Exploring Symmetric Measures for Variational Inference in Continual Learning

Jeremiah Ridge
jeremiah.ridge@wolfson.ox.ac.uk

Abstract

This study looks to examine the problem of continual learning through the VCL framework originally proposed by Nguyen et al. [12]. Specifically, this paper aims to understand how the choice of a symmetric or asymmetric measure of difference between two distributions, as employed in variational inference, impacts performance on continual learning tasks. The experiments conducted focus on discriminative tasks, and results show that symmetric measures can yield performance improvements over the original VCL framework under certain conditions.

1 Introduction

Online learning, and in particular continual learning, is a long-standing problem in computer science. It is a setting characterized by the need to continuously integrate new (sometimes non-i.i.d.) data to learn changing and/or novel tasks over time - often how real-world tasks manifest. However, a significant challenge plagues attempts to solve this domain: catastrophic forgetting [11] [6]. A 2018 paper by Nguyen et al. introduced the Variational Continual Learning (VCL) framework to address the problem of catastrophic forgetting in continual learning [12]. Stemming from the observation that Bayesian inference provided a ready basis for continual learning, VCL joins together two methods of approximate inference: online variational inference (VI) [4][14][3] and Monte Carlo VI for neural networks [2]. Additionally, Nguyen et al. incorporate the coreset data summarization method [1][7] to enhance VCL with a small episodic memory. Approximate inference techniques are necessary for subverting the intractability of exact Bayesian inference, in which the goal is to approximate the true posterior distribution. The choice of VI in particular relies on minimizing KL divergence over the set of allowed approximate posteriors (assumed to be the class of Gaussian distributions for the purposes of this study, as was the case of Nguyen et al.). KL divergence is generally the standard choice to measure the "distance" between distributions because it is mathematically convenient. Notably, however, KL divergence is not formally a metric of distance - one such reason being that it is not symmetric. While this asymmetry can be beneficial, this study hypothesizes that this feature may be sub-optimal in the case of continual learning, where the posterior being approximated for a new task might be substantially different from that of a previous one. The standard asymmetric measure could lead to models which fit to the distribution of the new task without prioritizing the need to preserve features of older, different tasks. This paper thus looks to explore the impact of leveraging symmetric measures of difference between two distributions, namely Jensen-Shannon (JS) divergence and Bhattacharyya distance, in the VCL framework.

2 Related Work

A recent survey [16] of continual learning (sometimes referred to as incremental learning or lifelong learning) categorized the existing frameworks into a set of five approaches, namely: regularization-based, replay-based, optimization-based, representation-based, and architecture-based. VCL falls under the umbrella of regularization-based approaches and as such it is no surprise that many of

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

the other frameworks chosen by the authors to serve as their baseline are similarly categorized. Specifically, the VCL paper selects maximum-likelihood and MAP estimation [6][10], Laplace propagation (LP) [15], elastic weight consolidation (EWC) [10], and synaptic intelligence (SI) [17] as their point of comparison. Regularization-based approaches can be generally described as the idea of adding regularization terms to balance between tasks as they arise. The category can be further divided into those focusing on weight (i.e. the variation of network parameters) regularization or function (i.e. the intermediate or final output of the prediction) regularization. All of the methods listed above are examples of weight regularization - VCL itself directly estimates a recursive approximate variational posterior. That being said, it is worth noting that many architectures combine features from multiple of these categories in a hybrid manner. Thus, while VCL can most broadly be categorized as a regularization-based approach, the addition of the coreset ties in principles from replay-based frameworks while the option of a multi-headed configuration touches on ideas from architecture-based approaches. The challenge posed by continual learning still holds the interest of the scientific community and many advancements have been achieved since the introduction of VCL. Regularization-based approaches continue to achieve state-of-the-art, such as Variational Auto-Regressive Gaussian Processes for Continual Learning (VAR-GPs) [8], which notably relied on VCL as its choice of baseline as recently as 2021. With respect to the particular question of study presented by this paper, there are some approaches which do not leverage approximate inference methods dependent on KL-divergence such as Kronecker factored online Laplace approximation [13]. However, none of them challenge the idea of symmetry with respect to this domain. As a result, this paper is well situated to examine the question of continual learning from a novel angle.

3 Proposed Approach

The goal of this study is to investigate the assumption that it is optimal to use an asymmetric measure, specifically KL-divergence, for evaluating approximate distributions in regularization-based approaches to online learning. This work will be carried out within the VCL framework as it is a well-established architecture within this class of approaches and continues to produce competitive results. The two symmetric measures that will be explored as alternatives to KL-divergence are JS divergence and Bhattacharyya distance, and the set of allowed approximate posteriors will be restricted to the set of Gaussians. In the pursuit of this goal, the original VCL architecture is first implemented as proposed in its debut paper [12] to establish performance baselines, though updated to leverage the PyTorch library as opposed to the authors' original choice of TensorFlow. The original paper repository [1], as well as a publicly available PyTorch version [2], served as a source of inspiration and assistance to this end. All code, data, and results from this study can be found at its repository here [3].

4 Theory

As given by Nguyen et al. [12], "variational continual learning employs a projection operator defined through a KL divergence minimization over the set of allowed approximate posteriors Q". In a general form,

$$q_t(\theta) = \arg\min_{q \in Q} KL\left(q(\theta) \middle\| \frac{1}{Z_t} q_{t-1}(\theta) p(D_t|\theta)\right), \text{ for } t = 1, 2, \ldots, T.$$

where Q is for our purposes the set of Gaussian distributions, Z_t an intractable normalizing constant which is not necessary for the optimization, and $q_0(\theta) =$ the prior, $p(\theta)$. In practice, KL takes the form:

$$KL(q, p) = \sum_k \frac{1}{2}\left(s_{1k}^{-2} + s_{0k}^{-2}(m_{1k} - m_{0k})^2 - 1 + \log\left(\frac{s_{1k}^2}{s_{0k}^2}\right)\right),$$

given $\mathbf{x} = [x_1, \ldots, x_K]$,

$$q(\mathbf{x}) = \mathcal{N}(\mathbf{x}; m_0, S_0) \text{ with } S_0 = \text{diag}([s_{01}^2, \ldots, s_{0K}^2]),$$

[1] https://github.com/nvcuong/variational-continual-learning/tree/master

[2] https://github.com/pihey1995/VariationalContinualLearning/tree/master

[3] https://github.com/ridgejo/Symmetric-VCL

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; m_1, S_1) \text{ with } S_1 = \text{diag}([s_{11}^2, \ldots, s_{1K}^2]),$$
$$\mathcal{N}(\mathbf{x}; m_0, S_0) = \prod_k \mathcal{N}(x_k; m_{0k}, s_{0k}^2),$$

and the independence of $x_1, \ldots, x_K$. This is evaluated layer-wise with the output for each layer then normalized by its respective dimensions before computing the sum. This formulation of the projection function allows the choice of measurement between distributions to be treated in a modular way. Thus, this study introduces JS divergence and Bhattacharyya distance into the VCL framework. JS divergence is a smoothed, symmetric version of KL divergence defined as

$$JSD(q \parallel p) = \frac{1}{2} KL(q \parallel M) + \frac{1}{2} KL(p \parallel M),$$

where $M = \frac{1}{2}(q + p)$ is a mixture distribution of q and p. It is important to note once more that this analysis is being carried out under the assumption that we are working with Gaussian distributions. KL divergence for Gaussians has a closed-form expression which can be computed analytically, and the mixture distribution M, being a linear combination of two Gaussians, is itself a Gaussian. In this restricted case, JS divergence therefore remains analytically tractable and can be evaluated in a layer-wise approach analogous to that shown for KL divergence. Bhattacharyya distance is another symmetric measure which provides a slightly more geometric interpretation of how close two distributions are to each other. Over Gaussian distributions it is defined as

$$BD(p, q) = \frac{1}{4} \left(\frac{(\mu_p - \mu_q)^2}{\sigma_p^2 + \sigma_q^2} \right) + \frac{1}{2} \ln \left(\frac{\sigma_p^2 + \sigma_q^2}{2\sigma_p \sigma_q} \right),$$

where μ, σ are the mean and variance of their respective distribution. As was the case with JS divergence, because of the simplifying assumption carried over from the original paper that all distributions fall within the set of Gaussians, Bhattacharyya distance is analytically tractable and can also be implemented in a layer-wise fashion. This study explores applications of these two symmetric measures both over the course of learning all tasks, as well as in a restricted case where KL divergence is used as each new task is learned but then swapped out for a symmetric measure when reincorporating information from the coreset.

5 Experiments

Keeping in mind the objective of evaluating solely the effect of symmetric vs asymmetric measures between Gaussians in the VCL framework, the structure of all experiments carried out mirrors those of the original paper by Nguyen et al. The focus is constrained only to the discriminative tasks given the scope of this study. For all experiments, Adam optimizer [9] with learning rate 10^{-3} is used to match the specifications of the VCL paper. As laid out in the "Proposed Approach", the VCL framework was first implemented in a manner faithful to the original paper in order to establish that performance equaled that reported by the authors. The dataset used for evaluation is the popular MNIST image corpus employed across two separate tasks: permuted MNIST and split MNIST. Beyond their use in the VCL paper, both tasks are well-established continual learning benchmarks. In permuted MNIST, each timestep consists of labeled images to which a fixed random permutation has been applied. In split MNIST, five binary classification tasks are presented in sequence: 0/1, 2/3, 4/5, 6/7, 8/9. As in the VCL paper, for permuted MNIST a fully-connected single-head network is used with two hidden layers of 100 units and for split MNIST a fully connected multi-head network is used with two hidden layers of 256 units. In both, ReLU activations are used. For all experiments, the following are evaluated: VCL, VCL with JS divergence (denoted JSD-VCL), VCL with Bhattacharyya distance (denoted BD-VCL), and VCL where Bhattacharyya distance is only applied when training on the coreset (denoted coresetBD-VCL). All of these methods are evaluated (when possible) in their base form, with a random coreset, and with a coreset selected by the k-center method [5]. All coresets are standardized to a size of 200, and cursory ablation testing did not find a strong effect of learning rate on the different methods. Results are reported in table 1 and additional graphical visualizations can be found in the appendix of supplementary materials.

The experimental methods JSD-VCL and BD-VCL both consistently performed worse than the original VCL architecture on permuted MNIST, with and without coresets. The coresetBD-VCL method achieved comparable performance to the best results of any of the original paper methods,

	Average Accuracy After All Tasks	
Method	Permuted MNIST	Split MNIST
VCL	90.4% , 90.0%*	98.1% , 97.0%*
VCL + Random	93.0%, 93.0%*	98.5%
VCL + k-Center	**93.4**%, 93.0%*	98.4%, 98.4%*
JSD-VCL	21.8%	96.3%
JSD-VCL + Random	77.9%	98.7%
JSD-VCL + k-Center	-	98.7%
BD-VCL	40.2%	97.3%
BD-VCL + Random	83.6%	98.9%
BD-VCL + k-Center	-	**99.1**%
coresetBD-VCL + Random	93.3%	98.4%
coresetBD-VCL + k-Center	**93.4**%	98.4%
random coreset only	-	96.4%
k-Center coreset only	61.8%	88.9%
EWC	84.0%*	63.1%*
SI	86.0%*	98.9%*
LP	82.0%*	61.2%*

Table 1: Average test accuracy for each method after all tasks on both permuted MNIST and split MNIST experiments. Results as reported by the original VCL paper are marked by a single asterisk (*). Results in bold reflect the highest performance for a given experiment.

though it is unclear whether this can be fully attributed to the experimental change. However, the opposite case was true for split MNIST, with JSD-VCL and BD-VCL recording performance gains over the original VCL framework. These mixed results do not entirely support the original hypothesis, but do suggest some interesting conclusions which are discussed in the next section.

6 Discussion and Conclusion

When comparing just the base methods (without coresets), it is apparent that the symmetric approaches actually suffer from catastrophic forgetting to a much higher degree - with performance decreasing almost immediately as new tasks are introduced (see attached appendix for visualizations). It seems that while symmetric measures don't perform as well on their own, they are, however, much more well suited to leverage information from episodic memory (the coreset) than asymmetric measures are. Where the addition of a coreset only yields performance gains on permuted MNIST of ~3% with KL divergence, when using Bhattacharyya distance or JS divergence this result jumps to ~43% or ~56%, respectively. This observation was the original inspiration behind the addition of coresetBD-VCL as an experimental framework. CoresetBD-VCL performs comparatively to the vanilla VCL + coreset methods, but does not offer any sort of real improvement over the latter. In fact, based off these results it would seem that the performance improvements in the case of a base VCL architecture and the permuted MNIST experiment come entirely from the addition of the coreset and are agnostic to which method is used during training on that coreset. This suggests that the influence of the method used during the bulk of training dominates that used for the coreset information, which is fairly unsurprising given the small coreset size tested. Further testing with larger coreset sizes is warranted, in particular to see if the asymptotic effect noted by Nguyen et al. holds for symmetric measures as well. Hyperparameter search for use with symmetric variants was also not investigated to its fullest extent by the current study, though no strong effect was found in brief ad hoc testing. Interestingly, for split MNIST both symmetric approaches (with coresets) report better average accuracy than all VCL variants from the original paper as well as SI, one of the baselines chosen by Nguyen et al. which outperformed their own method. This difference between the permuted and split MNIST experiments can likely be attributed to the change from a single to multi-headed architecture. It is also worth noting that the heuristic used in coreset construction does appear to have an affect on performance as well. While the original hypothesis that symmetric measures may be more robust to catastrophic forgetting in VCL seems to have been disproved, this work ultimately supports further research in this area. In particular, additional study on symmetric measures in multi-headed architectures, larger coreset sizes, and different coreset heuristics may prove fruitful.

References

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

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

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

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

[5] Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985. ISSN 0304-3975. doi: https://doi.org/10.1016/0304-3975(85)90224-5.

[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*, 2014a.

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

[8] Sanyam Kapoor, Theofanis Karaletsos, and Thang D. Bui. Variational auto-regressive gaussian processes for continual learning. In *International Conference on Machine Learning (ICML)*, 2021.

[9] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference for Learning Representations, San Diego*, 2017.

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

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

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

[13] Hippolyt Ritter, Aleksandar Botev, and David Barber. Online structured laplace approximations for overcoming catastrophic forgetting, 2018.

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

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

[16] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application, 2024.

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