BAYESIAN MODEL AVERAGING METHODS

ADVANCED ML (CS60075): TERM PROJECT REPORT

Shikhar Mohan, Adarsh Vemali, Parth Mall

Indian Institute of Technology Kharagpur

ABSTRACT

In this report we discuss and contrast between two bayesian optimisation algorithms and compare and analyze their effectiveness and performance on machine learning tasks. More specifically, we compare Stochastic Weighted Averaging Gaussian, a simple and generalised approach for uncertainty representation and calibration, against Cyclical Stochastic Gradient Markov Chain Monte Carlo(SG-MCMC), which targets distributions via a decreasing step-size scheme.

1. INTRODUCTION

1.1. A Simple Baseline for Bayesian Uncertainty in Deep Learning

In this paper they propose a different approach to Bayesian deep learning, they use the information contained in the SGD trajectory to efficiently approximate the posterior distribution over the weights of the neural network.

In particular, the contributions of the paper are the following:

1. In this work they have proposed **SWAG** (SWA-Gaussian), a scalable approximate Bayesian to draw inferences for deep learning. SWAG improves on Stochastic Weight Averaging(SWA), by additionally computes a low-rank plus diagonal approximation to the covariance of the iteration values, which is used together with the SWA mean, to define a Gaussian posterior approximation for the weights.

2. **SWAG** is motivated by the theoretical analysis of the stationary distribution of SGD iteration values, which suggests that the SGD trajectory contains useful information about the geometry of the posterior. They found that in the low-dimensional subspace spanned by SGD iterates the shape of the posterior distribution is approximately Gaussian within a basin of attraction. Further, SWAG is able to capture the geometry of this posterior remarkably well.

3. They show that SWAG can provide well-calibrated uncertainty estimates for neural networks across many settings in computer vision. In particular SWAG achieves higher test likelihood scores as compared to many SOTA approaches, including temperature scaling, MC-Dropout, SGLD, KFAC-Laplace and SWA on ImageNet and CIFAR datasets, on a range of architectures. They also demonstrate the effectiveness of the algorithm for tasks like out-of-domain detection, and transfer learning.

1.2. Cyclical Stochastic Gradient MCMC for Bayesian Deep Learning

In this paper, they propose to replace the traditional decreasing step-size schedule in Stochastic Gradient-Markov chain Monte Carlo with a cyclical variant.

SG-MCMC algorithms for inference with modern neural networks face several challenges:

i. In theory, it should converge to target distributions through a decreasing step size scheme, but suffers from bounded estimation errors in a short period.

ii. The loss surfaces for DNNs are highly multimodal [1]. It is our contention that this limitation arises from difficulties escaping local modes when using the small stepsizes that SG-MCMC methods typically require.

Note that the step-size in SG-MCMC controls the sampler's behavior in two key ways one, the magnitude deterministically drifts towards high density regions w.r.t. the current stochastic gradient, and two the level of injecting noise to randomly explore the parameter space. Hence, a small step-size reduces both abilities, resulting in a large number of iterations for the sampler to move across the modes.

This procedure can be viewed as SG-MCMC with warm restarts. The blue curve is the traditional decay, while the red curve shows the proposed cyclical schedule.

Cyclical SG-MCMC operates in two stages:

1. Exploration: when the step-size is large Fig 1(dashed red curves), we consider this stage as an effective burn-in mechanism, encouraging the sampler to take large moves and leave the local mode using the stochastic gradient. This step is similar to the mutation step in Genetic algorithms.

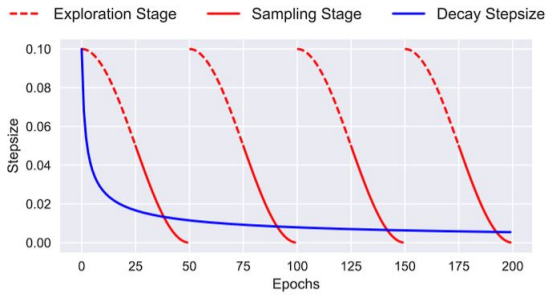

Fig. 1. Illustration of the proposed cyclical stepsize schedule (red) and the traditional decreasing stepsize schedule (blue) for SG-MCMC algorithms.

2. Sampling: when the step-size is small Fig 1(red curves), the sampler explores one local mode/valley. Hence, the samples are collected for local distribution estimation during this stage.

cSG-MCMC is a practical and efficient tool to provide significantly better mixing than the traditional models for complex distributions. The approach achieves similar performance to parallel MCMC with only a fraction of cost that parallel MCMC requires.

This approach provides a simple and automated approach to inference in modern Bayesian deep learning, with theoretical support, and promising results. This work is a step towards enabling MCMC approaches in Bayesian Deep Learning.

2. MOTIVATION

Machine learning models are finally used to make decisions. Realising and measuring uncertainty is crucial for decision making. e.g. in medical diagnoses and autonomous vehicles we want to protect against very rare but expensive mistakes. Deep learning models typically lack a representation of uncertainty, and provide overfit, overconfident and miscalibrated predictions.

Deep neural networks are often trained with stochastic optimization methods such as stochastic gradient descent (SGD) and its variants. Bayesian methods provide a principled alternative, which accounts for model uncertainty in weight space([2]), and achieves an automatic balance between fitting the data and model complexity, and previously had been a gold standard for inference with neural networks.

However, existing approaches are often highly sensitive to hyperparameter choices, and hard to scale to modern datasets and architectures, which limits their general applicability in modern deep learning.

3. RELATED WORKS

3.1. Bayesian Methods

Bayesian approaches depict uncertainty by establishing a distribution across model parameters. Marginalising these parameters to construct a full predictive distribution is known as Bayesian model averaging. Bayesian approaches were the state-of-the-art way to learn neural networks in the late 1990s [3, 4]. Modern neural networks, on the other hand, frequently include millions of parameters, and the posterior over these parameters (and hence the loss surface) is very non-convex, necessitating mini-batch techniques to reach a space of acceptable solutions [5]. Bayesian techniques have mainly proved intractable for current neural networks as a result of these factors. In this section, we look at some contemporary approaches to Bayesian deep learning.

Markov Chain Monte Carlo (MCMC): The Hamiltonian Monte Carlo work [3] made MCMC the standard for inference with neural networks. But HMC needs full gradients, something that is intractable for modern neural networks. Stochastic Gradient HMC (SGHMC) was introduced [6] extending HMC framework to use stochastic gradients in Bayesian inference, important for both scalability and exploring a space of solutions that provide good generalization. Stochastic gradient Langevin dynamics [7] uses first order Langevin dynamics in the stochastic gradient setting. While both SGHMC and SGLD asymptotically sample from the posterior in the limit of step sizes tending to zero, using finite learning rates introduces approximation errors [8] which make tuning SG MCMC methods tough.

Variational Inference: fitting a Gaussian variational posterior approximation over the weights of neural networks was suggested [9], which was further generalized [10] by proposing the *reparameterization trick* for training deep latent variable models: different variational inference methods using this trick were proposed for DNNs [11, 12]. While moderately sized networks enable variational methods to achieve strong performance, they are difficult to train on deep residual networks [13]. The insufficient data compression for DNNs provided by variational methods is argued to be the reason for difficulty in training [14]. Recent advances [15] on variational inference for deep learning mainly focus on smaller-datasets and architectures.

Dropout Variational Inference: [16] considered dropout at test time as approximation variational Bayesian inference using a spike and slab variational distribution. This technique is advanced by Concrete Dropout [17], which optimises dropout probability as well. These techniques are intriguing from a practical standpoint since they only involve ensembling dropout predictions at test time and have been effectively implemented to numerous downstream tasks [18, 19].

Laplace Approximations: a Gaussian posterior $N(\theta^*, I(\theta^*)^{-1})$ is assumed where θ^* is a MAP estimate and $I(\theta^*)^{-1}$ is the inverse of the Fisher information matrix. It was

most famously employed in [2] for Bayesian neural networks, where a diagonal approximation to the inverse of the Hessian was adopted for computational reasons. [20] discussed employing diagonal Laplace approximations in deep learning to combat catastrophic forgetting. For Laplace approximations, [21] advocated using either a diagonal or block Kronecker factored (KFAC) approximation to the Hessian matrix, and [22] successfully used the KFAC technique to online learning situations.

3.2. SGD Based Approximations

After examining the dynamics of SGD using stochastic calculus techniques, [8] suggested to utilise the iterates of averaged SGD as an MCMC sampler. [23] showed that under certain conditions a batch means estimator of the sample covariance matrix of the SGD iterates converges to $A = H(\theta)^{-1}C(\theta)H(\theta)^{-1}$, where $H(\theta)^{-1}$ is the inverse of the Hessian of the log likelihood and $C(\theta) = \mathbb{E}(\nabla \log p(\theta) \nabla \log p(\theta)^T)$ is the covariance of the gradients of the log likelihood. [23] demonstrates that a Gaussian approximation utilising A and the sample average of the iterates yields well calibrated confidence intervals for the parameters, and that the variance of the parameters achieves the Cramer Rao lower bound (the minimum possible variance).

4. METHODOLOGY

In this section, we discuss methodologies and the theoretical backgrounds pertaining to the two optimisation techniques for Bayesian Deep Networks, their respective methodologies and our comparison methodology as well. It is important to see that the main objective in Bayesian Deep Learning here is to treat the loss topology as a probability distribution and sample from it so as to perform model averaging. This is motivated by the fact that if we are to classify a new sample x^* using a model parameterized by θ and trained on dataset $\mathcal{D}$, we have

$$p(y|x^*, \mathcal{D}) = \int p(y|x^*, \theta) p(\theta|\mathcal{D}) \tag{1}$$

as the Bayesian Model Average (BMA). It is easy to see that our loss function being the negative log of posterior ends up being

$$\log p(\theta|\mathcal{D}) = -\log p(\mathcal{D}|\theta) - \log p(\theta) \tag{2}$$

Both of the techniques mentioned below traverse this loss landscape in an attempt to sample from them meaningfully enough to perform BMA.

4.1. Cyclic Stochastic Gradient MCMC

Like any MCMC method, this involves performing a Monte Carlo sampling over Markov Chains that are constructed and discarded stochastically. Here, the posterior distribution is given by $p(\theta|\mathcal{D}) \propto \exp(-U(\theta))$ where $U(\theta) = -\log p((D)|\theta) - \log p(\theta)$. Older MCMC methods such as SGLD [7] and SGHMC [6] have found considerable success in the past but cannot be scaled to the number of dimensions we work with in deep learning. Since there were very little functional differences between SGD and these SG-MCMC techniques, learning rates and thereby learning rate scheduling policies had a great impact on training. This led to these optimisation strategies being extremely dependent on how the LR-scheduling is set up.

To address these issues, a *cyclical* SG-MCMC algorithm was proposed. This SG-MCMC algorithm has two phases, an *exploration* phase where the optimiser looks around to find other modes and a *sampling* phase where the optimiser collects samples. This is achieved in an automatic mode by setting up a periodic step-size sequence as follows:

$$\alpha_k = \frac{\alpha}{2}\left[cos\Big(\frac{\pi \text{mod}(k-1, \lceil K/M \rceil)}{\lceil K/M \rceil}\Big) + 1\right] \tag{3}$$

It is easy to see that the stepsize α_k is periodic in k. It starts at α_0 and then achieves 0, before starting to increase yet again. The periods with a large stepsize can be seen as an exploration phase where the optimiser is encouraged to more aggressively discover and explore other modes where as periods with a small stepsize focus more on sampling. This can also be interpreted as an SG-MCMC with warm restarts.

4.2. SWAG

The Stochastic Weight Averaging: Gaussian (SWA: Gaussian or SWAG) algorithm essentially models a loss valley or a mode curve using its first two statistical moments. These modes are then used as mean and low-rank covariance to model the distribution of parameters in this valley as a gaussian. Sampling from this gaussian gives us the samples that we need to perform BMA accordingly. To go into more detail regarding the methodology of SWAG, we first take a look at SWA [24] then observe the additions made in SWA: Gaussian.

4.2.1. Stochastic Weight Averaging

The main idea of SWA is to start at a pretrained point in the landscape where sufficiently low loss has been obtained and then use SGD to traverse in this locality. If the SWA trainer reaches θ_i at the ith iteration, then the SWA-trained parameters $\theta_{\text{SWA}} = \frac{1}{T}\sum_{i=0}^{T} \theta_i$. A high constant learning rate is used to ensure that the SWA-SGD traverses different areas in the landscape and does not just sit around a single optimum. It has been shown that SWA captures a more centered solution which is much more robust to the shift between train and test distributions compared to SGD point estimates.

Model	DenseNet-161	ResNet-152
SGD	0.9094	0.8716
SWA	0.8655	0.8682
SWAG-Diag	0.8559	0.8584
SWAG	**0.8303**	**0.8205**
SWA-Temp	0.8359	0.8226

Table 1. NLL on ImageNet

4.2.2. SWA: Gaussian

As a precursor to the complete SWAG algorithm, we take a look at SWAG-Diagonal, where the second moment of parameters is used to obtain a diagonal covariance. The mean we obtain from SWA and the covariance we obtain from the second moment gives us the gaussian from which we sample to perform BMA. Note that the covariance matrix of the SWA iterates can be written as:

$$\Sigma = \frac{1}{T-1}\sum_{i=1}^{T}(\theta_i - \theta_{\text{SWA}})(\theta_S WA - \theta_i)^T \qquad (4)$$

However, we do not have the value of θ at hand, hence we approximate it with the then-running average, like $\Sigma = \frac{1}{T-1}\sum_{i=1}^{T}(\theta_i - \overline{\theta_i})(\overline{\theta_i} - \theta_i)^T = \frac{1}{T-1}DD^T$, where D is the deviation matrix comprised of columns $D_i = (\theta_i - \overline{\theta_i}$. We only use the columns obtained from the last K epochs to limit the rank of our covariance matrix. This low-rank covariance matrix ($\frac{1}{K-1}\dot{D}\dot{D}^T$) is then combined with the diagonal covariance Σ_{diag} obtained earlier. To sample from SWAG, we hence do the following

$$\theta' = \theta_{\text{SWA}} + \frac{1}{\sqrt{2}}\Sigma_{\text{diag}}^{\frac{1}{2}}z_1 + \frac{1}{\sqrt{2(K-1)}}\dot{D}z_2 \qquad (5)$$

where $z_1, z_2 \sim \mathcal{N}(0, I)$. These samples are then averaged to produce an approximate BMA.

5. EXPERIMENTS

In this section we state a few key experiments performed by the authors.

5.1. SWAG

Experiments on ImageNet using DenseNet-161 and ResNet-152 - Table 1, Table 2

Experiments on CIFAR-10 and CIFAR-100 using PreResNet-164 and WideResNet28x10 - Table 3, Table 4

5.2. Cyclical SG MCMC

Experiments on ImageNet - Table 5

Experiments on CIFAR-10 and CIFAR-100 - Table 6

Model	DenseNet-161	ResNet-152
SGD	77.79	78.39
SWA	**78.60**	78.92
SWAG-Diag	78.59	78.96
SWAG	78.59	**79.08**
SWA-Temp	78.60	78.92

Table 2. Accuracy on ImageNet

Model	PreResNet-164	WideResNet28x10
SGD	95.49	96.41
SWA	96.09	**96.46**
SWAG-Diag	96.03	96.41
SWAG	96.03	96.32
SWA-Dropout	**96.18**	96.39
SWA-Temp	96.09	96.46

Table 3. Accuracy on CIFAR-10

Model	PreResNet-164	WideResNet28x10
SGD	78.50	80.76
SWA	**80.19**	82.40
SWAG-Diag	80.18	**82.40**
SWAG	79.90	82.23
SWA-Dropout		82.30
SWA-Temp	80.19	82.40

Table 4. Accuracy on CIFAR-100

	NLL	Top1
SGD	0.9595	76.046
Snapshot-SGDM	0.8941	77.142
SGHMC	0.9308	76.274
cSGHMC	**0.8882**	77.114

Table 5. Comparisons on testing set of ImageNet

	CIFAR-10	CIFAR-100
SGD	5.29	23.61
Snapshot-SGDM	5.17	22.98
SGHMC	4.46	20.83
cSGHMC	4.39	20.81
SGLD	5.20	23.23
cSGLD	4.29	20.55
SGHMC	4.93	22.60
cSGHMC	**4.27**	**20.50**

Table 6. Comparisons of test error(%) CIFAR-10 and CIFAR-100

6. DISCUSSION

The presence of flat regions in loss landscapes containing several modes is a strong case for bayesian model averaging over MAP estimation (i.e. SGD to identify a single value of $theta$). Every collection of parameters in a modal valley provides a functionally distinct explanation for the data, which means that a MAP or any point estimate is indeed an arbitrary choice of parameters if generalisation is our aim. To incorporate all of these multiple parameterisations into the final model, it is much more natural to take all of these reasons into account and perform a probability-weighted line-integral of sorts across the loss landscape.

Both the methodologies use the loss landscape methods as a probability distribution and sample from it to perform bayesian model averaging. Both the works show that Bayesian model averaging within the gaussian subspace can improve predictions over SGD or SWA solutions. One ([25]) uses the information contained in the SGD trajectory to efficiently approximate the posterior distribution over the weights of the neural network while the other([26]) plays with the learning rate scheduler to explore various modes in the multimodal loss surface of DNNs.

7. FUTURE WORKS

Cyclical Stochastic Gradient MCMC for Bayesian Deep Learning - we propose two practical techniques to improve estimation efficiency:

1. A system temperature for exploration and exploitation
2. A weighted combination scheme for samples collected in different cycles to reflect their relative importance

Results should be found on other bigger network architectures like VGG-16, ResNet-162, ResNet-152 etc. to reinforce and compare the results. Further, the mentioned methods must be compared and contrasted on other varied tasks like Natural Language Processing and text classification.

8. REFERENCES

[1] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun, "The loss surfaces of multilayer networks," in *Artificial intelligence and statistics*. PMLR, 2015, pp. 192–204.

[2] David JC MacKay, "A practical bayesian framework for backpropagation networks," *Neural computation*, vol. 4, no. 3, pp. 448–472, 1992.

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

[4] David JC MacKay, "Bayesian interpolation," *Neural computation*, vol. 4, no. 3, pp. 415–447, 1992.

[5] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," *arXiv preprint arXiv:1609.04836*, 2016.

[6] Tianqi Chen, Emily Fox, and Carlos Guestrin, "Stochastic gradient hamiltonian monte carlo," in *International conference on machine learning*. PMLR, 2014, pp. 1683–1691.

[7] 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)*. Citeseer, 2011, pp. 681–688.

[8] Stephan Mandt, Matthew D Hoffman, and David M Blei, "Stochastic gradient descent as approximate bayesian inference," *arXiv preprint arXiv:1704.04289*, 2017.

[9] Alex Graves, "Practical variational inference for neural networks," *Advances in neural information processing systems*, vol. 24, 2011.

[10] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[11] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra, "Weight uncertainty in neural network," in *International Conference on Machine Learning*. PMLR, 2015, pp. 1613–1622.

[12] Durk P Kingma, Tim Salimans, and Max Welling, "Variational dropout and the local reparameterization trick," *Advances in neural information processing systems*, vol. 28, pp. 2575–2583, 2015.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[14] Léonard Blier and Yann Ollivier, "The description length of deep learning models," *arXiv preprint arXiv:1802.07044*, 2018.

[15] Christos Louizos and Max Welling, "Multiplicative normalizing flows for variational bayesian neural networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 2218–2227.

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

[17] Yarin Gal, Jiri Hron, and Alex Kendall, "Concrete dropout," *arXiv preprint arXiv:1705.07832*, 2017.

[18] Alex Kendall and Yarin Gal, "What uncertainties do we need in bayesian deep learning for computer vision?," *arXiv preprint arXiv:1703.04977*, 2017.

[19] Jishnu Mukhoti and Yarin Gal, "Evaluating bayesian deep learning methods for semantic segmentation," *arXiv preprint arXiv:1811.12709*, 2018.

[20] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al., "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[21] Hippolyt Ritter, Aleksandar Botev, and David Barber, "A scalable laplace approximation for neural networks," in *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*. International Conference on Representation Learning, 2018, vol. 6.

[22] Hippolyt Ritter, Aleksandar Botev, and David Barber, "Online structured laplace approximations for overcoming catastrophic forgetting," *arXiv preprint arXiv:1805.07810*, 2018.

[23] Xi Chen, Jason D Lee, Xin T Tong, and Yichen Zhang, "Statistical inference for model parameters in stochastic gradient descent," *arXiv preprint arXiv:1610.08637*, 2016.

[24] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson, "Averaging weights leads to wider optima and better generalization," *arXiv preprint arXiv:1803.05407*, 2018.

[25] Wesley Maddox, Timur Garipov, Pavel Izmailov, Dmitry Vetrov, and Andrew Gordon Wilson, "A simple baseline for bayesian uncertainty in deep learning," *arXiv preprint arXiv:1902.02476*, 2019.

[26] Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson, "Cyclical stochastic gradient mcmc for bayesian deep learning," *arXiv preprint arXiv:1902.03932*, 2019.