Natural Variational Continual Learning

Hanna Tseran [1,2]
hanna.tseran@gmail.com

Mohammad Emtiyaz Khan [1]
emtiyaz.khan@riken.jp

Tatsuya Harada [1,2]
harada@mi.t.u-tokyo.ac.jp

Thang D. Bui [3]
thang.buivn@gmail.com

[1] RIKEN Center for Advanced Intelligence project, Japan
[2] The University of Tokyo, Japan
[3] University of Sydney, Australia

Abstract

The goal of continual learning is to sequentially learn new skills without forgetting old ones. Recent continual-learning approaches have employed gradient-based approximate Bayesian inference methods to derive such algorithms. In this paper, we propose a natural-gradient method that unifies two recent approaches based on Laplace approximation and variational inference, respectively. Our method enables a plug-and-play implementation where the accuracy of the approximation can be traded off for the ease of implementation. Our method also enables a principled application of approximate Bayesian inference for continual learning, and gives competitive performance to the existing approaches.

1 Introduction

The ability to continually and quickly adapt to changing environments is central to building intelligent systems. For this purpose, it is important to remember useful past experiences while acquiring new ones. Such continual learning is crucial for many applications, such as robotics [4], where new tasks can appear during the training, and data from the previous tasks might be unavailable for retraining. Despite its importance, many popular learning paradigms cannot learn continually, and either suffer from catastrophic forgetting or require bespoke training algorithms or model changes to adapt to new environments.

There are various approaches to facilitate continual learning, e.g., by using an external memory, or by modifying the model/architecture whenever a new task is encountered. In this paper, our focus is on methods that modify the training algorithms so that they can learn continually. For example, the elastic weight consolidation (EWC) method [3] uses past experiences as a prior distribution within an approximate Bayesian framework by using a type of Laplace approximation. The past experiences then act as a regularizer to avoid forgetting. Another method called variational continual learning (VCL) [5] uses variational inference (VI) instead of the Laplace approximation. Our goal in this paper is to provide a new framework that enables an easy implementation and deployment of such continual-learning approaches.

We present a new framework based on natural-gradient methods for variational inference. Building on a recent work of [1], we show that both EWC and VCL can be easily implemented within this framework. Moreover, the accuracy of the approximation can be traded-off for the simplicity of the implementation. Our method, therefore, enables a principled application of approximate Bayesian inference methods while remaining simple to implement and deploy.

Continual Learning Workshop (NeurIPS 2018), Montréal, Canada.

2 Elastic-Weight Consolidation and Variational Continual Learning

A recent approach called the Elastic-Weight Consolidation (EWC) method has revived the interest in continual learning for neural networks [3]. The key idea there is to constrain model parameters to stay *close* to those estimated on previous tasks by using a regularizer. Specifically, denote the optimal parameter $\boldsymbol{\theta}_{t-1}$ found by maximizing an objective on the data $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_{t-1}$, where each $\mathcal{D}_i$ is from a past tasks i. Given the data $\mathcal{D}_t$ from a new task t, EWC estimates the new parameters by optimizing the following *regularized* objective functions:

$$\mathcal{L}^{(t)}_{\text{EWC}}(\boldsymbol{\theta}) := \log p(\mathcal{D}_t|\boldsymbol{\theta}) - \tfrac{1}{2}\lambda_t(\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1})^\top \mathbf{F}(\boldsymbol{\theta}_{t-1})(\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1}), \tag{1}$$

$$\text{where } \mathbf{F}(\boldsymbol{\theta}_{t-1}) := \sum_{n \in \mathcal{D}_{t-1}} \left[\nabla_\theta \log p(\mathcal{D}_{t-1,n}|\boldsymbol{\theta}) \nabla_{\theta^\top} \log p(\mathcal{D}_{t-1,n}|\boldsymbol{\theta})\right] \Big|_{\theta=\theta_{t-1}} \tag{2}$$

with $\mathbf{F}$ being the Fisher information matrix (FIM) and $\mathcal{D}_{t-1,n}$ is the n'th data example for the previous task. The second term in (1) is the regularization term. The objective can be seen as a type of Laplace approximation where we seek to find the mode of the posterior approximation with the likelihood $p(\mathcal{D}_t|\boldsymbol{\theta})$ and a Gaussian prior $\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1}, \mathbf{F}(\boldsymbol{\theta}_{t-1})^{-1})$. The Bayesian approach regularizes the learning on the new task by using the previous posterior approximation as the prior.

To optimize (1), stochastic-gradient methods are used in [3], e.g., we can use the following updates:

$$\text{EWC: } \boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \left[\hat{\mathbf{g}}_t(\boldsymbol{\theta}) - \frac{\lambda}{N}\mathbf{f}_{t-1} \circ (\boldsymbol{\theta} - \boldsymbol{\theta}_{t-1})\right], \tag{3}$$

$$\text{with } \hat{\mathbf{g}}_t(\boldsymbol{\theta}) := \frac{1}{M}\sum_{n \in \mathcal{M}} \nabla_\theta \log p(\mathcal{D}_{t,n}|\boldsymbol{\theta}), \text{ and } \mathbf{f}_{t-1} := \text{diag}\left[\mathbf{F}(\boldsymbol{\theta}_{t-1})\right], \tag{4}$$

where $\hat{\mathbf{g}}_t(\boldsymbol{\theta})$ is a stochastic-approximation of the gradient computed over a randomly sampled minibatch $\mathcal{M}$ from $\mathcal{D}_t$, N is the number of examples in $\mathcal{D}_t$, M is the size of the minibatch, and $\circ$ denotes element-wise multiplication. To simplify the computation, a diagonal approximation to the FIM is used. However, this algorithm still requires an explicit computation of the FIM after the iteration (3) converges. In contrast, as we show later, in our natural VCL approach, the FIM matrix is estimated during training and this additional step is not required.

Another recent approach called variational continual learning (VCL) uses variational inference for continual learning [5]. Instead of using a Laplace approximation, a variational approximation is obtained by optimizing the following variational lower bound:

$$\mathcal{L}^{(t)}_{\text{VCL}}(q_t(\theta)) := \mathbb{E}_{q_t(\theta)}\left[\log p(\mathcal{D}_t|\boldsymbol{\theta})\right] - \mathbb{D}_{KL}\left[q_t(\boldsymbol{\theta})||q_{t-1}(\boldsymbol{\theta})\right], \tag{5}$$

where $q_{t-1}(\boldsymbol{\theta})$ is the variational approximation at the previous step, and $\mathbb{D}_{KL}$ is the Kullback-Leibler divergence. In practice, the variational approximations are chosen to be in a simple family, e.g., a Gaussian approximation $q_t(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ with mean $\boldsymbol{\mu}_t$ and covariance $\boldsymbol{\Sigma}_t$. To simplify the computation, a diagonal approximation to the covariance is often used, i.e., $\boldsymbol{\Sigma} = \text{diag}(\boldsymbol{\sigma}^2)$ where $\boldsymbol{\sigma}$ is a vector. The mean and covariance can be obtained by using a stochastic-gradient method, e.g., by computing the gradients using the reparameterization trick [2] as shown below,

$$\text{VCL: } \boldsymbol{\mu} \leftarrow \boldsymbol{\mu} + \alpha \left[\hat{\mathbf{g}}_t(\hat{\boldsymbol{\theta}}) - \frac{1}{N}\boldsymbol{\sigma}^{-2}_{t-1} \circ (\boldsymbol{\mu} - \boldsymbol{\mu}_{t-1})\right], \tag{6}$$

$$\boldsymbol{\sigma} \leftarrow \boldsymbol{\sigma} + \alpha \left[\boldsymbol{\epsilon} \circ \hat{\mathbf{g}}_t(\hat{\boldsymbol{\theta}}) - \frac{1}{N}\left(-\boldsymbol{\sigma}^{-1} + \boldsymbol{\sigma} \circ \boldsymbol{\sigma}^{-2}_{t-1}\right)\right], \tag{7}$$

where $\hat{\boldsymbol{\theta}} := \boldsymbol{\mu} + \boldsymbol{\sigma} \circ \boldsymbol{\epsilon}$, i.e., $\hat{\boldsymbol{\theta}}$ is a sample from $\mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

The EWC update of (3) and the VCL update in (6) take similar forms, but there are notable differences. First, the stochastic gradient in VCL is computed at a sample $\hat{\boldsymbol{\theta}}$ from q_{t-1}, rather than a fixed parameter $\boldsymbol{\theta}$. This is due to the fundamental difference between the Laplace approximation and VI. The former is a local approximation at a fixed parameter, while the latter takes into account the neighborhood around the mean of the variational distribution [6]. For this reason, we expect VCL to be a better approximation than EWC.

The second difference is that the Fisher vector $\mathbf{f}_{t-1}$ is replaced by the variance $\boldsymbol{\sigma}^{-2}_{t-1}$. In variational inference, the variance is typically related to the Hessian of the log-likelihood (see [6] for details). In

EWC, the FIM is used instead of the Hessian. An advantage of the VCL approach is that, $\boldsymbol{\sigma}_{t-1}$ is estimated *during* the training, not after the training is finished. This computation in VCL, however, increases the cost. The EWC update is computationally simpler but might be less accurate than the VCL update. We will show that our natural VCL method is as simple as the EWC update but could also be as accurate as VCL.

Both EWC and VCL have their pros and cons. We will now present a natural-gradient based method which gives a unified framework for continual-learning algorithms similar to EWC and VCL. It also enables us to trade-off the computation cost for accuracy. We now describe our method.

3 Natural Variational Continual Learning

We present a natural-gradient method for VCL by using the recent work of [1] which proposes such a method for variational inference. Using their approach, the derivation of our method is straightforward, and below we only give the update and skip the derivations.

We first present an update obtained by using the Variational Online Gauss-Newton (VOGN) method presented in [1]. Given the prior $q_{t-1}(\boldsymbol{\theta}) := \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_t, \boldsymbol{\sigma}_{t-1}^2)$, we get the following update by applying the VOGN method which we call *Natural* VCL:

$$\text{Natural VCL: } \boldsymbol{\mu} \leftarrow \boldsymbol{\mu} + \alpha\, \boldsymbol{\sigma}^{-2} \left[\hat{\mathbf{g}}_t(\hat{\boldsymbol{\theta}}) - \frac{1}{N} \boldsymbol{\sigma}_{t-1}^{-2} \circ (\boldsymbol{\mu} - \boldsymbol{\mu}_{t-1}) \right], \tag{8}$$

which is similar to the VCL update but with an adaptive-learning rate equal to the variance $\boldsymbol{\sigma}^{-2}$. Unlike VCL, the variance is updated using an *online* estimate $\hat{\mathbf{f}}_t$ of the diagonal of the FIM:

$$\boldsymbol{\sigma}^{-2} \leftarrow \hat{\mathbf{f}}_t + \boldsymbol{\sigma}_{t-1}^{-2}, \quad \text{with } \hat{\mathbf{f}}_t \leftarrow \beta\, \hat{\mathbf{f}}_t + (1-\beta)\, \frac{N}{M} \sum_{n \in \mathcal{M}} \text{diag} \left[\mathbf{g}_{t,n}(\hat{\boldsymbol{\theta}}) \mathbf{g}_{t,n}(\hat{\boldsymbol{\theta}})^\top \right], \tag{9}$$

where β is a scalar step-size and $\mathbf{g}_{t,n}(\hat{\boldsymbol{\theta}}) := \nabla_\theta \log p(\mathcal{D}_{t,n}|\hat{\boldsymbol{\theta}})$. The vector $\hat{\mathbf{f}}_t$ is an *online* estimate of the diagonal of the FIM $\mathbf{F}(\hat{\boldsymbol{\theta}})$, and is used to compute the adaptive-learning rate to update $\boldsymbol{\mu}$.

The natural-VCL update can be seen as a regularized version of the EWC update shown in (3). This is because the variance of q_{t-1} is equal to $\boldsymbol{\sigma}_{t-1}^{-2} = \hat{\mathbf{f}}_{t-1} + \boldsymbol{\sigma}_{t-2}^{-2}$, i.e., the variance is equal to the online estimate of the Fisher plus a regularization term from the previous variance estimate. The natural-VCL algorithm is computationally a bit more attractive than the EWC method because the FIM is computed during the training and we do not need to run through the whole dataset to compute it after the training. This simplifies the computation in cases where revisiting the whole dataset is either costly, infeasible, or even perhaps impossible. However, computation in (9) might not always be easy to perform since it requires storage of individual gradients $\mathbf{g}_{t,n}$. Nevertheless, it does avoid the need to estimate FIM on the whole dataset $\mathcal{D}_t$.

The natural-VCL update is computationally much simpler than EWC similarly to the VCL update of (6). The learning rates are naturally adapted using the variance $\boldsymbol{\sigma}^{-2}$ and do not require additional adaptation using methods such as Adam/RMSprop. This leads to a simpler implementation and deployment of the method. However, the method above does not exactly optimize the VCL objective (5), rather uses a *Gauss-Newton* approximation [1] which only finds an approximate solution of the VCL objective. We will now describe two other variants of natural-VCL using which we can trade-off accuracy for ease of implementation.

To exactly optimize the VCL objective, we can use the Variational Online Newton (VON) method of [1], where the updates are modified to use Hessian instead of the FIM, as shown below:

$$\boldsymbol{\sigma}^{-2} \leftarrow \hat{\mathbf{h}}_t + \boldsymbol{\sigma}_{t-1}^{-2}, \text{ with } \hat{\mathbf{h}}_t \leftarrow \beta\, \hat{\mathbf{h}}_t + (1-\beta)\, \frac{N}{M} \sum_{n \in \mathcal{M}} \text{diag} \left[\nabla^2_{\theta\theta^\top} \log p(\mathcal{D}_{t,n}|\boldsymbol{\theta})|_{\theta=\hat{\theta}} \right], \tag{10}$$

This update optimizes the VCL objective and might perform better than EWC, but requires computation of the Hessian which might be costly.

We can obtain another version which is computationally much simpler than all the methods discussed so far, but makes a crude approxiation of the FIM by using a gradient-magnitude approximation. This is referred to as the Variational Adam algorithm or simply Vadam [1]. Applying this method to VCL,

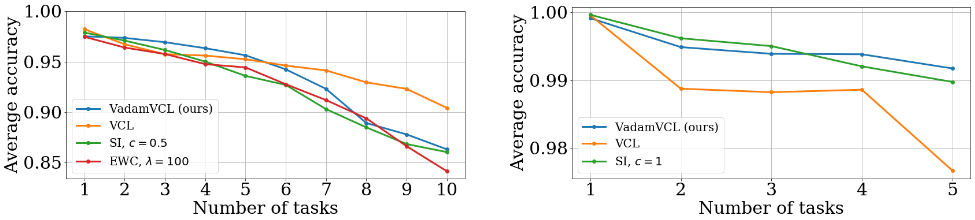

Figure 1: Results on the permuted MNIST (left) and split MNIST (right). EWC results in the right are omitted because of its poor performance. On permuted MNIST, VadamVCL performs similarly to SI and EWC but worse than VCL. For split MNIST, however, it performs better than VCL (and EWC which is omitted) and comparable to SI.

we get the following update:

$$\text{Vadam VCL: } \boldsymbol{\mu} \leftarrow \boldsymbol{\mu} + \alpha\, \boldsymbol{\sigma}^{-1} \left[\hat{\mathbf{g}}_t(\hat{\boldsymbol{\theta}}) - \frac{1}{N} \boldsymbol{\sigma}_{t-1}^{-2} \circ (\boldsymbol{\mu} - \boldsymbol{\mu}_{t-1}) \right] \tag{11}$$

$$\boldsymbol{\sigma}^{-2} \leftarrow \mathbf{s}_t + \boldsymbol{\sigma}_{t-1}^{-2}, \text{ with } \mathbf{s}_t \leftarrow \beta\, \mathbf{s}_t + (1-\beta)\, N \left[\hat{\mathbf{g}}_t(\hat{\boldsymbol{\theta}}) \right]^2 . \tag{12}$$

Here, the learning-rate is modified to be $\boldsymbol{\sigma}$ instead of its square. The update has a very similar form to the Vadam algorithm and is very simple to implement. However, we expect the performance to be worse than other methods we have presented earlier.

In summary, our natural-gradient framework for VCL is related to EWC and VCL, and can be used to trade-off accuracy and ease of implementation. Our method is a regularized version of EWC and therefore it is easy to obtain EWC as a special case. The version presented in (8) approximately optimizes the VCL objective but is simpler to implement than VCL. The natural-VCL update is also very similar to the cumulative Hessian update proposed by [7]. We can improve the accuracy of the proposed method by using (10) instead of (9), and we can obtain an easy to implement method by using (12). Overall, our method enables a plug-and-play implementation of continual learning methods where the accuracy of the approximation can be traded-off for the ease of implementation.

4 Experiments

We present some preliminary results comparing to existing methods such as EWC, VCL and Synaptic Intelligence (SI) [8]. We only present comparisons with Vadam VCL.

Permuted MNIST: Here, each task is the standard 10-class MNIST classification, with the pixels of the digits having undergone a fixed random permutation. Following the previous work of [5], we fit a fully-connected neural network with 2 hidden layers and 100 units each. We train the model on a sequence of 10 tasks corresponding to 10 random permutations. We run VadamVCL for 500 epochs with hyperparameters[1] $\beta_1 = 0.9, \beta_2 = 0.999, \lambda = 1, \gamma = 0.01, \eta = 0.001$. The average running accuracies for all methods are shown in the right figure in Figure 1.

Split MNIST: For this experiment MNIST digits dataset is split into 5 pairs: 0/1, 2/3, 4/5, 6/7, 8/9, The dataset received for each task is one of these pairs. For all methods is used a multi-head neural network with 2 hidden layers with 256 units each. VadamVCL parameters are set to $\beta_1 = 0.9, \beta_2 = 0.999, \lambda = 1, \gamma = 1, \eta = 0.0006$, and training on each task took 500 epochs. The average running accuracies for all methods are shown in left figure in Figure 1.

5 Conclusion

We presented the natural variational continual learning method. Our method allows a principled application of approximate Bayesian inference for continual learning. It enables a plug-and-play implementation where the accuracy can be traded off for the ease of implementation. We discussed various versions of our algorithm and discussed the relationship to EWC and VCL.

In the future, we hope to perform many numerical experiments to understand the effect of the approximation on the performance of the continual-learning task. We hope to implement all versions of our algorithms and compare them with existing methods.

[1] These hyperparameters are defined in [1]

References

[1] Mohammad Khan, Didrik Nielsen, Voot Tangkaratt, Wu Lin, Yarin Gal, and Akash Srivastava. Fast and scalable Bayesian deep learning by weight-perturbation in Adam. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2611–2620, 2018.

[2] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

[3] 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*, 114(13):3521–3526, 2017.

[4] Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. *arXiv preprint arXiv:1705.03550*, 2017.

[5] Cuong V. Nguyen, Yingzhen Li, Thang D. Bui, and Richard E. Turner. Variational continual learning. In *International Conference on Learning Representations*, 2018.

[6] Manfred Opper and Cédric Archambeau. The variational Gaussian approximation revisited. *Neural computation*, 21(3):786–792, 2009.

[7] Hippolyt Ritter, Aleksandar Botev, and David Barber. Online structured laplace approximations for overcoming catastrophic forgetting. In *International Conference on Learning Representations*, 2018.

[8] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pages 3987–3995, 2017.