Sparsity Increases Uncertainty Estimation in Deep Ensemble

Uyanga Dorjsembe*, Ju Hong Lee*, Bumghi Choi** and Jae Won Song***
* Dept. of Computer science, Inha University
** QHedge Inc
*** Value finders Inc
uyangamelo@gmail.com, juhong@inha.ac.kr, bgchoi666@gmail.com, jwsong@valuefinders.co.kr

Abstract

Deep neural networks have achieved almost human-level results in various tasks and have become popular in the broad artificial intelligence domains. Uncertainty estimation is an on-demand task caused by the black-box point estimation behavior of deep learning. The deep ensemble provides increased accuracy and estimated uncertainty; however, linearly increasing the size makes the deep ensemble unfeasible for memory-intensive tasks. To address this problem, we used model pruning and quantization with a deep ensemble and analyzed the effect in the context of uncertainty metrics. We empirically showed that the ensemble members' disagreement increases with pruning, making models sparser by zeroing irrelevant parameters. Increased disagreement implies increased uncertainty, which helps in making more robust predictions. Accordingly, an energy-efficient compressed deep ensemble is appropriate for memory-intensive and uncertainty-aware tasks.

1. Introduction

Deep neural networks make an overconfident [1], black-box point estimation because the neural network parameters comprise scalar matrices. Even in unseen data that come from another distribution, a neural network incorrectly predicts with high confidence. In deep learning, we require a scalable and straightforward uncertainty estimation method. Monte Carlo (MC) dropout uses a standard regularization technique dropout [2] with a Bernoulli mask in inference time by sampling to measure model uncertainty [3].

Ensemble [4] is a well-known method for improving the accuracy of machine learning tasks. Another sample-based uncertainty measuring method is the deep ensemble [5], which consists of randomly initialized, independently trained, and shared-architecture neural networks. The results showed excellent uncertainty estimation while increasing the accuracy [6]. Randomly initialized networks are close to each other in initialization, but they move further away in the function space during training. It is assumed that such function space diversity is the main reason for the excellent performance of the deep ensemble [7].

However, cost remains the main challenge for the deep ensemble as the training, test time and ensemble size linearly increase in parallel with the increase in ensemble members. The deep ensemble is not applicable in memory-intensive tasks because of its increased size.

In deep learning, model compression techniques [8] dramatically reduce the model size compared with relatively insignificant to no accuracy degradation. Model pruning increases the model sparsity by zeroing the irrelevant parameters. Low-rank factorization decomposes the matrix tensor by estimating the informative parameters. Knowledge distillation learns a distilled model and trains a more compact network to reproduce a more extensive or ensemble network. Model quantization converts the parameter floating-point into 8 bits or less.

2. Related Work

To the best of our knowledge, the majority of the studies aim to increase uncertainty quantification while decreasing the cost by leveraging knowledge distillation [9-12]. We assume that using the deep ensemble with knowledge distillation might lose the multimodality, which is the main advantage of its successful results. Correspondingly, because the baseline approach is a relatively simple and scalable method without any change in the model architecture, the compression method should also be as simple and scalable as the deep ensemble.

We already know that model pruning [13–17] has a trade-off between accuracy and size. This study analyzes the tradeoffs between the size, accuracy, and uncertainty estimation using a simple strategy comprising pruning and quantization with a deep ensemble (called a compressed deep ensemble).

Pruning sacrifices the long tail part of the training dataset. It is a tricky part of the dataset to be predicted by both humans and models because of noise-contaminated, multiple, or wrongly labeled objects [17]. If pruning identifies such instances, it is a favorable feature from an uncertainty perspective. Moreover, sparse initialization helps achieve greater diversity among initialization time units [18]. Thus, we believe that model pruning increases a non-zero parameter weight, and such an increased weight could positively affect the function space diversity.

3. Proposed Method

3.1 Problem Setup and High-Level Summary

We assume that the training dataset comprises of N data points **{x, y}**, where $\mathbf{x} \in \mathbf{R}^{d}$ represents d-dimensional features. For classification problems, the label $\mathbf{y} \in \{1, \ldots, K\}$ is assumed to be one of the K classes. We used a neural network to assign the prediction probability $\mathbf{p_{\theta}(y|\ x)}$ over the labels, where **θ** denotes the neural network's parameters. In this study, we created an ensemble that comprises **M** number of independent neural networks.

3.2 Deep Ensemble

The deep ensemble consists of three simple recipes to measure the uncertainty. Selecting a proper score rule as a training criterion is the first recipe. Independent, randomly initialized, shared architecture is the second recipe. They introduced adversarial training to smooth the predictive distributions, but it was not as effective as the abovementioned two recipes.

Popular loss functions meet the condition for the proper scoring rule; negative log-likelihood (NLL) for both regression and classification, root mean square error (RMSE) for regression, and Brier score for classification are all examples of proper scoring rules. Because a base learner trained on a bootstrap sample sees only 63% unique data points, instead of using the bootstrap, the entire dataset is favorable [19], even though the deep ensemble was theoretically motivated by the bootstrap. Ovadia et al. [6], [20–23] demonstrated that a deep ensemble consistently provides more reliable and practically useful uncertainty estimates. In the deep ensemble, all members are treated as a uniformly weighted mixture model, and the final prediction is the average of the predictions.

3.3 Model Pruning and Quantization

Model pruning decreases the model size by increasing the sparsity of the parameters by removing irrelevant parameters. In conventional pruning, the trained model is iteratively pruned until it attains the desired sparsity, and after pruning, the model with pruned parameters is re-trained. The neural network model has a function $\mathbf{f(x; W)}$, then pruned model has a function $\mathbf{f(x; W \odot M)}$, where $\mathbf{M} \in \{0, 1\}$ is a pruning mask to remove the parameter. Practically, the pruned parameters of **W** are set to zero or entirely removed. Using zero weights, element-wise multiplication is redundant. As a result, the computational footprint also decreases. Blalock et al. [15] empirically showed that a large, sparse model performs better than a small dense model. Model quantization is a model compression technique that converts the parameter representation from floating-point 32 bits to 8 bits or fewer. Quantization is complementary to pruning techniques and is harmless for accuracy. Thus, we pruned and quantized the deep ensemble to decrease the linearly increasing size.

3.4 Evaluation Metrics

We evaluated the standard deep ensemble as a baseline model and estimated accuracy, NLL, Brier score, zipped model size as a memory footprint, and compression ratio in the in-distribution data.

There is no ground truth in the out-of-distribution data; thus, we cannot measure the accuracy in these data. Furthermore, there was no standard uncertainty metric. Hence, we evaluated entropy, disagreement, and the confidence curve proposed in [6] as uncertainty metrics. As proposed in [5], ensemble disagreement counts the Kullback–Leibler divergence between the member network prediction and the mean prediction.

4. Experimental Setup

We first trained the deep ensemble using the MNIST [24] handwritten digits dataset. The hyperparameters and model summary are listed in Table 1. Subsequently, we applied a magnitude-based iterative pruning and 8-bit quantization for each member of deep ensembles using various sparsity levels: 25%, 50%, 75%, and 95% using the TensorFlow [25] Model Optimization Toolkit. We used 10,000 test samples from the MNIST dataset and 18,724 test samples from the NotMNIST [26] alphabet dataset as out-of-distribution data to evaluate the uncertainty.

<Table 1> Model summary and hyperparameters

Convolution layer 1 parameters	280 (26, 26, 28)
Convolution layer 2 parameters	22432 (9, 9, 32)
Fully connected layer parameters	5130 (512)
Training, re-training epochs	20, 5
Optimizer	Adam
Validation split	0.1
Ensemble size M	5

5. Experimental Results

Table 2 presents the results of the evaluation of the MNIST dataset. Accuracy was increased by 0.04%, and the entire ensemble size decreased by 4× by pruning with 25% sparsity and quantization. However, pruning with 50–75% sparsity and quantization slightly reduces the accuracy by 0.01% and 0.05%, while decreasing the ensemble size by 5× and 8×. Pruning with 95% sparsity degrades accuracy by ~1%.

Figure 1 presents the confidence curves of the in-distribution and out-of-distribution data. Confidence in the MNIST (Figure 1(a)) is higher than that of NotMNIST (Figure 1(b)), and if the confidence threshold τ=90%, the ensemble classifies more than 90% of the MNIST samples, but only 16–32% of the NotMNIST samples. An increase in sparsity was associated with a decrease in confidence in both test datasets.

The evaluation results of the out-of-distribution data are presented in Figure 2. All uncertainty metrics increased as sparsity increased; thus, pruning and quantization made the

<Table 2> Evaluation results of the MNIST dataset

		Accuracy ↑	NLL ↓	Brier score↓	Non-zero weights	Memory footprint ↓ (KBs)	Compression ratio ↑
Deep ensemble		99.3	0.027975	0.011383	138,929	518.06	1
Compressed deep ensemble	25% pruned	99.34	0.026542	0.010392	104,490	127.48	4
	50% pruned	99.29	0.023002	0.010677	69,775	103.27	5
	75% pruned	99.25	0.021455	0.011344	35,060	68.89	8
	95% pruned	98.03	0.068589	0.032339	7,290	29.08	18

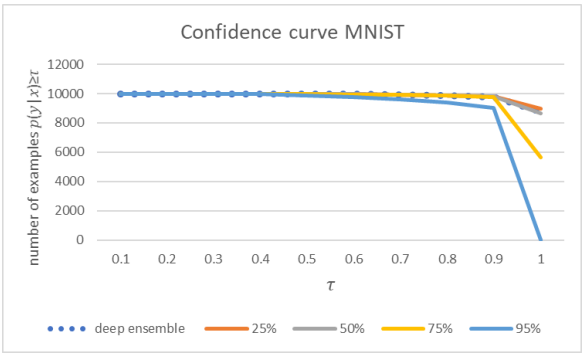

Figure 1(a) Confidence curve of the in-distribution data.

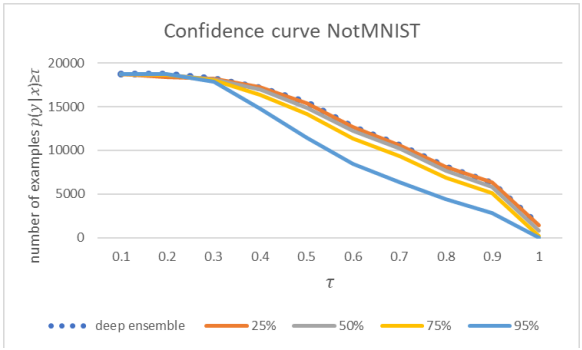

Figure 1(b) Confidence curve of out-of-distribution data.

deep ensemble more robust in unseen data, especially in test data that comes from another distribution. The predictive distribution moves to a uniform distribution as uncertainty increases, meaning that the model randomly guesses.

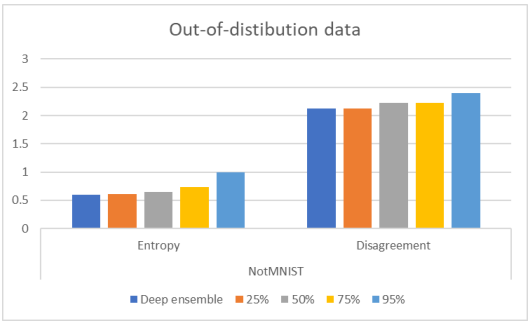

Figure 2 Uncertainty measurements of NotMNIST.

Baseline evaluation metrics did not change significantly with respect to the deep ensemble for pruning and quantization. The size of the deep ensemble decreased by a magnitude, and 50% of the parameters pruned and quantized deep ensemble were almost the same size as a single model. However, the ensemble's generalization error is lower than that of a single neural network, and it generalizes well in unseen data.

6. Discussion

We empirically showed that applying pruning and quantization into the deep ensemble decreased the ensemble size and increased the uncertainty metrics while showing a slight accuracy loss depending on the sparsity level. Pruning with 25–75% sparsity and quantization successfully addresses the problem associated with the linear increase in the size of the deep ensemble and shows solid performance. However, pruning with 95% sparsity noticeably degrades the accuracy.

There is a tradeoff between insignificant accuracy degradation, uncertainty, and memory footprint metric improvements. The pruned and quantized deep ensemble makes a less confident prediction and generalizes well by increasing the uncertainty metrics. Thus, pruning and quantization require a deep ensemble applicable to memory-intensive tasks in Internet-of-Things or mobile devices, while increased uncertainty metrics make the deep ensemble more robust in distribution shift.

Pruning after initialization should be studied further by applying the proposed strategy to real-world tasks. Moreover, creating a diverse deep ensemble by applying pruning with a pre-trained single model could be another possible research direction.

Acknowledgments: This research was funded by the National Research Foundation (NRF) of Korea, grant number 2017R1D1A1B03028929, 2019R1F1A1062094, and NRF-2020R1F1A1069361. The NRF was funded by the Korean Ministry of Education & the Ministry of Science and ICT.

References

[1] Guo, C.; et al. On Calibration of Modern Neural Networks. In Proceedings of the 34th International Conference on Machine Learning; Proceedings of Machine Learning Research; PMLR: International Convention Centre, Sydney, Australia, 2017; Vol. 70, pp 1321–1330.

[2] Srivastava, N.; et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. J. Mach. Learn. Res. 2014, 15 (1), 1929–1958.

[3] Gal, Y.; Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep

Learning. In Proceedings of The 33rd International Conference on Machine Learning; Proceedings of Machine Learning Research; PMLR: New York, New York, USA, 2016; Vol. 48, pp 1050–1059.
[4] Zhou, Z.-H. Ensemble Methods: Foundations and Algorithms, 1st ed.; Chapman & Hall, 2012.
[5] Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles. NIPS'17; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp 6405–6416.
[6] Ovadia, Y.; et al. Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift; 2019.
[7] Fort, S.; Hu, H.; Lakshminarayanan, B. Deep Ensembles: A Loss Landscape Perspective; 2020.
[8] Cheng, Y.; et al. A Survey of Model Compression and Acceleration for Deep Neural Networks; 2020.
[9] Tran, L.; et al. Hydra: Preserving Ensemble Diversity for Model Distillation; 2020.
[10] Malinin, A.; Mlodozeniec, B.; Gales, M. Ensemble Distribution Distillation; 2019.
[11] Malinin, A.; Gales, M. Predictive Uncertainty Estimation via Prior Networks. 32nd Conference on Neural Information Processing Systems NIPS'18; Montréal, Canada; 2018.
[12] Hu, R.; et al. The MBPEP: A Deep Ensemble Pruning Algorithm Providing High Quality Uncertainty Prediction. CoRR 2019, abs/1902.09238.
[13] Han, S.; Mao, H.; Dally, W. J. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding; 2016.
[14] Zhu, M.; Gupta, S. To Prune, or Not to Prune: Exploring the Efficacy of Pruning for Model Compression; 2017.
[15] Blalock, D.; et al. What Is the State of Neural Network Pruning?; 2020.
[16] Gao, S. A Discover of Class and Image Level Variance Between Different Pruning Methods on Convolutional Neural Networks. In 2020 IEEE International Conference on Smart Internet of Things (SmartIoT); 2020; pp 176–182.
[17] Hooker, S.; et al. What Do Compressed Deep Neural Networks Forget?; 2020.
[18] Goodfellow, I.; Bengio, Y.; Courville, A. Deep Learning; MIT Press, 2016; pp. 296.
[19] Nixon, J.; Lakshminarayanan, B.; Tran, D. Why Are Bootstrapped Deep Ensembles Not Better? In" I Can't Believe It's Not Better!" NeurIPS 2020 workshop; 2020.
[20] Gustafsson, F. K.; Danelljan, M.; Schön, T. B. Evaluating Scalable Bayesian Deep Learning Methods for Robust Computer Vision; 2020.
[21] Hubschneider, C.; Hutmacher, R.; Zöllner, J. M. Calibrating Uncertainty Models for Steering Angle Estimation. In 2019 IEEE Intelligent Transportation Systems Conference (ITSC); 2019; pp 1511–1518.
[22] Abdar, M.; et al. Review of Uncertainty Quantification in Deep Learning: Techniques, Applications and Challenges; 2021.
[23] Josiah, D.; et al. Quantifying Uncertainty in Deep Learning Systems, 2020.
[24] LeCun, Y.; Cortes, C. MNIST Handwritten Digit Database. 2010.
[25] Martín Abadi; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems; 2015.
[26] Kaggle notMNIST dataset, Available online: https://www.kaggle.com/lubaroli/notmnist.