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

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

The inverse of the Fisher information matrix simply follows from the formula for the exponential family: $F(\theta)^{-1} = \mathbb{E}[(T(c) - \theta)(T(c) - \theta)^T]$. It is a block diagonal matrix $F(\theta)^{-1} = \text{diag}(F_1(\theta_1)^{-1}, \ldots, F_{n_e}(\theta_{n_e})^{-1})$, where $F_i(\theta_i)^{-1} = \text{diag}(\theta_i - \theta_i^T)$. Sherman-Morrison formula reads $F_i(\theta_i) = \text{diag}(\theta_i)^{-1} + (1 - \sum_{j=1}^{m_i-1} [\theta]_{i, j})^{-1} 11^T$ and we have $F(\theta) = \text{diag}(F_1(\theta_1), \ldots, F_{n_e}(\theta_{n_e}))$.

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

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

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 $O(\sum_{i=1}^{n_e} (m_i - 1))$. In our preliminary study we did not observe any significant performance difference by this approximation.

### B. Derivation for Categorical Distribution

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

Our choice is $P_\theta = \mathcal{N}(\mu, \sigma^2) \times \cdots \times \mathcal{N}(\mu, \sigma^2)$ and $\theta = (\mu_1, \sigma^1_1, \ldots, \mu_{n_e}, \sigma^2_1, \ldots, \sigma^2_{n_e})$. Then, we have $T(c) = (T_1([c]_1), \ldots, T_{n_e}([c]_{n_e}))$ with $T_i([c]_i) = ([c]_i, \sigma^2_{c_i})$. It is the exponential parameterization as $\theta = (\mu, \mu^2 + \sigma^2) = \mathbb{E}[T_i([c]_i)]$ and $\theta = (\theta_1, \ldots, \theta_{n_e})$.

The inverse of the Fisher information matrix simply follows from the formula for the exponential family: $F(\theta)^{-1}$ is a block-diagonal matrix with block size 2 whose $i$-th block is $F_i(\theta_i)^{-1} = [\sigma^2_1, 2\mu_1\sigma^2_1; 2\mu_1\sigma^2_1, 4\mu_2^2\sigma^2_1 + 2\sigma^2_1]$. Since each block is a symmetric matrix of dimension 2, its eigen decomposition $F_i(\theta_i)^{-1} = \mathbf{E}\Lambda\mathbf{E}^T$ can be analytically obtained. With the decomposition, we have $F_i(\theta_i) = \mathbf{E}\Lambda^{-1}\mathbf{E}^T$ and $F(\theta)^{\frac{1}{2}} = \mathbf{E}\Lambda^{-\frac{1}{2}}\mathbf{E}^T$. Then, we have $F(\theta) = \text{diag}(F_1(\theta_1), \ldots, F_{n_e}(\theta_{n_e}))$ and $F(\theta)^{\frac{1}{2}} = \text{diag}(F_1(\theta_1)^{\frac{1}{2}}, \ldots, F_{n_e}(\theta_{n_e})^{\frac{1}{2}})$.

---

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

D. Restriction for the Range of $\theta$

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

For the categorical distribution, we set $\Theta = \{\theta_i^{\min}, \theta_i^{\max}\}_{i=1}^{m} \times \cdots \times \{\theta_n^{\min}, \theta_n^{\max}\}_{n=1}^{m}$, where $\theta_i^{\min} = \frac{1}{n_i(m_i-1)}$ and $\theta_i^{\max} = 1 - \frac{1}{n_i}$. 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 $\Theta$.

To force the parameter to live in $\Theta$, we apply the following steps after $\theta$ update:

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

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

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

For the integer variables, the parameters of the Gaussian distributions, $[\theta]_{i,1} := \mu_i$ and $[\theta]_{i,2} := \mu_i + \sigma_i^2$, 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 $\theta$ update we clip $[\theta]_{i,1}$ to $[\mu_i^{\min}, \mu_i^{\max}]$ and $[\theta]_{i,2}$ to $[\mu_i^{\min} + (\sigma_i^{\min})^2, [\mu_i^{\min} + (\sigma_i^{\max})^2]$. 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 $\alpha$, we ran ASNG on the selective squared error function with the varying $\alpha$ and initial step-size $\delta_0^\alpha$ for the step-sizes of $\epsilon_e = \{0.05, 0.0005\}$. Figure 1 shows the performance of ASNG with the different $\alpha$ settings. We observe that the hyper-parameter $\alpha$ is not sensitive for the performance, and ASNG reaches the target value for all settings.

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 $3 \times 10^{-4}$ 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
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 the local region. For this reason, the SSIM is also often used together with the PSNR, and more clearly assesses difference in the 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 $3 \times 3$ 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 $n_w = 60$ (# categorical variables) and $n_\theta = 214$ (dimension of $\theta$). A conceptual example of the symmetric CAE architecture and

---

**Figure 3.** The best cell structures discovered by ASNG-NAS in the classification task.
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 $n_c = 80$ ($\#$ categorical and integer variables) and $n_{\theta} = 174$ (dimension of $\theta$).

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

References


