Adaptive Stochastic Natural Gradient Method for One-Shot Neural Architecture Search: Supplementary Material

Youhei Akimoto [*1] **Shinichi Shirakawa** [*2] **Nozomu Yoshinari** [2] **Kento Uchida** [2] **Shota Saito** [2 3] **Kouhei Nishida** [4]

A. Proof of Proposition 3

Proof. It follows

$$\begin{aligned}
&\ln J(\theta') - \ln J(\theta) \\
&= \ln \int \frac{p_{\theta'}(\boldsymbol{c})}{p_\theta(\boldsymbol{c})} \frac{f(\boldsymbol{c})}{J(\theta)} p_\theta(\boldsymbol{c}) \\
&\geq \int \ln \left(\frac{p_{\theta'}(\boldsymbol{c})}{p_\theta(\boldsymbol{c})} \right) \frac{f(\boldsymbol{c})}{J(\theta)} p_\theta(\boldsymbol{c}) \\
&\geq \int_{\boldsymbol{c}: p_{\theta'}(\boldsymbol{c}) < p_\theta(\boldsymbol{c})} \ln \left(\frac{p_{\theta'}(\boldsymbol{c})}{p_\theta(\boldsymbol{c})} \right) \frac{f(\boldsymbol{c})}{J(\theta)} p_\theta(\boldsymbol{c}) \\
&\geq \frac{f^*}{J(\theta)} \int_{\boldsymbol{c}: p_{\theta'}(\boldsymbol{c}) < p_\theta(\boldsymbol{c})} \ln \left(\frac{p_{\theta'}(\boldsymbol{c})}{p_\theta(\boldsymbol{c})} \right) p_\theta(\boldsymbol{c}) \\
&= -\frac{f^*}{J(\theta)} \int_{\boldsymbol{c}: p_{\theta'}(\boldsymbol{c}) < p_\theta(\boldsymbol{c})} \ln \left(\frac{p_\theta(\boldsymbol{c})}{p_{\theta'}(\boldsymbol{c})} \right) p_\theta(\boldsymbol{c}) \\
&\geq -\frac{f^*}{J(\theta)} D_\theta(\theta', \theta) \ . \qquad \square
\end{aligned}$$

B. Derivation for Categorical Distribution

We derive the Fisher information matrix, its inverse and square root, and the natural gradient for the categorical distribution defined on $\mathcal{C} = [\![1, m_1]\!] \times \cdots \times [\![1, m_{n_{\boldsymbol{c}}}]\!]$.

In our parameterization $\theta = (\theta_1, \ldots, \theta_{n_{\boldsymbol{c}}})$, the probability of i-th ($i \in [\![1, n_{\boldsymbol{c}}]\!]$) categorical variable $[\boldsymbol{c}]_i$ to be $j \in [\![1, m_i - 1]\!]$ is $[\theta]_{i,j} = [\theta_i]_j$, and $1 - \sum_{j=1}^{m_i - 1} [\theta]_{i,j}$ is the probability of $[\boldsymbol{c}]_i = m_i$. For the sake of simplicity, we denote $[\theta]_{i,m_i} = 1 - \sum_{j=1}^{m_i - 1} [\theta]_{i,j}$. Then, $T(\boldsymbol{c}) = (T_1([\boldsymbol{c}]_1), \ldots, T_{n_{\boldsymbol{c}}}([\boldsymbol{c}]_{n_{\boldsymbol{c}}}))$, where $T_i : [\![1, m_i]\!] \to [0,1]^{m_i - 1}$ is the one-hot representation without the last element. It is the exponential parameterization as $\theta = \mathbb{E}[T(\boldsymbol{c})]$.

The inverse of the Fisher information matrix simply follows from the formula for the exponential family: $\mathbf{F}(\theta)^{-1} = \mathbb{E}[(T(\boldsymbol{c}) - \theta)(T(\boldsymbol{c}) - \theta)^{\mathrm{T}}]$. It is a block diagonal matrix $\mathbf{F}(\theta)^{-1} = \mathrm{diag}(\mathbf{F}_1(\theta_1)^{-1}, \ldots, \mathbf{F}_{n_{\boldsymbol{c}}}(\theta_{n_{\boldsymbol{c}}})^{-1})$, where $\mathbf{F}_i(\theta_i)^{-1} = \mathrm{diag}(\theta_i) - \theta_i \theta_i^{\mathrm{T}}$. Sherman-Morrison formula reads $\mathbf{F}_i(\theta_i) = \mathrm{diag}(\theta_i)^{-1} + (1 - \sum_{j=1}^{m_i - 1} [\theta_i]_j)^{-1} \mathbf{1}\mathbf{1}^{\mathrm{T}}$ and we have $\mathbf{F}(\theta) = \mathrm{diag}(\mathbf{F}_1(\theta_1), \ldots, \mathbf{F}_{n_{\boldsymbol{c}}}(\theta_{n_{\boldsymbol{c}}}))$.

As the Fisher information matrix is block-diagonal, and each block is of size $m_i - 1$, a naive computation of $\mathbf{F}(\theta)^{\frac{1}{2}}$ requires $O(\sum_{i=1}^{n_{\boldsymbol{c}}} (m_i - 1)^3)$. This is usually not expensive as $n_{\boldsymbol{c}} \gg m_i$. An alternative way that we employ in this paper is to replace $\mathbf{F}(\theta)^{\frac{1}{2}}$ with a tractable factorization A with $\mathbf{F}(\theta) = AA^{\mathrm{T}}$. Our choice of A is the block-diagonal matrix whose i-th block is square, of size $m_i - 1$, and

$$A_i = \mathrm{diag}(\theta_i)^{-\frac{1}{2}} + \frac{1}{\sqrt{[\theta]_{i,m_i}} + [\theta]_{i,m_i}} \mathbf{1} \sqrt{\theta_i}^{\mathrm{T}} \ ,$$

where $\sqrt{\theta_i}$ is a vector whose j-th element is the square root of $[\theta]_{i,j}$. Then, the product of A and a vector can be computed in $O(\sum_{i=1}^{n_{\boldsymbol{c}}} (m_i - 1))$. In our preliminary study we did not obverse any significant performance difference by this approximation.

C. Derivation for Gaussian Distribution

We derive the Fisher information matrix, its inverse and square root, and the natural gradient for the Gaussian distribution defined on $\mathcal{C} \subseteq \mathbb{R}^{n_{\boldsymbol{c}}}$.

Our choice is $P_\theta = \mathcal{N}(\mu_1, \sigma_1^2) \times \cdots \times \mathcal{N}(\mu_{n_{\boldsymbol{c}}}, \sigma_{n_{\boldsymbol{c}}}^2)$ and $\theta = (\mu_1, \mu_1^2 + \sigma_1^2, \ldots, \mu_{n_{\boldsymbol{c}}}, \mu_{n_{\boldsymbol{c}}}^2 + \sigma_{n_{\boldsymbol{c}}}^2)$. Then, we have $T(\boldsymbol{c}) = (T_1([\boldsymbol{c}]_1), \ldots, T_{n_{\boldsymbol{c}}}([\boldsymbol{c}]_{n_{\boldsymbol{c}}}))$ with $T_i([\boldsymbol{c}]_i) = ([\boldsymbol{c}]_i, [\boldsymbol{c}]_i^2)$. It is the exponential parameterization as $\theta_i = (\mu_i, \mu_i^2 + \sigma_i^2) = \mathbb{E}[T_i([\boldsymbol{c}]_i)]$ and $\theta = (\theta_1, \ldots, \theta_{n_{\boldsymbol{c}}})$.

The inverse of the Fisher information matrix simply follows from the formula for the exponential family. $\mathbf{F}(\theta)^{-1}$ is a block-diagonal matrix with block size 2 whose i-th block is $\mathbf{F}_i(\theta_i)^{-1} = [\sigma_i^2, 2\mu_i\sigma_i^2; 2\mu_i\sigma_i^2, 4\mu_i^2\sigma_i^2 + 2\sigma_i^4]$. Since each block is a symmetric matrix of dimension 2, its eigen decomposition $\mathbf{F}_i(\theta_i)^{-1} = \mathbf{E}\Lambda\mathbf{E}^{\mathrm{T}}$ can be analytically obtained. With the decomposition, we have $\mathbf{F}_i(\theta_i) = \mathbf{E}\Lambda^{-1}\mathbf{E}^{\mathrm{T}}$ and $\mathbf{F}(\theta_i)^{\frac{1}{2}} = \mathbf{E}\Lambda^{-\frac{1}{2}}\mathbf{E}^{\mathrm{T}}$. Then, we have $\mathbf{F}(\theta) = \mathrm{diag}(\mathbf{F}_1(\theta_1), \ldots, \mathbf{F}_{n_{\boldsymbol{c}}}(\theta_{n_{\boldsymbol{c}}}))$ and $\mathbf{F}(\theta)^{\frac{1}{2}} = \mathrm{diag}(\mathbf{F}_1(\theta_1)^{\frac{1}{2}}, \ldots, \mathbf{F}_{n_{\boldsymbol{c}}}(\theta_{n_{\boldsymbol{c}}})^{\frac{1}{2}})$.

[*]Equal contribution [1]University of Tsukuba & RIKEN AIP [2]Yokohama National University [3]SkillUp AI Co., Ltd. [4]Shinshu University. Correspondence to: Youhei Akimoto <akimoto@cs.tsukuba.ac.jp>, Shinichi Shirakawa <shirakawa-shinichi-bg@ynu.ac.jp>.

Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

D. Restriction for the Range of θ

To guarantee that the Fisher information matrix is nonsingular and the natural gradient is well defined, we restrict the domain Θ of the parameter of the probability distribution.

For the categorical distribution, we set $\Theta = [\theta_1^{\min}, \theta^{\max}]^{m_1-1} \times \cdots \times [\theta_{n_c}^{\min}, \theta^{\max}]^{m_{n_c}-1}$, where $\theta_i^{\min} = \frac{1}{n_c(m_i-1)}$ and $\theta^{\max} = 1 - \frac{1}{n_c}$. Then, a small yet positive probability for all combinations of categorical variables is guaranteed and the Fisher information matrix is nonsingular at any point of Θ.

To force the parameter to live in Θ, we apply the following steps after θ update:

$$[\theta]_{i,j} \leftarrow \max\{[\theta]_{i,j}, \theta_i^{\min}\} \text{ for all } i, j, \text{ then}$$

$$[\theta]_{i,j} \leftarrow [\theta]_{i,j} + \frac{1 - \sum_{k=1}^{m_i} [\theta]_{i,k}}{\sum_{k=1}^{m_i} \left([\theta]_{i,k} - \theta_i^{\min}\right)} \left([\theta]_{i,j} - \theta_i^{\min}\right) \quad .$$

The first line guarantees $[\theta]_{i,j} \geq \theta_i^{\min}$. The second line ensures $\sum_{j=1}^{m_i} [\theta]_{i,j} = 1$, while keeping $[\theta]_{i,j} \geq \theta_i^{\min}$.

For the integer variables, the parameters of the Gaussian distributions, $[\theta]_{i,1} := \mu_i$ and $[\theta]_{i,2} := \mu_i^2 + \sigma_i^2$, are forced to be in a compact set as follows. The range of the mean value of each integer variable is $[\mu_i^{\min}, \mu_i^{\max}]$, which is the same as the range of the integer variable. The standard deviation is forced to be no smaller than $\sigma_i^{\min} = 1/4$ and no greater than $\sigma_i^{\max} = (\mu_i^{\max} - \mu_i^{\min})/2$. To keep the parameters inside these ranges, after every θ update we clip $[\theta]_{i,1}$ to $[\mu_i^{\min}, \mu_i^{\max}]$ and $[\theta]_{i,2}$ to $[[\theta]_{i,1}^2 + (\sigma_i^{\min})^2, [\theta]_{i,1}^2 + (\sigma_i^{\max})^2]$. If the variables are real-value, rather than integer, then $\sigma_i^{\min}$ may be set smaller depending on the meaning of the variable.

E. Experimental Details

E.1. Toy Problem

To check the robustness of ASNG for the hyper-parameter α, we ran ASNG on the selective squared error function with the varying α and initial step-size δ_θ^0 for the step-sizes of $\epsilon_{\boldsymbol{x}} = \{0.05, 0.0005\}$. Figure 1 shows the performance of ASNG with the different α settings. We observe that the hyper-parameter α is not sensitive for the performance, and ASNG reaches the target value for all settings.

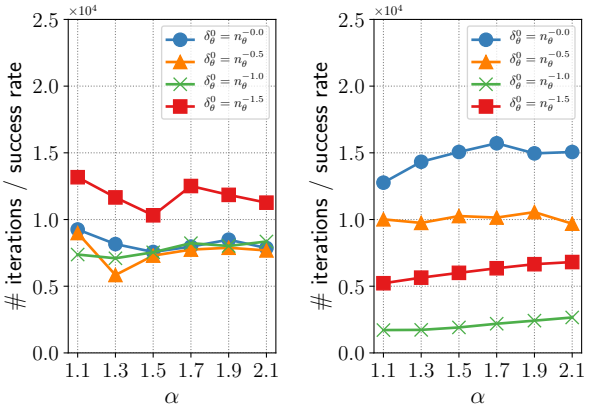

Figure 1. Performance of ASNG with the different α settings on the selective squared error function for $\epsilon_{\boldsymbol{x}} = 0.05$ (left) and 0.0005 (right). Median values over 100 runs are reported.

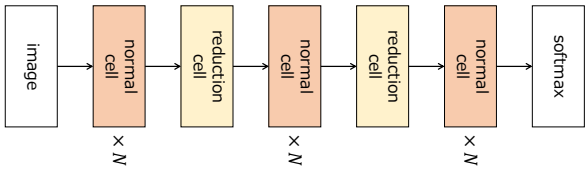

Figure 2. Overall model structure for the classification task.

E.2. Image Classification

Dataset: We use the CIFAR-10 dataset which consists of 50,000 and 10,000 RGB images of 32×32, for training and testing. All images are standardized in each channel by subtracting the mean and then dividing by the standard deviation. We adopt the standard data augmentation for each training mini-batch: padding 4 pixels on each side, followed by choosing randomly cropped 32×32 images and by performing random horizontal flips on the cropped images. We also apply the cutout (DeVries & Taylor, 2017) to the training data.

Search Space: Figure 2 shows the overall model structure for the classification task. We optimize the architecture of the normal and reduction cells by ASNG-NAS. In the retraining phase, we construct the CNN using the optimized cell architecture with an increased number of cells N and channels. In the reduction cell, all operations applied to the inputs of the cell have a stride of 2, and the number of channels is doubled to keep the dimension of the output roughly constant.

Training Details: In the architecture search phase, we fix affine parameters of batch normalizations for the purpose of absorbing effect of the dynamic change in architecture. We apply weight decay of 3×10^{-4} and clip the norm of gradient at 5. In the retraining phase, we make all batch normalizations have learnable affine parameters because the architecture no longer changes. We apply the ScheduledDropPath (Zoph et al., 2018) dropping out each path between nodes, and the drop path rate linearly increases from 0 to 0.3 during the training. We also add the auxiliary classifier (Szegedy et al., 2016) with the weight of 0.4 that is connected from the second reduction cell. The total loss

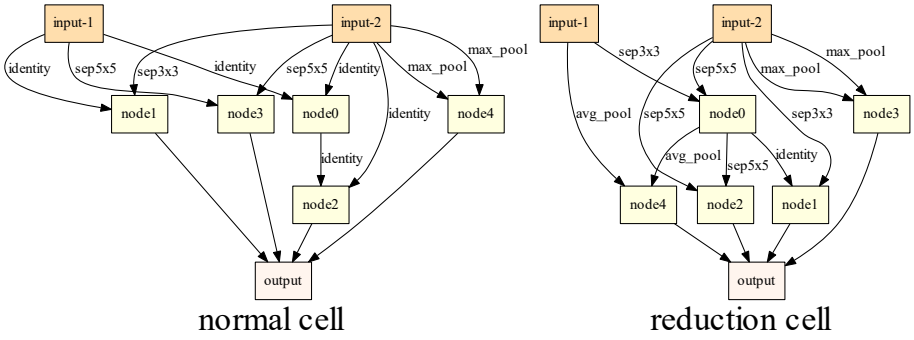

Figure 3. The best cell structures discovered by ASNG-NAS in the classification task.

is a weighted sum of the losses of the auxiliary classifier and output layer. Other settings are the same as the architecture search phase.

The best cell structures: The best cell structures that achieve the error rate of 2.66% is displayed in Figure 3.

E.3. Inpainting

Dataset: The CelebA is a large-scale human face image dataset that contains 202,599 RGB images. We select 101,000 and 2,000 images for training and test, respectively, in the same way as Suganuma et al. (2018). All images were cropped to properly contain the entire face by using the provided the bounding boxes, and resized to 64×64 pixels. All images are normalized by dividing by 255, and we perform data augmentation of random horizontal flipping on the training images. We adopt three masks, Center, Pixel, and Half, to make corrupted images. The purpose of the task is to recover a clean image from the corrupted image as much as possible. The masks in random pixel and half image masks were randomly generated for each training mini-batch and each test image.

Evaluation Measure: The PSNR is the metric evaluating the error between the ground truth and restored images and corresponds to the mean squared error (MSE). But the PSNR are not very well matched to perceived visual quality because the PSNR can not distinguish between the large difference on local region and the small difference on overall region. For this reason, the SSIM is also often used together with the PSNR, and more clearly assesses difference in each local region. We quantize the generated image within $[0, 255]$ followed by to calculate the PSNR and SSIM value. The setting of SSIM is based on Wang et al. (2004).

Search Space: We employ the convolutional autoencoder (CAE), which is similar to RED-Net (Mao et al., 2016), as a base architecture. RED-Net consists of a chain of convolution layers and symmetric deconvolution layers as the encoder and decoder parts, respectively. The encoder and decoder parts perform the same counts of downsampling and upsampling with a stride of 2, and a skip connection between the convolutional layer and the mirrored deconvolution layer can exist. For simplicity, each layer employs either a skip connection or a downsampling, and the decoder part is employed in the same manner. In the skip connected deconvolution layer, the input feature maps from the encoder part are added to the output of the deconvolution operation, followed by ReLU. In the other layers, the ReLU activation is performed after the convolution and deconvolution operations. We prepare six types of layers: the combination of the kernel sizes $\{1 \times 1, 3 \times 3, 5 \times 5\}$ and the existence of the skip connection. The layers with different settings do not share weight parameters.

To represent a symmetric CAE, it is enough to represent the encoder part. We consider N_{c} hidden layers and the output layer. We encode the type, channel size, and connections of each hidden layer. The kernel size and stride of the output deconvolution layer are fixed with 3×3 and 1, respectively, but the connection is determined by a categorical variable. To ensure the feed-forward architecture and to control the network depth, the connection of the i-th layer is only allowed to be connected from $(i-1)$ to $\max(0, i-b)$-th layers, where b ($b > 0$) is called the level-back parameter. Namely, the categorical variable representing the connection of the i-th layer has $\min(i, b)$ categories. Obviously, the first hidden layer always connects with the input, and we can ignore this part. With this representation, there can exist *inactive* layers that do not connect to the output layer. Therefore, this model can represent variable length architectures by the fixed-dimensional variables. We choose $N_c = 20$ and the level-back parameter of $b = 5$.

ASNG-NAS (Cat) encodes the type and channel size of each hidden layer by categorical variables with 6 and 3 categories, respectively. We select the output channel size of each hidden layer from $\{64, 128, 256\}$. It amounts to $n_{\mathbf{c}} = 60$ (# categorical variables) and $n_\theta = 214$ (dimension of θ). A conceptual example of the symmetric CAE architecture and

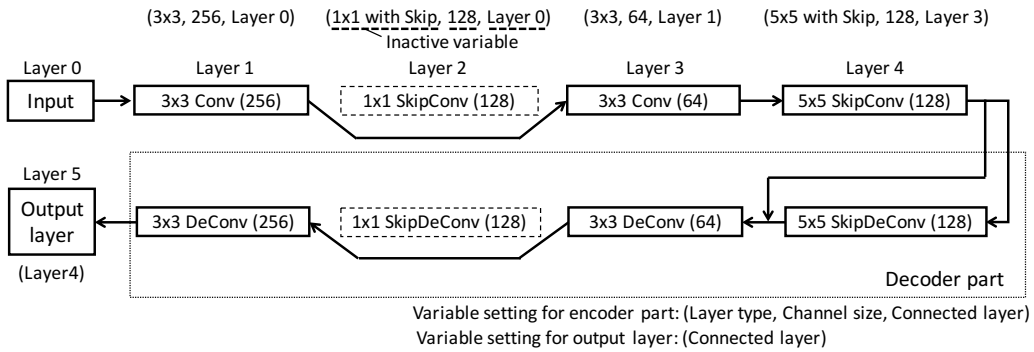

Figure 4. A conceptual example of the decoded symmetric CAE architecture and the corresponding categorical variables. The decoder part is automatically decided from the encoder structure as a symmetric manner.

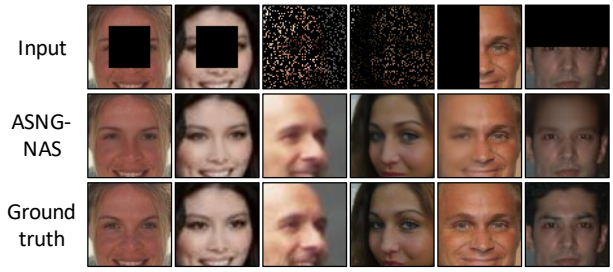

Figure 5. Example of inpainting results obtained by ASNG-NAS.

the corresponding representation by the categorical variables is shown in Figure 4. ASNG-NAS (Int) encodes the kernel size and the channel size by integers in $[\![1, 3]\!]$ (corresponding to $\{1 \times 1, 3 \times 3, 5 \times 5\}$) and $[\![64, 256]\!]$. The existence of skip connection is determined by a categorical variable with 2 categories. It amounts to $n_{\boldsymbol{c}} = 80$ (# categorical and integer variables) and $n_\theta = 174$ (dimension of θ).

Example of Inpainting Result: Figure 5 shows the example of inpainting results obtained by ASNG-NAS.

References

DeVries, T. and Taylor, G. W. Improved Regularization of Convolutional Neural Networks with Cutout. *arXiv preprint:1708.04552*, 2017.

Mao, X., Shen, C., and Yang, Y. Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2802–2810, 2016.

Suganuma, M., Ozay, M., and Okatani, T. Exploiting the Potential of Standard Convolutional Autoencoders for Image Restoration by Evolutionary Search. In *The 35th International Conference on Machine Learning (ICML)*, volume 80, pp. 4778–4787, 2018.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.

Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13 (4):600–612, 2004.

Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. Learning Transferable Architectures for Scalable Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8697–8710, 2018.