Novel Uncertainty Framework for Deep Learning Ensembles

Tal Kachman, Michal Moshkovitz, Michal Rosen-Zvi
IBM Research

Abstract

Deep neural networks have become the default choice for many machine learning tasks, such as classification and regression. Dropout, a method commonly used to improve the accuracy of deep neural networks, generates an ensemble of thinned networks with extensive weight sharing. Recent studies [1, 2] show that dropout can be viewed as approximate Bayesian inference, and can be utilized as a practical tool to obtain uncertainty estimates of the network. We propose a novel statistical mechanics-based framework for dropout. We use this framework to propose a new generic algorithm that focuses on estimating the variance of the loss, as measured by the ensemble of thinned networks. Our approach can be applied to a wide range of deep neural network architectures and machine learning tasks. In classification, this algorithm allows the generation of a don't-know answer, which can increase the reliability of the classifier. Empirically, we demonstrate that our algorithm improves state-of-the-art AUC results on publicly available benchmarks.

1 Introduction

Deep learning (DL) algorithms are being used to successfully solve real-world classification problems from various fields, including recognizing handwritten digits and identifying the presence of key diagnostic features in medical images [11, 9]. A typical classification challenge for a DL algorithm involves training the algorithm on an example data set, then using a separate set of test data to evaluate its performance. The aim is to provide answers that are as accurate as possible, as measured by the true positive rate (TPR) and the true negative rate (TNR). Many DL classifiers, particularly those using a softmax function in the very last layer, yield a continuous score, h. A step function is then used to map this continuous score to each of the possible categories that are being classified. TPR and TNR scores can then be generated for each separate example that is being predicted. This is done by setting a threshold parameter that is applied when mapping h to the decision. Values above this threshold are mapped to positive predictions, while values below it are mapped to negative predictions. The ROC curve is then generated from these pairs of TPR/TPN scores. The performance of binary classifiers is often evaluated by calculating the area under the ROC curve (AUC) [3].

Many studies show that the AUC achieved by DL algorithms is higher than most, if not all, of the alternative classifiers. Unfortunately, DL algorithms are notorious for being "black box" models, as it is difficult to obtain insight into how the algorithm arrived at its conclusion. One way to mitigate this problem is to provide a measure of the classification uncertainty, or the confidence one has in the classification prediction, along with the prediction of the outcome. An additional benefit of assessing uncertainty in DL algorithms is that such classifiers are frequently permitted to return a "don't know" answer for very low confidence predictions in the test data, and can be judged only on the responses in the "do know" portion of the dataset. This permits the algorithm to generate higher quality predictions overall, while leaving humans to interpret the samples for which it would generate poor quality predictions, similar to a triage system. Recent studies combine the dropout technique for DL regularization with Bayesian modeling, to derive uncertainty estimates in DL classifiers [1, 2, 5, 6].

Third workshop on Bayesian Deep Learning (NeurIPS 2018), Montréal, Canada.

The three main contributions of this work are the following. First, we introduce a statistical mechanic framework that assigns probability distributions over the ensemble of thinned networks of the dropout, i.e., neural networks with a subset of neurons removed, based on the performance of their cross-entropy loss function across the test dataset. This framework has a flexible variable, β, which represents the inverse temperature. When set to zero, it results in a uniform distribution assumption over the thinned networks, and the framework collapses to a Bayesian one. In contrast, a finite inverse temperature β results in a non-uniform distribution, and the framework enables interpretation and reasoning regarding uncertainty. Second, we present a new algorithm that is based on estimations of the loss variance in the case of a finite β through Monte Carlo sampling, called Loss Variance Monte Carlo Estimate (LoVME). Finally, we illustrate the benefits of deriving uncertainty through the LoVME algorithm in scenarios where the classifier can yield a don't-know answer. We use the MNIST [8] and CIFAR [7] datasets to show the performance of our algorithm, and compare our results to state-of-the-art algorithms for uncertainty in DL.

2 Statistical Mechanics Framework for Deep Learning Ensemble

In this section we view dropout through the lens of statistical mechanics. This enable us to design a novel uncertainty framework for deep learning ensembles. For ease of notation, throughout this section we follow a single example for which we want to estimate uncertainty. For the uncertainty estimation, we use the dropout method, which induces an ensemble of thinned networks. Each thinned network i is defined by its loss $\mathcal{L}_i$ and the number of neurons that it contains, N_i. We would like to find the distribution over the thinned NNs, which we denote $p(\mathcal{L}_\text{i}, N_i)$ or simply p_i. We can estimate the expected size of the network, as it depends on the dropout value and the expected loss using the training error. Thus, we can approximate the following two expectations

$$\mathbb{E}[\mathcal{L}] = \sum_i p\left(\mathcal{L}_i, N_i\right) \mathcal{L}_i, \quad \mathbb{E}[N] = \sum_i p\left(\mathcal{L}_i, N_i\right) N_i. \tag{1}$$

According to the principle of maximum entropy [4], the distribution with maximal entropy (i.e., the distribution that maximizes $-\sum p_i \log p_i$) best represents the current state of knowledge among all distributions that satisfy Equation (1). The resulting distribution is the one with the probability of the i-th thinned NN being

$$p_i = \frac{1}{Z} e^{-(\beta \mathcal{L}_i + \eta N_i)}, \tag{2}$$

where β and η are some parameters that depend on $\mathbb{E}[\mathcal{L}]$ and $\mathbb{E}[N]$, and Z is a normalization factor. In statistical mechanics, distribution (2) is called Gibbs distribution, β is referred to as the temperature, η as the chemical potential, and Z as the partition function.

3 Loss Variance Monte Carlo Estimate

The variance of the loss can be deduced from the partition function Z,

$$\mathbb{V}\text{ar}[\mathcal{L}] = \frac{\partial^2 \log Z}{\partial \beta^2} \tag{3}$$

This variance is a measure of the uncertainty; thus, to estimate the uncertainty we would like to compute the partition function. Unfortunately, the partition function contains an exponential number of terms and is therefore impossible to calculate directly. To remedy this situation, we present an algorithm that approximates this value using the importance sampling technique. This algorithm, Loss Variance Monte Carlo Estimate (LoVME), can numerically evaluate the needed value in an efficient manner.

We used the probability defined in Equation (2) as a non-uniform sampler. This approach gives less weight to thinned networks with high loss, and allows us to avoid sampling only in a small region of the thinned NN. To ensure that we pick each state with its associated probability measure, we generate a Markov chain Monte Carlo in an ergodic way, such that the rates obey a detailed balance. The detailed balance assumption allows us to sample thinned networks with Metropolis-Hastings algorithms, with the following acceptance rule when moving from thinned network i to j

$$\begin{cases} e^{-\beta\left((\mathcal{L}_i - \mathcal{L}_j) + \frac{\eta}{\beta}(N_i - N_j)\right)} & \text{if } \mathcal{L}_i - \mathcal{L}_j + \frac{\eta}{\beta}\left(N_i - N_j\right) > 0 \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

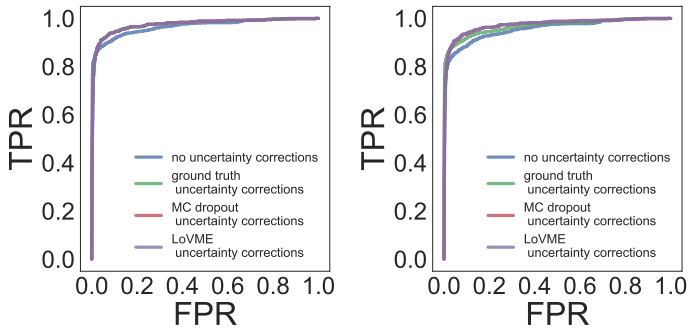

Figure 1: ROC curves when algorithms are allowed to return "don't know"

4 Experiments

To demonstrate the effectiveness of the LoVEMe algorithm, we focus on the natural scenario where an algorithm is allowed to return "don't know" on a small fraction of examples. These examples are in turn classified by humans. Thus, one can set a certain threshold over an uncertainty measure above which an example is too ambiguous. To estimate the uncertainty, we trained a LeNET [10] network in PyTorch [12], using dropout with probability $p = 0.5$ [1]. We used a cross-entropy softmax loss function in both the training and testing phases. We applied it to two different datasets: MNIST [8] and CIFAR-10 [7]. For MNIST, we tested against a partially perturbed subset, where we introduced both random rotation and noise to 10% randomly chosen images in the test set.

The basic uncertainty measure, which we call the *"ground truth" measure*, is calculated using the variance over different initializations of the weights. Specifically, we trained an ensemble of 4000 LeNET networks with different initializations of the weights. For each ensemble we performed a classification of the test portion in the dataset, and calculated the variance over the 4000 networks of the probability outcome from the softmax cross-entropy for the correct class. For each dataset, we generated an ensemble of thinned NNs using the LoVMe algorithm to estimate the uncertainty. Each of these ensembles typically converged within 300 Monte Carlo steps.

4.1 Results

We show that removing high uncertainty examples increases the AUC when using our LoVME method compared to Monte Carlo dropout and the naïve ground truth. Figure 1 shows the Receiver Operating Characteristic (ROC) curves and Table 1 shows the different values of the AUC for each of these three methods.

	W/O uncertainty corrections	Ground truth w/ uncertainty corrections	MC dropout uncertainty w/ corrections	LoVME w/ uncertainty corrections
MNIST	0.967	**0.974**	**0.974**	**0.974**
CIFAR10	0.956	0.967	0.972	**0.973**

Table 1: The AUC values of the ROC curves of Figure 1. In bold, the best algorithm for each dataset.

5 Conclusions

We presented a method for estimating the uncertainty of a DL classifier using a novel framework that assigns probability distributions to thinned networks using ideas from statistical mechanics. Using this framework, we designed the Loss Variance Monte Carlo Estimate (LoVME) algorithm and illustrated the benefits of this algorithm on the MNIST [8] and CIFAR [7] datasets.

[1] Code to be available for both the network and the LoVMe algorithm.

References

[1] Yarin Gal. Uncertainty in deep learning. *University of Cambridge*, 2016.

[2] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[3] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

[4] Edwin T Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.

[5] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015.

[6] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, pages 5580–5590, 2017.

[7] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[8] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[9] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.

[10] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[11] Paulo J Lisboa and Azzam FG Taktak. The use of artificial neural networks in decision support in cancer: a systematic review. *Neural networks*, 19(4):408–415, 2006.

[12] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.