Why Are Bootstrapped Deep Ensembles Not Better?

Anonymous Author(s)
Affiliation
Address
email

Abstract

Ensemble methods have been incredibly consistent in reaching state of the art performance. One of the popular ways to construct an ensemble is to train individual members on a bootstrap sample of the original dataset. While bootstrapping is very popular in the literature on decision trees and random forests, it has been relatively less popular for ensembles of deep neural networks. We investigate a major hypothesis for the weak performance of bootstrap ensembles - dataset size - and find that the number of unique datapoints in the training set is the main determinant of ensemble performance.

1 Introduction

Let $\boldsymbol{x}_n \in \mathbb{R}^D$ denote D-dimensional features and y_n denote the corresponding target. $y \in \mathbb{R}$ for regression and $y \in [1, \ldots, K]$ for classification. Assume we have a parametric model $p(y|\boldsymbol{x}, \boldsymbol{\theta})$ for the conditional distribution, and a prior $p(\boldsymbol{\theta})$ over parameters. The Bayesian posterior over parameters is given by

$$p(\boldsymbol{\theta}|\{\boldsymbol{x}_n, y_n\}_{n=1}^N) \propto p(\boldsymbol{\theta}) \prod_{n=1}^N p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}). \tag{1}$$

Computing the posterior over $\boldsymbol{\theta}$ is computationally challenging when $p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta})$ is parametrized using a deep neural network. While computing the posterior is challenging, it is usually easy to compute the MAP solution, which corresponds to the mode of the posterior. The MAP solution can be written as the minimizer of the following loss (negative log likelihood + negative log prior):

$$L(\boldsymbol{\theta}, \{\boldsymbol{x}_n, y_n\}_{n=1}^N) = -\log p(\boldsymbol{\theta}) - \sum_{n=1}^N \log p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}). \tag{2}$$

$$\hat{\boldsymbol{\theta}}_{\mathsf{MAP}} = \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \{\boldsymbol{x}_n, y_n\}_{n=1}^N). \tag{3}$$

The MAP solution is computationally efficient, but only gives a point estimate and not a distribution over parameters. There are two popular ways to extend MAP to get multiple solutions:

- *Diverse random seeds*: Initialize at M different values, use different mini-batch ordering for each ensemble member and repeat the minimization in equation 2 which could lead to M different solutions, if the loss is non-convex.
- *Bootstrap aggregation a.k.a. Bagging* [Breiman, 1996]: which trains members on different bootstrap samples of the original dataset.

Concretely, bagging first samples a bootstrap dataset of N points with replacement from the original training set of N unique points, and then trains a neural network on this resampled dataset. Concep-

Submitted to 1st I Can't Believe It's Not Better Workshop (ICBINB@NeurIPS 2020). Do not distribute.

tually, it is equivalent to drawing a set of instance weights $\boldsymbol{w} = [w_1, \ldots, w_N]$ from a Multinomial distribution N times with mean parameter $[1/N, ..., 1/N]$. Note that $\sum_n w_n = N$.

Once we have the weights $\boldsymbol{w}$, we minimize the weighted loss function:

$$L(\boldsymbol{\theta}, \{\boldsymbol{x}_n, y_n\}_{n=1}^N, \boldsymbol{w}) = -\log p(\boldsymbol{\theta}) - \sum_{n=1}^{N} w_n \log p(y_n|\boldsymbol{x}_n, \boldsymbol{\theta}). \tag{4}$$

$$\hat{\boldsymbol{\theta}}_{\mathsf{WMAP}} = \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \{\boldsymbol{x}_n, y_n\}_{n=1}^N, \boldsymbol{w}). \tag{5}$$

where $\hat{\boldsymbol{\theta}}_{\mathsf{WMAP}}$ denotes the "weighted MAP" solution obtained by minimizing the weighted loss. Note that setting $w_n = 1 \forall n$ in equation 4 recovers the usual unweighted case in equation 2.

Bootstrap is theoretically well-motivated [Efron, 1992] and popular in the literature on ensembles of decision trees (cf. random forests [Breiman, 2001]), but it has been found to hurt performance for ensembles of deep neural networks. Lakshminarayanan et al. [2017], Lee et al. [2015] report that using the entire dataset for all the networks in the ensemble, which corresponds to a fixed value of $\boldsymbol{w} = [1, \ldots, 1]$, works better than training the individual networks on bootstrap versions of the original dataset. Lakshminarayanan et al. [2017] hypothesize that the poor empirical performance could be due to the fact that the bootstrap sample only contains 63.2% of unique points.[1] Lakshminarayanan et al. [2017] also point to a parallel in the literature on decision tree ensembles where earlier work on random forests, e.g. Breiman's random forests [Breiman, 2001] used bootstrap, while latter work such as extremely randomized trees (ERT) [Geurts et al., 2006] found that bootstrap was not necessary when using other sources of randomization.

We investigate this hypothesis in detail by implementing variants of bootstrap that increase the number of unique points. We systematically vary the fraction of unique points in the range $[0.632, 1]$ and investigate the accuracy as well as calibration of the resulting ensemble on popular image classification benchmarks such as CIFAR-10 and CIFAR-100. We find that when we increase the number of unique data points, the accuracy of the ensemble (as well as the individual members) peaks at 1, corresponding to using the full dataset.

2 Methods + Experiment Results

We evaluate a Wide-Resnet [Zagoruyko and Komodakis, 2016] ensemble on CIFAR-100 and CIFAR-10 [Krizhevsky, 2009], as well as a small CNN ensemble on Fashion-MNIST [Xiao et al., 2017]. We evaluate accuracy and calibration of individual networks as well as ensembles. All ensembles trained have 10 members, unless indicated otherwise.

2.1 Evaluation

We evaluate calibration using the expected calibration error (ECE) [Naeini et al., 2015].

Expected Calibration Error (ECE). To approximate the calibration error in expectation, ECE Naeini et al. [2015] discretizes the probability interval into a fixed number of bins, and assigns each predicted probability to the bin that encompasses it. The calibration error is the difference between the fraction of predictions in the bin that are correct (accuracy) and the mean of the probabilities in the bin (confidence). Intuitively, the accuracy estimates $\mathbb{P}(Y = y \mid \hat{p} = p)$, and the average confidence is a setting of p. ECE computes a weighted average of this error across bins:

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{n_b}{N} \left|\mathrm{acc}(b) - \mathrm{conf}(b)\right|,$$

where n_b is the number of predictions in bin b, N is the total number of data points, and $\mathrm{acc}(b)$ and $\mathrm{conf}(b)$ are the accuracy and confidence of bin b, respectively.

[1] From Lakshminarayanan et al. [2017, §2.4]: *"The bootstrap draws N times uniformly with replacement from a dataset with N items. The probability an item is picked at least once is $1 - (1 - 1/N)^N$, which for large N becomes $1 - e^{-1} \approx 0.632$. So, the expected number of unique data points in a bootstrap sample is $0.632N$."*

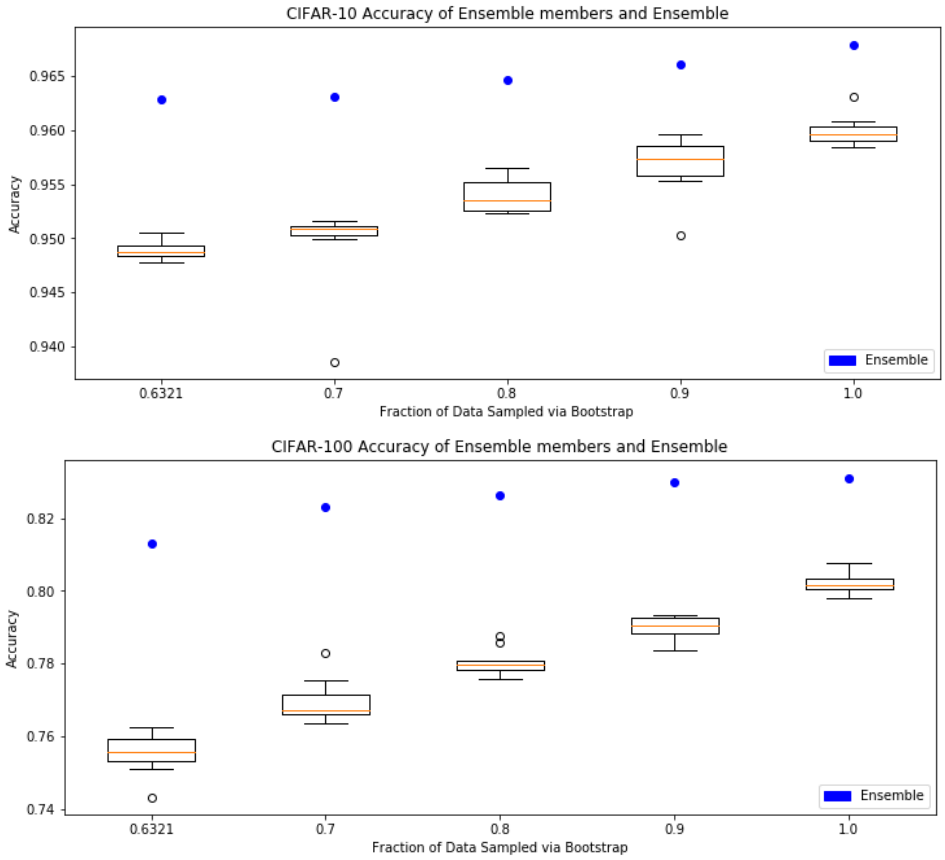

Figure 1: Wide Resnet trained on CIFAR-10 (top) and CIFAR-100 (bottom) Bootstrap Ensembles as we vary the number of unique points in the sampled dataset. The box-and-whisker plot indicates the performance distribution of the 10 members of the ensemble before averaging. The ensemble at 1.0 trains on the standard dataset. The ensemble at 0.6321 trains on a dataset sampled through the vanilla bootstrap. It has equal size to the original dataset and is sampled with replacement. All other fractions of the data sample with replacement beyond the number of original datapoints in the dataset.

2.2 Supersampling the dataset does not lead to improved performance

We evaluate a Wide Resnet on data from bootstrap draws from a dataset as well as on the standard dataset. As mentioned earlier, the bootstrap sampled data contains 63.2% unique points. We evaluate reweighting by supersampling the training dataset for each ensemble member. The bootstrap is modified to sample data with replacement until a given percentage of the training dataset has been sampled. We sample up to 70%, 80%, 90%, and 99% of the training data. In this case diversity is in the form of a reweighting of the dataset which comes out of the randomness in the datapoint selection. Finally, we also report results for deep ensembles [Lakshminarayanan et al., 2017] which correspond to the case where each ensemble member is trained on the entire 100% of the dataset, and the source of diversity is random initialization and randomness in SGD [Fort et al., 2019].

In Figure 1 we show the distribution of ensemble member performance alongside ensemble performance in a box-and-whisker plot on CIFAR-10 and CIFAR-100. The median ensemble member's accuracy is marked by the orange line, with the minimum and maximum ensemble members marked by a dot or an end line depending on whether they are within 1.5 * the Inter Quartile Range (25th-75th percentile range) of the data. Models trained on bootstrap dataset (left most point corresponding to 0.632 marker) achieve a much smaller accuracy on their validation set than models trained on the full dataset, which is to be expected. While the ensemble of the bootstrap sample members performs better than the single model, the gains from ensembling do not overcome the drop in accuracy due to dataset size, underperforming an ensemble trained on the full dataset. As the number of unique

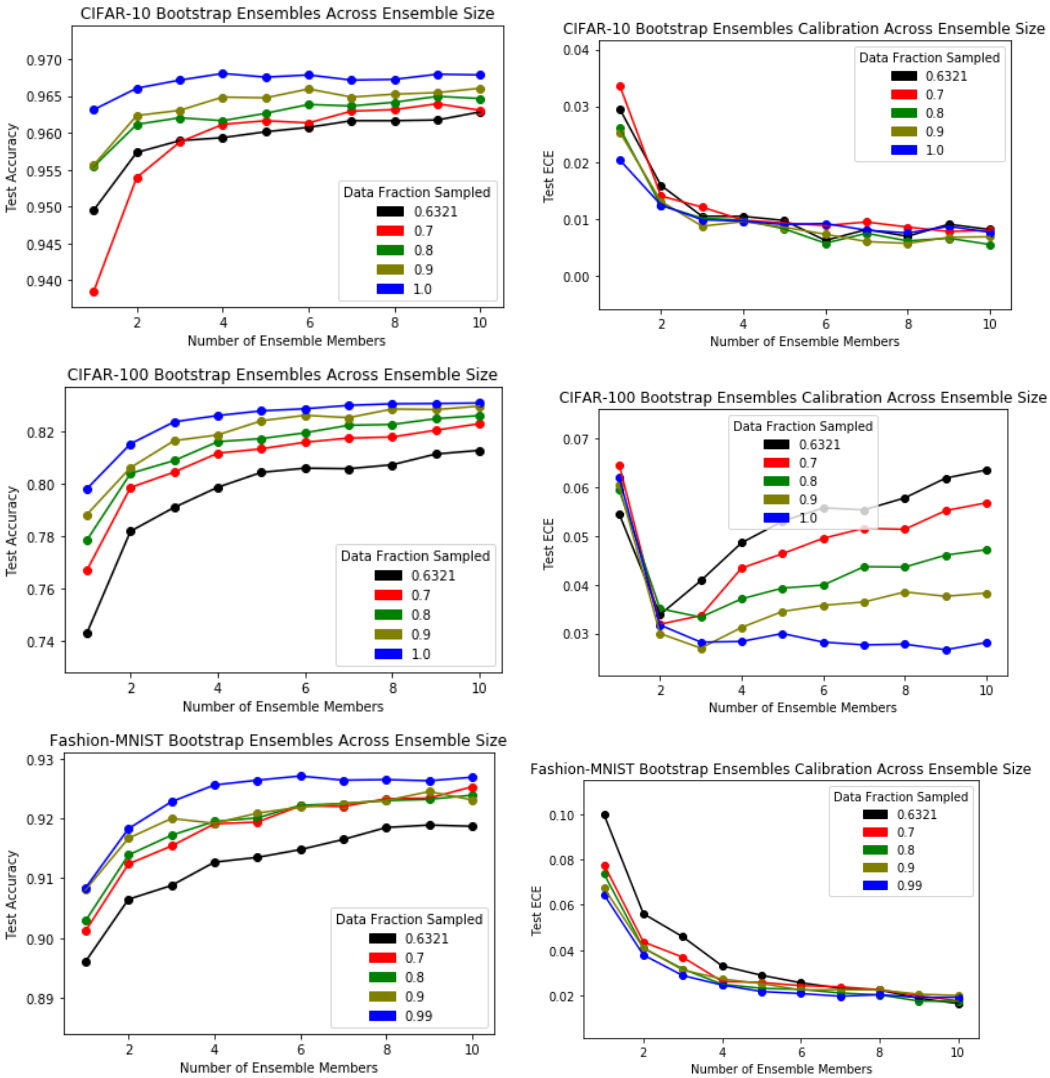

Figure 2: Bootstrap ensembles are dominated by standard deep ensembles on both accuracy and calibration error across the number of ensemble members.

datapoints in the training dataset increases, both the individual ensemble member performance and overall ensemble performance consistently improve peaking at the use of the full dataset.

In Figure 2 we investigate ensemble size and its impact on calibration error and accuracy for varying fraction of unique datapoints. The trend is that ensembles trained with the bootstrap are dominated in accuracy by ensembles trained without it at every number of ensemble members from 2 up to 10. We also observe that deep ensembles require fewer networks to achieve the same performance. With calibration, the impact of full dataset training is not as dramatic. While the general trend is also for ensembles trained with more unique training datapoints to outperform models trained with a heavier bootstrap, CIFAR 10 trained with > 4 ensemble members and Fashion-MNIST trained with > 7 members show a negligible difference between ECE of ensembles trained on a varying number of unique samples. We also notice an interesting phenomenon with ECE on CIFAR-100 merits further investigation: where deep ensembles are better calibrated with increasing members but bootstrap ensembles are better calibrated for $M = 2$ but their ECE becomes worse for higher M.

3 Discussion

We investigate the effect of bootstrapping strategies on the accuracy and calibration of ensembles. We find that the number of unique points significantly affects the accuracy of the ensemble as well

as individual members. Using the entire dataset typically performs best w.r.t. calibration, and interestingly, bootstrap ensembles with fewer unique data points can still be well-calibrated. For future work, we plan to look at calibration under dataset shift [Ovadia et al., 2019] and detecting out-of-distribution inputs [Hendrycks and Gimpel, 2016, Lakshminarayanan et al., 2017].

References

L. Breiman. Random forests. *Machine learning*, 2001.

Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

Bradley Efron. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics*, pages 569–593. Springer, 1992.

Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

Stefan Lee, Senthil Purushwalkam, Michael Cogswell, David Crandall, and Dhruv Batra. Why M heads are better than one: Training a diverse ensemble of deep networks. *arXiv preprint arXiv:1511.06314*, 2015.

Mahdi Pakdaman Naeini, Gregory F Cooper, and Milos Hauskrecht. Obtaining Well Calibrated Probabilities Using Bayesian Binning. In *AAAI*, pages 2901–2907, 2015.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D Sculley, Sebastian Nowozin, Joshua V Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. *arXiv preprint arXiv:1906.02530*, 2019.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.