Bayesian Optimization Using Hamiltonian Dynamics for Sparse Artificial Neural Networks

Mohamed Fakhfakh, Bassem Bouaziz, Faïez Gargouri, Lotfi Chaâri

BAYESIAN OPTIMIZATION USING HAMILTONIAN DYNAMICS FOR SPARSE ARTIFICIAL NEURAL NETWORKS

Mohamed Fakhfakh[1,2]*, Bassem Bouaziz*[2]*, Faiez Gargouri*[2] *and Lotfi Chaari*[1]

[1]University of Toulouse, INP, IRIT , France
firstname.lastname@toulouse-inp.fr
[2]University of Sfax - MIRACL laboratory - Tunisia
firstname.lastname@isims.usf.tn

ABSTRACT

Artificial Neural Networks (ANN) are being widely used in supervised Machine Learning (ML) to analyse signals or images for many applications. Using a learning database, one of the main challenges is to optimize the network weights. This optimization step is generally performed using a gradient-based approach with a back-propagation strategy. For the sake of efficiency, regularization is generally used. When non-smooth regularizers are used especially to promote sparse networks, this optimization becomes challenging. Classical gradient-based optimizers cannot be used due to differentiability issues. In this paper, we propose an MCMC-based optimization scheme formulated in a Bayesian framework. Hamiltonian dynamics are used to design an efficient sampling scheme. Promising results show the usefulness of the proposed method to allow ANNs with low complexity levels reaching high accuracy rates.

***Index Terms*—** Artificial Neural Networks, optimization, deep learning, LSTM, MCMC, Hamiltonian dynamics.

1. INTRODUCTION

Deep Learning (DL) [1] has grown at a remarkable rate, attracting a great number of researchers and practitioners. It has become one of the most popular research directions in many applications such as recognition, medical diagnoses, self-driving cars, recommendation systems, etc [2]. The essence of most architectures is to build an optimization model and learn the parameters from the given data. From the perspective of the gradient information in optimization, optimization methods can be divided into three categories [3]: 1) first-order optimization methods such as stochastic gradient; 2) high-order optimization methods, mainly Newton's algorithm; and 3) heuristic derivative-free optimization methods. In parallel, Bayesian techniques have demonstrated their ability to provide efficient optimization algorithms with better convergence. Recently, sampling using Hamiltonian dynamics [4] has been investigated developing the so called Hamiltonian Monte Carlo (HMC) sampling. A more sophisticated algorithm has been proposed in [5, 6] called non-smooth Hamiltonian Monte Carlo (ns-HMC) sampling. This method solves the problem of HMC schemes that cannot be used in the case of exponential distributions with non-differentiable energy function. In this paper, we investigate the use of ns-HMC for the learning process of Artificial Neural Networks (ANNs). Specifically, we propose a Bayesian optimization method to minimise the target cost function and derive the optimal weights vector. Indeed, we demonstrate that using the proposed method leads to high accuracy results, which cannot be reached using competing optimizers. The rest of this paper is organized as follows. The addressed problem is formulated in Section 2. The proposed efficient Bayesian optimization scheme is developed in Section 3 and validated in Section 4. Finally, conclusions and future work are drawn in Section 5.

2. PROBLEM FORMULATION

One of the most well-known processes in designing an efficient artificial neural network is weights optimization. For example, to solve an image classification problem, the ANN weights vector W is updated during the learning phase by minimizing the difference between the ground truth and the labels estimated by the network. Regularization can also be done for the sake of efficiency in order to have a more accurate weights arrangement. Smooth regularizers are used in this scenario, such as the $\ell 2$ norm. Gradient-based algorithms could still be used. However, if one wants to promote sparse networks [7], sparse regularizations must be used such as the ℓ_1 norm. Gradient-based optimization schemes are therefore sub-optimal since the target cost function is no longer differentiable. In this paper, we propose a method to allow weights optimization under non-smooth regularizations. Let us denote by x an input to be presented to the ANN. The estimated label will be denoted by $\widehat{y}(x, W)$ as a non-linear function of the input x and the weights vector $W \in \mathbb{R}^N$, while the ground truth label will be denoted by y. Using a quadratic error with

an ℓ_1 regularization with M input data for the learning step, the weights vector can be estimated as:

$$\widehat{W} = \arg\min_W \mathcal{L}(W) = \arg\min_W \sum_{m=1}^{M} \|\widehat{y}(x^m; W) - y^{(m)}\|_2^2 + \lambda\|W\|_1, \quad (1)$$

where λ is a regularization parameter balancing the solution between the data fidelity and regularization terms, and M is the number of learning data. Since the optimization problem in (1) is not differentiable, the use of gradient-based algorithms with back-propagation is not possible. In this case, the learning process is costly and very complicated. In Section 3, we present a method to efficiently estimate the weights vector without increase of learning complexity. The optimization problem in (1) is formulated and solved in a Bayesian framework.

3. BAYESIAN OPTIMIZATION

As stated above, the weights optimization problem is formulated in a Bayesian framework. In this sense, the problem parameters and hyperparameters are assumed to follow probability distributions. More specifically, a likelihood distribution is defined to model the link between the target weights vector and the data, while a prior distribution is defined to model the prior knowledge about the target weights.

3.1. Hierarchical Bayesian model

According to the principle of minimizing the error between the reference label y and the estimated one $\widehat{y}$, and assuming a quadratic error (first term in (1)), we define the likelihood distribution as

$$f\left(y; W, \sigma\right) \propto \prod_{m=1}^{M} \exp\left(-\frac{1}{2\sigma^2}\|\widehat{y}(x^m; W) - y^{(m)}\|^2\right), \quad (2)$$

where σ^2 is a positive parameter to be set.
As regards the prior knowledge on the weights vector W, we propose the use of a Laplace distribution in order to promote the sparsity of the neural network:

$$f(W; \lambda) \propto \prod_{k=1}^{N} \exp\left(-\frac{\|W^{[k]}\|_1}{\lambda}\right), \quad (3)$$

where λ is a hyperparameter to be fixed or estimated. By adopting a Maximum A Posteriori (MAP) approach, we first need to express the posterior distribution. Based on the defined likelihood and prior, this posterior writes:

$$\begin{aligned} f(W; y, \sigma, \lambda) &\propto f(y; W, \sigma) f(W; \lambda) \\ &\propto \prod_{m=1}^{M} \exp\left(-\frac{1}{2\sigma^2}\|\widehat{y}(x^m; W) - y^{(m)}\|^2\right) \\ &\times \prod_{k=1}^{N} \exp\left(-\frac{\|W^{[k]}\|_1}{\lambda}\right). \end{aligned} \quad (4)$$

It is clear that this posterior is not straightforward to handle in order to derive a closed-form expression of the estimate $\widehat{W}$. For this reason, we resort to a stochastic sampling approach in order to numerically approximate the posterior, and hence to calculate an estimator for $\widehat{W}$. The following Section details the adopted sampling procedure.

3.2. Hamiltonian Sampling

Let us denote $\alpha = \dfrac{\lambda}{\sigma^2}$ and $\theta = \{\sigma^2, \lambda\}$. For a weight W^k we define the following energy function

$$E_\theta^k(W^k) = \frac{\alpha}{2} \sum_{m=1}^{M} \|\widehat{y}(x^m; W) - y^{(m)}\|_2^2 + \|W^k\|_1. \quad (5)$$

The posterior in (4) can therefore be reformulated as

$$f(W; y, \theta) \propto \exp\left(-\sum_{k=1}^{N} E_\theta^k(W^k)\right). \quad (6)$$

To sample according to this exponential posterior, and since direct sampling is not possible due to the form of the energy function E_θ^k, Hamiltonian sampling is adopted. Indeed, Hamiltonian dynamics [4] strategy has been widely used in the literature to sample from high dimensional vectors. However, sampling using Hamiltonian dynamics requires computing the gradient of the energy function, which is not possible in our case due to the ℓ_1 term. To overcome this difficulty, we resort to a non-smooth Hamiltonian Monte Carlo (ns-HMC) strategy as proposed in [5]. More specifically, we use the plug and play procedure developed in [6]. Indeed, this strategy requires to calculate the proximity operator only at an initial point, and uses the shift property [8] to deduce the proximity operator during the iterative procedure [6, Algorithm 1]. As regards the proximity operator calculation, let us denote by $G_{\mathcal{L}}(W^k)$ the gradient of the quadratic term of the loss function $\mathcal{L}$ with respect to the weight W^k. Let us also denote by $\varphi(W^k) = \|W^k\|_1$. Following the standard definition of the proximity operator [8], we can write for a point z

$$\text{prox}_{E_\theta^k}(z) = p \Leftrightarrow z - p \in \partial E_\theta^k(p). \quad (7)$$

Straightforward calculations lead to the following expression of the proximity operator:

$$\text{prox}_{E_\theta^k}(z) = \text{prox}_{\varphi}\left(z - \frac{\alpha}{2} G_{\mathcal{L}}(W^k)\right). \quad (8)$$

Since prox_{φ} is nothing but the soft thresholding operator [8], the proximity operator in (8) can be easily calculated once a single gradient step is applied (back-propagation) to calculate $G_{\mathcal{L}}(W^k)$. The main steps of the proposed method are detailed in Algorithm 1.

Algorithm 1: Main steps of the proposed Bayesian optimization.

- Fix the hyperparameters λ and σ;
- Initialize with some W_0;
- Perform one back-propagation step to provide an initialization for $G_{\mathcal{L}}(W_0)$;
- Compute $\text{prox}_{E_\theta}(W_0)$ according to (8);
- Use the Gibbs sampler in [6, Algorithm 1] until convergence;

After convergence, Algorithm 1 provides chains of coefficients sampled according to the target distribution of each W^k. These chains can be used to compute an MMSE (minimum mean square error) estimator (after discarding the samples corresponding to the burn-in period).
It is worth noting that hyperprior distributions can be put on λ and σ in order to integrate them in the hierarchical Bayesian model. These hyperparameters can therefore be estimated from the data at the expense of some additional complexity.

4. EXPERIMENTAL VALIDATION

In order to validate the proposed method, two (2) images classification experiments are conducted using two different datasets: The first dataset includes CT (Computed Tomography) images for COVID-19 classification, while the second includes Lentigo classification on real RCM (Reflectance confocal microscopy) data. For the sake of comparison, two kinds of optimizers are used : *i)* MCMC-based, specifically a random walk Metropolis Hastings (rw-MH) algorithm, and *ii)* gradient-based such as Adam, and SGD.

4.1. Used CNN architecture

To perform the classification task, the CNN architecture employed in this study has three convolutional (Conv-32, Conv-64 and Conv-128) and two fully-connected (FC-128, and FC-softamx) layers similar to LeNet. Each convolutional layer includes filters with 3×3 Kernels in addition to 2×2 max-poolong layers, with stride size equal to 1. As deep neural networks can easily overfit when trained with small datasets, the used CNN is extended with two regularizing techniques : Batch Normalization and Dropout (the dropout rate is set by cross validation to $p = 0.35$). As regards coding, we used python programming language with Keras and Tensorflow libraries on an Intel(R) Core(TM) i7 3630QM CPU 2.40GHZ architecture with 8 Go memory.

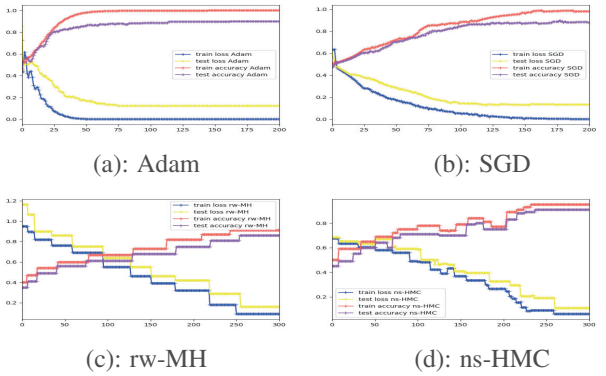
4.2. COVID-19 CT image classification

This section studies the performance of the proposed optimization scheme on a classification problem for Covid-19 detection using CT images of size 230. Two classes are considered: normal and Covid-19. A publicly available dataset of CT scans is used[1], involving 1252 CT scans of SARS-CoV-2 infected subjects, in addition to 1230 normal CT scans. A training and test sets are used, involving 720 and 200 images, respectively. Once the model trained, Table 1 reports the accuracy, loss for the four optimization methods, and this based on the test set. Computational time is also reported. The reported accuracy and loss values indicate a better performance of the used CNN when trained using the proposed optimizations scheme. As regards the computational, an almost twice faster convergence is ensured using the proposed optimizer.

Table 1. Experiment 1: Results for CT image classification.

Optimizers	Comp. time (hrs)	Accuracy	Loss
Prop. method	**0.30**	**0.90**	**0.10**
rw-MH	1.08	0.85	0.17
Adam	0.52	0.87	0.12
SGD	0.53	0.88	0.13

To further assess the convergence behaviour, Fig. 1 displays loss and accuracy curves obtained with the competing optimizers.



(a): Adam (b): SGD

(c): rw-MH (d): ns-HMC

Fig. 1. Experiment 1: Train and test curves for Adam (a), SGD (b), rw-MH (c), and the proposed ns-HMC method (d).

Despite the used regularization (dropout and batch normalization), an overfitting behavior can be reported by analysing the displayed curves of the SGD and Adam optimizers. This effect is less visible using the MCMC-based optimizers, which may be explained by a better exploration of the searching space using such methods. This is mainly

[1]https://www.kaggle.com/plameneduardo/sarscov2-ctscan-dataset

due to the Bayesian formulation. Moreover, the used efficient sampling clearly helps accelerating convergence, which confirms the reported computational time in Table 1. Indeed, ns-HMC sampling integrates a gradient information related to the geometry of the target distribution, which finally leads to a faster convergence of the used sampler. It is worth noting that the curves irregularity for Bayesian techniques (proposed method rw-MH) are due to the random sampling effect. No monotonic behaviour is expected.

4.3. Lentigo image classification

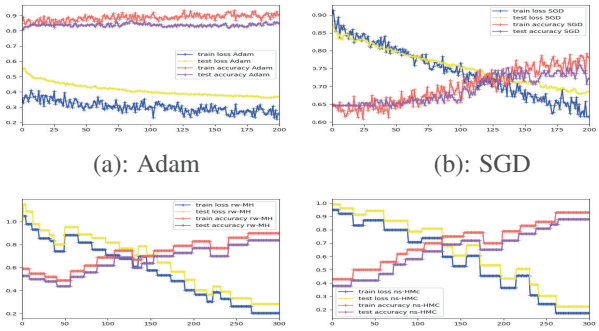
This section evaluates the validation of the proposed method with lentigo detection on real RCM data. The dataset is provided from Lab. Pierre Fabre. In this experiment, the data include 428 RCM images which high spatial resolutions and annotation on each image into two healthy and lentigo classes. The images were acquired with a Vivascope 1500 apparatus. Each RCM image shows a field of view of 500×500 μm with 1000×1000 pixels. A selection of 45 women aged 60 years were recruited. All participants have offered their informed consent to the RCM skin test. The reported scores in Table 2 indicate that the proposed method clearly outperforms the competing optimizers in terms of learning precision, and hence classification performance. Furthermore, the competing optimizers do not perform well to learn the used CNN on the real RCM data. This confirms the ability of the proposed method to allow simple networks reaching high accuracy levels, in contrast to standard optimizers, even when regularization is used. The gain in terms of computational time using the proposed method is more important on this experiment. Indeed, Fig. 2 displays the loss and accuracy curves for all optimizers. It can be easily noticed that standard optimizers need more epochs to converge in comparison to experiments in Section 4.2.

Table 2. Experiment 2: Results for lentigo image classification.

Optimizers	Comp. time (hrs)	Accuracy	Loss
Prop. method	**0.41**	**0.89**	**0.22**
rw-MH	1.32	0.83	0.30
Adam	1.09	0.84	0.36
SGD	1.13	0.73	0.68

5. CONCLUSION

In this paper, we proposed a novel Bayesian optimization method to fit weights for artificial neural networks where sparsity constraints are applied. Our results demonstrate the good performance of the proposed method in comparison with standard optimizers, even when combined with classical regularization techniques. Moreover, the proposed technique allows simple networks to enjoy high accuracy and generalization properties, mainly due to a better exploration of the target searching space. Future work will focus on proposing a distributed or parallel implementation to further accelerate convergence.



(a): Adam (b): SGD

(c): rw-MH (d): ns-HMC

Fig. 2. Experiment 2: Train and test curves for Adam (a), SGD (b), rw-MH (c), and the proposed HMC method (d).

6. REFERENCES

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] M. Fakhfakh, B. Bouaziz, F. Gargouri, and L. Chaari, "Prognet: Covid-19 prognosis using recurrent and convolutional neural networks," *The Open Medical Imaging Journal*, vol. 12, pp. 11–22, 2020.

[3] R. Zaheer and H. Shaziya, "A study of the optimization algorithms in deep learning," in *2019 Third International Conference on Inventive Systems and Control (ICISC)*. IEEE, 2019, pp. 536–539.

[4] S. Brooks, A. Gelman, G. Jones, and X.L. Meng, *Handbook of Markov Chain Monte Carlo*, CRC press, 2011.

[5] L. Chaari, J.-Y. Tourneret, C. Chaux, and H. Batatia, "A Hamiltonian Monte Carlo method for non-smooth energy sampling," *IEEE Trans. on Signal Process.*, vol. 64, no. 21, pp. 5585 – 5594, Jun. 2016.

[6] L. Chaari, J.-Y. Tourneret, and H. Batatia, "A plug and play Bayesian algorithm for solving myope inverse problems," *European Signal Processing Conference EUSIPCO*, pp. 742–746, Sept 2018.

[7] U. Evci, F. Pedregosa, A. N. Gomez, and E. Elsen, "The difficulty of training sparse neural networks," *ArXiv*, vol. abs/1906.10732, 2019.

[8] C. Chaux, P.L. Combettes, J.C. Pesquet, and V.R. Wajs, "A variational formulation for frame-based inverse problems," *Inverse Problems*, vol. 23, no. 4, pp. 1495, 2007.