Continual Learning via Bit Level Information Preserving (Appendix)

Yujun Shi[1] Li Yuan[1] Yunpeng Chen[2] Jiashi Feng[1]
[1]National University of Singapore [2] YITU Technology
{shi.yujun,yuanli}@u.nus.edu yunpeng.chen@yitu-inc.com elefjia@nus.edu.sg

1. Detailed Elaboration on Information Gain Estimation

In this section, we elaborate on **how to obtain Eqn.(8) mentioned in the main text**.

To start with, recall that we have introduced in the main text that $p(\theta_{0:t-1})$ and $p(\theta_{0:t})$ can be approximated by:

$$p(\theta_{0:t-1}) = \mathcal{N}(\theta^*_{0:t-1}, (tmF_{0:t-1})^{-\frac{1}{2}}) \tag{1}$$

and

$$p(\theta_{0:t}) = \mathcal{N}(\theta^*_{0:t}, ((t+1)mF_{0:t})^{-\frac{1}{2}}) \tag{2}$$

respectively. In addition, relation between $F_{0:t-1}$ and $F_{0:t}$ is:

$$F_{0:t} = \frac{tF_{0:t-1} + F_t}{t+1}. \tag{3}$$

Next, recall the definition of information gain on the quantized parameter $Q(\theta, N)$ is:

$$IG(Q(\theta, N), \mathcal{D}_t) = H(Q(\theta_{0:t-1}, N)) - H(Q(\theta_{0:t}, N)). \tag{4}$$

To calculate $IG(Q(\theta, N), \mathcal{D}_t)$, we first connect $H(Q(\theta, N))$ with $h(\theta)$, where $h(\theta)$ is differential entropy defined as:

$$h(\theta) = -\int p(\theta) \ln p(\theta) d\theta. \tag{5}$$

$H(Q(\theta, N))$ and $h(\theta)$ can be connected by the following lemma.

Lemma 1. *Consider a random variable X with density function $p(x)$ with support of $[-1 + \frac{1}{2^N}, 1 - \frac{1}{2^N}]$ and assume $p(x) \ln p(x)$ is Riemann integrable. If we quantize X by N bits with Q defined in the main text, then we have:*

$$H(Q(X, N)) \approx \frac{1}{\ln 2} h(X) + N - 1. \tag{6}$$

Proof. Firstly, we can divide the support of X, which is $[-1+\frac{1}{2^N}, 1-\frac{1}{2^N}]$, into $2^N - 1$ bins with equal length of $\delta_N = \frac{1}{2^{N-1}}$. Next, denote the center of the i-th bin as x_i and we have: $x_i = \frac{i}{2^{N-1}} - 1$. We write $P(Q(X, N) = x_i)$ with shorthand of $P_i(Q(X, N))$ and we have:

$$\begin{aligned} P_i(Q(X, N)) &= \int_{x_i - \frac{\delta_N}{2}}^{x_i - \frac{\delta_N}{2}} p(x) dx \quad \text{(property of probability density function)} \\ &\approx p(x_i)\delta_N \quad \text{(Mean Value Theorem as } \delta_N \to 0.) \end{aligned} \tag{7}$$

With the above Eqn. (7), we then rewrite the Shannon entropy of $Q(X, N)$ as:

$$\begin{aligned} H(Q(X,N)) &= -\sum_{i=1}^{2^N-1} P_i(Q(X,N)) \log_2 P_i(Q(X,N)) \\ &\approx -\sum_{i=1}^{2^N-1} p(x_i)\delta_N \log_2 p(x_i)\delta_N \quad \text{(applying (7))} \\ &= -\sum_{i=1}^{2^N-1} \delta_N p(x_i) \log_2 p(x_i) - \sum_{i=1}^{2^N-1} \delta_N p(x_i) \log_2 \delta_N \\ &\approx \frac{1}{\ln 2} h(X) + (N-1) \quad \text{(Riemann integrable as } \delta_N \to 0; \delta_N = \frac{1}{2^{N-1}}) \end{aligned} \tag{8}$$

□

Therefore, the information gain can be rewritten as:

$$IG(Q(\theta, N), \mathcal{D}_t) = \frac{1}{\ln 2}(h(\theta_{0:t-1}) - h(\theta_{0:t})). \tag{9}$$

With the posterior approximation Eqn. 1 and Eqn. 2, $h(\theta_{0:t-1})$ and $h(\theta_{0:t})$ can be calculated in closed-form by $\frac{1}{2} - \frac{1}{2}\ln(2\pi m t F_{0:t-1})$ and $\frac{1}{2} - \frac{1}{2}\ln(2\pi m(t+1)F_{0:t})$ respectively. That means that $IG(Q(\theta, N), \mathcal{D}_t)$ can be approximated by:

$$\begin{aligned} IG(Q(\theta,N),\mathcal{D}_t) &= \frac{1}{\ln 2}(h(\theta_{0:t-1}) - h(\theta_{0:t})) \\ &\approx \frac{1}{2\ln 2} \ln \frac{m(t+1)F_{0:t}}{mtF_{0:t-1}} \\ &= \frac{1}{2}\log_2 \frac{tF_{0:t-1} + F_t}{tF_{0:t-1}}. \end{aligned} \tag{10}$$

In this way, we derive the Eqn.(8) mentioned in the main text.

2. Implementation Details

2.1. Parameter Range

To perform parameter quantization, we normally have to constraint parameters to be within a certain interval. In the main text, to more conveniently elaborate our method, we assume model parameters are constrained within $[-1, 1]$ and then quantize them accordingly. However, in real scenarios, parameters of different layers normally distributed in different manner. Therefore, we constraint model parameters in the interval of $[-\frac{C}{\sqrt{n}}, \frac{C}{\sqrt{n}}]$, where C is a pre-defined hyper-parameter, and $n = num\_input\_dimension$ for fully connected layers and $n = kernel\_size \times kernel\_size \times num\_input\_channel$ for convolution layers. C is set to be 20 for the mini-ImageNet experiments. and 6 for all the other experiments. This strategy is inspired by previous literature on model parameter initialization [3, 2].

2.2. Trainig Strategy

According to the main text, when trainig on a new task, Straight Through Estimator (STE) is used to perform quantization aware training. However, since we quantize the model to 20 bits in all our experiments and the difference is insignificant between whether or not using STE. Therefore, we do not quantize parameters during training and only quantize parameters before doing bit freezing.

2.3. Information Gain

Empirically, we find that using the following formulation:

$$IG(Q(\theta, N), \mathcal{D}_t) = \frac{1}{2}\log_2 \frac{tF_{0:t-1} + F_t}{(t+1)F_{0:t-1}} \tag{11}$$

which is slightly different from Eqn. (10), can produce better results. Therefore, we adopt this slightly modified version to estimate information gain.

For more details, please refer to our released implementation.

3. More Detailed Experiment Results

3.1. More Detailed 20-split mini-ImageNet Results

In this section, we add comparison of model size for the 20-split mini-ImageNet experiment in Tab. 1. All models are variants of AlexNet. From the table, we show that the model size of all methods are kept approximate to ensure fair comparisons.

3.2. Experiments on ResNet

In addition, to demonstrate the effectiveness of our method on models with Batch Norm layers and residual connections, we evaluate our method on ResNet-18 model with 20-split mini-ImageNet. The result is shown in Tab. 2.

3.3. Discussion and Ablation Study of F_0

As mentioned in the main text, F_0 is an important hyper-parameter for our method. According to our method, smaller F_0 corresponds to more bits being frozen when learning subsequent tasks and vice versa. Therefore, by setting F_0 to be small, our method tends to suffer less forgetting while being less capable of adapting new tasks and vice versa. We ablate using different F_0 with our 20-split mini-ImageNet experiment in Tab. 3. Through this ablation study, we can see that setting $F_0 = 5 \times 10^{-16}$ can best balance between preventing forgetting (BWT) and adapting new tasks (ACC).

Table 1. Experiment Results on 20-Split mini-ImageNet. RB is size of replay buffer, MS is model size. Results are averaged over 5 runs; mean $\pm$ std is reported. Results denoted by (†) are provided by [1].

Methods	BWT (%)	ACC (%)	RB (MB)	MS (MB)
LWF	-45.93 $\pm$ 1.05	29.30 $\pm$ 0.64	-	104.1
A-GEM†	-15.23 $\pm$ 1.45	52.43 $\pm$ 3.10	110.1	102.6
HAT†	-0.04 $\pm$ 0.03	59.45 $\pm$ 0.05	-	123.6
ACL†	-3.71 $\pm$ 1.31	57.66 $\pm$ 1.44	-	113.1
ACL-R†	0.00 $\pm$ 0.00	62.07 $\pm$ 0.51	8.5	113.1
BLIP	-1.05 $\pm$ 0.42	65.69 $\pm$ 0.87	-	104.78

Table 2. Experiment Results on 20-split mini-ImageNet with ResNet-18 and AlexNet. MS is Model Size. Arch is model architecture. Results are averaged over 5 random seeds; mean $\pm$ std is reported.

Methods	Arch	BWT (%)	ACC (%)	MS (MB)
BLIP	AlexNet	-1.05 $\pm$ 0.42	65.69 $\pm$ 0.87	104.78
BLIP	ResNet	-0.72 $\pm$ 0.46	65.94 $\pm$ 1.36	42.76

Table 3. Ablation study on F_0 with AlexNet and 20-split mini-ImageNet. All results are averaged over 5 random seeds. mean $\pm$ std is reported.

F_0	BWT (%)	ACC (%)
1×10^{-14}	-3.60 $\pm$ 0.59	65.23 $\pm$ 0.87
5×10^{-15}	-3.01 $\pm$ 0.76	65.10 $\pm$ 1.16
1×10^{-15}	-1.55 $\pm$ 0.53	65.76 $\pm$ 0.81
5×10^{-16}	-1.05 $\pm$ 0.42	65.69 $\pm$ 0.87
1×10^{-16}	-0.26 $\pm$ 0.32	64.78 $\pm$ 0.96
5×10^{-17}	-0.17 $\pm$ 0.29	64.29 $\pm$ 0.82

4. More Visualization on Bit Freezing Process

In this section, we provide more results on the bit freezing process visualization on different layers of our PPO agent in Fig 1. From the results, the similar phenomenon of more bits getting frozen as mentioned in the main text can be observed.

From the heatmap, we can also see that for some parameters, no bits are frozen throughout the whole continual learning process. This means that no information gain is observed on these parameters. This phenomenon might correspond to the nature of ReLU activation function, whose gradient is 0 if the input is negative. However, it is beyond the scope of this work and we do not provide further discussion on this phenomenon.

In addition, this bit freezing process is further visualized in histogram in Fig. 2.

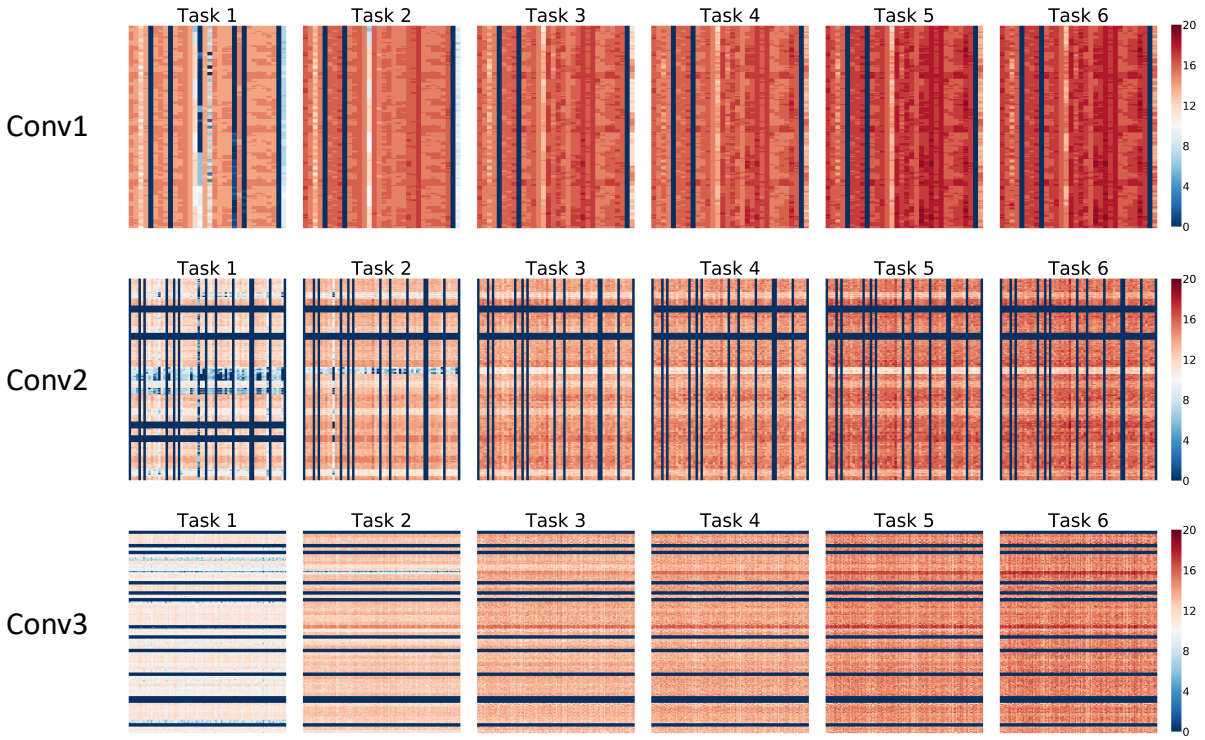

Figure 1. **Bit freezing visualization** We visualized bit freezing process as continual learning proceeds on all the convolution layers of our PPO agent. Each pixel in a heat map represents the number of frozen bits of the corresponding entry (parameter) in weight matrix of a convolution layer. Each parameter has a total of 20 bits. From darker blue to darker red denotes more bits being frozen.

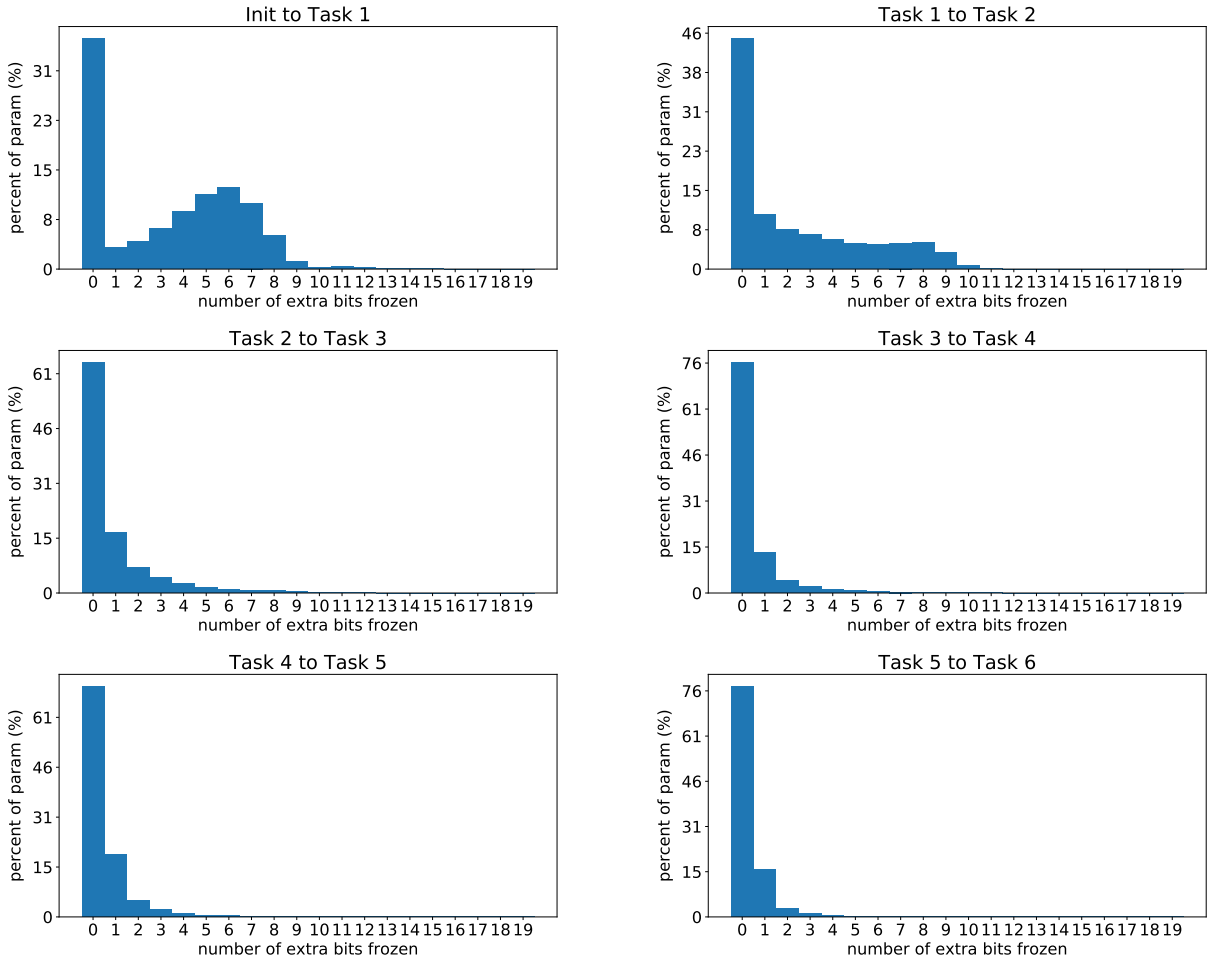

Figure 2. **Bit freezing visualization in histogram** Each histogram in this figure shows what percentage of parameters have how many extra bits frozen after learning one task.

5. Additional Acknowledgement

The authors would also like to thank Jun Hao Liew, Kuangqi Zhou, Minda Hu, Zihang Jiang, Bingyi Kang and all reviewers for helpful feedback and discussions.

References

[1] Sayna Ebrahimi, Franziska Meier, Roberto Calandra, Trevor Darrell, and Marcus Rohrbach. Adversarial continual learning. In *The European Conference on Computer Vision (ECCV)*, 2020. 3

[2] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010. 2

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 2