Minimal Random Code Learning: Getting Bits Back from Compressed Model Parameters

Marton Havasi
Department of Engineering
University of Cambridge
mh740@cam.ac.uk

Robert Peharz
Department of Engineering
University of Cambridge
rp587@cam.ac.uk

José Miguel Hernández-Lobato
Department of Engineering
University of Cambridge,
Microsoft Research,
Alan Turing Institute
jmh233@cam.ac.uk

Abstract

While deep neural networks are a highly successful model class, their large memory footprint puts considerable strain on energy consumption, communication bandwidth, and storage requirements. Consequently, model size reduction has become an utmost goal in deep learning. Following the classical bits-back argument, we encode the network weights using a random sample, requiring only a number of bits corresponding to the Kullback-Leibler divergence between the sampled variational distribution and the encoding distribution. By imposing a constraint on the Kullback-Leibler divergence, we are able to explicitly control the compression rate, while optimizing the expected loss on the training set. The employed encoding scheme can be shown to be close to the optimal information-theoretical lower bound, with respect to the employed variational family. On benchmarks LeNet-5/MNIST and VGG-16/CIFAR-10, our approach yields the best test performance for a fixed memory budget, and vice versa, it achieves the highest compression rates for a fixed test performance.

1 Introduction

Traditional approaches to model compression usually rely on three main techniques: pruning, quantization and coding. For example, Deep Compression (Han et al., 2016) proposes a pipeline employing all three of these techniques in a systematic manner. From an information-theoretic perspective, the central routine is *coding*, while pruning and quantization can be seen as helper heuristics to reduce the entropy of the empirical weight-distribution, leading to shorter encoding lengths (Shannon, 1948). Also, the recently proposed Bayesian Compression (Louizos et al., 2017) falls into this scheme, despite being motivated by the so-called *bits-back* argument (Hinton & Van Camp, 1993) which theoretically allows for higher compression rates.[1] While the bits-back argument certainly motivated the use of variational inference in Bayesian Compression, the downstream encoding is still akin to Deep Compression (and other approaches). In particular, the variational distribution is merely used to derive a *deterministic set of weights*, which is subsequently encoded with Shannon-style coding. This approach, however, does not fully exploit the coding efficiency postulated by the bits-back argument.

[1] Recall that the bits-back argument states that, assuming a large dataset and a neural network equipped with a weight-prior p, the effective coding cost of the network weights is $\mathrm{KL}(q||p) = \mathbb{E}_q[\log \frac{q}{p}]$, where q is a variational posterior. However, in order to realize this effective cost, one needs to encode *both* the network weights and the training targets, while it remains unclear whether it can also be achieved for network weights alone.

Workshop on Compact Deep Neural Network Representation with Industrial Applications (NIPS 2018), Montréal, Canada.

In this paper, we step aside from the pruning-quantization pipeline and propose a novel coding method which approximately realizes bits-back efficiency. In particular, we refrain from constructing a deterministic weight-set but rather encode a *random weight-set* from the full variational posterior. This is fundamentally different from first drawing a weight-set and subsequently encoding it – this would be no more efficient than previous approaches. Rather, the coding scheme developed here is allowed to pick a random weight-set which can be cheaply encoded. By using results from Harsha et al. (2010), we show that such a coding scheme always exists and that the bits-back argument indeed represents a theoretical lower bound for its coding efficiency. Moreover, we propose a practical scheme which produces an approximate sample from the variational distribution and which can indeed be encoded with this efficiency. Since our algorithm learns a distribution over weight-sets and derives a random message from it, while minimizing the resulting code length, we dub it *Minimal Random Code Learning* (MIRACLE).

2 Motivation

All preceding works (Han et al., 2015; Louizos et al., 2017; Reagen et al., 2018; Chen et al., 2015) essentially use the following coding scheme, or a (sometimes sub-optimal) variant of it. After a deterministic weight-set $\boldsymbol{w}^*$ has been obtained, involving potential pruning and quantization techniques, one interprets $\boldsymbol{w}^*$ as a sequence of i.i.d. variables and assumes the coding distribution (i.e. a dictionary) $p'(w) = \frac{1}{N}\sum_{i=1}^{N} \delta_{w_i^*}$, where δ_x denotes the Dirac delta at x. According to Shannon's source coding theorem (Shannon, 1948), $\boldsymbol{w}^*$ can be coded with no less than $N\mathrm{H}[p']$ nats (H denotes the Shannon entropy), which is asymptotically achieved by Huffman coding (Huffman, 1952), like in Han et al. (2016). However, note that the Shannon lower bound can also be written as

$$N\mathrm{H}[p'] = -\sum_{i=1}^{N} \log p'(w_i^*) = -\log p'(\boldsymbol{w}^*) = \int \delta_{\boldsymbol{w}^*}(\boldsymbol{w}) \log \frac{\delta_{\boldsymbol{w}^*}(\boldsymbol{w})}{p'(\boldsymbol{w})} d\boldsymbol{w} = \mathrm{KL}(\delta_{\boldsymbol{w}^*}||p'), \quad (1)$$

where we set $p'(\boldsymbol{w}) = \prod_i p'(w_i)$. Thus, these Shannon-style coding schemes are in some sense optimal, when the variational family is *restricted to point-measures*, i.e. deterministic weights. By extending the variational family to comprise more general distributions q, the coding length $\mathrm{KL}(q||p)$ could potentially be drastically reduced. In the following, we develop one such method which exploits the uncertainty represented by q in order to encode a *random* weight-set with short coding length.

3 Minimal Random Code Learning

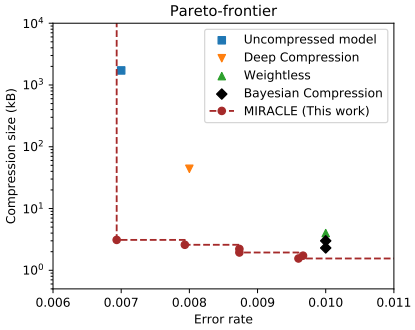
Consider the scenario where we want to train a neural network but our memory budget is constrained. As illustrated in the previous section, a variational approach offers – in principle – a simple and elegant solution. Now, similar to Louizos et al. (2017), we first fix a suitable network architecture, select an encoding distribution p and a parameterized variational family q_ϕ for the network weights $\boldsymbol{w}$. We consider, however, a slightly different variational objective related to the β-VAE (Higgins et al., 2017) in order to be able to constrain the compression size using the penalty factor β:

$$\mathcal{L}(\phi) = \underbrace{\mathbb{E}_{q_\phi}[\log p(\mathcal{D}|\boldsymbol{w})]}_{\text{negative loss}} - \beta \underbrace{\mathrm{KL}(q_\phi||p)}_{\text{model complexity}} . \quad (2)$$

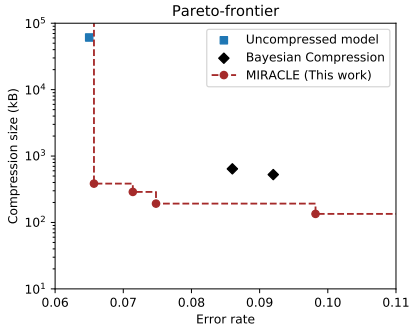
This objective directly reflects our goal of achieving both a good training performance (loss term) and being able to represent our model with a short code (model complexity), at least according to the bits-back argument. After obtaining q_ϕ by maximizing (2), a weight-set drawn from q_ϕ will perform comparable to a deterministically trained network, since the variance of the negative loss term will be comparatively small to the mean. , and since the KL term regularizes the model. Thus, our declared goal is to *draw a sample from* q_ϕ such that this sample can be *encoded as efficiently as possible*. It turns out that the expected message length $\mathrm{E}[|M|]$ that allows for sampling q_ϕ is bounded by the mutual information between the data D and the weights $\boldsymbol{w}$ (Harsha et al., 2010; Havasi et al., 2018):

$$\mathbb{E}_D[|M|] \geq \mathrm{H}[M] \geq \mathrm{I}[D:M] \geq \mathrm{I}[D:\boldsymbol{w}] = \mathbb{E}_D[\mathrm{KL}(q_\phi||p)] . \quad (3)$$

Harsha et al. (2010) provide a constructive proof that this lower-bound can be well approximated using a variant of rejection sampling. However, this algorithm is in fact intractable, because it



(a) LeNet-5 on MNIST



(b) VGG-16 on CIFAR-10

Figure 1: The error rate and the compression size for various compression methods. The Pareto-frontier is denoted for MIRACLE (This work). Lower left is better.

requires keeping track of the acceptance probabilities over the whole sample domain. We propose a method to produce an approximate sample from q_ϕ that can be cheaply encoded. First, $K = \exp(\mathrm{KL}(q_\phi||p))$ samples are drawn from p, using the shared random generator. Subsequently, we craft a discrete proxy distribution $\tilde{q}$, which has support only on these K samples, and where the probability mass for each sample is proportional to the importance weights $a_k = \frac{q_\phi(\boldsymbol{w}_k)}{p(\boldsymbol{w}_k)}$. Finally, we draw a sample from $\tilde{q}$ and return its index k^*. Since any number $0 \leq k^* < K$ can be easily encoded with $\mathrm{KL}(q_\phi||p)$ nats, we achieve our aimed coding efficiency. *Decoding* the sample is easy: simply draw the $k^{*\text{th}}$ sample $\boldsymbol{w}_{k^*}$ from the shared random generator (e.g. by resetting the random seed). While this algorithm is remarkably simple and easy to implement, it can be shown that it produces a close-to unbiased sample from q_ϕ (Chatterjee & Diaconis, 2018; Havasi et al., 2018).

Furthermore, an immediate caveat is that the number K of required samples grows exponentially in $\mathrm{KL}(q_\phi||p)$, which is clearly infeasible for encoding a practical neural network. To deal with this issue, the weights are randomly split into groups each with a small, fixed allowance of nats such that drawing $\exp(\mathrm{KL}(q_{\phi_{\text{block}}}||p_{\text{block}})) \approx 10^6$ samples can be done efficiently.

4 Experimental Results

The experiments[2] were conducted on two common benchmarks, LeNet-5 on MNIST and VGG-16 on CIFAR-10, using a Gaussian distribution with diagonal covariance matrix for q_ϕ. As baselines, we used three recent state-of-the-art methods, namely Deep Compression (Han et al., 2016), Weightless encoding (Reagen et al., 2018) and Bayesian Compression (Louizos et al., 2017). The performance of the baseline methods are quoted from their respective source materials.

We see that MIRACLE is Pareto-better than the competitors: for a given test error rate, we achieve better compression, while for a given model size we achieve lower test error (Figure 1).

5 Conclusion and Future Work

In this paper we followed through the philosophy of the bits-back argument for the goal of coding model parameters. Our algorithm is backed by solid recent information-theoretic insights, yet it is simple to implement. We demonstrated that it outperforms the previous state-of-the-art.

An important question remaining for future work is how efficient MIRACLE can be made in terms of memory accesses and consequently for energy consumption and inference time. There lies clear potential in this direction, as any single weight can be recovered by its group-index and relative index within each group. By smartly keeping track of these addresses, and using pseudo-random generators as algorithmic lookup-tables, we could design an inference machine which is able to directly run our compressed models, which might lead to considerable savings in memory accesses.

[2] The code is publicly available at `https://github.com/cambridge-mlg/miracle`

Acknowledgements

We want to thank Christian Steinruecken, Olivér Janzer, Kris Stensbo-Smidt and Siddharth Swaroop for their helpful comments.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 797223 — HYBSPN. Marton Havasi acknowledges funding from EPSRC.

References

S. Chatterjee and P. Diaconis. The sample size required in importance sampling. *The Annals of Applied Probability*, 28(2):1099–1135, 2018.

W. Chen, J. Wilson, S. Tyree, K. Weinberger, and Y. Chen. Compressing neural networks with the hashing trick. In *Proceedings of ICML*, pp. 2285–2294, 2015.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1135–1143, 2015.

Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. *International Conference on Learning Representations (ICLR)*, 2016.

P. Harsha, R. Jain, D. McAllester, and J. Radhakrishnan. The communication complexity of correlation. *IEEE Transactions on Information Theory*, 1(56):438–449, 2010.

Marton Havasi, Robert Peharz, and José Miguel Hernández-Lobato. Minimal random code learning: Getting bits back from compressed model parameters. *arXiv preprint arXiv:1810.00440*, 2018.

I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. Beta-VAE: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

G. E. Hinton and D. Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pp. 5–13. ACM, 1993.

David A Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, 1952.

C. Louizos, K. Ullrich, and M. Welling. Bayesian compression for deep learning. In *Proceedings of NIPS*, pp. 3288–3298, 2017.

B. Reagen, U. Gupta, R. Adolf, M. M. Mitzenmacher, A. M. Rush, G.-Y. Wei, and D. Brooks. Weightless: Lossy weight encoding for deep neural network compression. *International Conference on Machine Learning*, 2018.

C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3): 379–423, 1948.