Variational Continual Learning in Deep Models

Cuong V. Nguyen **Yingzhen Li** **Thang D. Bui** **Richard E. Turner**
Department of Engineering, University of Cambridge
{vcn22,yl494,tdb40,ret26}@cam.ac.uk

Abstract

This short paper develops variational continual learning (VCL), a simple but general framework for continual learning that fuses online variational inference (VI) and recent advances in Monte Carlo VI for neural networks. The framework can successfully train both deep discriminative models and deep generative models in complex continual learning settings where existing tasks evolve over time and entirely new tasks emerge. Experimental results show that variational continual learning outperforms state-of-the-art continual learning methods on a variety of tasks, avoiding catastrophic forgetting in a fully automatic way.[1]

1 Introduction

Continual learning is a very general form of online learning in which data continuously arrive in a possibly non i.i.d. way, tasks may change over time, and entirely new tasks can emerge [1, 2, 3]. What is more, continual learning systems must adapt to perform well on the entire set of tasks in an online way that avoids revisiting all previous data at each stage. This is a key problem in machine learning since real world tasks continually evolve over time and the size of datasets often prohibits frequent batch updating. Moreover, practitioners are often interested in solving a set of related tasks that benefit from being handled jointly in order to leverage multi-task transfer.

The ubiquity of deep learning means that it is important to develop deep continual learning methods. However, it is challenging to strike a balance between adapting to recent data and retaining knowledge from old data. Too much plasticity leads to catastrophic forgetting [4, 5, 6] and too much stability leads to an inability to adapt. Recently there has been a resurgence of interest in this area. One approach trains individual models on each task and then carries out a second stage of training to combine them [7]. A more elegant and more flexible approach maintains a single model and uses a single type of regularized training that prevents drastic changes in the parameters which have a large influence on prediction, but allows other parameters to change more freely [8, 9, 10]. Our approach here follows this venerable work, but is arguably more principled, extensible and automatic.

This paper is built on the observation that there already exists an extremely general framework for continual learning: Bayesian inference. Critically, Bayesian inference retains a distribution over model parameters that indicates the plausibility of any setting given the observed data. When new data arrive, we combine what previous data have told us about the model parameters (the previous posterior) with what the current data are telling us (the likelihood). Multiplying and renormalizing yields the new posterior, from which point we can recurse. Critically, the previous posterior constrains parameters that strongly influence prediction, preventing them from changing drastically, but it allows other parameters to change. The wrinkle is that exact Bayesian inference is typically intractable and so approximations are required. Fortunately, there is an extensive literature on approximate inference for neural networks. We merge online variational inference (VI) [11, 12, 13] with Monte Carlo VI for neural networks [14] to yield *variational continual learning* (VCL). In addition, we extend VCL

[1] The full version of this paper, including additional experiments and discussions, is available at: https://arxiv.org/abs/1710.10628.

Second workshop on Bayesian Deep Learning (NIPS 2017), Long Beach, CA, USA.

to include a small episodic memory by combining VI with the coreset data summarization method [15, 16]. We demonstrate that the framework is general, applicable to both deep discriminative models and deep generative models, and that it yields excellent performance.

2 Continual Learning by Approximate Bayesian Inference

Consider a discriminative model that returns a probability distribution over an output y given an input $\boldsymbol{x}$ and parameters $\boldsymbol{\theta}$, that is $p(y|\boldsymbol{\theta}, \boldsymbol{x})$. In continual learning, the goal is to learn the model parameters from a set of sequentially arriving datasets $\{\boldsymbol{x}_t^{(n)}, y_t^{(n)}\}_{n=1}^{N_t}$. Following a Bayesian approach, a prior $p(\boldsymbol{\theta})$ is placed over $\boldsymbol{\theta}$. The posterior after seeing T datasets is recovered by applying Bayes' rule:

$$p(\boldsymbol{\theta}|\mathcal{D}_{1:T}) \propto p(\boldsymbol{\theta}) \prod_{t=1}^{T} \prod_{n_t=1}^{N_t} p(y_t^{(n_t)}|\boldsymbol{\theta}, \boldsymbol{x}_t^{(n_t)}) = p(\boldsymbol{\theta}) \prod_{t=1}^{T} p(\mathcal{D}_t|\boldsymbol{\theta}) \propto p(\boldsymbol{\theta}|\mathcal{D}_{1:T-1}) p(\mathcal{D}_T|\boldsymbol{\theta}).$$

Here we have used the shorthand $\mathcal{D}_t = \{y_t^{(n)}\}_{n=1}^{N_t}$ and the input dependence has been suppressed to lighten notation. Importantly, a recursion has been identified whereby the posterior after seeing the T-th dataset is produced by taking the posterior after seeing the $(T-1)$-th dataset, multiplying by the likelihood and renormalizing. In other words, online updating emerges naturally from Bayes' rule.

In most cases the posterior is intractable and approximation is required, even when forming the first posterior $p(\boldsymbol{\theta}|\mathcal{D}_1) \approx q_1(\boldsymbol{\theta}) = \mathrm{proj}(p(\boldsymbol{\theta})p(\mathcal{D}_1|\boldsymbol{\theta}))$. Here $q(\boldsymbol{\theta}) = \mathrm{proj}(p^*(\boldsymbol{\theta}))$ denotes a projection operation that takes an intractable un-normalized distribution $p^*(\boldsymbol{\theta})$ and returns a tractable normalized approximation $q(\boldsymbol{\theta})$. Subsequent approximations can be produced recursively by combining the approximate posterior distribution with the likelihood and projecting: $p(\boldsymbol{\theta}|\mathcal{D}_{1:T}) \approx q_T(\boldsymbol{\theta}) = \mathrm{proj}(q_{T-1}(\boldsymbol{\theta})p(\mathcal{D}_T|\boldsymbol{\theta}))$. In this way, online updating is supported.

The field of approximate inference provides several choices for the projection operation including Laplace approximation, variational KL minimization, moment matching, and importance sampling, which respectively lead to Laplace propagation [17], online VI [11, 12, 13], assumed density filtering [18] and sequential Monte Carlo [19] in the continual learning setting. Our paper uses the online VI approach as it typically outperforms the other methods for complex models in the static setting [20] and yet it has not been applied to continual learning of neural networks.

Variational Continual Learning (VCL) and Episodic Memory Enhancement. VCL employs a projection operator defined through a KL divergence minimization over the set of allowed approximate posteriors $\mathcal{Q}$,

$$q_t(\boldsymbol{\theta}) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}\big(q(\boldsymbol{\theta}) \parallel \tfrac{1}{Z_t} q_{t-1}(\boldsymbol{\theta})\, p(\mathcal{D}_t|\boldsymbol{\theta})\big), \text{ for } t = 1, 2, \ldots, T. \tag{1}$$

The zeroth approximate distribution is defined to be the prior, $q_0(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$. Z_t is the intractable normalizing constant of $p^*(\boldsymbol{\theta}) = q_{t-1}(\boldsymbol{\theta})\, p(\mathcal{D}_t|\boldsymbol{\theta})$ and is not required to compute the optimum.

VCL will perform exact Bayesian inference if the true posterior is a member of the approximating family $\mathcal{Q}$ at every step. Typically this will not be the case and we might worry that performing repeated approximations may accumulate errors causing the algorithm to forget old tasks. Furthermore, the minimization at each step is also approximate (e.g. due to employing an additional Monte Carlo approximation) and so additional information may be lost. In order to mitigate this potential problem, we extend VCL to include a small representative set of data from previously observed tasks that we call the coreset. The coreset is analogous to an episodic memory [21] that retains key information from previous tasks which the algorithm can revisit in order to refresh its memory of them.

Algorithm 1 describes coreset VCL. The algorithm maintains a pair $(C_t, \tilde{q}_t)$ comprising the coreset and the variational distribution for non-coreset data at each time step t. The distribution $\tilde{q}_t$ is continuously updated using eq. (2). When performing prediction, $\tilde{q}_t$ and C_t are combined to form the final variational distribution q_t using eq. (3). This step ensures recent exposure to data from each task and helps mitigate any residual forgetting. Treating the non-coreset and coreset data points separately in the algorithm also ensures the data are not over-represented in the posterior approximation.

VCL in Deep Discriminative Models. VCL requires specification of $q(\boldsymbol{\theta})$ where $\boldsymbol{\theta}$ in the current case is a D dimensional vector formed by stacking the network's biases and weights. For simplicity we use a Gaussian mean-field approximate posterior $q_t(\boldsymbol{\theta}) = \prod_{d=1}^{D} \mathcal{N}(\theta_{t,d}; \mu_{t,d}, \sigma_{t,d}^2)$. Training the network using the VFE approach in eq. (1) is equivalent to maximizing the variational lower bound:

$$\mathcal{L}_{\mathrm{VCL}}^t(q_t(\boldsymbol{\theta})) = \sum_{n=1}^{N_t} \mathbb{E}_{\boldsymbol{\theta} \sim q_t(\boldsymbol{\theta})}\big[\log p(y_t^{(n)}|\boldsymbol{\theta}, \mathbf{x}_t^{(n)})\big] - \mathrm{KL}(q_t(\boldsymbol{\theta})||q_{t-1}(\boldsymbol{\theta})) \tag{4}$$

Algorithm 1 Coreset VCL

Input: Prior $p(\boldsymbol{\theta})$, coreset size K from each task.
Output: Variational and predictive distributions at each step $\{q_t(\boldsymbol{\theta}), p(y^*|\boldsymbol{x}^*, \mathcal{D}_{1:t})\}_{t=1}^T$.

Initialize the coreset and variational approximation: $C_0 \leftarrow \emptyset$, $q_0 \leftarrow p_0$.
for $t = 1 \dots T$ **do**
  Observe the next dataset $\mathcal{D}_t$.
  $C_t \leftarrow$ update the coreset using C_{t-1} and $\mathcal{D}_t$.
  Update the variational distribution for non-coreset data points:

$$\tilde{q}_t(\boldsymbol{\theta}) \leftarrow \arg\min_{q \in \mathcal{Q}} \mathrm{KL}\big(q(\boldsymbol{\theta}) \parallel \tfrac{1}{Z} \tilde{q}_{t-1}(\boldsymbol{\theta})\, p(\mathcal{D}_t \cup C_{t-1} \setminus C_t | \boldsymbol{\theta})\big). \tag{2}$$

  Compute the final variational distribution (only used for prediction, and not propagation):

$$q_t(\boldsymbol{\theta}) \leftarrow \arg\min_{q \in \mathcal{Q}} \mathrm{KL}\big(q(\boldsymbol{\theta}) \parallel \tfrac{1}{Z} \tilde{q}_t(\boldsymbol{\theta}) p(C_t | \boldsymbol{\theta})\big). \tag{3}$$

  Perform prediction at test input $\boldsymbol{x}^*$: $p(y^*|\boldsymbol{x}^*, \mathcal{D}_{1:t}) = \int q_t(\boldsymbol{\theta}) p(y^*|\boldsymbol{\theta}, \boldsymbol{x}^*) \mathrm{d}\boldsymbol{\theta}$.
end for

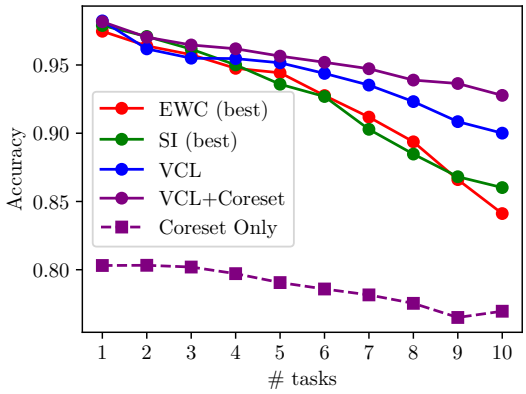

Figure 1: Average test set accuracy on all observed tasks for Permuted MNIST.

w.r.t. the variational parameters $\{\mu_{t,d}, \sigma_{t,d}\}_{d=1}^D$. Whilst the KL term can be computed in closed-form, the expected log-likelihood requires further approximation. Here we take the usual approach of employing simple Monte Carlo and the *reparameterization* trick to compute the gradients [22, 23]. At the first time step, the prior, and therefore $q_0(\boldsymbol{\theta})$ is chosen to be a multivariate Gaussian [14, 24].

VCL in Deep Generative Models. We now extend VCL to encompass variational auto-encoders (VAEs) [23, 25]. Consider a model $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})p(\mathbf{z})$, for observed data $\mathbf{x}$ and latent variables $\mathbf{z}$. The prior over latent variables $p(\mathbf{z})$ is typically Gaussian, and the distributional parameters of $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$ are defined by a deep neural network. In the batch setting, the standard VAE approach learns the parameters $\boldsymbol{\theta}$ by approximate maximum likelihood estimation, which is unsuitable for continual learning as it does not return parameter uncertainty estimates that are critical for weighting the information learned from old data. So, instead the VCL approach will approximate the posterior distribution over parameters after observing the t-th dataset. The approximate posterior q_t is obtained by maximizing the variational lower bound:

$$\mathcal{L}_{\mathrm{VCL}}^t(q_t(\boldsymbol{\theta}), \boldsymbol{\phi}) = \mathbb{E}_{q_t(\boldsymbol{\theta})}\Big\{ \textstyle\sum_{n=1}^{N_t} \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_t^{(n)})}\big[\log \frac{p(\mathbf{x}_t^{(n)}|\mathbf{z}, \boldsymbol{\theta}) p(\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_t^{(n)})}\big] \Big\} - \mathrm{KL}(q_t(\boldsymbol{\theta}) || q_{t-1}(\boldsymbol{\theta})) \tag{5}$$

w.r.t. q_t and $\boldsymbol{\phi}$, and the encoder network $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_t^{(n)})$ is parameterized by $\boldsymbol{\phi}$ which is task-specific.

3 Experiments

We now discuss two experiments for VCL. For additional experiments and discussions, see [26].

Deep Discriminative Models. We test VCL on the permuted MNIST benchmark for continual learning [6, 9, 10]. The dataset received at each time step consists of labeled MNIST images whose

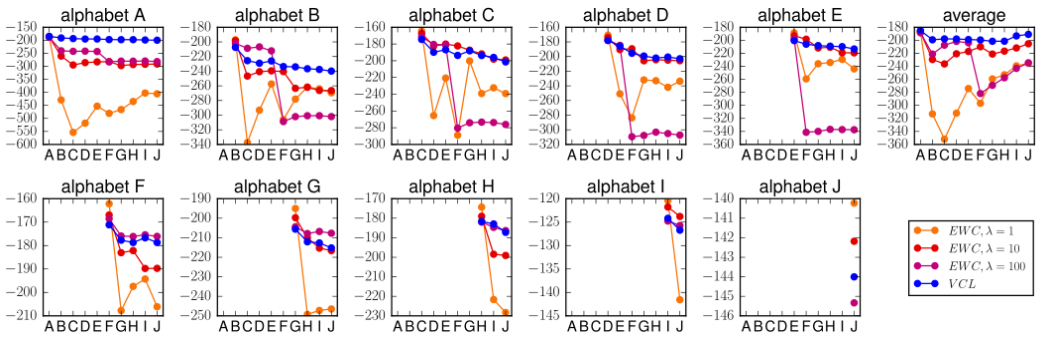

Figure 2: Test-LL results for VAE on notMNIST (small) dataset. The higher the better.

pixels have undergone a fixed random permutation. We compare VCL to Elastic Weight Consolidation (EWC) [9] and Synaptic Intelligence (SI) [10] approaches. The models consist of two fully connected hidden layers, each of which contains 100 hidden units. We evaluate both VCL and VCL with a coreset randomly selected from each task.

From results in fig. 1, VCL outperforms the EWC and SI by large margins, even though they benefited from an extensive hyper-parameter search (see the appendix of the full paper [26] for the hyper-parameter search of EWC and SI). After 10 tasks, VCL achieves 90% accuracy, while EWC and SI only achieve 84% and 86% respectively. The results also show that using the coreset alone is not useful for this task, but combining it with VCL leads to a modest improvement over VCL to 93% accuracy.

Deep Generative Models. We consider generating 10 characters A to J of the notMNIST dataset when 10 datasets containing images of each character are received in sequence. The generative model consists of shared and task-specific components, each represented by a one hidden layer neural network with 500 hidden units. The dimensionality of the latent variable and the intermediate representation are 50 and 500, respectively. We use task-specific encoders that are neural networks with symmetric architectures to the generator.

We compare VCL to EWC (with hyper-parameters $\lambda = 1, 10, 100$). The methods are quantitatively evaluated in fig. 2 by an importance sampling estimate of the test log-likelihood (test-LL). Initially, VCL performs slightly worse on the most recent task when compared to EWC. This is likely due to the fact that VCL optimizes a distribution over $\boldsymbol{\theta}$ while EWC computes point estimates, and the latter strategy often produces higher test-LL. However, importantly VCL has a superior long-term memory of previous tasks which leads to better overall performance. We note that EWC benefited here from carefully tuning the hyper-parameter. This again reveals a key advantage of VCL: its objective function is hyper-parameter free.

4 Conclusion

Approximate Bayesian inference provides a natural framework for continual learning. VCL, developed in this paper, is an approach in this vein that extends online VI to handle general continual learning tasks and complex neural network models. VCL can be enhanced by including a small episodic memory that leverages coreset algorithms from statistics. We demonstrated how the framework can be applied to both discriminative and generative models, giving state-of-the-art performance compared to previous approaches, even though it has no free parameters in its objective function.

Acknowledgments

The authors would like to thank Brian Trippe, Siddharth Swaroop, and Matej Balog for insightful comments and discussions. Cuong V. Nguyen is supported by EPSRC grant EP/M0269571. Yingzhen Li is supported by the Schlumberger FFTF Fellowship. Thang D. Bui is supported by the Google European Doctoral Fellowship. Richard E. Turner is supported by Google as well as EPSRC grants EP/M0269571 and EP/L000776/1.

References

[1] Jeffrey C. Schlimmer and Douglas Fisher. A case study of incremental concept induction. In *The National Conference on Artificial Intelligence*, 1986.

[2] Richard S. Sutton and Steven D. Whitehead. Online learning with random representations. In *International Conference on Machine Learning*, 1993.

[3] Mark B. Ring. CHILD: A first step towards continual learning. *Machine Learning*, 1997.

[4] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 1989.

[5] Roger Ratcliff. Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions. *Psychological Review*, 1990.

[6] Ian J. Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. In *International Conference on Learning Representations*, 2014.

[7] Sang-Woo Lee, Jin-Hwa Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Overcoming catastrophic forgetting by incremental moment matching. In *Advances in Neural Information Processing Systems*, 2017.

[8] Zhizhong Li and Derek Hoiem. Learning without forgetting. In *European Conference on Computer Vision*, 2016.

[9] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 2017.

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

[11] Zoubin Ghahramani and H. Attias. Online variational Bayesian learning. In *NIPS Workshop on Online Learning*, 2000.

[12] Masa-Aki Sato. Online model selection based on the variational Bayes. *Neural Computation*, 2001.

[13] Tamara Broderick, Nicholas Boyd, Andre Wibisono, Ashia C. Wilson, and Michael I. Jordan. Streaming variational Bayes. In *Advances in Neural Information Processing Systems*, 2013.

[14] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, 2015.

[15] Olivier Bachem, Mario Lucic, and Andreas Krause. Coresets for nonparametric estimation – the case of DP-means. In *International Conference on Machine Learning*, 2015.

[16] Jonathan Huggins, Trevor Campbell, and Tamara Broderick. Coresets for scalable Bayesian logistic regression. In *Advances in Neural Information Processing Systems*, 2016.

[17] Alex J. Smola, S.V.N. Vishwanathan, and Eleazar Eskin. Laplace propagation. In *Advances in Neural Information Processing Systems*, 2004.

[18] Peter S. Maybeck. *Stochastic models, estimation, and control*. Academic Press, 1982.

[19] Jun S. Liu and Rong Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 1998.

[20] Thang Bui, Daniel Hernández-Lobato, Jose Hernandez-Lobato, Yingzhen Li, and Richard Turner. Deep Gaussian processes for regression using approximate expectation propagation. In *International Conference on Machine Learning*, 2016.

[21] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, 2017.

[22] Tim Salimans and David A. Knowles. Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 2013.

[23] Diederik P. Kingma and Max Welling. Stochastic gradient VB and the variational auto-encoder. In *International Conference on Learning Representations*, 2014.

[24] Alex Graves. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems*, 2011.

[25] Danilo J. Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, 2014.

[26] Cuong V. Nguyen, Yingzhen Li, Thang D. Bui, and Richard E. Turner. Variational continual learning. *arXiv:1710.10628*, 2017.