Sampling-based Bayesian Inference with gradient uncertainty

Chanwoo Park Jae Myung Kim Seok Hyeon Ha Jungwoo Lee
Department of Electrical and Computer Engineering
Seoul National University
`{cpark, goldkim92, hash1108}@cml.snu.ac.kr, junglee@snu.ac.kr`

Abstract

Deep neural networks(NNs) have achieved impressive performance, often exceed human performance on many computer vision tasks. However, one of the most challenging issues that still remains is that NNs are overconfident in their predictions, which can be very harmful when this arises in safety critical applications. In this paper, we show that predictive uncertainty can be efficiently estimated when we incorporate the concept of gradients uncertainty into posterior sampling. The proposed method is tested on two different datasets, MNIST for in-distribution confusing examples and notMNIST for out-of-distribution data. We show that our method is able to efficiently represent predictive uncertainty on both datasets.

1 Introduction

Recent deep neural networks (NN) methods have achieved human-level or superhuman performance at various tasks in computer vision, natural language processing, and robotics. But they often fail to estimate predictive uncertainty which can be crucial for some tasks such as medical diagnosis or autonomous driving, and tend to be overconfident even when their predictions are incorrect. In order for these critical applications to be successfully employed, they must be able to provide how certain they are about cancer detection and road sign recognition [2]. Bayesian NNs, which find a posterior distribution over network parameters, are the state-of-the-art methods for estimating predictive uncertainty [17, 16].

1.1 Bayes's Theorem

$$p(\theta|D) = \frac{p(D|\theta)\, p(\theta)}{p(D)} = \frac{p(D|\theta)\, p(\theta)}{\int p(D|\theta)\, p(\theta)\, d\theta} \tag{1}$$

$p(D|\theta)$ is the likelihood of the occurrence of dataset D given a model with parameters θ, $p(\theta)$ the prior, and $p(D)$ the data distribution. Bayesian NNs compute the posterior distribution over the parameters to estimate predictive uncertainty. However exact Bayesian inference is not feasible for NN since computing marginal likelihoods in high dimension is analytically intractable, thus we need to approximate the inference problem.

1.2 Related Works

There have been two different major approaches for the approximation. One is variational Bayesian methods, which approximate Bayesian inference by introducing simpler, tractable distribution $q(\theta)$ to approximate posterior distribution [13, 3, 11, 4, 10]. This method minimizes the Kullback-Leibler (KL) divergence between p and q, $KL(q(\theta)||p(\theta|D))$, which is optimized by maximizing the evidence lower bound (ELBO). On the other hand, Markov Chain Monte Carlo (MCMC) methods

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

have been successfully applied to Bayesian NNs. MCMC is non-parametric and asymptotically exact which iteratively draw samples from unknown true distribution to approximate expectation [8, 9, 12]. However, traditional MCMC methods require the full dataset in each iteration to generate proposals and calculate the acceptance probability which make them prohibitively expensive for large datasets.

Stochastic gradient MCMC (SG-MCMC) has gained keen interest recently which uses minibatches of the data to generate samples and ignore the acceptance step, thus scales well to large datasets. Welling & Teh(2011) developed stochastic-gradient Langevin dynamics(SGLD) by incorporating Langevin dynamics into stochastic optimization to insert adaptively scaled Gaussian noise, which is the first sampling algorithm based on stochastic gradients [19]. Chen et al.(2014) suggested Stochastic gradient Hamiltonian Monte Carlo (SGHMC) where they introduced auxiliary momentum term to rapidly explore the parameter space [7]. Recent studies have been proposed to improve convergence and sampling efficiency of SGLD and SGHMC for the past years [1, 15, 18, 6].

2 Our methods

In this paper we try to sample from an approximate posterior distribution using gradient uncertainty. Let $X^{(l)} = (X_{l,1}, X_{l,2}, ..., X_{l,m})$ denotes the sequence of gradient vector of i_{th} example in l_{th} minibatch where $X_{l,i} = \nabla_\theta J(\theta; x^{(i)}, y^{(i)})$ and m is the minibatch size. We define a new measure to quantify gradient uncertainty.

$$Gradient\,Uncertainty = \sum_{i \neq j} \langle X_i, X_j \rangle \tag{2}$$

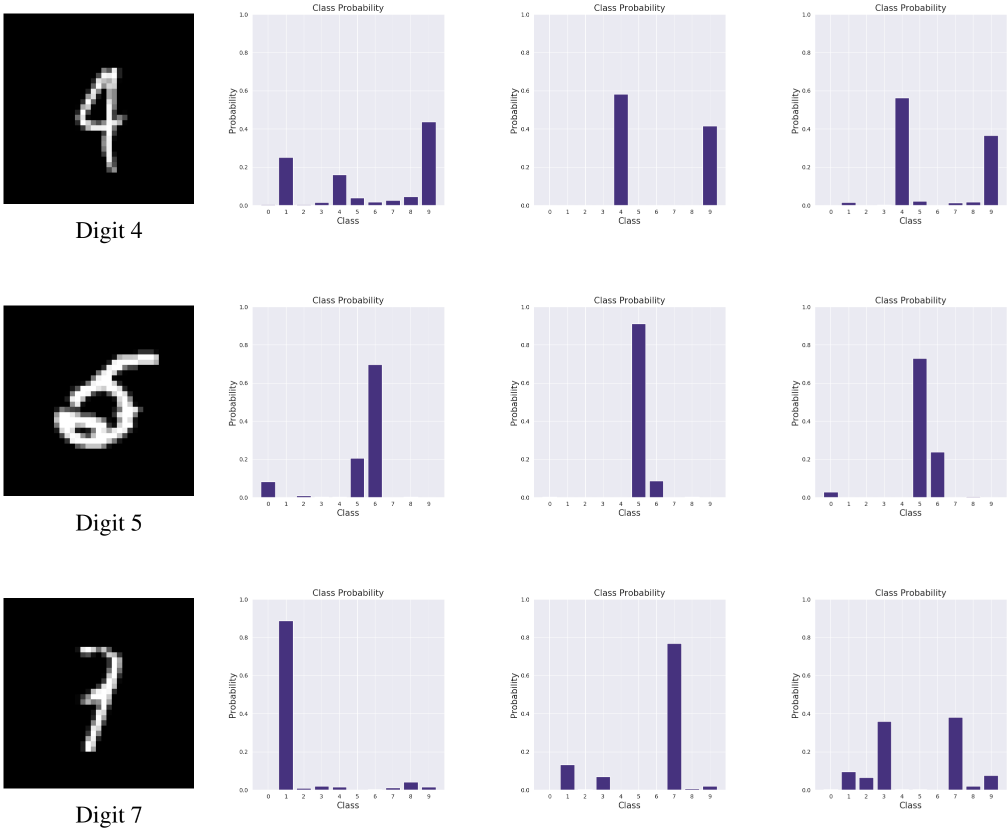
To the extent of our knowledge, this is the first work using stochastic gradient uncertainty for Bayesian sampling algorithm. The main idea is that we use gradient uncertainty as an indicator that the parameters are near the local optimum. The proposed sampling method is outlined in Alg 1.

Algorithm 1 Sampling algorithm

```
Initialize: Random $\theta_1$
for $l$ = 1,2,...,T do
    for $i$ = 1,2,...,m do                                        ▷ m : minibatch size
        $X_{l,i} \leftarrow \nabla_\theta J(\theta; x^{(i)}, y^{(i)})$
    end for
    $\tilde{X}_l \leftarrow \frac{1}{m}\sum_{i=1}^{m} X_{l,i}$
    $gradient\ uncertainty \leftarrow \sum_{i \neq j} \langle X_i, X_j \rangle$
    if $gradient\ uncertainty < threshold$ then
        Sample $\theta_l$
        $\theta_{l+1} \leftarrow \theta_l - \epsilon\tilde{X}_l + N(0, \sigma^2)$       ▷ $\epsilon$ : learning rate
    else
        $\theta_{l+1} \leftarrow \theta_l - \epsilon\tilde{X}_l$
    end if
end for
```

When the algorithm reaches local extrema, the exact gradient should be either zero or very close to zero. However, because of unbiased but noisy stochastic gradient, we assume the stochastic gradient near local modes follows zero-mean Gaussian distribution. Since stochastic gradients from a single training example, X_i, are randomly scattered around zero in each dimension of the gradient vector, the sum of inner product between different gradient vectors in a minibatch, Eq. (2), gives a sufficiently small value, whereas the sum of inner product can be large if the gradient vectors are not following zero-mean Gaussian but pointing in the similar direction.

We evaluate *gradient uncertainty* from the minibatch and sample parameters when it is lower than a certain threshold, meaning that we assume it has reached near local mode. That is, the magnitude of mean of gradients is sufficiently small and directions of gradients are diverse. It is computationally efficient because we calculate *gradient uncertainty* only from data subsets.



(a) SGLD (b) Dropout (c) Ours

Figure 1: MNIST Confusing examples

In most NN, the loss functions are non-convex and thus gradient-based optimization might get stuck in a local mode. We add random Gaussian noise to a proposal every time we sample a set of parameters, so that our method can escape from local mode and explore the parameter space.

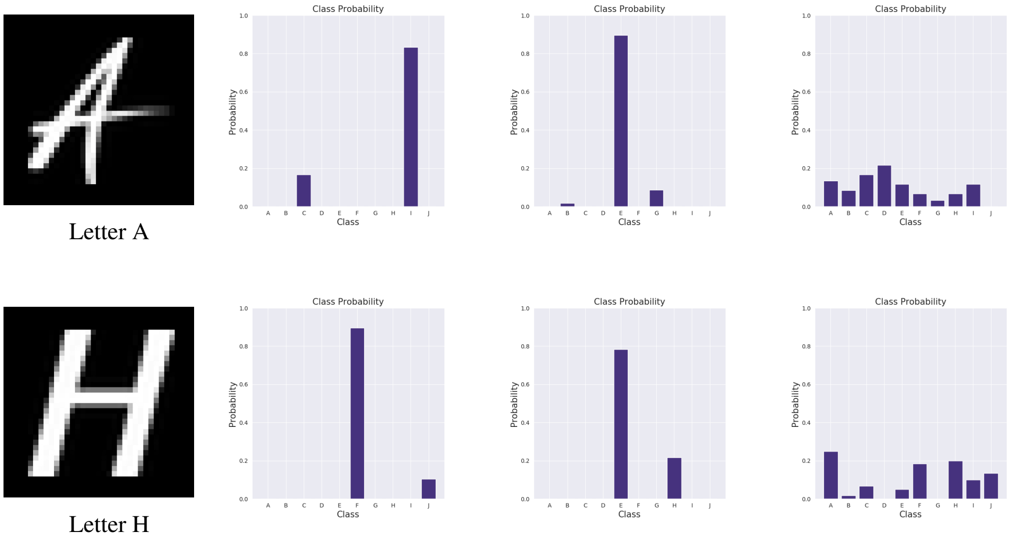
3 Experiments

We conduct two sets of experiments with MNIST [14] and notMNIST datasets for uncertainty estimation. The notMNIST dataset contains front glyphs for the 10 class letters(A-J), which can be treated as out-of-distribution data for the networks trained on original MNIST. The notMNIST dataset consists of 19K hand-cleaned instances and 500K uncleaned instances [5].

3.1 MNIST : Confusing examples

In the first experiment, we train networks on MNIST and compare the ability of three different methods to quantify uncertainty when confusing examples are given. We take the same five-layer convolutional NN which are trained on MNIST, and then evaluate the predictive uncertainty with 60 Monte Carlo samples. The results are compared with Dropout Bayesian Approximation [10] and SGLD [19]. This is shown in Figure 1.

In our approach the predictive results of given confusing digits are diverse and the probability is concentrated on the most likely candidates, i.e. in our cases ’4’, ’5’, ’7’ can be misinterpreted as ’9’, ’6’, ’3’ respectively. On the other hand, other methods show an overconfident prediction by assigning higher probability to incorrect classes.



Letter A

Letter H

(a) SGLD

(b) Dropout

(c) Ours

Figure 2: notMNIST Out-of-distribution examples

3.2 notMNIST : Out-of-distribution data

We further experimented with the same networks trained on MNIST to evaluate the uncertainty estimation on out-of-distribution data, which is notMNIST dataset in our case. Figure 2 shows the comparison results with the same baseline methods. It is shown that the proposed approach generates a fairly flat predictive posterior, while others are rather sharp. This result shows that our method captures reliable uncertainty when classifying unseen data.

To compare the effectiveness of uncertainty estimation on out-of-distribution data, we use the entropy of the predictive distribution, $H(Y|X)$. The predictive distribution for a given image X and the entropy can be expressed as:

$$p(Y|X) = \int p(Y|X, \theta)p(\theta|D)d\theta \tag{3}$$

$$H(Y|X) = -\int p(y|X)\ln(p(y|X))dy \tag{4}$$

We approximate $H(Y|X)$ using 60 Monte Carlo samples over the notMNIST dataset. Entropy comparison with baseline methods can be seen in Table 1. Our method shows higher predictive entropy on out-of-distribution data, i.e., our model gives less confident predictions on them. In both sets of experiments, we observe that our approach well represents predictive uncertainty while SGLD and Dropout generally produce overconfident probabilities.

	Ours	**SGLD**	**Dropout**
In-distribution Test Accuracy (%)	98.05	95.34	99.39
Out-of-distribution Predictive Entropy	**1.355**	0.107	0.665

Table 1: Entropy Comparison

4 Conclusion

Overconfident prediction is a major obstacle for many deep learning architectures to be deployed in safety critical applications. This paper presents a new methodology for uncertainty estimation which uses stochastic gradient uncertainty as an indicator to sample. We experimented our model on MNIST and notMNIST datasets. In both cases, we have shown that our method efficiently sample from posterior using stochastic gradients based computation. It is an important future work to compare quality of uncertainty estimates with other state-of-the-art SG-MCMC and variational methods. Also, applying adaptive learning rate based on gradient uncertainty when performing SG-MCMC will be another line of future work.

Acknowledgments

This work is in part supported by Basic Science Research Program (NRF-2017R1A2B2007102) through NRF funded by MSIP, Technology Innovation Program (10051928) funded by MOTIE, Bio-Mimetic Robot Research Center funded by DAPA (UD130070ID), INMAC, and BK21-plus.

References

[1] Sungjin Ahn, Anoop Korattikara, and Max Welling. Bayesian posterior sampling via stochastic gradient fisher scoring. *arXiv preprint arXiv:1206.6380*, 2012.

[2] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

[3] David Barber and Christopher M Bishop. Ensemble learning in bayesian neural networks. *NATO ASI SERIES F COMPUTER AND SYSTEMS SCIENCES*, 168:215–238, 1998.

[4] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

[5] Yaroslav Bulatov. Notmnist dataset. *Google (Books/OCR), Tech. Rep.[Online]. Available: http://yaroslavvb. blogspot. it/2011/09/notmnist-dataset. html*, 2011.

[6] Changyou Chen, David Carlson, Zhe Gan, Chunyuan Li, and Lawrence Carin. Bridging the gap between stochastic gradient mcmc and stochastic optimization. In *Artificial Intelligence and Statistics*, pages 1051–1060, 2016.

[7] Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *International Conference on Machine Learning*, pages 1683–1691, 2014.

[8] Siddhartha Chib and Edward Greenberg. Understanding the metropolis-hastings algorithm. *The american statistician*, 49(4):327–335, 1995.

[9] Petros Dellaportas, Jonathan J Forster, and Ioannis Ntzoufras. On bayesian model and variable selection using mcmc. *Statistics and Computing*, 12(1):27–36, 2002.

[10] 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.

[11] Alex Graves. Practical variational inference for neural networks. In *Advances in neural information processing systems*, pages 2348–2356, 2011.

[12] Kenneth M Hanson. Markov chain monte carlo posterior sampling with the hamiltonian method. In *Medical Imaging 2001: Image Processing*, volume 4322, pages 456–468. International Society for Optics and Photonics, 2001.

[13] Geoffrey E Hinton and Drew 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*, pages 5–13. ACM, 1993.

[14] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2, 2010.

[15] Chunyuan Li, Changyou Chen, David E Carlson, and Lawrence Carin. Preconditioned stochastic gradient langevin dynamics for deep neural networks. In *AAAI*, volume 2, page 4, 2016.

[16] David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

[17] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

[18] Sam Patterson and Yee Whye Teh. Stochastic gradient riemannian langevin dynamics on the probability simplex. In *Advances in Neural Information Processing Systems*, pages 3102–3110, 2013.

[19] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 681–688, 2011.