Supplementary Material
ReNAS: Relativistic Evaluation of Neural Architecture Search

Yixing Xu[1], Yunhe Wang[1], Kai Han[1], Yehui Tang[14], Shangling Jui[2], Chunjing Xu[1], Chang Xu[3]
[1]Noah's Ark Lab, Huawei Technologies, [2]Huawei Technologies
[3]The University of Sydney, [4]Peking University
{yixing.xu, yunhe.wang}@huawei.com; c.xu@sydney.edu.au

1. Proof of Theorem 1

We first give the definition of σ-admissibility of the ranking loss function ℓ:

Definition 1. ***(σ-admissibility)*** *Given $\mathcal{F}$ as a class of real-valued functions on $\mathcal{X}$. Denote ℓ as the ranking loss function and $\sigma > 0$. Then ℓ is σ-admissible with respect to $\mathcal{F}$, if for all $f_1, f_2 \in \mathcal{F}$ and all $(x, y), (x', y') \in (\mathcal{X} \times \mathcal{Y})$, we have:*

$$|\ell(f_1, (x, y), (x', y')) - \ell(f_2, (x, y), (x', y'))| \leq \sigma(|f_1(x) - f_2(x)| + |f_1(x') - f_2(x')|). \quad (1)$$

Then, we will get the following generalization error bound for a given ranking loss function ℓ:

Lemma 1. *Given $\mathcal{A}$ as the symmetric ranking algorithm whose outputs of samples on a training dataset $\mathcal{D} \in (\mathcal{X} \times \mathcal{Y})^n$ is $f_{\mathcal{D}} = \arg\min_{f \in \mathcal{F}} \hat{R}_\ell^\lambda(f)$, in which $n \in \mathbb{N}$ is the number of training samples. Denote c_x and c_f as the upper bound of the inputs and weights such that for all $x \in \mathcal{X}$ and $f : \mathcal{X} \to \mathbb{R}$ we have $|x| \leq c_x$ and $\|f\|_2 \leq c_f$. Also given ℓ as the ranking loss function that satisfy $0 \leq \ell(f, (x, y), (x', y')) \leq L$ for all $f : \mathcal{X} \to \mathbb{R}$ and $(x, y), (x', y') \in (\mathcal{X} \times \mathcal{Y})^2$. Then for any $0 < \delta < 1$, with probability at least $1 - \delta$ we have:*

$$R_\ell(f_{\mathcal{D}}) < \hat{R}_\ell(f_{\mathcal{D}}) + \frac{8\sigma c_x^2 c_f^2}{\lambda n} + (\frac{4\sigma c_x^2 c_f^2}{\lambda} + L)\sqrt{\frac{2\ln(1/\delta)}{n}}. \quad (2)$$

Proof. Given the assumption of using a two layer neural network with ReLU activation function, we can denote the output of the neural network as:

$$f(x) = W_2 \eta(W_1 \cdot x), \quad (3)$$

in which W_1 and W_2 are the parameters of the given network, and η indicates the ReLU activation function. Also denote $\|f\|_2 = \sqrt{\|W_1\|_2^2 + \|W_2\|_2^2}$ as the ℓ_2-norm of the parameters, we then have:

$$\begin{aligned} |f(x)| &= |W_2 \eta(W_1 \cdot x)| \\ &\leq \big||W_2| \eta(|W_1 \cdot x|)\big| \\ &= |W_2 \cdot W_1 \cdot x| \\ &= |W_2 \cdot W_1| \cdot |x| \\ &\leq \frac{1}{2}(\|W_1\|_2^2 + \|W_2\|_2^2)|x| \\ &\leq \frac{1}{2} c_x c_f \|f\|_2. \end{aligned} \quad (4)$$

Thus, given Fcn. 4 mentioned above, and Theorem.8, Fcn.6 and Theorem.11 in [1], we can successfully prove Lemma 1. □

After that, we prove that a hinge ranking loss is 1-admissible with respect to $\mathcal{F}$ and an MSE loss is $(\frac{c_x c_f L}{2\sqrt{\lambda}} + 1)$-admissible with respect to $\mathcal{F}$.

Theorem 1. *Given $\mathcal{F}$ as a class of real-valued functions on $\mathcal{X}$. Denote ℓ as the ranking loss function and $\sigma > 0$. Then $\ell_h(f, (x, y), (x', y')) = [(a - (f(x) - f(x')) \cdot sign(y - y'))]_+$ is 1-admissible with respect to $\mathcal{F}$,* e.g. *for all $f_1, f_2 \in \mathcal{F}$ and all $(x, y), (x', y') \in (\mathcal{X} \times \mathcal{Y})$, we have:*

$$|\ell_h(f_1, (x, y), (x', y')) - \ell_h(f_2, (x, y), (x', y'))| \leq |f_1(x) - f_2(x)| + |f_1(x') - f_2(x')|. \quad (5)$$

Proof. Without loss of generality, we assume that $\ell_h(f_1, (x, y), (x', y')) \geq \ell_h(f_2, (x, y), (x', y'))$. Note that when $\ell_h(f_1, (x, y), (x', y')) = \ell_h(f_2, (x, y), (x', y'))$, we simply have:

$$|\ell_h(f_1, (x, y), (x', y')) - \ell_h(f_2, (x, y), (x', y'))| = 0 \leq |f_1(x) - f_2(x)| + |f_1(x') - f_2(x')|, \quad (6)$$

thus the following prove is based on $\ell_h(f_1, (x, y), (x', y')) > \ell_h(f_2, (x, y), (x', y'))$, and can be divided into following situations:

(1) $(f_1(x) - f_1(x')) \cdot \text{sign}(y - y') \leq a$ and $(f_2(x) - f_2(x')) \cdot \text{sign}(y - y') \leq a$. Then we have:

$$\begin{aligned}
&|\ell_h(f_1, (x, y), (x', y')) - \ell_h(f_2, (x, y), (x', y'))| \\
=&|a - (f_1(x) - f_1(x')) \cdot \text{sign}(y - y') \\
&- a + (f_2(x) - f_2(x')) \cdot \text{sign}(y - y')| \\
=&\text{sign}(y - y')|f_1(x) - f_2(x) + f_1(x') - f_2(x')| \\
\leq&|f_1(x) - f_2(x) + f_1(x') - f_2(x')| \\
\leq&|f_1(x) - f_2(x)| + |f_1(x') - f_2(x')|. \qquad (7)
\end{aligned}$$

(2) $(f_1(x) - f_1(x')) \cdot \text{sign}(y - y') \leq a$ and $(f_2(x) - f_2(x')) \cdot \text{sign}(y - y') > a$. Then we have:

$$\begin{aligned}
&|\ell_h(f_1, (x, y), (x', y')) - \ell_h(f_2, (x, y), (x', y'))| \\
=&|a - (f_1(x) - f_1(x')) \cdot \text{sign}(y - y') - 0| \\
<&|a - (f_1(x) - f_1(x')) \cdot \text{sign}(y - y') \\
&- (a - (f_2(x) - f_2(x')) \cdot \text{sign}(y - y'))| \\
\leq&|f_1(x) - f_2(x)| + |f_1(x') - f_2(x')|. \qquad (8)
\end{aligned}$$

Therefore, in all situations we have:

$$\begin{aligned}
&|\ell_h(f_1, (x, y), (x', y')) - \ell_h(f_2, (x, y), (x', y'))| \leq \\
&\qquad |f_1(x) - f_2(x)| + |f_1(x') - f_2(x')|, \qquad (9)
\end{aligned}$$

and thus $\ell_h(f_1, (x, y), (x', y'))$ is 1-admissible with respect to $\mathcal{F}$. □

Theorem 2. *Given $\mathcal{F}$ as a class of real-valued functions on $\mathcal{X}$. Denote ℓ as the ranking loss function and $\sigma > 0$. Then $\ell_{mse}(f, (x, y), (x', y')) = \frac{1}{2}((f(x) - y)^2 + (f(x') - y')^2)$ is $(\frac{c_x c_f L}{2\sqrt{\lambda}} + 1)$-admissible with respect to $\mathcal{F}$,* e.g. *for all $f_1, f_2 \in \mathcal{F}$ and all $(x, y), (x', y') \in (\mathcal{X} \times \mathcal{Y})$, we have:*

$$\begin{aligned}
&|\ell_h(f_1, (x, y), (x', y')) - \ell_h(f_2, (x, y), (x', y'))| \leq \\
&(\frac{c_x c_f L}{2\sqrt{\lambda}} + 1)\big(|f_1(x) - f_2(x)| + |f_1(x') - f_2(x')|\big). \qquad (10)
\end{aligned}$$

Proof.

$$\begin{aligned}
&|\ell_{\text{mse}}(f_1, (x, y), (x', y')) - \ell_{\text{mse}}(f_2, (x, y), (x', y'))| \\
=&\frac{1}{2}|(f_1(x) - y)^2 + (f_1(x') - y')^2 \\
&+ (f_2(x) - y)^2 + (f_2(x') - y')^2| \\
=&\frac{1}{2}|f_1^2(x) - 2f_1(x) + f_1^2(x') - 2f_1(x') \\
&- f_2^2(x) + 2f_2(x) - f_2^2(x') + 2f_2(x')| \\
=&\frac{1}{2}|f_1^2(x) - f_2^2(x) + f_1^2(x') - f_2^2(x') \\
&- 2(f_1(x) - f_2(x)) - 2(f_1(x') - f_2(x'))| \\
\leq&\frac{1}{2}(|f_1^2(x) - f_2^2(x)| + |f_1^2(x') - f_2^2(x')| \\
&+ 2|(f_1(x) - f_2(x))| + 2|(f_1(x') - f_2(x'))|) \\
=&\frac{1}{2}(|f_1(x) + f_2(x)| + 2)|f_1(x) - f_2(x)| + \\
&\frac{1}{2}(|f_1(x') + f_2(x')| + 2)|f_1(x') - f_2(x')|. \qquad (11)
\end{aligned}$$

Given Fcn. 4 mentioned above, we have:

$$|f(x)| \leq \frac{1}{2} c_x c_f \|f\|_2. \qquad (12)$$

Also note that:

$$\ell_{\text{mse}} = \hat{R}_{\ell_{\text{mse}}} + \lambda\|f\|_2^2 \leq L. \qquad (13)$$

Since $\hat{R}_{\ell_{\text{mse}}} > 0$, we have:

$$\|f\|_2^2 \leq \frac{L^2}{\lambda}. \qquad (14)$$

Applying Eq. 14 to Eq. 12, we have:

$$|f(x)| \leq \frac{1}{2} c_x c_f \|f\|_2 \leq \frac{c_x c_f L}{2\sqrt{\lambda}}. \qquad (15)$$

Finally, applying Eq. 15 to Eq. 11, we can derive the $(\frac{c_x c_f L}{2\sqrt{\lambda}} + 1)$-admissibility of ℓ_{mse}:

$$\begin{aligned}
&|\ell_{\text{mse}}(f_1, (x, y), (x', y')) - \ell_{\text{mse}}(f_2, (x, y), (x', y'))| \\
\leq&\frac{1}{2}(|f_1(x) + f_2(x)| + 2)|f_1(x) - f_2(x)| \\
&+ \frac{1}{2}(|f_1(x') + f_2(x')| + 2)|f_1(x') - f_2(x')| \\
\leq&(\frac{c_x c_f L}{2\sqrt{\lambda}} + 1)\big(|f_1(x) - f_2(x)| + |f_1(x') - f_2(x')|\big), \\
&\qquad (16)
\end{aligned}$$

and thus finish the proof. □

Combining the above definition, lemma and theorems, we have proved Theorem 1 in the main paper.

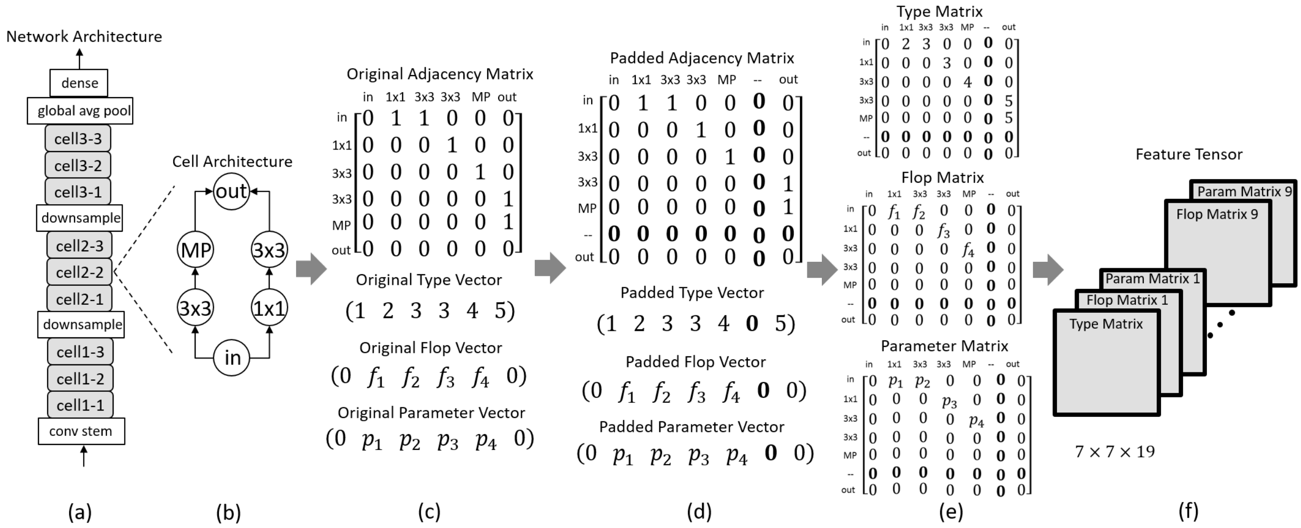

Figure 1: An example of encoding neural network architecture into feature tensor. **(a)**: The skeleton of the neural network architecture. **(b)**: A specific cell architecture with 6 nodes. **(c)**: The corresponding adjacency matrix $\mathcal{A}$, type vector $\mathbf{t}$, FLOP vector $\mathbf{f}$ and parameter vector $\mathbf{p}$ of the cell. **(d)**: Padding adjacency matrix $\mathcal{A}$ to 7×7 and vectors accordingly. Note that the zero-padding is added at penultimate row and column, since the last row and column represents the output node. **(e)**: Vectors are broadcasted into matrix, and an element wise multiplication is made with the adjacency matrix to get the type matrix, FLOP matrix and parameter matrix. **(f)**: There are 9 cells in the network, thus producing 9 different FLOP matrices and parameter matrices. All the cells share the same type matrix. We concatenate all the matrices to get the final $19 \times 7 \times 7$ tensor.

2. An Example of Deriving Feature Tensor

In this section, we give an example of the process of deriving feature tensor from cell-based search space NAS-Bench-101. We use an architecture with 6 nodes in a cell, and the process is shown in Fig. 1.

3. More Experiments on NAS-Bench-101

In this section, we conduct more experiments on NAS-Bench-101 dataset to further demonstrate the usefulness of the proposed ReNAS method.

In the following we give an intuitive representation of the best architectures selected by the performance predictor with different number of training samples as shown in Fig. 2. The best cell architecture searched by EA algorithm using proposed predictor trained with random selected training samples is shown in column 2 of Fig. 2. Note that the rank-1 architecture in NAS-Bench-101 dataset cannot be selected by the predictor even when using 90% of the training data. This is because when using pairwise ranking based loss function, there are $n(n-1)/2$ training pairs and it is inefficient to train them in a single batch. Thus, mini-batch updating method is used and a single architecture is compared with limited architectures in one epoch, which causes the lack of global information about this architecture especially when the number of training samples is large. In fact, the mini-batch size b is set to 1024 in the experiment, and it is a compromise between effectiveness and efficiency.

This is the same reason that the performance of the architecture found by the predictor trained with 90% dataset is marginally better than that trained with 0.1% dataset. Specifically, we divide the architectures into two parts. The first part is the architectures trained with 0.1% and 1% dataset, and the second part is the rest. Note that in the first part the number of training sample is on the same order of magnitude with the mini-batch size b, thus the global information of a single model is easy to obtain and the performance becomes better when there are more training data. In the second part, the number of training sample is significantly larger than b. On one hand, increasing the number of samples helps training. On the other hand, the global ranking information is harder to get. Thus, the performance is marginally better when using more training samples.

Finally, there are some common characteristics among these architectures. The first is that the distance between input and output node is at most 2, which shows the significance of skip-connection. The second is that 3×3 operation appears in each architecture. Based on these observations, we separate the NAS-Bench-101 dataset based on the distance between input node and output node, and whether the 3×3 operation is used. Some statistics are shown in Tab. 1.

It shows that the shorter the distance between input n-

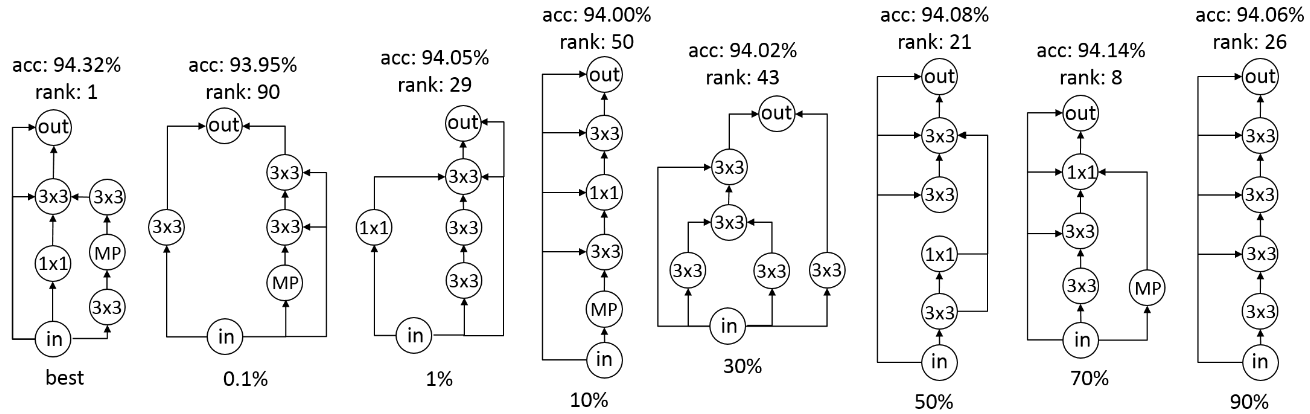

Figure 2: The best architectures found by the predictor with different ratio of training samples.

Table 1: Statistics on NAS-Bench-101 dataset. '3 × 3' refers to whether the model uses this operation. 'Distance' refers to the distance between input node and output node. '#model' refers to the number of models. 'Best acc' refers to the performance of the best architecture among '#model' number of models on CIFAR-10 dataset. 'Average acc' refers to the average performance of '#model' number of models on CIFAR-10 dataset.

3×3	Distance	#model	Best acc	Average acc
yes	1	68552	**94.32**	**91.97**
	2	153056	94.05	91.02
	3	110863	93.68	89.31
	4	27227	92.36	87.40
	5	2516	90.54	86.51
	6	211	88.87	84.91
no	1	12468	91.62	88.40
	2	26282	90.81	86.69
	3	17735	90.24	83.53
	4	4282	88.95	80.20
	5	400	88.16	78.84
	6	32	86.71	74.93

Table 2: Predictors trained and evaluated with the whole NAS-Bench-101 dataset and sub dataset. The experiments are repeated 20 times to alleviate the randomness of the results.

Datasets	accuracy(%)	ranking(%)
whole dataset	93.95 ± 0.11	0.02
sub dataset	$\mathbf{94.02 \pm 0.14}$	**0.01**

ode and output node, the better the performance is. Besides, 3 × 3 operation helps the architecture to perform better. Based on the observation above, we may form a better search space of NAS-Bench-101 dataset by using only 68552 models with 3 × 3 operation and skip-connect between input node and output node. An experiment of training and evaluating performance predictor is conducted on this sub search space and the results show that the predictor trained and evaluated within the sub search space performs better than the previous one as shown in Tab. 2. It shows that a better search space helps to produce a better performance predictor.

References

[1] Shivani Agarwal and Partha Niyogi. Generalization bounds for ranking algorithms via algorithmic stability. *Journal of Machine Learning Research*, 10(Feb):441–474, 2009. 1