The effects of training and evaluation schemes in VCL

1083152

Abstract

This paper critically examines variational continual learning (VCL). Centrally, we contend that the interleaved training and evaluation strategy employed in the original VCL paper unfairly favours the models with coresets. Hence, we first reassess the impact of coresets in VCL. As the results show no substantial advantages, we secondly explore a coreset-free training scheme modification to improve VCL performance: applying learning rate decay to the shared weights. Our findings show that complete decay (parameter freezing) is the most effective strategy, as it makes the model more stable and prevents catastrophic forgetting.

1 Introduction

Continual learning (CL) is a machine learning paradigm where a stream of data is used to continuously extend an existing model's knowledge. Unlike traditional machine learning assumptions, this stream need not be i.i.d. and tasks may change over time. These relaxed assumptions are appropriate for many real-world settings, as the world is inherently dynamic, and ignoring this fact may lead to concept drift [1]. The primary challenge lies in striking a balance between adapting to new data and retaining knowledge from previous tasks, often termed the 'stability-plasticity problem' [2]. Too much stability hinders adaptabilty and leads to model drift, whereas too much plasticity can cause a decline in performance on previously learned tasks, known as the catastrophic forgetting problem [3].

Variational Continual Learning (VCL) [4] is an approach that addresses this problem by leveraging online variational inference [5] and enhancing it with a small episodic memory called a *coreset*. The key idea behind VCL is to maintain a posterior distribution over the model parameters, allowing the model to capture uncertainty and thus limit updates to important parameters.

It was derived from the source code of the VCL paper [4], however, that they employed an interleaved training/evaluation scheme (ITES) to evaluate the coreset methods. This means that the reported accuracies in [4] were obtained by sequentially training on the task 1 coreset data, evaluating on task 1, training on the task 2 coreset data, evaluating on task 2, and so forth. This strategy benefits models that suffer from catastrophic forgetting, which is precisely the problem that coresets aim to address.

Therefore, we start by reinvestigating the impact of coresets in VCL when using a more sensible sequential training/evaluation scheme (STES), where we first train the prediction model on all coreset data and only then evaluate it on all test data. Further, we explore the efficacy of LR decay on the shared weights – a coreset-free training scheme modification – for improving VCL performance.

2 Related Work

Most recent approaches in continual learning can be categorized into one of three families: replay methods, regularization-based methods and parameter isolation methods [6]. Replay methods store a subset of the data from previous tasks or generate synthetic data with generative models. This data is then used to either explicitly retrain the model on previous tasks, such as in iCaRL [7] and Experience Replay [8, 9], or to regularize the learning process, like in GEM and A-GEM [10, 11]. Regularization-based methods, on the other hand, introduce an extra regularization term in the

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

loss function that penalizes changes to important parameters. The baselines employed in the VCL paper [4] are all prime examples of this family. These include EWC [12], SI [13] and LP [14]. Finally, parameter isolation methods aim to dedicate different model parameters to each task to avoid catastrophic forgetting. Two important examples are PNN [15] and PathNet [16].

VCL [4] borrows from the ideas of all three families. In essence, it regularizes the learning process by maintaining a posterior distribution over the model parameters. Further, it employs parameter isolation due to the multi-head architecture of the Bayesian neural network. Finally, it uses the coreset to store data from previous tasks and periodically retrain the posterior.

Like every replay method in continual learning, VCL [4] must deal with the question which data to store in a buffer, usually of limited size [7, 17]. It is the first continual learning paper that uses coresets – originally devised for data summarization [18] – to solve this problem. However, this paper provides no conclusive evidence for their advantages in VCL due to the skewed evaluation strategy. Some later publications reported a similar performance benefit [19, 20], but we are unaware of any previous work that endorses coresets while providing an open-source setup that does not use ITES.

LR decay is a standard technique in deep learning known to help both optimization and generalization [21, 22]. In the context of CL, it has been shown to mitigate catastrophic forgetting [23, 17]. To the best of our knowledge, no previous work has investigated the effect of using a lower LR for only the shared weights of a multi-head architecture. This approach is related to the parameter isolation family, as it aims to combat catastrophic forgetting by making the shared weights more stable.

3 Proposed Approach

In this paper, we examine two research questions. The first investigates to which degree the benefits of coresets reported in the VCL paper [4] stem from ITES versus from actual benefits. We hypothesize that the reported performance when using coresets is greatly inflated by the evaluation strategy, but that there is also a small actual benefit. To study this, we contrast ITES with our STES (described in Section 1). We do this by replicating the original paper's experimental setup for the Split MNIST task and comparing the average test accuracies of both strategies.

The second research question is whether and to which extent we can improve (no-coreset) VCL using LR decay on the shared weights. We hypothesize that moderate decay will improve performance by preventing catastrophic forgetting, but that extreme decay will make the model unable to adapt to new tasks. To study this, we reuse the same experimental setup, but use the Split FashionMNIST dataset instead because it is more challenging and can thus better highlight the effects of different approaches. Following [17], we use an exponential decay schedule for the LR and compare the test accuracies for decay rates ranging from 1 (no decay) to 0 (parameter freezing).

All code for this paper is implemented from scratch in PyTorch and is available in the following anonymous GitHub repository: `https://anonymous.4open.science/r/UDL-7F3E`.

4 Theory

Consider a discriminative model that, given an input x and parameters $\boldsymbol{\theta}$, outputs a probability distribution over the classes $p(y \mid x, \boldsymbol{\theta})$. In traditional machine learning, one would try to find the (fixed) parameters $\boldsymbol{\theta}$ that maximize the likelihood of the observed data. In variational inference, however, one tries instead to find the posterior distribution $p(\boldsymbol{\theta} \mid \mathcal{D}_{1:T})$ over the parameters given a prior $p(\boldsymbol{\theta})$ and the data $\mathcal{D}_{1:T} = \{\mathcal{D}_t \mid 1 \leq t \leq T\}$.

Assuming $\boldsymbol{\theta}$ are the weights and biases of a neural network, we use a Gaussian mean-field approximation for the posterior [24]. Let $\mathcal{Q} = \{\mathcal{N}(\boldsymbol{\theta} \mid \boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma})^2)\}$ be the set of all potential such approximations. By extending the derivation in [4], we show the following theorem in Appendix A.

Theorem 1. *The iterative method with $q_0(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$ and the following update rule yields approximations $q_t(\boldsymbol{\theta}) \approx p(\boldsymbol{\theta} \mid \mathcal{D}_{1:t})$:*

$$q_t(\boldsymbol{\theta}) = \arg\min_{q \in \mathcal{Q}} KL\left(q(\boldsymbol{\theta}) \,\middle\|\, \frac{1}{p(\mathcal{D}_t)} q_{t-1}(\boldsymbol{\theta}) p(\mathcal{D}_t \mid \boldsymbol{\theta})\right) \tag{1}$$

Note that this minimization problem cannot be solved exactly. The following theorem gives an objective function to optimize w.r.t. $\boldsymbol{\psi} = (\boldsymbol{\mu}, \boldsymbol{\sigma})$, for which we don't need the intractable $p(\mathcal{D}_t)$.

Theorem 2. *Minimizing (1) w.r.t. $q \in \mathcal{Q}$ is equivalent to maximizing the following ELBO w.r.t. $\boldsymbol{\psi}$:*

$$\mathcal{L}_{VCL}^{t}(\boldsymbol{\psi}) \coloneqq \sum_{n=1}^{N_t} \mathbb{E}_{\boldsymbol{\theta} \sim q_{\boldsymbol{\psi}}(\boldsymbol{\theta})} \left[\log p(y_t^{(n)} \mid x_t^{(n)}, \boldsymbol{\theta}) \right] - KL\left(q_{\boldsymbol{\psi}}(\boldsymbol{\theta}) \parallel q_{t-1}(\boldsymbol{\theta})\right) \tag{2}$$

The first term in the ELBO is the log-likelihood of the data $\mathcal{D}_t$ under the current posterior $q_{\boldsymbol{\psi}}(\boldsymbol{\theta})$. The second term is the KL divergence between the current posterior and the previous posterior, which acts as a regularization term to prevent catastrophic forgetting. The ELBO is a lower bound on $p(\mathcal{D}_t)$, and maximizing it is equivalent to minimizing the KL divergence in (1).

The second term can be efficiently computed by using the closed-form formula for the KL divergence between two multivariate diagonal Gaussian distributions, that was derived in the UDL lectures. For the first term, we employ the local parametrization trick and Monte Carlo sampling to approximate its gradient [25], which leads to the following unbiased and tractable estimator for $\nabla_{\boldsymbol{\psi}} \mathcal{L}_{VCL}^{t}(\boldsymbol{\psi})$.

Theorem 3. *The following is an unbiased estimator for $\nabla_{\boldsymbol{\psi}} \mathcal{L}_{VCL}^{t}(\boldsymbol{\psi})$:*

$$\hat{G}(\boldsymbol{\psi}) = \frac{1}{K} \sum_{k=1}^{K} \nabla_{\boldsymbol{\psi}} I(g(\boldsymbol{\psi}, \boldsymbol{\epsilon}^{(k)})) - \nabla_{\boldsymbol{\psi}} KL\left(q_{\boldsymbol{\psi}}(\boldsymbol{\theta}) \parallel q_{t-1}(\boldsymbol{\theta})\right) \tag{3}$$

$$\textit{where} \quad I(\boldsymbol{\theta}) = \sum_{n=1}^{N_t} \log p(y_t^{(n)} \mid x_t^{(n)}, \boldsymbol{\theta}), \quad g(\boldsymbol{\psi}, \boldsymbol{\epsilon}) = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon}^{(k)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

Note that both $I(g(\boldsymbol{\psi}, \boldsymbol{\epsilon}^{(k)}))$ and $\mathrm{KL}\left(q_{\boldsymbol{\psi}}(\boldsymbol{\theta}) \parallel q_{t-1}(\boldsymbol{\theta})\right)$ can be computed in closed-form and are differentiable w.r.t. $\boldsymbol{\psi} = (\boldsymbol{\mu}, \boldsymbol{\sigma})$. Thus, we can use PyTorch's automatic differentiation engine to compute the gradient estimates and optimize the ELBO using stochastic gradient descent.

5 Experiments

5.1 Sequential vs Interleaved Evaluation

In this experiment, we replicate the VCL paper's experimental setup for Split MNIST and compare the results of the following five training/evaluation schemes: (1) VCL, (2) VCL+RandomCS+STES, (3) VCL+KCenterCS+STES, (4) VCL+RandomCS+ITES, and (5) VCL+KCenterCS+ITES. We use the same hyperparameters and optimizer as the original paper, which are summarized in Table 1. Further, we also use the same model architecture, which is a fully connected neural network with two hidden layers, each of which contains 256 hidden units with ReLU activations. The last layer consists of a separate head for each task that transforms the shared features into a softmax distribution. The used performance metric is the average reported test accuracy on all observed tasks, to allow comparison with [4]. All reported results are the average over 10 runs with different random seeds.

Table 1: Hyperparameters

Hyperparameter	Value
Objective	$\mathcal{L}_{VCL}^{t}$
Optimizer	Adam
Learning Rate	10^{-3}
Epochs	120
Batch Size	Full
Coreset Size	1000
# Hidden Layers	2
Hidden Dimension	256
Nonlinearity	ReLU
# Gradient Samples	10

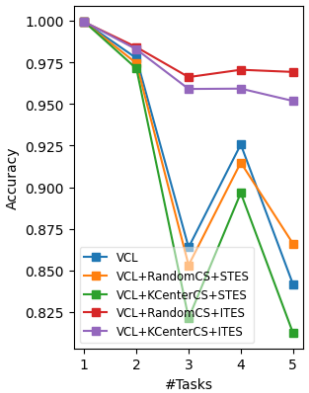

Figure 1: Average test accuracy on all observed tasks in the Split MNIST experiment

Figure 1 shows the evolution of this performance metric under different training/evaluation schemes. Unfortunately, we were unable to replicate the high performance reported for vanilla VCL in [4]. We attribute this to minor differences in the experimental setup, such as variations in the defaults of PyTorch and TensorFlow. Nonetheless, the comparative analysis of different strategies remains insightful.

Specifically, the models using ITES report a better performance across the board, whereas those using STES report comparable accuracies. Notably, there is no discernible advantage to using coresets when using STES. This result supports a stronger interpretation of our original hypothesis, as it implies that the reported benefits of coresets in [4] may stem solely from the evaluation strategy. We conclude that there is insufficient proof for the benefit of coresets in VCL and advise against their use in practice, at least until new, convincing evidence is available.

5.2 Learning Rate Decay on Shared Weights

In this experiment, we investigate the performance of VCL under different LR decay schedules for the shared weights. Following [17], we use exponential decay schedules with decay rates γ ranging from 1 (no decay) to 0 (parameter freezing). We use the same hyperparameters (Table 1) and model architecture as before, but replace Split MNIST by Split FashionMNIST. The LR of the heads is still 10^{-3}, while the LR of the shared weights is decayed after every task according to $\text{lr}_t = \gamma^t \cdot 10^{-3}$. We still use the test accuracy as the performance metric, but now report the accuracy on separate tasks and the last observed task as well to gain more insight into the effects of the different LR schedules.

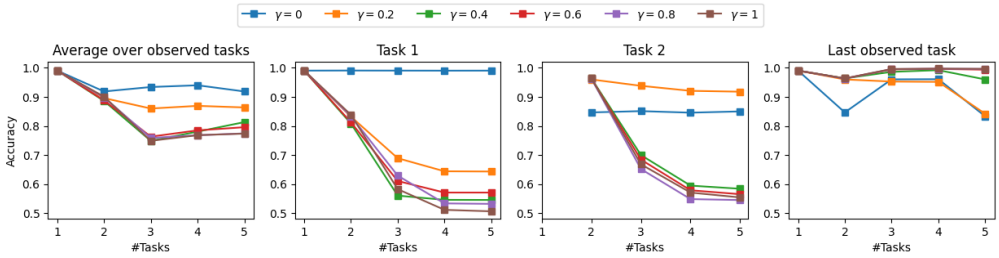

Figure 2: Test set accuracies for the Split FashionMNIST experiment. The x-axis always denotes the number of tasks that have been observed.

Figure 2 shows the test accuracies for the Split FashionMNIST experiment. There are three observations to be made. First, every $\gamma > 0$ results in catastrophic forgetting, as the models' performances on task 1 decrease significantly as more tasks are observed. Only $\gamma = 0.2$ alleviates this problem somewhat on the later tasks when the LR has already decayed significantly. Second, the schedule with $\gamma = 0$ (parameter freezing) is still reasonably good at adapting to new tasks. This is likely due to the fact that the shared weights learned on the first task generate shared features that are also effective for later tasks. Third, the schedule with $\gamma = 0$ performs the best overall due to its resistance to catastrophic forgetting. This refutes our hypothesis from Section 3 that only moderate decay would improve performance. We conclude that the parameter isolation in VCL introduced by the multi-head architecture already gives the model enough plasticity to adapt to new tasks, and that the shared weights should be kept fixed to make the model more stable and prevent catastrophic forgetting.

6 Discussion and Conclusion

Our first experiment investigated the impact of the evaluation scheme on the reported test accuracy and reexamined the benefits of coresets in VCL. The results support the hypothesis that ITES greatly inflates the reported performance of the coreset methods: under a more sensible evaluation strategy, no advantage to coresets was observed. We conclude that there is currently insufficient evidence for the benefit of coresets to justify their use in VCL.

We recognize, however, that our study is not without limitations. Small discrepancies in experimental setups may have influenced our results, as indicated by our inability to replicate the performance of vanilla VCL reported in [4]. Additionally, the results are based on a single experimental setup, and it is possible that coresets could be beneficial in other settings. For these reasons, further work is necessary to reach more conclusive results. Logical next steps would be to implement STES in the codebase of [4], to inquire with the authors of follow-up papers whether they discovered any benefits to using coresets and to evaluate the algorithm on more challenging datasets and with larger models.

In our second experiment, LR decay on shared weights was explored as an alternative method to improve VCL performance. Exceeding this original goal, the results suggest that simply freezing the shared weights and thus eliminating the need for variational inference, is a much more effective strategy for continual learning. This is a promising preliminary result, as it is a simple method that does not require any additional overhead. However, further work is necessary to extend this result to more realistic settings. To conclude, we identify two avenues for future research. First, the LR decay strategy whose effects were dwarfed in this paper by those of the parameter isolation strategy, could be further investigated in other settings. Second, the idea of parameter isolation could be explored in combination with other continual learning algorithms, such as iCaRL [7] and GEM [10].

A Derivations

Theorem 1. *The iterative method with $q_0(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$ and the following update rule yields approximations $q_t(\boldsymbol{\theta}) \approx p(\boldsymbol{\theta} \mid \mathcal{D}_{1:t})$:*

$$q_t(\boldsymbol{\theta}) = \arg\min_{q \in \mathcal{Q}} KL\left(q(\boldsymbol{\theta}) \,\middle\|\, \frac{1}{p(\mathcal{D}_t)} q_{t-1}(\boldsymbol{\theta}) p(\mathcal{D}_t \mid \boldsymbol{\theta})\right) \tag{1 revisited}$$

Proof. For the base case $t = 0$ of this iteration, $p(\boldsymbol{\theta} \mid \mathcal{D}_{1:t})$ degenerates to the prior $p(\boldsymbol{\theta})$. Following [4], we choose the prior $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} \mid \mathbf{0}, \boldsymbol{I})$, which is in $\mathcal{Q}$ and can hence be exactly represented by $q_0(\boldsymbol{\theta})$. So $q_0(\boldsymbol{\theta}) = p(\boldsymbol{\theta}) = p(\boldsymbol{\theta} \mid \mathcal{D}_{1:0})$ is exact.

Now we will derive the update rule. Using Bayes' rule, the posterior $p(\boldsymbol{\theta} \mid \mathcal{D}_{1:t})$ can be rewritten in a recursive manner:

$$p(\boldsymbol{\theta} \mid \mathcal{D}_{1:t}) = \frac{p(\mathcal{D}_{1:t} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{D}_{1:t})} = \frac{p(D_t \mid \boldsymbol{\theta}) p(\mathcal{D}_{1:t-1} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{D}_t) p(\mathcal{D}_{1:t-1})} = \frac{p(D_t \mid \boldsymbol{\theta})}{p(\mathcal{D}_t)} p(\boldsymbol{\theta} \mid \mathcal{D}_{1:t-1})$$

Suppose we have a good approximation $q_{t-1}(\boldsymbol{\theta}) \approx p(\boldsymbol{\theta} \mid \mathcal{D}_{1:t-1})$, it follows that

$$p(\boldsymbol{\theta} \mid \mathcal{D}_{1:t}) \approx \frac{p(D_t \mid \boldsymbol{\theta})}{p(\mathcal{D}_t)} q_{t-1}(\boldsymbol{\theta})$$

We can leverage this fact to obtain the next approximation $q_t(\boldsymbol{\theta}) \approx p(\boldsymbol{\theta} \mid \mathcal{D}_{1:t})$. As q_t is constrained to be in $\mathcal{Q}$ (i.e. a Gaussian mean-field approximation [24]), we would like to find the $q_t \in \mathcal{Q}$ that best approximates $p(\mathcal{D}_t)^{-1} p(D_t \mid \boldsymbol{\theta}) q_{t-1}(\boldsymbol{\theta})$. One interpretation of this is to find the $q_t \in \mathcal{Q}$ with minimal KL divergence w.r.t. this expression, which leads to the minimization problem in the theorem statement.

□

Theorem 2. *Minimizing (1) w.r.t. $q \in \mathcal{Q}$ is equivalent to maximizing the following ELBO w.r.t. $\boldsymbol{\psi}$:*

$$\mathcal{L}^t_{VCL}(\boldsymbol{\psi}) \coloneqq \sum_{n=1}^{N_t} \mathbb{E}_{\boldsymbol{\theta} \sim q_{\boldsymbol{\psi}}(\boldsymbol{\theta})} \left[\log p(y_t^{(n)} \mid x_t^{(n)}, \boldsymbol{\theta})\right] - KL\left(q_{\boldsymbol{\psi}}(\boldsymbol{\theta}) \,\|\, q_{t-1}(\boldsymbol{\theta})\right) \tag{2 revisited}$$

Proof. $\mathcal{Q}$ contains all diagonal multivariate Gaussian distributions over $\boldsymbol{\theta}$, i.e. precisely all distributions of the form $q_{\boldsymbol{\psi}}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} \mid \boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma})^2)$ parametrized by $\boldsymbol{\psi} = (\boldsymbol{\mu}, \boldsymbol{\sigma})$. Therefore, the optimum to (1) is $q_{\boldsymbol{\psi}^*}$ where

$$\boldsymbol{\psi}^* = \arg\min_{\boldsymbol{\psi}} \mathrm{KL}\left(q_{\boldsymbol{\psi}}(\boldsymbol{\theta}) \,\middle\|\, \frac{1}{p(\mathcal{D}_t)} q_{t-1}(\boldsymbol{\theta}) p(\mathcal{D}_t \mid \boldsymbol{\theta})\right)$$

Now define

$$\mathcal{K}(\boldsymbol{\psi}) \coloneqq \mathrm{KL}\left(q_{\boldsymbol{\psi}}(\boldsymbol{\theta}) \,\middle\|\, \frac{1}{p(\mathcal{D}_t)} q_{t-1}(\boldsymbol{\theta}) p(\mathcal{D}_t \mid \boldsymbol{\theta})\right)$$

Then, by the definition of the KL divergence, we have

$$\begin{aligned}
\mathcal{K}(\boldsymbol{\psi}) &= \int q_{\boldsymbol{\psi}}(\boldsymbol{\theta}) \log\left(\frac{q_{\boldsymbol{\psi}}(\boldsymbol{\theta}) p(\mathcal{D}_t)}{q_{t-1}(\boldsymbol{\theta}) p(\mathcal{D}_t \mid \boldsymbol{\theta})}\right) d\boldsymbol{\theta} \\
&= \int q_{\boldsymbol{\psi}}(\boldsymbol{\theta}) \log(p(\mathcal{D}_t))\, d\boldsymbol{\theta} + \int q_{\boldsymbol{\psi}}(\boldsymbol{\theta}) \log\left(\frac{q_{\boldsymbol{\psi}}(\boldsymbol{\theta})}{q_{t-1}(\boldsymbol{\theta})}\right) d\boldsymbol{\theta} - \int q_{\boldsymbol{\psi}}(\boldsymbol{\theta}) \log p(\mathcal{D}_t \mid \boldsymbol{\theta})\, d\boldsymbol{\theta} \\
&= \log(p(\mathcal{D}_t)) + \mathrm{KL}\left(q_{\boldsymbol{\psi}}(\boldsymbol{\theta}) \,\|\, q_{t-1}(\boldsymbol{\theta})\right) - \mathbb{E}_{\boldsymbol{\theta} \sim q_{\boldsymbol{\psi}}(\boldsymbol{\theta})} \left[\log p(\mathcal{D}_t \mid \boldsymbol{\theta})\right] \\
&= \log(p(\mathcal{D}_t)) + \mathrm{KL}\left(q_{\boldsymbol{\psi}}(\boldsymbol{\theta}) \,\|\, q_{t-1}(\boldsymbol{\theta})\right) - \mathbb{E}_{\boldsymbol{\theta} \sim q_{\boldsymbol{\psi}}(\boldsymbol{\theta})} \left[\log\left(\prod_{n=1}^{N_t} p(y_t^{(n)} \mid x_t^{(n)}, \boldsymbol{\theta})\right)\right] \\
&= \log(p(\mathcal{D}_t)) + \mathrm{KL}\left(q_{\boldsymbol{\psi}}(\boldsymbol{\theta}) \,\|\, q_{t-1}(\boldsymbol{\theta})\right) - \mathbb{E}_{\boldsymbol{\theta} \sim q_{\boldsymbol{\psi}}(\boldsymbol{\theta})} \left[\sum_{n=1}^{N_t} \log p(y_t^{(n)} \mid x_t^{(n)}, \boldsymbol{\theta})\right] \\
&= \log(p(\mathcal{D}_t)) + \mathrm{KL}\left(q_{\boldsymbol{\psi}}(\boldsymbol{\theta}) \,\|\, q_{t-1}(\boldsymbol{\theta})\right) - \sum_{n=1}^{N_t} \mathbb{E}_{\boldsymbol{\theta} \sim q_{\boldsymbol{\psi}}(\boldsymbol{\theta})} \left[\log p(y_t^{(n)} \mid x_t^{(n)}, \boldsymbol{\theta})\right]
\end{aligned}$$

Where the last step follows from linearity of expectation. Now define the ELBO as in the theorem statement:

$$\mathcal{L}_{VCL}^t(\boldsymbol{\psi}) \coloneqq \sum_{n=1}^{N_t} \mathbb{E}_{\boldsymbol{\theta}\sim q_{\boldsymbol{\psi}}(\boldsymbol{\theta})} \left[\log p(y_t^{(n)} \mid x_t^{(n)}, \boldsymbol{\theta})\right] - \mathrm{KL}\left(q_{\boldsymbol{\psi}}(\boldsymbol{\theta}) \parallel q_{t-1}(\boldsymbol{\theta})\right) \tag{4}$$

Then we have

$$\mathcal{K}(\boldsymbol{\psi}) = \log(p(\mathcal{D}_t)) - \mathcal{L}_{VCL}^t(\boldsymbol{\psi}) \tag{5}$$

Notice that $\mathcal{L}_{VCL}^t(\boldsymbol{\psi})$ is indeed an ELBO, as it is a lower bound on $\log(p(\mathcal{D}_t))$. This follows directly from the previous equation and the fact that $\mathcal{K}(\boldsymbol{\psi})$, like all KL divergences, is non-negative.

Further, because $\log(p(\mathcal{D}_t))$ is a constant w.r.t. $\boldsymbol{\psi}$, it follows from (5) that the minimum of $\mathcal{K}(\boldsymbol{\psi})$ coincides with the maximum of $\mathcal{L}_{VCL}^t(\boldsymbol{\psi})$. We conclude that the $q \in \mathcal{Q}$ that minimizes (1) is $q_{\boldsymbol{\psi}^*}$ where $\boldsymbol{\psi}^*$ maximizes the ELBO $\mathcal{L}_{VCL}^t(\boldsymbol{\psi}^*)$. □

Theorem 3. *The following is an unbiased estimator for* $\nabla_{\boldsymbol{\psi}}\mathcal{L}_{VCL}^t(\boldsymbol{\psi})$*:*

$$\hat{G}(\boldsymbol{\psi}) = \frac{1}{K}\sum_{k=1}^{K} \nabla_{\boldsymbol{\psi}} I(g(\boldsymbol{\psi}, \boldsymbol{\epsilon}^{(k)})) - \nabla_{\boldsymbol{\psi}} KL\left(q_{\boldsymbol{\psi}}(\boldsymbol{\theta}) \parallel q_{t-1}(\boldsymbol{\theta})\right) \tag{3 revisited}$$

where $\quad I(\boldsymbol{\theta}) = \sum_{n=1}^{N_t} \log p(y_t^{(n)} \mid x_t^{(n)}, \boldsymbol{\theta}), \quad g(\boldsymbol{\psi}, \boldsymbol{\epsilon}) = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon}^{(k)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

Proof. Define $I(\boldsymbol{\theta})$ as in the theorem statement. Then, by linearity of expectation, the first term in the ELBO (equation 2) is $\mathbb{E}_{\boldsymbol{\theta}\sim q_{\boldsymbol{\psi}}(\boldsymbol{\theta})}[I(\boldsymbol{\theta})]$. Now let $H(\boldsymbol{\psi}) \coloneqq \nabla_{\boldsymbol{\psi}}\mathbb{E}_{\boldsymbol{\theta}\sim q_{\boldsymbol{\psi}}(\boldsymbol{\theta})}[I(\boldsymbol{\theta})]$. Reparametrizing $\boldsymbol{\theta}$ as $g(\boldsymbol{\psi}, \boldsymbol{\epsilon})$ with $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon}) = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, we have

$$H(\boldsymbol{\psi}) = \nabla_{\boldsymbol{\psi}} \int q_{\boldsymbol{\psi}}(\boldsymbol{\theta}) I(\boldsymbol{\theta})\, d\boldsymbol{\theta} = \nabla_{\boldsymbol{\psi}} \int p(\boldsymbol{\epsilon}) I(g(\boldsymbol{\psi}, \boldsymbol{\epsilon}))\, d\boldsymbol{\epsilon}$$

This integral satisfies the conditions of the Leibniz integral rule [26], hence we can interchange the gradient and the integral:

$$H(\boldsymbol{\psi}) = \int p(\boldsymbol{\epsilon}) \nabla_{\boldsymbol{\psi}} I(g(\boldsymbol{\psi}, \boldsymbol{\epsilon}))\, d\boldsymbol{\epsilon} = \mathbb{E}_{\boldsymbol{\epsilon}\sim p(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\psi}} I(g(\boldsymbol{\psi}, \boldsymbol{\epsilon}))]$$

So the full gradient of the ELBO is

$$G(\boldsymbol{\psi}) \coloneqq \nabla_{\boldsymbol{\psi}}\mathcal{L}_{VCL}^t(\boldsymbol{\psi}) = \mathbb{E}_{\boldsymbol{\epsilon}\sim p(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\psi}} I(g(\boldsymbol{\psi}, \boldsymbol{\epsilon}))] - \nabla_{\boldsymbol{\psi}} \mathrm{KL}\left(q_{\boldsymbol{\psi}}(\boldsymbol{\theta}) \parallel q_{t-1}(\boldsymbol{\theta})\right)$$

The first term can be approximated in an unbiased way using Monte Carlo sampling and the second term can be computed directly. This yields the estimator given in the theorem statement. □

References

[1] Geoffrey I Webb, Roy Hyde, Hong Cao, Hai Long Nguyen, and Francois Petitjean. Characterizing concept drift. *Data Mining and Knowledge Discovery*, 30(4):964–994, 2016.

[2] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

[3] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

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

[5] Tamara Broderick, Nicholas Boyd, Andre Wibisono, Ashia C Wilson, and Michael I Jordan. Streaming variational bayes. *Advances in neural information processing systems*, 26, 2013.

[6] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

[7] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[8] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019.

[9] David Isele and Akansel Cosgun. Selective experience replay for lifelong learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[10] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

[11] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

[12] 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.

[13] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017.

[14] Alex J Smola, SVN Vishwanathan, and Eleazar Eskin. Laplace propagation. In *Advances in Neural Inf. Proc. Systems (NIPS)*, pages 441–448, 2004.

[15] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[16] Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A Rusu, Alexander Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017.

[17] Pietro Buzzega, Matteo Boschini, Angelo Porrello, and Simone Calderara. Rethinking experience replay: a bag of tricks for continual learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2180–2187. IEEE, 2021.

[18] Olivier Bachem, Mario Lucic, and Andreas Krause. Practical coreset constructions for machine learning. *arXiv preprint arXiv:1703.06476*, 2017.

[19] Sebastian Farquhar and Yarin Gal. Towards robust evaluations of continual learning. *arXiv preprint arXiv:1805.09733*, 2018.

[20] Sebastian Farquhar and Yarin Gal. A unifying bayesian view of continual learning. *arXiv preprint arXiv:1902.06494*, 2019.

[21] Kaichao You, Mingsheng Long, Jianmin Wang, and Michael I Jordan. How does learning rate decay help modern neural networks? *arXiv preprint arXiv:1908.01878*, 2019.

[22] Rong Ge, Sham M Kakade, Rahul Kidambi, and Praneeth Netrapalli. The step decay schedule: A near optimal, geometrically decaying learning rate procedure for least squares. *Advances in neural information processing systems*, 32, 2019.

[23] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. Understanding the role of training regimes in continual learning. *Advances in Neural Information Processing Systems*, 33:7308–7320, 2020.

[24] Cheng Zhang, Judith Bütepage, Hedvig Kjellström, and Stephan Mandt. Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):2008–2026, 2018.

[25] Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. *Advances in neural information processing systems*, 28, 2015.

[26] Harley Flanders. Differentiation under the integral sign. *The American Mathematical Monthly*, 80(6):615–627, 1973.