A Bayesian Optimization Framework for Neural Network Compression - Supplementary Material

Xingchen Ma*[a], Amal Rannen Triki * †[a], Maxim Berman[a], Christos Sagonas[b], Jacques Cali ‡[c], and Matthew B. Blaschko[a]

[a]ESAT-PSI, KU Leuven, Belgium
[b]Onfido, London, UK
[c]Blue Prism, London, UK

In this supplementary material, we provide more analysis that support the method presented in the main paper. Section A shows that controlling the L_2 distance allows a control over the risk of the compressed function as claimed in Section 3.1. Section B justifies the number of samples we used to estimate the function distance introduced in section 3.1. Section C shows a visualization of the proposed acquisition function and how to select the parameter γ.

A. Function distance and generalization

In this section, we show that compression with the L_2 knowledge distillation objective can lead to a generalization bound on the performance of the compressed network as a function of the performance of the uncompressed network. To do so, we assume that the loss used in evaluating the risk is Lipschitz-continuous.

Definition 1 (Lipschitz-continuity)**.** *A function $f : \mathcal{X} \mapsto \mathcal{Y}$ is K-Lipshitz continuous if:*

$$\forall x_1, x_2 \in \mathcal{X}, \|f(x_1) - f(x_2)\|_{\mathcal{Y}} \leq K\|x_1 - x_2\|_{\mathcal{X}}, \quad (1)$$

where $\|.\|_{\mathcal{X}}$ and $\|.\|_{\mathcal{Y}}$ are norms on the spaces $\mathcal{X}$ and $\mathcal{Y}$.

Definition 2 (Generalization error)**.** *Let f be a function mapping an input space $\mathcal{X}$ to a target space Y, and $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ an i.i.d. sample drawn from a distribution P over $\mathcal{X} \times \mathcal{Y}$. We define:*

- *the risk as $\mathcal{R}(f) = \mathbb{E}_{(x,y)\sim P}\left[\ell(f(x), y)\right]$,*
- *the empirical risk as $\hat{\mathcal{R}}_n(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i)$,*
- *the generalization error as the gap between the empirical risk and the true risk:*

$$\mathcal{E}_n(f) = \mathcal{R}(f) - \hat{\mathcal{R}}_n(f) \quad (2)$$

Proposition 1. *Let f^* be the function encoded by a neural network optimized to minimize the empirical risk, such that f^* achieves a small generalization error, and $\tilde{f}_\theta$ the function encoded by a compressed version of this network. Then, if the loss function ℓ is K-Lipschitz continuous, controlling the function distance:*

$$\mathcal{L}(\tilde{f}_\theta, f^*) := \mathbb{E}_{x\sim P}(\|\tilde{f}_\theta(x) - f^*(x)\|_2^2) = \|f^* - \tilde{f}_\theta\|_{2,P}^2 \quad (3)$$

allows a control over the true risk of $\tilde{f}_\theta$.

Proof. For simplicity, we consider a target space included in $\mathbb{R}^d$ and we use the ℓ_2 vector norm on this space. If ℓ is K-Lipschitz continuous with respect to its first input, then we can write:

$$\forall x \in \mathcal{X}, |\ell(f^*(x), y) - \ell(\tilde{f}_\theta(x), y)| \leq K\|f^*(x) - \tilde{f}_\theta(x)\|_2 \quad (4)$$

Therefore, using Jensen inequality, we have:

$$(\mathcal{R}(f^*) - \mathcal{R}(\tilde{f}_\theta))^2 \leq \mathbb{E}_P[|\ell(f^*(x), y) - \ell(\tilde{f}_\theta(x), y)|^2] \quad (5)$$

$$\leq K^2\mathbb{E}_P(\|\tilde{f}_\theta(x) - f^*(x)\|_2^2) \quad (6)$$

$$\Leftrightarrow |\mathcal{R}(f^*) - \mathcal{R}(\tilde{f}_\theta)| \leq K\|f^* - \tilde{f}_\theta\|_{2,P} \quad (7)$$

To sum up, we have:

$$\mathcal{R}(\tilde{f}_\theta) \leq \mathcal{R}(f^*) + K\|f^* - \tilde{f}_\theta\|_{2,P} \quad (8)$$

$$\leq \hat{\mathcal{R}}_n(f^*) + \mathcal{E}_n(f^*) + K\|f^* - \tilde{f}_\theta\|_{2,P}. \quad (9)$$

Therefore, if f^* is a minimizer of the empirical risk such that the generalization error is small, then it is sufficient to control the distance $\mathcal{L}(\tilde{f}_\theta, f^*)$ to achieve a control over the risk of $\tilde{f}_\theta$. □

*Authors with equal contribution
†This author is currently affiliated with Deepmind.
‡Contribution to this research project was entirely made while this co-author was at Onfido, UK.

B. Distribution of the function norm estimates

In order to use the function norm in Bayesian optimization, we need to check that the sample mean used to estimate $\mathcal{L}(\tilde{f}_\theta, f^*)$ follows a Gaussian distribution. Figure 1 and 2 show the Q-Q (quantile-quantile) and density plots for the distribution of the estimated function norm under different sampling size respectively. These experiments are realized with a Resnet18 and the corresponding compressed model obtained with random compression parameters. These figures are generated with 500 norms totally. Both Figure 1 and Figure 2 show that a small sample size (less than 20) results in a distribution skewed to the left. When the sampling size is larger than 50, the observed empirical distribution is well centered. In our experiments, we consistently use 50 as our sampling size, as it balances the computation cost and the requirements of a GP model.

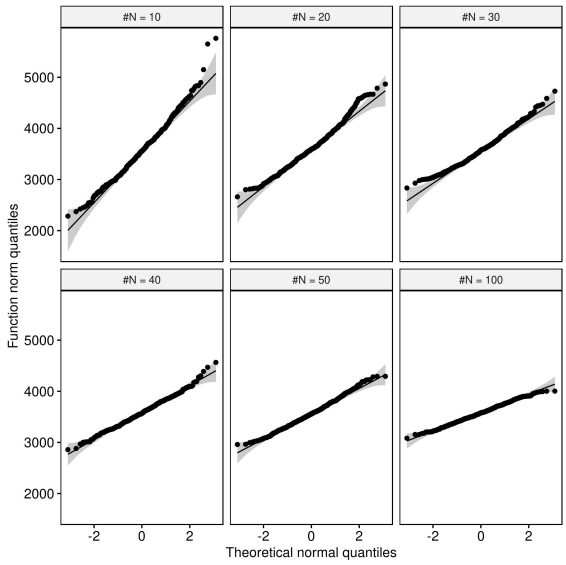

Figure 1. Q-Q plot for the norm under different sampling size

To complete Section 4.1 of the paper, Figure 3 shows the relationship between the estimated function norm and the top1 error rate for different layers in Resnet101.

C. Visualization of our acquisition function

Figure 4 shows a visualization of our acquisition function to the *sinone* function. The objective function is shown in the top row, and the acquisition function is shown in the lower row. The blue vertical line shows the current best point and the red vertical line shows the next exploring point. In this optimization, we add Gaussian noise (sigma equals to 0.03) to *sinone* function. [1]

[1] Animation compatible with Adobe Reader.

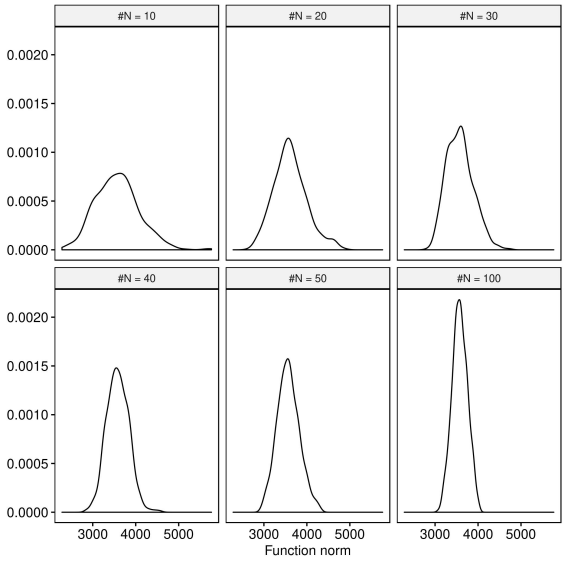

Figure 2. Density plot of the estimated norm with different sampling sizes

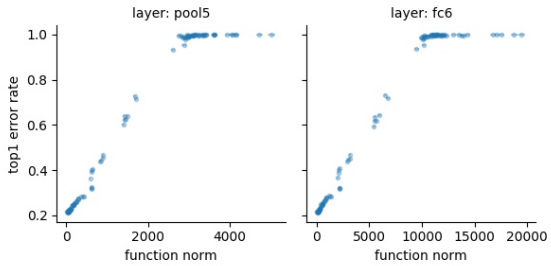

Figure 3. Estimated norm vs Top-1 error rate in Resnet101

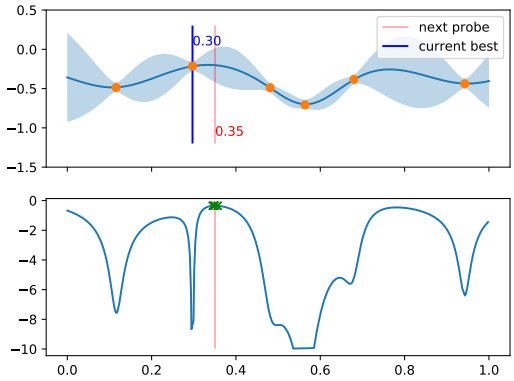

Figure 4. Visualization of our acquisition function for *sinone* function.

C.1. Selection of γ

To select γ in our VGG experiments, we optimized accuracy given a fixed compression ratio budget. We may write

this as a constrained optimization problem:

$$\arg\min_{\theta} \mathcal{L}(\tilde{f}_\theta, f^*), \quad \text{s.t. } R(\tilde{f}_\theta, f^*) \leq B. \tag{10}$$

Taking the Lagrangian of this constrained optimization problem, we can perform a minimax procedure using Bayesian optimization to solve a minimization procedure over θ, and then alternating with a maximization over the Lagrange multiplier from the constraint. As multiple Bayesian optimization iterations are used, we implemented a caching mechanism that simply reweights the linear combination of $\mathcal{L}(\tilde{f}_\theta, f^*)$ and $R(\tilde{f}_\theta, f^*)$ computed in previous iterations by the current value of the Lagrange multiplier.