(c(\mathcal{A}), \tilde{f}(c(\mathcal{A}))), (c(\mathcal{A} \cup \{v\}), \tilde{f}(c(\mathcal{A} \cup v))), (c_{L^N}, f^\star)$, is not in the convex hull, which is contradictory to convex hull definition. Therefore, $F(\mathcal{A} \cup \{v\}) - F(\mathcal{A}) \geq 0$ holds.

By concavity, we have $\forall a < b, s > 0, \tilde{f}(a+s) - \tilde{f}(s) \geq \tilde{f}(b+s) - \tilde{f}(b)$. In order to show $F(\mathcal{A} \cup \{v\}) - F(\mathcal{A}) \geq F(\mathcal{B} \cup \{v\}) - F(\mathcal{B})$, we just need to show $\tilde{f}(c(\mathcal{A} \cup \{v\})) - \tilde{f}(c(\mathcal{A})) \geq \tilde{f}(c(\mathcal{B} \cup \{v\})) - \tilde{f}(c(\mathcal{B}))$. By setting $a = c(\mathcal{A}), s =$

$c(v), b = c(\mathcal{B})$,

$$\tilde{f}(c(\mathcal{A} \cup \{v\})) - \tilde{f}(c(\mathcal{A})) = \tilde{f}(c(\mathcal{A}) + c(v)) - \tilde{f}(c(\mathcal{A})) \tag{4}$$

$$\tilde{f}(c(\mathcal{B} \cup \{v\})) - \tilde{f}(c(\mathcal{B})) = \tilde{f}(c(\mathcal{B}) + c(v)) - \tilde{f}(c(\mathcal{B})), \tag{5}$$

and with $\mathcal{A} \subseteq \mathcal{B}$, $a < b$, and $c(v) \geq 0$ shows $c \geq 0$. Again with concavity, we have $\tilde{f}(c(\mathcal{A}) + c(v)) - \tilde{f}(c(\mathcal{A})) \geq \tilde{f}(c(\mathcal{B}) + c(v)) - \tilde{f}(c(\mathcal{B}))$. Therefore, $F(\mathcal{A} \cup \{v\}) - F(\mathcal{A}) \geq F(\mathcal{B} \cup \{v\}) - F(\mathcal{B})$ holds.

□

1.2. Proof of Theorem 1

Theorem 1. *If F is a nondecreasing set function satisfying diminishing return property and $F(\emptyset) = 0$, then the CEG algorithm achieves a constant ratio $\frac{1}{2}(1 - \frac{1}{e})$ of the optima:*

$$\begin{aligned} &\max\{F(\tilde{\mathcal{S}}_{UC}), F(\tilde{\mathcal{S}}_{APR}), F(\tilde{\mathcal{S}}_{AMR})\} \\ &\geq \frac{1}{2}(1 - \frac{1}{e}) \max_{MAdds(\mathcal{S}) \leq B_m, Param(\mathcal{S}) \leq B_p} F(\mathcal{S}) \end{aligned} \tag{6}$$

Proof. [2] state that the better solution of the unit-cost greedy and benefit-cost greedy algorithms solving the Budgeted MAX-COVER problem is at least a constant factor $\frac{1}{2}(1 - \frac{1}{e})$ of the optimal solution, proved for the special case of the Budgeted MAX-COVER problem. [3] prove this result for arbitrary nondecreasing submodular functions. By applying this result to our architecture searching case, we have

$$\max\{F(\tilde{\mathcal{S}}_{UC}), F(\tilde{\mathcal{S}}_{APR})\} \geq \frac{1}{2}(1 - \frac{1}{e}) \max_{MAdds(\mathcal{S}) \leq B_m, Param(\mathcal{S}) \leq B_p} F(\mathcal{S}) \tag{7}$$

$$\max\{F(\tilde{\mathcal{S}}_{UC}), F(\tilde{\mathcal{S}}_{AMR})\} \geq \frac{1}{2}(1 - \frac{1}{e}) \max_{MAdds(\mathcal{S}) \leq B_m, Param(\mathcal{S}) \leq B_p} F(\mathcal{S}) \tag{8}$$

Eq. 7 states that the better solution of the unit cost greedy and accuracy parameter ratio greedy can achieve a constant factor $\frac{1}{2}(1 - \frac{1}{e})$ of the optimal solution under the MAdds and parameter budget constraint. Similarly, Eq. 8 states that the better solution of the unit cost greedy and accuracy MAdds ratio greedy can achieve a constant factor $\frac{1}{2}(1 - \frac{1}{e})$ of the optimal solution under the MAdds and parameter budget constraint. Apply maximization to both side of Eq.7 and Eq.8, we have,

$$\begin{aligned} &\max\{F(\tilde{\mathcal{S}}_{UC}), F(\tilde{\mathcal{S}}_{APR}), F(\tilde{\mathcal{S}}_{AMR})\} \\ &\geq \frac{1}{2}(1 - \frac{1}{e}) \max_{MAdds(\mathcal{S}) \leq B_m, Param(\mathcal{S}) \leq B_p} F(\mathcal{S}) \end{aligned} \tag{9}$$

□

2. Experimental Results

In this section, we describe our architecture solution paths and show our final network architectures in detail.

2.1. Searching and Adapting Details

As our main purpose is to look for low cost mobile neural networks and depth-wise based blocks have been shown to have high computational efficiency, we derive basic blocks from MobileNetV2[5] blocks by using different expansion ratios and group convolutions instead of searching block skeletons from scratch. Methods searching from scratch [4, 6] typically find models that are memory intensive and computation intensive in practice. Mentioned in the main paper, We have $L = 6$ different basic blocks to pick from and $N = 36$ number of positions that can be filled for building networks under parameter and MAdds constraints for our low cost architecture search. A special case of inserting all positions with basic block of type 5 on ImageNet can be seen in Table 1. For architecture search, we start from an initial small basic network (seen in Table 2), giving us the ability to evaluate accuracy on the validation set. From the small basic network, we are searching network architectures under parameter and MAdds constraints by RCAS. The difference between architectures on ImageNet and CIFAR-100 is the input image size, stride of conv0 and RC2, and the last fully connected layer's output channels c_{final}. By changing the input image size to 32×32, the stride of conv0 and RC2 to 1, and the output channels of the last fully connected layer to 100, we can construct the network architecture on CIFAR-100 given the network architecture on ImageNet. Similarly, by changing the input image size to 224×224, the stride of conv0 and RC2 to 2, and the output channels of the last fully connected layer to 1000, we can construct the network architecture on ImageNet adapted from network architecture on CIFAR-100.

Name	Input	Operator	Expansion (t)	Expansion Group (g_e)	Stride (s)	Projection Group (g_p)	Output Channels	Repetition
conv0	$224 \times 224 \times 3$	conv2d	-	-	2	-	32	-
MB1	$112 \times 112 \times 32$	MB	1	1	1	1	16	1
RC2	$112 \times 112 \times 16$	Type 5	6	1	2	1	24	1
RC2(1-5)	$56 \times 56 \times 24$	Type 5	6	1	1	1	24	5
RC3	$56 \times 56 \times 24$	Type 5	6	1	2	1	32	1
RC3(1-5)	$28 \times 28 \times 32$	Type 5	6	1	1	1	32	5
RC4	$28 \times 28 \times 32$	Type 5	6	1	2	1	64	1
RC4(1-5)	$14 \times 14 \times 64$	Type 5	6	1	1	1	64	5
RC5	$14 \times 14 \times 64$	Type 5	6	1	1	1	96	1
RC5(1-5)	$14 \times 14 \times 96$	Type 5	6	1	1	1	96	5
RC6	$14 \times 14 \times 96$	Type 5	6	1	2	1	160	1
RC6(1-5)	$7 \times 7 \times 160$	Type 5	6	1	1	1	160	5
RC7	$7 \times 7 \times 160$	Type 5	6	1	1	1	320	1
RC7(1-5)	$7 \times 7 \times 320$	Type 5	6	1	1	1	320	5
conv8	$7 \times 7 \times 320$	conv2d 1x1	-	-	-	1	1280	-
pool9	$7 \times 7 \times 1280$	avgpool 7x7	-	-	-	1	1280	-
fc10	$1 \times 1 \times 1280$	FC	-	-	-	-	1000	-

Table 1: The architecture of a special case of filling all positions with basic block of type 5 on ImageNet. Each line gives a sequence of 1 or more identical (modulo stride) layers with repetition times. All layers in the same module or sequence have the same number of output channels. When the dimension of the input to the block is not the same as the output, i.e., RC2, we simply skip the residual connection as in MobileNetV2. MB block refers to the first MobileNetV2 block in MobileNetV2 with expansion $t = 1$, expansion group $g_e = 1$, projection group $g_p = 1$ and stride $s = 1$. During architecture search, we set conv0, MB1, conv8, pool9 and fc10 as basic layers for building convolutional neural networks and only search basic blocks from RC2-RC7(1-5), which will be picked to fill $N = 36$ positions for building networks under parameter and MAdds constraints.

Name	Input	Operator	Expansion (t)	Expansion Group (g_e)	Stride (s)	Projection Group (g_p)	Output Channels (C_{out})	Repetition (n_i)
conv0	$224 \times 224 \times 3$	conv2d	-	-	2	-	32	-
MB1	$112 \times 112 \times 32$	MB	1	1	1	1	16	1
RC2	$112 \times 112 \times 16$	Type 5	6	1	2	1	24	1
RC3	$56 \times 56 \times 24$	Type 5	6	1	2	1	32	1
RC4	$28 \times 28 \times 32$	Type 5	6	1	2	1	64	1
RC6	$14 \times 14 \times 96$	Type 5	6	1	2	1	160	1
conv8	$7 \times 7 \times 96$	conv2d 1x1	-	-	-	1	1280	-
pool9	$7 \times 7 \times 1280$	avgpool 7x7	-	-	-	1	1280	-
fc10	$1 \times 1 \times 1280$	FC	-	-	-	-	c_{final}	-

Table 2: The architecture of an initial starting network on ImageNet by filling 4 positions with type 5 and adding basic layers, conv0, MB1, conv8, pool9, fc10. Starting from this network, we aim to fill remaining 32 positions with 6 types of basic blocks under parameter and MAdds constraints.

2.2. The RCNet Architecture

On CIFAR-100, the architecture search path by RCAS is

$$[5, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$
$$[5, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$
$$[5, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 6, 0, 0, 0, 0, 0, 0, 5, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$
$$[5, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 5, 5, 0, 0, 0, 0, 6, 0, 0, 0, 0, 0, 0, 5, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$
$$[5, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 5, 5, 0, 0, 0, 0, 6, 0, 0, 6, 0, 0, 0, 5, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$
$$[5, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 5, 5, 0, 0, 0, 0, 6, 0, 0, 6, 0, 4, 0, 5, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$
$$[5, 0, 0, 0, 0, 0, 5, 2, 0, 0, 0, 5, 5, 0, 0, 0, 0, 6, 0, 0, 6, 0, 4, 0, 5, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$
$$[5, 3, 0, 0, 0, 0, 5, 2, 0, 0, 0, 5, 5, 0, 0, 0, 0, 6, 0, 0, 6, 0, 4, 0, 5, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$
$$[5, 3, 0, 0, 0, 0, 5, 2, 0, 3, 0, 5, 5, 0, 0, 0, 0, 6, 0, 0, 6, 0, 4, 0, 5, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$
$$[5, 3, 0, 0, 0, 0, 5, 2, 0, 3, 0, 5, 5, 0, 3, 0, 0, 6, 0, 0, 6, 0, 4, 0, 5, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$
$$[5, 3, 0, 0, 0, 0, 5, 2, 0, 3, 0, 5, 5, 0, 3, 0, 0, 6, 0, 0, 6, 0, 4, 0, 5, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$
$$[5, 3, 0, 0, 0, 0, 5, 2, 0, 3, 0, 5, 5, 0, 3, 0, 0, 6, 0, 0, 6, 0, 4, 6, 5, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$
$$[5, 3, 0, 0, 0, 0, 5, 2, 0, 3, 0, 5, 5, 0, 3, 0, 0, 6, 0, 0, 6, 0, 4, 6, 5, 4, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0]$$
$$[5, 3, 0, 0, 0, 0, 5, 2, 0, 3, 0, 5, 5, 0, 3, 3, 0, 6, 0, 0, 6, 0, 4, 6, 5, 4, 6, 0, 4, 0, 0, 0, 0, 0, 0, 0]$$
$$[5, 3, 0, 0, 0, 0, 5, 2, 0, 3, 3, 5, 5, 0, 3, 3, 0, 6, 0, 0, 6, 0, 4, 6, 5, 4, 6, 0, 4, 0, 0, 0, 0, 0, 0, 0]$$
$$[5, 3, 0, 0, 0, 0, 5, 2, 0, 3, 3, 5, 5, 0, 3, 3, 0, 6, 0, 0, 6, 0, 4, 6, 5, 4, 6, 0, 4, 1, 0, 0, 0, 0, 0, 0]$$
$$[5, 3, 0, 1, 0, 0, 5, 2, 0, 3, 3, 5, 5, 0, 3, 3, 0, 6, 0, 0, 6, 0, 4, 6, 5, 4, 6, 0, 4, 1, 0, 0, 0, 0, 0, 0]$$
$$[5, 3, 0, 1, 1, 0, 5, 2, 0, 3, 3, 5, 5, 0, 3, 3, 0, 6, 0, 0, 6, 0, 4, 6, 5, 4, 6, 0, 4, 1, 0, 0, 0, 0, 0, 0]$$
$$[5, 3, 0, 1, 1, 0, 5, 2, 0, 3, 3, 5, 5, 0, 3, 3, 0, 6, 0, 1, 6, 0, 4, 6, 5, 4, 6, 0, 4, 1, 0, 0, 0, 0, 0, 0]$$
$$[5, 3, 0, 1, 1, 1, 5, 2, 0, 3, 3, 5, 5, 0, 3, 3, 0, 6, 0, 1, 6, 0, 4, 6, 5, 4, 6, 0, 4, 1, 0, 0, 0, 0, 0, 0]$$
$$[5, 3, 5, 1, 1, 1, 5, 2, 0, 3, 3, 5, 5, 0, 3, 3, 0, 6, 0, 1, 6, 0, 4, 6, 5, 4, 6, 0, 4, 1, 0, 0, 0, 0, 0, 0]$$
$$[5, 3, 5, 1, 1, 1, 5, 2, 0, 3, 3, 5, 5, 0, 3, 3, 0, 6, 0, 1, 6, 0, 4, 6, 5, 4, 6, 0, 4, 1, 0, 6, 0, 0, 0, 0]$$
$$[5, 3, 5, 1, 1, 1, 5, 2, 0, 3, 3, 5, 5, 0, 3, 3, 0, 6, 0, 1, 6, 2, 4, 6, 5, 4, 6, 0, 4, 1, 0, 6, 0, 0, 0, 0]$$
$$[5, 3, 5, 1, 1, 1, 5, 2, 0, 3, 3, 5, 5, 0, 3, 3, 0, 6, 0, 1, 6, 2, 4, 6, 5, 4, 6, 1, 4, 1, 0, 6, 0, 0, 0, 0]$$
$$[5, 3, 5, 1, 1, 1, 5, 2, 0, 3, 3, 5, 5, 0, 3, 3, 0, 6, 3, 1, 6, 2, 4, 6, 5, 4, 6, 1, 4, 1, 0, 6, 0, 0, 0, 0]$$

where the first line means the initial starting network architecture and the last line means the searched final network architecture under given parameter and MAdds constraints. Each line (current architecture) fills one more block based on the searched previous architecture by RCAS. $1 - 6$ in one line means the type of basic block is picked to fill $N = 36$ positions for building network architecture and 0 means no block will be inserted to those positions, considered as identity operation.

Given parameter constraint $2.4M$ and MAdds constraint $91.1M$ (MobileNetV2 parameters and MAdds on CIFAR-100), our network architecture searched on CIFAR-100 is described in Table 3.

Name	Input	Operator	Expansion (t)	Expansion Group (g_e)	Stride (s)	Projection Group (g_p)	Output Channels (C_{out})	Repetition (n_i)
conv0	$32 \times 32 \times 3$	conv2d	-	-	1	-	32	-
MB1	$32 \times 32 \times 32$	MB	1	1	1	1	16	1
RC2	$32 \times 32 \times 16$	Type 5	6	1	1	1	24	1
RC2(1)	$32 \times 32 \times 24$	Type 3	6	2	1	2	24	1
RC2(2)	$32 \times 32 \times 24$	Type 1	3	1	1	1	24	1
RC2(3)	$32 \times 32 \times 24$	Type 1	3	1	1	1	24	1
RC3	$32 \times 32 \times 24$	Type 5	6	1	2	1	32	1
RC3(1)	$16 \times 16 \times 32$	Type 2	3	1	1	2	32	1
RC3(2)	$16 \times 16 \times 32$	Type 3	6	2	1	2	32	1
RC3(3)	$16 \times 16 \times 32$	Type 3	6	2	1	2	32	1
RC3(4)	$16 \times 16 \times 32$	Type 5	6	1	1	1	32	1
RC4	$16 \times 16 \times 32$	Type 5	6	1	2	1	64	1
RC4(1)	$8 \times 8 \times 64$	Type 3	6	2	1	2	64	1
RC4(2)	$8 \times 8 \times 64$	Type 3	6	2	1	2	64	1
RC4(3)	$8 \times 8 \times 64$	Type 6	12	2	1	4	64	1
RC5	$8 \times 8 \times 64$	Type 1	3	1	1	1	96	1
RC5(1)	$8 \times 8 \times 96$	Type 6	12	2	1	4	96	1
RC5(2)	$8 \times 8 \times 96$	Type 4	6	1	1	2	96	1
RC5(3)	$8 \times 8 \times 96$	Type 6	12	2	1	4	96	1
RC6	$8 \times 8 \times 96$	Type 5	6	1	2	1	160	1
RC6(1)	$4 \times 4 \times 160$	Type 4	6	1	1	2	160	5
RC6(2)	$4 \times 4 \times 160$	Type 6	12	2	1	4	160	5
RC6(3)	$4 \times 4 \times 160$	Type 4	6	1	1	2	160	5
RC6(4)	$4 \times 4 \times 160$	Type 1	3	1	1	1	160	5
conv8	$4 \times 4 \times 160$	conv2d 1x1	-	-	-	1	1280	-
pool9	$4 \times 4 \times 1280$	avgpool 4x4	-	-	-	1	1280	-
fc10	$1 \times 1 \times 1280$	FC	-	-	-	-	100	-

Table 3: The searched network architecture on CIFAR-100 under MobileNetV2 parameter and MAdds constraints.

On ImageNet, the architecture search path by RCAS is,

$$[5,0,0,0,0,0,5,0,0,0,0,0,5,0,0,0,0,0,0,0,0,0,0,5,0,0,0,0,0,0,0,0,0,0]$$
$$[5,1,0,0,0,0,5,0,0,0,0,0,5,0,0,0,0,0,0,0,0,0,0,5,0,0,0,0,0,0,0,0,0,0]$$
$$[5,1,0,0,0,0,5,0,0,0,0,0,5,0,0,0,0,0,0,6,0,0,0,5,0,0,0,0,0,0,0,0,0,0]$$
$$[5,1,0,0,0,0,5,0,0,0,1,0,5,0,0,0,0,0,0,6,0,0,0,5,0,0,0,0,0,0,0,0,0,0]$$
$$[5,1,0,0,0,0,5,0,0,0,1,0,5,2,0,0,0,0,0,6,0,0,0,5,0,0,0,0,0,0,0,0,0,0]$$
$$[5,1,0,0,0,0,5,0,0,0,1,0,5,2,0,0,0,0,0,6,0,4,0,5,0,0,0,0,0,0,0,0,0,0]$$
$$[5,1,0,0,0,0,5,0,0,0,1,0,5,2,0,4,0,0,0,6,0,4,0,5,0,0,0,0,0,0,0,0,0,0]$$
$$[5,1,0,0,0,0,5,0,0,0,1,0,5,2,0,4,0,0,0,6,0,4,0,5,0,0,0,6,0,0,0,0,0,0]$$
$$[5,1,0,0,0,0,5,0,0,0,1,0,5,2,0,4,0,0,0,6,0,4,6,5,0,0,0,6,0,0,0,0,0,0]$$
$$[5,1,0,0,0,0,5,0,0,0,1,0,5,2,0,4,0,0,0,6,0,4,6,5,3,0,0,6,0,0,0,0,0,0]$$
$$[5,1,0,0,0,0,5,0,0,0,1,0,5,2,0,4,0,0,0,6,0,4,6,5,3,0,3,6,0,0,0,0,0,0]$$
$$[5,1,0,0,0,0,5,0,0,0,1,0,5,2,0,4,0,0,1,6,0,4,6,5,3,0,3,6,0,0,0,0,0,0]$$
$$[5,1,0,0,0,0,5,0,0,0,1,0,5,2,0,4,0,0,1,6,0,4,6,5,3,0,3,6,3,0,0,0,0,0]$$
$$[5,1,0,0,0,0,5,0,0,0,1,0,5,2,0,4,0,0,1,6,0,4,6,5,3,0,3,6,3,0,0,0,6,0]$$
$$[5,1,0,4,0,0,5,0,0,0,1,0,5,2,0,4,0,0,1,6,0,4,6,5,3,0,3,6,3,0,0,0,6,0]$$
$$[5,1,0,4,0,0,5,0,0,0,1,0,5,2,0,4,0,0,1,6,0,4,6,5,3,0,3,6,3,0,0,0,6,0]$$

Given parameter constraint $3.4M$ and MAdds constraint $300M$ (MobileNetV2 parameters and MAdds on ImageNet), our network architecture searched on ImageNet is described in Table 4. A graphic of the architecture search and final architecture can be found on the supplement webpage.

Name	Input	Operator	Expansion (t)	Expansion Group (g_e)	Stride (s)	Projection Group (g_p)	Output Channels	Repetition
conv0	$224 \times 224 \times 3$	conv2d	-	-	2	-	32	-
MB1	$112 \times 112 \times 32$	MB	1	1	1	1	16	1
RC2	$112 \times 112 \times 16$	Type 5	6	1	2	1	24	1
RC2(1)	$56 \times 56 \times 24$	Type 1	3	2	1	2	24	1
RC2(2)	$56 \times 56 \times 24$	Type 4	6	1	1	2	24	1
RC3	$56 \times 56 \times 24$	Type 5	6	1	2	1	32	1
RC3(1)	$28 \times 28 \times 32$	Type 1	3	1	1	1	32	1
RC4	$28 \times 28 \times 32$	Type 5	6	1	2	1	64	1
RC4(1)	$14 \times 14 \times 64$	Type 2	3	1	1	2	64	1
RC4(2)	$14 \times 14 \times 64$	Type 4	6	1	1	2	64	1
RC5	$14 \times 14 \times 64$	Type 1	3	1	1	1	96	1
RC5(1)	$14 \times 14 \times 96$	Type 6	12	2	1	4	96	1
RC5(2)	$14 \times 14 \times 96$	Type 4	6	1	1	2	96	1
RC5(3)	$14 \times 14 \times 96$	Type 6	12	2	1	4	96	1
RC6	$14 \times 14 \times 96$	Type 5	6	1	2	1	160	1
RC6(1)	$7 \times 7 \times 160$	Type 3	6	2	1	2	160	1
RC6(2)	$7 \times 7 \times 160$	Type 3	6	2	1	2	160	1
RC6(3)	$7 \times 7 \times 160$	Type 6	12	2	1	4	160	1
RC6(4)	$7 \times 7 \times 160$	Type 3	6	2	1	2	160	1
RC7	$7 \times 7 \times 160$	Type 6	12	2	1	4	320	1
conv8	$7 \times 7 \times 320$	conv2d 1x1	-	-	-	1	1280	-
pool9	$7 \times 7 \times 1280$	avgpool 7x7	-	-	-	1	1280	-
fc10	$1 \times 1 \times 1280$	FC	-	-	-	-	1000	-

Table 4: The searched network architecture on ImageNet under MobileNetV2 parameter and MAdds constraints.

Name	Input	Operator	Expansion (t)	Expansion Group (g_e)	Stride (s)	Projection Group (g_p)	Output Channels (C_{out})	Repetition (n_i)
conv0	$224 \times 224 \times 3$	conv2d	-	-	2	-	32	-
MB1	$112 \times 112 \times 32$	MB	1	1	1	1	16	1
RC2	$112 \times 112 \times 16$	Type 5	6	1	2	1	24	1
RC2(1)	$56 \times 56 \times 24$	Type 3	6	2	1	2	24	1
RC2(2)	$56 \times 56 \times 24$	Type 1	3	1	1	1	24	1
RC2(3)	$56 \times 56 \times 24$	Type 1	3	1	1	1	24	1
RC2(4)	$56 \times 56 \times 24$	Type 1	3	1	1	1	24	1
RC2(5)	$56 \times 56 \times 24$	Type 5	6	1	1	1	24	1
RC3	$56 \times 56 \times 24$	Type 5	6	1	2	1	32	1
RC3(1)	$28 \times 28 \times 32$	Type 2	3	1	1	2	32	1
RC3(2)	$28 \times 28 \times 32$	Type 3	6	2	1	2	32	1
RC3(3)	$28 \times 28 \times 32$	Type 3	6	2	1	2	32	1
RC3(4)	$28 \times 28 \times 32$	Type 5	6	1	1	1	32	1
RC4	$28 \times 28 \times 32$	Type 5	6	1	2	1	64	1
RC4(1)	$14 \times 14 \times 64$	Type 3	6	2	1	2	64	1
RC4(2)	$14 \times 14 \times 64$	Type 3	6	2	1	2	64	1
RC4(3)	$14 \times 14 \times 64$	Type 6	12	2	1	4	64	1
RC5	$14 \times 14 \times 64$	Type 1	3	1	1	1	96	1
RC5(1)	$14 \times 14 \times 96$	Type 6	12	2	1	4	96	1
RC5(2)	$14 \times 14 \times 96$	Type 4	6	1	1	2	96	5
RC5(3)	$14 \times 14 \times 96$	Type 6	12	2	1	4	96	5
RC6	$14 \times 14 \times 96$	Type 5	6	1	2	1	160	1
RC6(1)	$7 \times 7 \times 160$	Type 4	6	1	1	2	160	5
RC6(2)	$7 \times 7 \times 160$	Type 6	12	2	1	4	160	5
RC6(3)	$7 \times 7 \times 160$	Type 4	6	1	1	2	160	5
RC6(4)	$7 \times 7 \times 160$	Type 1	3	1	1	1	160	5
RC7	$7 \times 7 \times 160$	Type 6	12	2	1	4	320	1
conv8	$7 \times 7 \times 320$	conv2d 1x1	-	-	-	1	1280	-
pool9	$7 \times 7 \times 1280$	avgpool 7x7	-	-	-	1	1280	-
fc10	$1 \times 1 \times 1280$	FC	-	-	-	-	1000	-

Table 5: The searched network architecture on CIFAR-100 adapts to ImageNet.

2.3. Adapting to ImageNet

We showed the architecture detail of RCNet searched on CIFAR-100 and ImageNet in the above subsection. In this subsection, we show the network architecture on ImageNet by adapting the searched RCNet on CIFAR-100. Table 5 lists the adapted RCNet. This adapted model offers 72.1% top-1 accuracy with similar complexity, comparable to searching directly on ImageNet. Without searching directly on ImageNet, the RCAS algorithm can find a comparable architecture given parameter and MAdds constraints ~ 4 times faster. Even though RCNet searched on ImageNet and RCNet adapted from CIFAR-100 show similar strong performance on ImageNet, the architectures are not very alike layer by layer. This may indicate the existence of many locally optimal solutions for architecture search under parameter and MAdds constraints.

References

[1] Ronald L. Graham. An efficient algorithm for determining the convex hull of a finite planar set. *Info. Pro. Lett.*, 1:132–133, 1972. 1

[2] Samir Khuller, Anna Moss, and Joseph Seffi Naor. The budgeted maximum coverage problem. *Information processing letters*, 70(1):39–45, 1999. 2

[3] Andreas Krause and Carlos Guestrin. *A note on the budgeted maximization of submodular functions*. 2005. 2

[4] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018. 2

[5] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018. 2

[6] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. 2