SURROGATED ASSISTED BAYESIAN NEURAL NETWORK FOR GEOLOGICAL MODELS

Sean Luo

Supervisors: Rohitash Chandra, Richard Scalzo

School of Mathematics and Statistics
UNSW Sydney

January 2022

Submitted in partial fulfillment of the requirements of the degree of
Bachelor of Science with Honours

Plagiarism statement

I declare that this thesis is my own work, except where acknowledged, and has not been submitted for academic credit elsewhere.

I acknowledge that the assessor of this thesis may, for the purpose of assessing it:

- Reproduce it and provide a copy to another member of the University; and/or,
- Communicate a copy of it to a plagiarism checking service (which may then retain a copy of it on its database for the purpose of future plagiarism checking).

I certify that I have read and understood the University Rules in respect of Student Academic Misconduct, and am aware of any potential plagiarism penalties which may apply.

By signing this declaration I am agreeing to the statements and conditions above.

Signed: ______________________ Date: ______________

Acknowledgements

By far the greatest thanks must go to my supervisor for the guidance, care and support they provided.

Thanks to my family for supporting me to advance in higher education.

Thanks go to Fred Flintstone and Robert Taggart for allowing this thesis style to be shamelessly copied.

Sean Luo, Day Month Year.

Abstract

This thesis is an extension to the surrogate assisted parallel tempering bayesian deep learning framework that uses a proposal surrogate to sample from that simulates the expensive langevin proposal which requires expensive forward passes.

the surrogate to generate weight and its corresponding numerator and denominator during accept reject to avoid all the extra forward pass and backward loss compare to normal DNN training. The input would be a window of past model parameters and a random vector from normal distrbution, The output would be a new set of model parameters, the corresponding numerator and denominator of accept reject step.

Contents

Chapter 1

Introduction (Currently a collection of messy notes)

In the past decade there has being some progression in the field of bayesian deep learning. Bayesian deep learning offers an intrinsic way to ensemble models that helps to quantify the posterior uncertainty in model parameters and prediction. Applying tradition bayesian inference techniques such as Monte Carlo sampling in deep neural networks have multiple challenges, mainly revolving around computational expenses due to the evaluation speed of large dataset, exponentially increasing rejection rate in MCMC sampler as the parameter space dimensions grows with large model, difficulty to evaluate uncertainty in realtime data, efficient proposal convergence and others. Variational bayesian inference methods are faster than sampling, but it does not offer an exact approximation of the target distribution.
Ideas explored:

Pure Bayesian:

1. Using stochastic mini batches with Langevin dynamic by Welling et al [1].
2. Using langevin dynamic Dynamic as proposal to predict time series by chandra et al [2]
3. Using parallel tempering to speed up exploration and exploitation by chandra [3]
4. Using surrogate models to simulate the posterior likelihood $p(D|\theta)$ to speed up evaluation, by chandra. [4]
5. Surrogate models are selected from variantional inference category, including, GAN, VAE, and Normaling Flow.
6. Train a WGAN to approximate the joint prior distribution of parameter and state, and sample from latent space instead. This is useful when the domain has high dimension of state and input parameters that each have different prior distributions. [5]. youtube video of paper presentation
7.

Modification to frequentist models that turns into bayesian:

1. Using dropout as intrinsic ensemble (bayesian?).
2. Using stochastic weight averaging withe gaussian noise (SWAG) for bayesian model averaging by approximating a normal weight posterior initialized on a pretrained solution. Bayesian model averaging in general prevents overfitting and overconfidence - Maddox et al [6]

Other assumptions and constraints

1. weight decay as normal prior on weights

1.1 Langevin Dynamics

In standard LD or 1 Leap frog step of Hamilton Dynamics, there is 1 distinctive forward+backward pass in a leap frog step:

1. w_last forward (for posterior likelihood $p(data|w\_last)$)
2. w_last backward (for proposal distribution mean w_last_bar,$w\_star \sim N(w\_last\_bar, \sigma)$)
3. w_star forward (for posterior likelihood $p(data|w\_star)$)
4. w_star backward (for proposal distribution mean w_star_bar,$w\_last \sim N(w\_star\_bar, \sigma)$)

w_last are reused from last iteration.

By using a surrogate model to approximate the posterior likelihood $p(data|w_star)$ we replaced step 3's forward pass.

Now we propose to replace step 4's backward pass with a GAN that generates the weights.

The believed benefits includes:

1. faster evaluation.

For this GAN to be benificial, it has several constraints:

1. shallower than original network.
2. approximates the full posterior $p(w|data)$.

1.2 Generative Models

The target loss function for generative model in continuous case (we care for continuous case only as we plan to use it to generate the weights) in unsupervised setting is $L(D) \approxeq \frac{1}{N}\sum_{i=1}^{N} -\log(p_\theta(\tilde{x}^{(i)})) + c$. Where $\tilde{x}^{(i)} = x^i + u$ with $u \sim U(0, a)$ and $c = -M \times \log a$, a is dependent on data discretization and M is the dimension of x. This loss function encourages the network to learn to distribute data across the domain of the standard normal distribution. When there is an underlying distribution to approximate, we need to utilizes distance/divergence measures in loss function to have convergent distribution.

VAE is a shrinkage training model that has two shrinker that trains to a latent space of mean and variance of normal distribution. from the two latent vector it then trains to simulate a proper input.

GAN have discriminator and generator, generator samples from standard normal and maps it via a network to output space, discriminator compare the overall generated result to dataset. The training is conducted via a maximizing loss for discriminator and minimizing loss on generator.

Normalizing Flow is composed of multiple inversible layers that helps to provide a tractable posterior likelihood on weights. Due to the decomposition of Jacobian matrix during gradient calculation, there is a significant reduction in memory requirements. Specifically the GLOW model proposed by OpenAI [7] uses 1x1 convolution to simulate channel switching to boost training performance. Another paper from USTC improves upon this by using matrix exponentials (no idea what it means at the moment)

In flow and autoencoder we are dealing with distributions directly in the latent space (flow is autoencoder with only one network) and Arjovsky (NYU) [8] et al proposed in 2017 to simulate the Wasserstein distance (in general Earth Mover

Distance) instead as a loss measure when training GAN.(may be it can be applied to VAE or NF? Or maybe it already is, base on how I see the implementation).

It is stated that divergence measures such as Jensen or KL fails to pickup the low dimensional manifold of a lot of problems' distributions (Q: are the distribution of weights of DNN parameters supported by low dimensional manifolds?), further in practice, other measures when training GAN produces in saturated discriminator that fails to provided any sensible gradient afterwards, that leads to mode collapse. WGAN resolves this as further training of critic/(score based discriminator instead of class based discriminator) still provides reasonable gradient. WGAN's weakness include failure when using momentum based optimizer (use RMSProp instead), and the clamping of hyperparameters to maintain lipschitz (what does this mean?) seems to be troublesome as it is problem specific. This is addressed by Gulrajani [9] using gradient penalty normalisation to replace the clipping that allows the loss to be normally distributed instead of bernoulli distrbuted at the two clipping points. This paper also states that WGAN has comparable epoch performance to WCGAN while offering better stability, the caveat is a slower wallclock time.

More info here GAN vs WGAN

short read on improved WGAN - Gradient Penalty

1.3 low dimensional manifold

(Imagine your patterned bedsheets. They've got a nice plaid grid look to them, very easy to predict what the next few centimeters of material look like.

Now imagine someone tied them in a knot, tore them a little, balled them up, and made them a crumpled mess, then handed them to you. Now if you were to look at the plaid pattern, it would be very difficult to discern how the pattern is, or predict what will be the color of any point of the arbitrary 3D ball of mess you've got. But if you untangled it, you could clearly see the pattern again.

The data is a bunch of colors in an arbitrary 3D ball of mess with very difficult to discern patterns. But the data lies on a low dimensional manifold (the 2D bedsheet). If you could figure out the manifold (flatten the bedsheet), the data will be easy to model.) *explaination on reddit*

(stil have problem understanding the math behind it, but implementation wise i think it just removes the bounded final activation and uses the score as a loss)

1.4 Notes:

Swag : approximates local loss distribution (posterior?) using standard normal momentum distribution. Propose -¿ Accept Reject: replacing forward pass for proposal Surrogate by Chandra et al: evaluate proposal and spit out its posterior likelihood i.e.

$min(1, pi(w*|x)/pi(wi|x) * q(wi|w*)/q(w*|wi))$

$min(1, p(x|w*)/p(x|wi) * p(w*|x)/p(wi|x) * q(wi|w*)/q(w*|wi))$

$p(w|x)$ and $q(wi|w*)$ are both easy to compute, the first is a loss, the second is proposal distribution (using tractable distribution like normal that s easy to evaluate) $p(x|w)$ requires forward pass, so we evaluate it using surrogate

$https://www.lpsm.paris/pageperso/merle/slides_3.1.TV.pdf$ stuff for distance understanding list to read Tractable Approximate Gaussian Inference for Bayesian Neural Networks

Adversarial Variational Bayes: Unifying Variational Autoencoders and Generative Adversarial Networks

GAN vs VAE

Distributed Bayesian optimisation framework for deep neuroevolution

Bridging the Gap Between f-GANs and Wasserstein GANs Stronger and Faster Wasserstein Adversarial Attacks

Modulating Surrogates for Bayesian Optimization

GENERATIVE ADVERSARIAL LEARNING OF MARKOV CHAINS

on the quantitative analysis of decoder based generative models

Packages of interest include pymc, pyro, pytorch, (maybe sydney-machine-learning/pingala?)

1.5 Proposal

Chapter 2

Sampling

Chapter 3

Method

Chapter 4

Experiment

Chapter 5

Analysis

Chapter 6

Discussion

Chapter 7

Conclusion

References

[1] Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *ICML*, 2011.

[2] Rohitash Chandra, Lamiae Azizi, and Sally Cripps. Bayesian neural learning via langevin dynamics for chaotic time series prediction. In *ICONIP*, 2017.

[3] Rohitash Chandra, Konark Jain, Ratneel Vikash Deo, and Sally Cripps. Langevin-gradient parallel tempering for bayesian neural learning. *CoRR*, abs/1811.04343, 2018.

[4] Rohitash Chandra, Konark Jain, and Arpit Kapoor. Surrogate-assisted parallel tempering for bayesian neural learning. *CoRR*, abs/1811.08687, 2018.

[5] Nikolaj T. Mücke, Benjamin Sanderse, Sander Bohté, and Cornelis W. Oosterlee. Markov chain generative adversarial neural networks for solving bayesian inverse problems in physics applications, 2021.

[6] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[7] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[8] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017.

[9] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.