A PROBABILISTIC APPROACH TO SELF-SUPERVISED LEARNING USING CYCLICAL STOCHASTIC GRADIENT MCMC

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper we present a practical Bayesian self-supervised learning method with Cyclical Stochastic Gradient Hamiltonian Monte Carlo (cSGHMC). Within this framework, we place a prior over the parameters of a self-supervised learning model and use cSGHMC to approximate the high dimensional and multimodal posterior distribution over the embeddings. By exploring an expressive posterior over the embeddings, Bayesian self-supervised learning produces interpretable and diverse representations. Marginalizing over these representations yields an improvement in semi-supervised learning and out-of-distribution detection tasks. We provide experimental results on multiple semi-supervised classification tasks on Cifar10 and Cifar100. Moreover, we demonstrate the effectiveness of the proposed method in out-of-distribution detection using the SVHN dataset.

1 INTRODUCTION

Self-supervised learning is a learning strategy where the data themselves provide the labels (Jing & Tian (2020)). The aim of self-supervised learning is to learn useful representations of the input data without relying on human annotations (Zbontar et al. (2021)). Since they do not rely on annotated data, they have been used as an essential step in many areas such as natural language processing, computer vision and biomedicine(Jospin et al. (2020)).

Contrastive methods (Chen et al. (2020)) are one of the promising self-supervised learning approaches which learn representations by maximizing the similarity between embeddings obtained from different distorted versions of an image (Zbontar et al. (2021)). Several tricks are proposed to overcome the issue of feature collapse. These include using negative samples in simCLR (Zbontar et al. (2021)) and stop gradient in BYOL (Grill et al. (2020)).

Self-supervised models are often trained using stochastic optimization methods which approximate the *distribution* over the parameters with a *point mass* ignoring the uncertainty in the parameter space. Indeed if the regularizer imposed on the model parameters is viewed as the the log of a prior on the distribution of the parameters, optimizing the cost function may be viewed as a maximum *a-posteriori* (MAP) estimate of model parameters (Li et al. (2016)). Bayesian methods provide principled alternatives that model the whole posterior over the parameters and account for model uncertainty in the parameter space (Zhang et al. (2020)).

Exact Bayesian learning of Deep Neural Networks is generally intractable, hence Bayesian deep learning models use approximation methods like variational inference (VI, Blundell et al. (2015)) or MCMC methods (Neal (2012)) to capture the posterior over the parameters and estimate model uncertainty. While VI methods usually approximate a single mode, MCMC methods are used for sampling from different modes (Jospin et al. (2020)). In a recent line of work, Stochastic Gradient Markov Chain Monte Carlo (SG-MCMC) methods (Welling & Teh (2011), Chen et al. (2014), Ma et al. (2015)) were proposed which couple MCMC with SGD to provide a promising sampling approach to inference in Bayesian deep learning for large datasets (Welling & Teh (2011)). In another work, Zhang et al. (2020) proposed Cyclical Stochastic Gradient MCMC (cSG-MCMC) in order to explore a highly multimodal parameter space given a realistic computational budget (Zhang et al. (2020)).

In this paper we aim to adapt Bayesian supervised learning concepts to self-supervised learning to make a self-supervised learning model fully probabilistic using cSG-MCMC. Our motivation comes from the fact that the posterior distribution over the parameters of a self-supervised learning model may be multimodal and thus insufficiently represented by a single point estimate. By exploring the posterior distribution over the parameters instead of point mass we aim to improve performance in downstream tasks. Moreover, the optimization step in cSG-MCMC involves injection of Gaussian noise to the parameter update of SGD which helps to alleviate the feature collapse issue in the contrastive methods and make the features more informative (Li et al. (2016)).

In this paper, we propose a simple Bayesian formulation for self-supervised learning with a specific family of cSG-MCMC methods called Cyclical Stochastic Gradient Hamiltonian Monte Carlo (cSGHMC) (Zhang et al. (2020)). Within this framework, we use BYOL as a self-supervised learning model which allows to incorporate Bayesian learning of an approximate posterior over the parameters instead of MAP. Our experimental results indicate that by integrating a Bayesian learning we can achieve better performance in downstream tasks including classification and out-of-distribution detection. The simplicity of the proposed approach is one of its greatest strengths.

2 Problem Statement

Given a dataset $\mathcal{D}$, a self-supervised learning model $\mathcal{F}_\theta$ parameterized by θ, aims to produce a representation Z_θ by solving a predefined proxy task. In this paper we wish to learn a distribution over the embeddings Z_θ by placing a prior over the parameters θ and using Bayesian learning instead of MAP estimation. For learning the representations we use BYOL, a recent self-supervised learning method based on contrastive learning. To obtain the distribution over the embeddings, we use cSGHMC. In the following, first we describe the self-supervised learning model to learn representations. Then, we describe cSGHMC and highlight how it allows to obtain a distribution over the embeddings.

2.1 Self supervised learning

The aim of contrastive learning is to learn representations by contrasting two augmented views of an image. Particularly BYOL learns representations by reducing a contrastive loss between two neural networks referred to as online network $\mathcal{F}_\theta$ (parameterized by θ) and target network $\mathcal{F}_\xi$ (parameterized by ξ). Each network consists of three components, an encoder $f(.)$ (e.g., Resnet-18), a projection head $g(.)$ (e.g., an MLP) and a prediction head $q(.)$ (e.g., an MLP). For a given minibatch $X = \{x_i\}_{i=1}^N$ sampled from a dataset $\mathcal{D}$ it produces two distorted views, $t(X)$ and $t'(X)$, via a distribution of data augmentations $\mathcal{T}$. The two batches of distorted views then are fed to the online network and the target network, producing batches of embeddings, Z_θ and Z_ξ, respectively. These features are then transformed with the projection heads into Y_θ and Y_ξ. The online network then outputs a prediction Q_θ of Y_ξ using the prediction head $q_\theta(.)$. Finally the following mean squared error between the normalized predictions $\bar{Q}_\theta$ and target projections Y_ξ is defined:

$$\mathcal{L}_{\theta,\xi} = \|\bar{Q}_\theta - \bar{Y}_\xi\|^2 = 2 - 2.\frac{\langle \bar{Q}_\theta, \bar{Y}_\xi \rangle}{\|\bar{Q}_\theta\|.\|\bar{Y}_\xi\|}. \tag{1}$$

$\tilde{\mathcal{L}}_{\theta,\xi}$ is computed by separately feeding $t'(X)$ to the online network $\mathcal{F}_\theta$ and $t(X)$ to the target network $\mathcal{F}_\xi$. Then, at each training step, a stochastic optimization step is performed to minimize

$$\mathcal{L}_{\theta,\xi}^{\text{BYOL}} = \mathcal{L}_{\theta,\xi} + \tilde{\mathcal{L}}_{\theta,\xi} \tag{2}$$

The gradient is taken only with respect to θ. So, the parameter update is as follows:

$$\begin{aligned} \theta &\leftarrow \text{optimizer}(\theta, \nabla_\theta \mathcal{L}_{\theta,\xi}^{\text{BYOL}}). \\ \xi &\leftarrow \tau\xi + (1-\tau)\theta, \end{aligned} \tag{3}$$

where the weights ξ are an exponential moving average of the online network's parameters θ with a target decay rate $\tau \in [0, 1]$. At the end of training, the encoder $f_\theta(.)$ is used for the downstream task. During training only the parameters θ of the online network $\mathcal{F}_\theta$ are updated.

2.2 POSTERIOR SAMPLING USING CSGHMC

In the Bayesian paradigm, for a given dataset $\mathcal{D} = \{x_i\}_{i=1}^n$ and a θ-parameterized model, the following *a-posterior distribution* over θ is computed using Bayes' rule as: $p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta)$, where $p(\theta)$ is a *prior* assigned to the parameters θ and $p(\mathcal{D}|\theta)$ is the likelihood.

In MAP optimization, the prior has the role of a regularizer and the likelihood has the role of a cost function. An optimizer is optimized to find the MAP solution which is amenable to the parameter update:

$$\Delta\theta = -\frac{\ell}{2}(\frac{n}{N}\sum_{i=1}^{N}\nabla_\theta \log p(x_i|\theta) + \nabla_\theta \log p(\theta)), \tag{4}$$

for a given randomly sampled mini-batch $X = \{x_i\}_{i=1}^N \subset \mathcal{D}$ and learning rate ℓ.

In contrast to MAP optimization, in the Bayesian paradigm the model explores the distribution over the model parameters. Welling & Teh (2011) showed that this distribution can be approximated using Stochastic Gradient Langevin Dynamics (SGLD) by injecting Gaussian noise to the parameter updates of SGD so that they do not collapse to just the MAP solution. This leads to the following parameter update:

$$\Delta\theta = -\frac{\ell}{2}(\frac{n}{N}\sum_{i=1}^{N}\nabla_\theta \log p(x_i|\theta) + \nabla_\theta \log p(\theta)) + \epsilon; \;\; \epsilon \propto \mathcal{N}(0, \ell I) \tag{5}$$

SGHMC (Chen et al. (2014)) is an improved counterpart of SGLD which introduces a momentum variable $\boldsymbol{m}$. The posterior sampling is done using the following update rule:

$$\begin{aligned} \boldsymbol{m} &= \beta\boldsymbol{m} - \frac{\ell}{2}(\frac{n}{N}\sum_{i=1}^{N}\nabla_\theta \log p(x_i|\theta) + \nabla_\theta \log p(\theta)) + \epsilon; \;\; \epsilon \propto \mathcal{N}(0, (1-\beta)\ell I) \\ \theta &= \theta + \boldsymbol{m} \end{aligned} \tag{6}$$

where β is the momentum term. Equations 5 and 6 guarantee convergence to the true posterior as long as the learning rate ℓ decreases towards zero. Zhang et al. (2020) showed that replacing the traditional decreasing learning rate schedule in SGHMC with a cyclical variant allows to explore multimodal posterior distributions and developed cSGHMC. In this paper we apply cSGHMC to take samples from the posterior distribution.

3 POSTERIOR OVER REPRESENTATIONS

To infer a posterior over the embeddings, we place a prior α over the parameters θ of the online network $\mathcal{F}_\theta$. By placing a distribution over θ, we induce a distribution over an infinite space of online networks $\mathcal{F}_\theta$. This results in a distribution over embeddings Z_θ. Sampling from this distribution corresponds to sampling from the following conditional posterior:

$$Z_\theta \propto p(\theta|X) \propto p(X|\theta)p(\theta|\alpha), \tag{7}$$

where X is a mini-batch. Equation 7 can be interpreted intuitively as follows. We sample weights θ from the prior $p(\theta|\alpha)$. Then we condition on this sample of the weights form a particular online network $\mathcal{F}_\theta$ which is used to produce embedding Z_θ by minimizing the loss function $\mathcal{L}^{\text{Byol}}_{\theta,\xi}$.

To take samples from the posterior in equation 7 we use cSGHMC and the update rule in equation 6. Algorithm 1 describes our proposed method to sample from the posterior. Algorithm 1 produces samples from the posterior over the parameters θ of the online network $\mathcal{F}_\theta$ and consequently a distribution over embeddings Z_θ, since we compute the gradients of the loss with respect to the parameters that we are sampling.

Our proposed probabilistic approach is a natural Bayesian generalization of MAP optimization. Indeed, if one performs MAP optimization using SGD in Algorithm 1 instead of posterior sampling, one approximates the whole posterior over θ with a point mass. Sampling from the whole posterior over embeddings versus approximating this with a point mass allows estimating uncertainties in the embedding space.

Algorithm 1 Probabilistic Self-Supervised Learning

Require: ℓ initial learning rate, β momentum term, K number of training iterations
Ensure: sequence $\theta^1, \theta^2, ...$
for $k = 1 : K$ **do**
- sample mini-batch $X = \{x_i\}_{i=1}^N$ and augmentations $t \sim \mathcal{T}, t' \sim \mathcal{T}$
- compute $\mathcal{L}_{\theta,\xi}^{\text{BYOL}}$ based on equation 2

$\ell \leftarrow C(k)\ell$ ▷ update learning rate using cyclic modulation
$\boldsymbol{m} \leftarrow \beta \boldsymbol{m} - \frac{\ell}{2}\nabla_\theta \mathcal{L}_{\theta,\xi}^{\text{BYOL}} + \epsilon; \;\; \epsilon \propto \mathcal{N}(0, (1-\beta)\ell I)$
$\theta \leftarrow \theta + \boldsymbol{m}$ ▷ update parameters θ using update rule 6
$\xi \leftarrow \tau\xi + (1-\tau)\xi$
if end of cycle **then**
yield θ
end if
end for

Marginalizing over representations: Once pretraining is done, we can marginalize the posterior over θ for downstream tasks. To compute the predictive distribution for a new instance x^* we use a model average over all collected samples with respect to the posterior over θ.

$$p(y^*|x^*, \mathcal{D}) = \int p(y^*|x^*, \theta)p(\theta|\mathcal{D})d\theta \approx \frac{1}{T}\sum_{t=1}^{T} p(y^*|x^*, \theta^{(k)}), \theta^{(k)} \propto p(\theta|\mathcal{D}). \tag{8}$$

We will see that this model average is effective for semi-supervised learning performance as well as out-of-distribution detection task.

4 Experiments

In this section, we present our experimental results. We evaluate proposed method on several tasks including semi-supervised learning and out-of-distribution detection. First, we describe our experimental setup. Then, we evaluate our model using semi-supervised setting, and lastly, we evaluate our model using out-of-distribution examples. We implemented the code using PyTorch (Paszke et al. (2017)).

4.1 Experimental Setup

We conduct our experiments on Cifar-10 and Cifar-100 (Krizhevsky & Hinton (2009)). For self-supervised learning model we use ResNet-18 (He et al. (2016)) as the encoder. Following the original setting of BYOL, we use 2-layer MLPs as the projection and prediction heads. We use terms 'representations' or 'embeddings' interchangeably as the output of the encoder. The representations are used for downstream tasks. We use the same set of data augmentations in Grill et al. (2020) on both Cifar-10 and Cifar-100, consists of random cropping and resizing with a random horizontal flip, followed by a color distortion and a grayscale conversion.

For cSGHMC, we place a $\mathcal{N}(0, I)$ prior on the parameters of the online network and use a cosine decay learning rate schedule (Zhang et al. (2020)) with cycle length of 50. We use momentum term of 0.9 and a batch size of 64. For pretraining on Cifar-10 we use initial learning rate $\ell = 0.5$ and inject Gaussian noise at epoch 40. For pretraining on Cifar-100 we use initial learning rate $\ell = 0.4$ and inject Gaussian noise at epoch 47. We collect 3 samples at the end of each cycle. We pretrain models on 95% of training data of Cifar-10 and Cifar-100 without labels, leaving 5% as validation set to monitore overfitting. In cSGHMC it is important to scale the noise and prior with the length of the dataset (Florian et al. (2020)).

We compare with a self-supervised model trained with SGD optimizer. For the SGD baseline, we use SGD with the same momentum term. We use fixed learning rate schedule with $\ell = 0.03$ and the same batch size. We do not use any weight decay. We pretrained all models on Cifar-10 and Cifar-100 for the same number of epochs 1500 and unless otherwise stated we take parameter samples starting from epoch 1200 resulting in 21 samples in Cifar-10 and from epoch 1300 resulting

Table 1: Top-1 accuracy in semi-supervised learning on Cifar-10 using 1%, 10%, 50% and 100% training examples (%).

Dataset	CIFAR-10			
Method	1%	10%	50%	100%
BYOL	73.79 ± 0.23	78.17 ± 0.29	80.29 ± 0.20	81.35 ± 0.09
BBYOL	74.04 ± 0.67	78.26 ± 0.57	$\mathbf{81.58 \pm 0.37}$	82.85 ± 0.30
BYOL-ENS	$\mathbf{74.41 \pm 0.31}$	$\mathbf{78.39 \pm 0.23}$	80.35 ± 0.32	81.66 ± 0.24
BBYOL-ENS	74.32 ± 0.60	78.29 ± 0.67	81.37 ± 0.35	$\mathbf{83.21 \pm 0.29}$

Table 2: Top-1 accuracy in semi-supervised learning on Cifar-100 using 10%, 50% and 100% training examples (%).

Dataset	CIFAR-100		
Method	10%	50%	100%
BYOL	40.87 ± 0.10	48.92 ± 0.22	51.71 ± 0.39
BBYOL	39.58 ± 0.86	48.34 ± 0.48	51.33 ± 0.38
BYOL-ENS	$\mathbf{41.45 \pm 0.10}$	49.53 ± 0.26	52.50 ± 0.23
BBYOL-ENS	40.51 ± 0.92	$\mathbf{49.84 \pm 0.48}$	$\mathbf{53.31 \pm 0.42}$

in 15 samples in Cifar-100. We refer to BYOL pretrained with cSGHMC as BBYOL, and its deterministic counterpart pretrained with SGD as BYOL. We also take samples in MAP optimization during pertaining at the same epochs as in cSGHMC and take marginalizing over these parameters as another baseline. If we marginalize over the representations in downstream task, we use terms BBYOL-ENS and BYOL-ENS respectively.

4.2 Semi-Supervised Evaluation

One of the common ways to evaluate the quality of representations learnt via a self-supervised learning model is in semi-supervised learning task. In this task a pretrained model is fine-tuned on subsets of original training dataset with labels. We use subsets of size 1%, 10%, 50% and 100% of Cifar-10 labeled training data. Table 1 indicates results obtained on the Cifar-10 test set averaged over 4 runs. First we observe that representations obtained from BBYOL improve performance in all subsets of labeled data. The improvement is higher when we fine tun on larger portion of data (e.g. 50% and 100%). The second observation is that marginalizing over the representations in BYOL-ENS also improves performance specially in low data regime. It is due to the nature of contrastive loss which induces diversity in the parameter space. Whenever the loss is not too high, marginalizing over these representations improve the performance. Further, BBYOL-ENS significantly outperforms other baselines, when we fine tune on 100% of labeled data. In order to investigate the generalizability of our findings, we also provide the results obtained on the Cifar-100 test set in Table 2. We fine-tune a pretarined model on Cifar-100 on the splits of 10%, 50% and 100% of training data with labels and report results on the test set averaged over 3 runs. Consistent with our previous results, marginalizing over representations improve performance in both BYOL-ENS and BBYOL-ENS with a better performance for BBYOL-ENS when fine-tuned on larger splits of data. We should highlight this point that results on Cifar-100 (and also Cifar-10) can be improved if we use a larger architecture, but our goal here is not benchmarking against other self-supervised learning models, rather we want to show the effectiveness of our proposed method compared to the baselines which were pretrained with the same architecture and setting.

4.3 Out-of-Distribution Detection

To further explore the diversity of embeddings, we consider the out-of-distribution (OOD) detection task. We fine-tune a pretrained model on CIFAR-10 over 100% split of its train dataset and test on SVHN dataset (Netzer et al. (2011)). We estimate the entropy of the predictive distribution on SVHN and plot its empirical CDF for BBYOL-ENS and BYOL in Figure 1. For the unseen data, we

expect the model indicates max entropy and the probablity of low entropy should be lower (Zhang et al. (2020)).

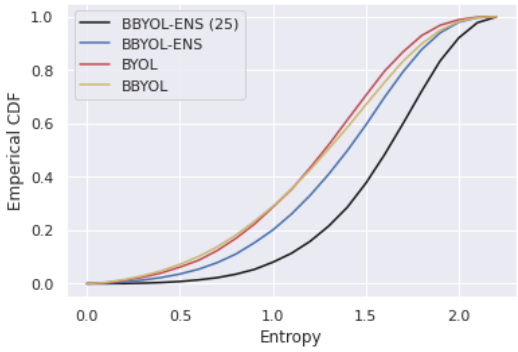

Figure 1: Empirical CDF for the predictive entropy. Models are fine-tunned on 100% split of Cifar-10 and tested on SVHN. BYOL-ENS (25) is marginalizing over 25 embeddings.

In Figure 1, BBYOL-ENS (25) indicates marginalizing over 25 samples where we took one sample at each cycle starting from epoch 300. We observe that all models assign high entropy to unseen data, since self-supervised learning improves model uncertainty (Hendrycks et al. (2019)). It is seen that BBYOL-ENS assigns higher entropy to unseen data, specially when we take more samples in different cycles. It indicates that the embeddings produced by sampling from the posterior come from different modes and provide different characterization of training data. Indeed different embeddings can provide different predictions on OOD, leading to better uncertainty estimation.

Table 3: Results for OOD detection for various settings. Models are fine-tunned over 100% labeled data of CIFAR-10 and tested on SVHN. All values are in percentages. $\uparrow$ indicates larger value is better, and $\downarrow$ indicates lower value is better.

Method	NLL $\downarrow$	AUROC $\uparrow$
BYOL	5.25	46.92
BYOL-ENS	4.77	46.90
BBYOL	4.63	**47.73**
BBYOL-ENS	**4.22**	47.25

5 Discussion and Future Work

In this paper, we propose to explore the distribution over the representations using cSGHMC instead of estimating the posterior with a MAP solution. Our experimental results indicate that samples taken from the posterior provide meaningfully different representations which leads to improved accuracy and uncertainty estimation in downstream tasks. Capturing the posterior over the embeddings can also provide the possibility of uncertainty estimation in the embedding space, opening new directions for practitioners in the field.

For future work we would like to further investigate the following: (i) We would like to analyze the notion of uncertainty in the embedding space and its relation with uncertainty in the predictive space. (ii) We would like to explore the possibility of using a Bayesian self-supervised learning model as a prior for a Bayesian downstream task.

References

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pp. 1613–1622. PMLR, 2015.

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In Eric P. Xing and Tony Jebara (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1683–1691, Bejing, China, 22–24 Jun 2014. PMLR. URL https://proceedings.mlr.press/v32/cheni14.html.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Wenzel Florian, Roth Kevin, Veeling Bastiaan S, Światkowski Jakub, Tran Linh, Mandt Stephan, Snoek Jasper, Salimans Tim, Jenatton Rodolphe, and Nowozin Sebastian. How good is the bayes posterior in deep neural networks really? *arXiv preprint arXiv:2002.02405*, 2020.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. *Advances in neural information processing systems*, 32, 2019.

Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):4037–4058, 2020.

Laurent Valentin Jospin, Wray L. Buntine, Farid Boussaïd, Hamid Laga, and Mohammed Bennamoun. Hands-on bayesian neural networks - a tutorial for deep learning users. *CoRR*, abs/2007.06823, 2020. URL https://arxiv.org/abs/2007.06823.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009.

Chunyuan Li, Andrew Stevens, Changyou Chen, Yunchen Pu, Zhe Gan, and Lawrence Carin. Learning weight uncertainty with stochastic gradient mcmc for shape classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5666–5675, 2016.

Yi-An Ma, Tianqi Chen, and Emily B. Fox. A complete recipe for stochastic gradient MCMC. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 2917–2925, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/9a4400501febb2a95e79248486a5f6d3-Abstract.html.

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

Yuval Netzer, Tao Wang, Adam Coates, A. Bissacco, Bo Wu, and A. Ng. Reading digits in natural images with unsupervised feature learning. 2011.

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. In *NIPS 2017 Workshop on Autodiff*, 2017. URL https://openreview.net/forum?id=BJJsrmfCZ.

Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pp. 681–688, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pp. 12310–12320. PMLR, 2021.

Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical stochastic gradient mcmc for bayesian deep learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkeS1RVtPS.