

Optimization and Generalization of Regularization-Based Continual Learning: a Loss Approximation Viewpoint

Dong Yin ^{*1}, Mehrdad Farajtabar ^{†1}, Ang Li ^{‡1}, Nir Levine ^{§1}, and Alex Mott ^{¶1}

¹DeepMind, Mountain View

February 10, 2021

Abstract

Neural networks have achieved remarkable success in many cognitive tasks. However, when they are trained sequentially on multiple tasks without access to old data, their performance on early tasks tend to drop significantly. This problem is often referred to as *catastrophic forgetting*, a key challenge in *continual learning* of neural networks. The regularization-based approach is one of the primary classes of methods to alleviate catastrophic forgetting. In this paper, we provide a novel viewpoint of regularization-based continual learning by formulating it as a second-order Taylor approximation of the loss function of each task. This viewpoint leads to a unified framework that can be instantiated to derive many existing algorithms such as Elastic Weight Consolidation and Kronecker factored Laplace approximation. Based on this viewpoint, we study the optimization aspects (*i.e.*, convergence) as well as generalization properties (*i.e.*, finite-sample guarantees) of regularization-based continual learning. Our theoretical results indicate the importance of accurate approximation of the Hessian matrix. The experimental results on several benchmarks provide an empirical validation of our theoretical findings.

1 Introduction

Neural networks are achieving human-level performance on many cognitive tasks including image classification (Krizhevsky et al., 2012) and speech recognition (Hinton et al., 2006). However, as opposed to humans, their acquired knowledge is comparably volatile and can be easily dismissed. Especially, the *catastrophic forgetting* phenomenon refers to the case when a neural network forgets the past tasks if it is not allowed to retrain or reiterate on them again (Goodfellow et al., 2013; McCloskey and Cohen, 1989).

Continual learning is a research direction that deals with the catastrophic forgetting problem. Recent works tried to tackle this issue from a variety of perspectives. Most approaches can be categorized into three classes, *i.e.*, the regularization-based (Kirkpatrick et al., 2017; Zenke et al., 2017; Ritter et al., 2018; Farajtabar et al., 2020), expansion-based (Rusu et al., 2016; Yoon et al., 2018), and replay-based methods (Lopez-Paz and Ranzato, 2017; Chaudhry et al., 2018).

Although various algorithms have been proposed, few efforts were spent on the theoretical foundations of continual learning. Take the regularization-based approach as an example; in these algorithms, we usually construct a quadratic regularization function which penalizes the updates to the weights that are important to prior tasks. In most works, the regularization functions are designed based on heuristics without a theoretically rigorous understanding of how they affect the optimization and generalization of the learning algorithm, and how they mitigate catastrophic forgetting. We believe a solid theoretical foundation can enable better design and development of efficient continual learning algorithms.

In this paper, we tackle the optimization and generalization aspects of regularization-based continual learning. We study a unified framework for these algorithms from the perspective of loss landscapes. More specifically, in our framework at the end of each task we approximate the loss function by estimating its second-order Taylor expansion.

^{*}dongyin@google.com

[†]farajtabar@google.com

[‡]anglili@google.com

[§]nirlevine@google.com

[¶]alexmott@google.com

The approximation is used as a surrogate added to the loss function of the future tasks. We show that many existing algorithms such as Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017) and Kronecker factored Laplace approximation (Ritter et al., 2018) are special cases of our framework.

More importantly, equipped with this loss function approximation viewpoint, we are able to conduct rigorous analysis of the regularization-based algorithms. In summary, we make the following theoretical contributions:

- To study the optimization and generalization properties of regularization-based continual learning, we provide a loss approximation framework that unifies a few important continual learning methods.
- We leverage the unified framework to prove a *sufficient and worst-case necessary* condition under which by conducting gradient descent on the approximate loss function, we can still minimize the actual loss function. We identify the accuracy of Hessian matrix approximation, the moving distance in parameter space between tasks, and high order derivatives as important factors in the optimization procedure of the algorithms.
- We further provide convergence analysis of regularization-based continual learning for both non-convex and convex loss functions.
- We also investigate the generalization property of the algorithms under the loss approximation framework and provide finite-sample guarantees.

In addition to the theoretical contributions, we also present experimental evidence on three common continual learning benchmarks to validate our theoretical findings using the loss approximation viewpoint.

Notation Throughout this paper, for any positive integer N , we define $[N] := \{1, 2, \dots, N\}$. We use $\|\cdot\|_2$ to denote the ℓ_2 norm of vectors and the operator norm of matrices.

2 Related Work

Continual learning algorithms Generally speaking, three classes of algorithms exist to overcome catastrophic forgetting: *regularization-based*, *replay-based*, and *expansion-based* methods.

The *regularization-based* approaches are the main focus of this paper. These methods usually impose limiting constraints on the weight updates of the neural network according to some relevance score for previous knowledge (Kirkpatrick et al., 2017; Zenke et al., 2017; Ritter et al., 2018; Mirzadeh et al., 2020a; Farajtabar et al., 2020). They can also be combined with other methods (Nguyen et al., 2017; Titsias et al., 2019). A large fraction of regularization-based algorithms construct a quadratic regularization function at the end of each task and combine it with the loss of future tasks (Kirkpatrick et al., 2017; Zenke et al., 2017; Ritter et al., 2018; Aljundi et al., 2018; Park et al., 2019). In this paper, we show that our loss function approximation viewpoint provides a general framework for understanding these methods.

The *expansion-based* methods allocate new neurons or layers or modules to accommodate new tasks while utilizing the shared representation learned from previous ones (Rusu et al., 2016; Xiao et al., 2014; Yoon et al., 2018; Li et al., 2019; Jerfel et al., 2019). The *replay-based* methods store previous data or, alternatively, train a generative model of them and replay samples from them interleaved with samples drawn from the current task (Shin et al., 2017; Kamra et al., 2017; Zhang et al., 2019; Rios and Itti, 2018; Lüders et al., 2016; Lopez-Paz and Ranzato, 2017).

Theoretical understanding for continual learning One prior work that uses second-order Taylor expansion to approximate the loss functions of previous tasks is the Efficient Lifelong Learning Algorithm (ELLA) (Ruvolo and Eaton, 2013). We emphasize that ELLA is actually a special case in our framework. ELLA aims to construct a latent dictionary such that the model parameters for each task can be represented as a sparse linear combination of the items in the dictionary. Using the terminology of neural networks, this is equivalent to a one hidden layer linear network (*i.e.*, the activation function is identity) with multi-head structure, where each head corresponds to each task. In contrast, our framework does not rely on specific neural network architectures and thus can be applied to more general problems. As for the theoretical guarantees, Ruvolo and Eaton (2013) only showed the convergence of the *approximate loss*, whereas in this paper, we focus on mitigating catastrophic forgetting, *i.e.*, the gap between the approximate loss and the actual loss and how this gap affects the optimization and generalization of the learning problem.

Another line of theoretical research focuses on regret analysis for continual learning (Alquier et al., 2017; Finn et al., 2019; Wu et al., 2019). These studies aim to analyze the behavior of the algorithm *during the course of training*. In comparison, we focus on the forgetting problem, *i.e.*, the performance of the *final model*. A recent work (Bennani and Sugiyama, 2020) studies the generalization property of the OGD algorithm (Farajtabar et al., 2020), which does not use the quadratic regularization loss. Another important but different topic in continual/multi-task learning is the generalization to unseen tasks (Baxter, 1998; Maurer et al., 2016; Pentina and Ben-David, 2015). A few other works try to obtain a better understanding of continual learning via empirical methods (Nguyen et al., 2019; Farquhar and Gal, 2019; Mirzadeh et al., 2020b).

3 Problem Formulation

In this section, we present the continual learning formulation studied in this paper. We consider a sequence of K supervised learning tasks \mathcal{T}_k , $k \in [K]$. For task \mathcal{T}_k , there is an unknown distribution \mathcal{D}_k over the space of feature-label pairs $\mathcal{X} \times \mathcal{Y}$. Let $\mathcal{W} \subseteq \mathbb{R}^d$ be a model parameter space, and for the k -th task, let $\ell_k(w; x, y) : \mathcal{W} \mapsto \mathbb{R}$ be the loss function of w associated with data point (x, y) . The population loss function of task \mathcal{T}_k is defined as

$$L_k(w) := \mathbb{E}_{(x,y) \sim \mathcal{D}_k} \ell_k(w; x, y).$$

Our general objective is to learn a parametric model with minimized population loss over all the K tasks. More specifically, in continual learning the learner follows the following protocol: When learning on the k -th task, the learner obtains access to n_k data points $(x_{k,i}, y_{k,i})$, $i \in [n_k]$ sampled i.i.d. according to \mathcal{D}_k and we define

$$\hat{L}_k(w) := \frac{1}{n_k} \sum_{i=1}^{n_k} \ell_k(w; x_{k,i}, y_{k,i})$$

as the empirical loss function; the learner then updates the model parameter w using these n_k training data, and after the training procedure is finished, the learner loses access to them, but can store some side information about the task. The goal is to avoid forgetting previous tasks when trained on new tasks by utilizing the side information.

4 Regularization-Based Approaches and Loss Approximation Viewpoint

As mentioned in Section 2, one important approach to solving continual learning problems is to use regularization. In these algorithms, at the end of each task, we construct a regularization function of the model parameters, usually quadratic, and when training future tasks, we minimize the regularized loss function. In this section, we derive a unified framework for regularization-based continual learning through the lens of loss function approximation, and demonstrate that many existing algorithms can be considered as variants of our framework.

To measure the effectiveness of a continual learning algorithm, we use a simple criterion that after each task, we want the average population loss over all the tasks to be small. This means that, for every $k \in [K]$, after training on \mathcal{T}_k , we want to find w to minimize $\frac{1}{k} \sum_{k'=1}^k L_{k'}(w)$. Since optimizing the loss function is the key to training a good model, we begin with a straightforward method for continual learning: storing the second-order Taylor expansion of the empirical loss function, and using it as an approximation of the loss function for an old task when training on new tasks. We start with a simple setting. Suppose that there are two tasks, and at the end of \mathcal{T}_1 , we obtain a model \hat{w}_1 . Then we compute the gradient and Hessian matrix of $\hat{L}_1(w)$ at \hat{w}_1 , and construct the second-order Taylor expansion of $\hat{L}_1(w)$ at \hat{w}_1 :

$$L_1^{\text{prox}}(w) = \hat{L}_1(\hat{w}_1) + (w - \hat{w}_1)^\top \nabla \hat{L}_1(\hat{w}_1) + \frac{1}{2} (w - \hat{w}_1)^\top \nabla^2 \hat{L}_1(\hat{w}_1) (w - \hat{w}_1).$$

When training on \mathcal{T}_2 , we try to minimize $\frac{1}{2}(L_1^{\text{prox}}(w) + \hat{L}_2(w))$. The basic idea of this design is that, we hope in a neighborhood around \hat{w}_1 , the quadratic function $L_1^{\text{prox}}(w)$ stays as a good approximation of $\hat{L}_1(w)$, and thus approximately still minimizes the average of the empirical loss functions $\frac{1}{2}(\hat{L}_1(w) + \hat{L}_2(w))$, which in the limit generalizes to the population loss function $\frac{1}{2}(L_1(w) + L_2(w))$.

Formally, let \hat{w}_k be the model that we obtain at the end of the k -th task. We define the approximation of the sum of the first k empirical loss functions as

$$L_k^{\text{prox}}(w) = \sum_{k'=1}^k \underbrace{\hat{L}_{k'}(\hat{w}_{k'})}_{(a)} + \underbrace{(w - \hat{w}_{k'})^\top \nabla \hat{L}_{k'}(\hat{w}_{k'})}_{(b)} + \underbrace{\frac{1}{2}(w - \hat{w}_{k'})^\top H_{k'}(w - \hat{w}_{k'})}_{(c)}, \quad (1)$$

where, $H_{k'}$ denotes the Hessian matrix $\nabla^2 \hat{L}_{k'}(\hat{w}_{k'})$ or its approximation. We construct $L_k^{\text{prox}}(w)$ at the end of task k , and when training on task $k+1$, we minimize $\frac{1}{k+1}(L_k^{\text{prox}}(w) + \hat{L}_{k+1}(w))$.

One important **assumption** that we make throughout this work is the validity of the second-order **Taylor expansion** as an **approximation to the loss functions**. While this might seem as a crude approximation in general, we argue this has merit for our setup. More specifically, we rely on the wealth of studies on overparameterized models where the loss tends to be very well behaved and almost convex in a reasonable neighborhood around the minima (Saxe et al., 2013; Choromanska et al., 2015; Goodfellow et al., 2014). Moreover, there is a similar observation in the recently proposed NTK regime (Jacot et al., 2018). In continual learning, similar assumptions are made by most approaches that rely on approximating the posterior on the weights by a Gaussian (Kirkpatrick et al., 2017) or a first-order approximation of the loss surface around the optimum (Farajtabar et al., 2020) or an explicit Taylor expansion over the previous minima (Mirzadeh et al., 2020b).

We now proceed to discuss how several existing algorithms such as EWC (Kirkpatrick et al., 2017), Kronecker approximation (Ritter et al., 2018), SI (Zenke et al., 2017), and MAS (Aljundi et al., 2018) can be considered as variants of our framework. First, the constant term (a) in Eq. (1) does not depend of w and thus can be ignored. Moreover, at the end of each task, the gradient term $\hat{L}_{k'}(\hat{w}_{k'})$ is usually small, so term (b) in Eq. (1) is usually skipped in practice. In the following, we show that several existing algorithms are basically using different approaches to approximate the Hessian matrix in term (c), and thus are special cases in our framework.

Elastic Weight Consolidation (EWC) The EWC algorithm uses a diagonal matrix whose diagonal elements are the Fisher information of the model weights to construct the regularization term. The intuition of EWC is that the weights with higher Fisher information are more important to current task, and thus should undergo less change in future tasks. As we know, if the loss function is the negative log-likelihood, and we obtained the ground truth probabilistic model, then the Fisher information matrix is equivalent to the Hessian matrix (see *e.g.*, Chapter 4.5 of Keener (2011)). Hence, the EWC algorithm can be considered as a method that uses the diagonal elements of the Hessian matrix to approximate the full Hessian in Eq. (1).

Kronecker factored Laplace approximation The basic idea of the Kronecker factored Laplace approximation algorithm by Ritter et al. (2018) is similar to that of EWC, with the key difference being that it uses Kronecker product approximation of the Fisher information matrix (Martens and Grosse, 2015; Grosse and Martens, 2016). In other words, this approximation takes the off-diagonal elements of the Hessian matrix into consideration, and thus usually performs better than EWC.

Finally, the SI and MAS algorithms construct the diagonal regularization function based on the concept of intelligent synapses. Although the basic idea is different from Hessian/Fisher matrix approximation, these algorithms can also be analyzed in our framework as the regularization function is quadratic. More specifically, our analysis mainly relies on an assumption that the difference between the approximate Hessian matrix (the diagonal regularization matrix in SI and MAS) and the ground truth Hessian matrix is bounded in operator norm (see Assumption 2 in Section 5), and our analysis can be applied to SI and MAS as long as this assumption is satisfied.

Equipped with the loss approximation viewpoint, we are ready to perform theoretical analysis of regularization-based algorithms in order to get deeper understanding of their behavior. We provide analyses from both optimization (Section 5) and generalization (Section 6) aspects.

5 Optimization Analysis

As we can see, the key idea in regularization-based algorithms is to approximate the loss functions of previous tasks using quadratic functions. This leads to the following theoretical question: *By running gradient descent algorithm on an approximate loss function, can we still minimize the actual loss that we are interested in?* Intuitively, the answer to this question depends on several factors:

- (i) Approximation error of the Hessian matrix. A more accurate approximation of the Hessian matrix leads to better approximation the loss function, and thus it becomes easier to minimize the actual loss given an approximation of it.
- (ii) The moving distance in the parameter space. We perform Taylor expansion at the end of each task. This quadratic approximation can become more and more inaccurate as the model parameters move away from the end-of-task points. Therefore, if we only need to move a small distance between tasks, *i.e.*, tasks are similar, then we may be able to achieve better performance.
- (iii) High order derivatives. According to the properties of Taylor expansion, the error between the Taylor approximation and the actual function depends on the high order derivatives. For quadratic approximation, when the third derivative of the loss function is small, it becomes easier to keep minimizing the actual loss.

In the following, we rigorously demonstrate these points. For the purpose of theoretical analysis, we make a few simplifications to our setup. Without loss of generality, we study the training process of the last task \mathcal{T}_K , and still use \hat{w}_k to denote the model parameters that we obtain at the end of the k -th task. We use the loss function approximation in Eq. (1), but in this section, for simplicity we ignore the finite-sample effect and replace the empirical loss function with the population loss function, *i.e.*, we define

$$\tilde{L}_{K-1}(w) = \sum_{k=1}^{K-1} L_k(\hat{w}_k) + (w - \hat{w}_k)^\top \nabla L_k(\hat{w}_k) + \frac{1}{2}(w - \hat{w}_k)^\top H_k(w - \hat{w}_k),$$

where H_k represents $\nabla^2 L_k(\hat{w}_k)$ or its approximation.¹ We discuss the finite sample (generalization) effect in the next section. As mentioned, during the training of the last task, we have access to the approximate loss function

$$\tilde{F}(w) := \frac{1}{K}(\tilde{L}_{K-1}(w) + L_K(w)),$$

whereas the actual loss function that we care about is

$$F(w) := \frac{1}{K} \sum_{k=1}^K L_k(w).$$

We also focus on gradient descent instead of its stochastic counterpart. In particular, let $w_0 := \hat{w}_{K-1}$ be the initial model parameter for the last task. We run the following update for $t = 1, 2, \dots, T$:

$$w_t = w_{t-1} - \eta \nabla \tilde{F}(w_{t-1}). \quad (2)$$

We use the following standard notions for differentiable function $f : \mathcal{W} \mapsto \mathbb{R}$.

Definition 1. f is μ -smooth if $\|\nabla f(w) - \nabla f(w')\|_2 \leq \mu \|w - w'\|_2, \forall w, w' \in \mathcal{W}$.

Definition 2. f is ρ -Hessian Lipschitz if $\|\nabla^2 f(w) - \nabla^2 f(w')\|_2 \leq \rho \|w - w'\|_2, \forall w, w' \in \mathcal{W}$.

We make the assumptions that the loss functions are smooth and Hessian Lipschitz. We note that the Hessian Lipschitz assumption is standard in analysis of non-convex optimization (Nesterov and Polyak, 2006; Jin et al., 2017), and that the Hessian Lipschitz constant is an upper bound for the third derivative.

Assumption 1. We assume that $L_k(w)$ is μ -smooth and ρ -Hessian Lipschitz $\forall k \in [K]$.

We also assume that the error between the matrices H_k and $\nabla^2 L_k(\hat{w}_k)$ is bounded.

Assumption 2. We assume that for every $k \in [K]$, $\|H_k\|_2 \leq \mu$, where μ is defined in Assumption 1, and that $\|H_k - \nabla^2 L_k(\hat{w}_k)\|_2 \leq \delta$ for some $\delta \geq 0$.

¹In other words, $\tilde{L}_{K-1}(w)$ is the *population* version of $L_{K-1}^{\text{prox}}(w)$ in Section 4.

5.1 Sufficient and Worst-Case Necessary Condition for One-Step Update

We begin with analyzing a single step during training. Our goal is to understand by running a single step of gradient descent on $\tilde{F}(w)$, whether we can minimize the actual loss function $F(w)$. More specifically, we have the following result.

Theorem 1. *Under Assumptions 1 and 2, and suppose that in the t -th iteration, we observe*

$$\|\nabla \tilde{F}(w_{t-1})\|_2 \geq \frac{c}{K} \sum_{k=1}^{K-1} \delta \|w_{t-1} - \hat{w}_k\|_2 + \rho \|w_{t-1} - \hat{w}_k\|_2^2, \quad (3)$$

for some $c > 1$, and the learning rate satisfies $\eta \leq \frac{2(1-1/c)}{\mu}$, then we have

$$F(w_t) \leq F(w_{t-1}) - \eta \left(1 - \frac{1}{c} - \frac{\mu\eta}{2}\right) \|\nabla \tilde{F}(w_{t-1})\|_2^2.$$

We prove Theorem 1 in Appendix A. Here, we emphasize that this result does not assume any convexity of the loss functions. The theorem provides a sufficient condition Eq. (3), under which by running gradient descent on \tilde{F} , we can still minimize the true loss function F . Intuitively, when the right hand side of Eq. (3) is smaller, this condition is more likely to be satisfied. This implies that when (i) the approximation error of the Hessian matrix, *i.e.*, δ is small, or (ii) the moving distance in the parameter space $\|w_{t-1} - \hat{w}_k\|_2$ is small or (iii) the third derivative ρ is small, condition Eq. (3) becomes easier to hold. Therefore, this theorem validates our hypotheses at the beginning of this section. In addition, we note that Theorem 1 also implies that as training goes on and $\|\nabla \tilde{F}(w_t)\|_2$ decreasing, it is beneficial to decrease the learning rate η , since when c decreases, the upper bound on η that guarantees the decay of F (*i.e.*, $2(1 - 1/c)/\mu$) also decreases. We notice that the importance of learning rate decay for continual learning has been observed in some empirical study recently (Mirzadeh et al., 2020a).

In Proposition 1 below, we will see that this condition is also necessary in the worst-case scenario, at least for the case where $K = 2$. More specifically, we can construct cases in which (3) is violated and the gradients of $F(w)$ and $\tilde{F}(w)$ have opposite directions.

Proposition 1. *Suppose that $K = 2$, $d = 1$, $\mathcal{W} = [0, 1]$. Then, there exists \hat{w}_1 , $L_1(w)$, $\tilde{L}_1(w)$, and $L_2(w)$ such that if $|\tilde{F}'(w)| < \frac{1}{2}[\delta|w - \hat{w}_1| + \rho(w - \hat{w}_1)^2]$, then $\tilde{F}'(w) \cdot F'(w) < 0$.*

We prove Proposition 1 in Appendix B.

5.2 Convergence Analysis

Although the condition in Eq. (3) provides us with insights on the dynamics of the training algorithm, it is usually hard to check this condition in every step, since we may not have good estimates of δ and ρ . A practical implementation is to choose a constant learning rate along with an appropriate number of training steps. In this section, we provide bounds on the convergence behavior of our algorithm with a constant learning rate and T iterations, both for non-convex and convex loss functions. These results imply that sometimes early stopping may be helpful, especially when δ or ρ is large or the tasks are more distinct from each other, *i.e.*, the moving distance between tasks is large. We begin with a convergence analysis for non-convex loss functions in Theorem 2, in which we use the common choice of learning rate $1/\mu$ for gradient descent on smooth functions (Bubeck, 2014).

Theorem 2 (non-convex). *Let $F_0 := F(w_0)$, $F^* := \min_{w \in \mathcal{W}} F(w)$, $\tilde{F}_0 := \tilde{F}(w_0)$, and $\tilde{F}^* := \min_{w \in \mathcal{W}} \tilde{F}(w)$. Then, under Assumptions 1 and 2, after running T iterations of the gradient descent update (2) with learning rate $\eta = 1/\mu$, we have*

$$\frac{1}{T} \sum_{t=1}^T \|\nabla F(w_{t-1})\|_2 \leq \frac{\alpha}{\sqrt{T}} + \beta + \gamma_1 \sqrt{T} + \gamma_2 T,$$

where $\alpha = \sqrt{2\mu(F_0 - F^*)}$, $\beta = \frac{\sqrt{3}}{K} \sum_{k=1}^{K-2} \delta \|w_0 - \hat{w}_k\|_2 + 2\rho \|w_0 - \hat{w}_k\|_2^2$, $\gamma_1 = \delta \sqrt{\frac{3}{\mu}(\tilde{F}_0 - \tilde{F}^*)}$, and $\gamma_2 = \frac{4\rho}{\mu}(\tilde{F}_0 - \tilde{F}^*)$.

We prove Theorem 2 in Appendix C. Unlike standard optimization analysis, the average norm of the gradients does not always decrease as T increases, when $\delta \neq 0$ or $\rho \neq 0$. As we can see, the coefficients γ_1 and γ_2 in the last two terms that “hurt” convergence increase as δ and ρ increase, respectively.

When the loss functions are convex, we can prove a better guarantee which does not have the $\mathcal{O}(\sqrt{T})$ and $\mathcal{O}(T)$ terms as in Theorem 2. More specifically, we have the following assumption and theorem.

Assumption 3. $L_k(w)$ is convex and $H_k \succeq 0, \forall k$.

Theorem 3 (convex). Suppose that Assumptions 1, 2, and 3 hold, and define $F^* = \min_{w \in \mathcal{W}} F(w)$, $w^* \in \arg \min_{w \in \mathcal{W}} F(w)$, $\tilde{w}^* \in \arg \min_{w \in \mathcal{W}} \tilde{F}(w)$, and $\tilde{D} := \|w_0 - \tilde{w}^*\|_2$. After running T iterations of the gradient descent update (2) with learning rate $\eta = 1/\mu$, we have

$$F(w_T) - F^* \leq \frac{\alpha}{T} + \beta,$$

where $\alpha = 2\mu\tilde{D}^2$, and $\beta = \frac{1}{2K} \sum_{k=1}^{K-1} \delta(\|w^* - \hat{w}_k\|_2^2 + 2\tilde{D}^2 + 2\|\tilde{w}^* - \hat{w}_k\|_2^2) + \rho(\|w^* - \hat{w}_k\|_2^3 + 4\tilde{D}^3 + 4\|\tilde{w}^* - \hat{w}_k\|_2^3)$.

We prove Theorem 3 in Appendix D. As we can see, if $\delta \neq 0$ or $\rho \neq 0$, we still cannot guarantee the convergence to the true minimum of F , due to the inexactness of \tilde{F} . On the other hand, if the loss functions are quadratic and we save the full Hessian matrices, i.e., $\delta = \rho = 0$, as we have full information about previous loss functions, we can recover the standard $\mathcal{O}(1/T)$ convergence rate for gradient descent on convex and smooth functions.

6 Generalization Analysis

While the previous section provides with important insights on the convergence behavior of the algorithm, one aspect that is less studied is the finite-sample analysis of continual learning. More specifically, what we eventually care about is the average *population* loss across all the tasks, i.e., $F(w) = \frac{1}{K} \sum_{k=1}^K L_k(w)$, but what we have at the end of the last task is the *empirical* loss of the last task, i.e., $\hat{L}_K(w) := \frac{1}{n_K} \sum_{i=1}^{n_K} \ell_K(w; x_{K,i}, y_{K,i})$, as well as the *empirical quadratic approximation*:

$$L_{K-1}^{\text{prox}}(w) = \sum_{k=1}^{K-1} \hat{L}_k(\hat{w}_k) + (w - \hat{w}_k)^\top \nabla \hat{L}_k(\hat{w}_k) + \frac{1}{2} (w - \hat{w}_k)^\top H_k (w - \hat{w}_k).$$

Thus, our generalization analysis aims to bound the gap between $F(w)$ and $\frac{1}{K}(\hat{L}_K(w) + L_{K-1}^{\text{prox}}(w))$. Since our focus in this section is the finite-sample effect, we will not consider the approximation error of the empirical Hessian matrix, and instead, we assume that H_k here is the exact empirical Hessian matrix. The extension of our results to an approximate empirical Hessian matrix is straightforward.

Assumption 4. In Eq. (4), $H_k = \nabla^2 \hat{L}_k(\hat{w}_k) = \frac{1}{n_k} \sum_{i=1}^{n_k} \nabla^2 \ell(\hat{w}_k; x_{k,i}, y_{k,i})$.

In addition, we make the standard boundedness assumptions that are commonly used in uniform convergence analysis (Mohri et al., 2018; Foster et al., 2018).

Assumption 5. For any $k \in [K]$, $w \in \mathcal{W}$ and $(x, y) \in \mathcal{X} \times \mathcal{Y}$, $|\ell_k(w; x, y)| \leq b$, $\|\nabla \ell_k(w; x, y)\|_2 \leq g$, and $\|\nabla^2 \ell_k(w; x, y)\|_2 \leq h$.

With these assumptions, we are ready to state our generalization bound.

Theorem 4. Under Assumptions 4 and 5, with probability at least $1 - \delta$ over the random training examples $(x_{k,i}, y_{k,i})$, $i \in [n_k]$, $k \in [K]$, we have

$$F(w) \leq \underbrace{\frac{1}{K}(\hat{L}_K(w) + L_{K-1}^{\text{prox}}(w))}_{\text{regularized training loss}} + \underbrace{\frac{2}{K} \left(bR_K^{(\ell)} + \sum_{k=1}^{K-1} (bR_k^{(\ell)} + gR_k^{(g)} + hR_k^{(h)}) \right)}_{\text{finite-sample effect}} + \underbrace{\mathcal{O}\left(\frac{\log(\frac{K}{\delta})}{\sqrt{n}}\right) + \frac{\rho}{2K} \sum_{k=1}^{K-1} \|w - \hat{w}_k\|_2^3}_{\text{loss approximation error}},$$

where

$$\begin{aligned} R_k^{(\ell)} &= \mathbb{E}_\sigma \left[\frac{1}{n_k} \sup_w \sum_{i=1}^{n_k} \sigma_i \ell_k(w; x_{k,i}, y_{k,i}) \right], \\ R_k^{(g)} &= \mathbb{E}_\sigma \left[\frac{1}{n_k} \sup_w \left\| \sum_{i=1}^{n_k} \sigma_i \nabla \ell_k(w; x_{k,i}, y_{k,i}) \right\|_2 \right], \\ R_k^{(h)} &= \mathbb{E}_\sigma \left[\frac{1}{n_k} \sup_w \left\| \sum_{i=1}^{n_k} \sigma_i \nabla^2 \ell_k(w; x_{k,i}, y_{k,i}) \right\|_2 \right] \end{aligned}$$

are the (normed) Rademacher complexities for the value, gradient and Hessian matrix of the loss functions (σ denotes a sequence of i.i.d. Rademacher random variables); and $n = \min_{k \in [K]} n_k$.

We prove Theorem 4 in Appendix E. As we can see, the gap between $F(w)$ and the regularized training loss consists of two parts. The first part is related to the Rademacher complexity terms, which represent the finite-sample effect, and can approach zero as the number of training data for each task increases. The second part is the loss function approximation error, which does not decay when the number of training examples increases. This again demonstrates that the accuracy of the loss function approximation plays a key role in regularization-based continual learning.

7 Experiments

In this section, we present experimental results validating our theoretical findings. More specifically, our experiments mainly validate two claims: (i) an accurate approximation of the Hessian matrix leads to better overall performance and (ii) training a large number of steps may not always be helpful, and sometimes early stopping is needed. All the experiments are implemented with JAX (Bradbury et al., 2018).

7.1 Experiment Setup

Datasets. We use three standard continual learning benchmarks created based on MNIST (LeCun et al., 1998) and CIFAR-100 (Krizhevsky and Hinton, 2009) datasets, *i.e.*, Permuted MNIST (Goodfellow et al., 2013), Rotated MNIST (Lopez-Paz and Ranzato, 2017), and Split CIFAR (Chaudhry et al., 2018). In Permuted MNIST, for each task, we choose a random permutation of the pixels of MNIST images, and reorder all the images according to the permutation. We use 20-task Permuted MNIST in the experiments. In Rotated MNIST, for each task, we rotate the MNIST images by a particular angle. In our experiments, we choose a 20-task Rotated MNIST, with the rotation angles being 0, 10, 20, ..., 190 degrees. For Split CIFAR, we split the 100 labels of the CIFAR-100 dataset to 20 disjoint subsets, each subset corresponding to one coarse label.² For each task, we use the CIFAR-100 data whose labels belong to a particular subset.

Architecture. For Permuted MNIST and Rotated MNIST, we use a multilayer perceptron (MLP) with two hidden layers, each having 256 units. For Split CIFAR, we use a convolutional neural network (CNN) with two convolutional layers with 3×3 kernel and 64 and 128 output channels, respectively. We add max pooling with stride 2 after each convolutional layer. We add one hidden fully connected layer with 512 units. Moreover, for this model, we use a multi-head structure similar to what has been used by Chaudhry et al. (2018); Farajtabar et al. (2020): instead of having 100 logits in the output layer, we use separate heads for different tasks, and each head corresponds to the classes of the associated task. During training, for each task, we only optimize the cross-entropy loss over the logits and labels of the corresponding output head. For both MLP and CNN models, we use ReLU activation function.

Algorithms. We implement three regularization-based algorithms, namely EWC, Synaptic Intelligence (SI), and Kronecker factored Laplace approximation (Kronecker for short), which can be analyzed in our framework. We omit the constant and gradient terms ((a) and (b) in Eq. (1)), and following Kirkpatrick et al. (2017); Zenke et al. (2017); Ritter et al. (2018), we add an extra coefficient λ in the quadratic regularization function as a hyper parameter, *i.e.*, we minimize $\frac{1}{k+1}(\hat{L}_{k+1}(w) + \lambda L_k^{\text{prox}}(w))$. In our experiments, we tune λ and report the results with the choice of λ that produces the best average test accuracy over all tasks. The hyper parameter choice for all the results are provided in Appendix F. Following prior works (Kirkpatrick et al., 2017; Chaudhry et al., 2018; Farajtabar et al., 2020), we choose a learning rate of 10^{-3} and a batch size of 10. For all the results that we report, we present the average result over 5 independent runs, as well as the standard deviation.

7.2 Results

We evaluate the EWC, SI and Kronecker algorithms on the three benchmarks. As mentioned, the Kronecker approximation algorithm makes use of the off-diagonal elements of the Hessian matrix, and thus provides a better approximation compared to EWC and SI which use diagonal quadratic regularization. We note that Ritter et al. (2018) have already observed that the Kronecker algorithm outperforms EWC and SI, and presented results on Permuted MNIST. Here, we include Permuted MNIST for completeness and also provide additional results on Rotated MNIST and Split CIFAR. The average test accuracy over the 20 tasks are presented in Tables 1, 2, and 3

²CIFAR-100 contains 20 coarse labels, and each of them contains 5 fine-grained labels.

for the three benchmarks, respectively. In these tables, we also provide the results for the *vanilla SGD* algorithm, which sequentially trains over all tasks without storing any side information. In the caption of each table, we also provide the average test accuracy for the *multi-task* algorithm, which has access to all the data from all the tasks simultaneously. In all of these tables, we present the results as a function of the number of epochs that we train for each task. As we can see, Kronecker algorithm consistently performs the best. These results confirm our theoretical finding that an accurate Hessian approximation is important for regularization-based algorithms.

Table 1: Average test accuracy on Permuted MNIST. Test accuracy for multi-task algorithm is 96.8 ± 0.0 .

epoch	Average accuracy \pm Std. (%)			
	Vanilla	EWC	SI	Kronecker
1	59.1 ± 0.9	62.8 ± 0.7	62.2 ± 0.9	82.9 ± 0.2
2	56.8 ± 0.9	59.8 ± 0.4	63.7 ± 0.7	86.6 ± 0.2
4	57.1 ± 0.6	57.4 ± 0.7	69.4 ± 0.3	90.1 ± 0.1
8	55.5 ± 1.1	56.1 ± 0.7	76.5 ± 0.5	93.0 ± 0.1
16	54.8 ± 1.4	56.3 ± 1.3	80.7 ± 0.5	94.8 ± 0.1
32	52.6 ± 1.6	56.9 ± 1.9	78.2 ± 1.2	96.0 ± 0.0

Table 2: Average test accuracy on Rotated MNIST. Test accuracy for multi-task algorithm is 97.2 ± 0.1 .

epoch	Average accuracy \pm Std. (%)			
	Vanilla	EWC	SI	Kronecker
1	41.9 ± 0.6	47.7 ± 0.5	42.1 ± 0.3	69.1 ± 0.5
2	43.5 ± 0.6	50.2 ± 0.4	44.3 ± 0.3	70.0 ± 0.3
4	44.6 ± 0.3	53.9 ± 0.9	47.4 ± 0.3	71.9 ± 0.7
8	45.7 ± 0.5	57.9 ± 1.4	50.5 ± 0.3	75.5 ± 0.8
16	46.2 ± 0.5	60.9 ± 1.4	54.8 ± 0.6	78.8 ± 0.8
32	46.6 ± 0.5	62.4 ± 2.5	58.3 ± 0.8	81.9 ± 0.9

Table 3: Average test accuracy on Split CIFAR. Test accuracy for multi-task algorithm is 62.6 ± 0.2 .

epoch	Average accuracy \pm Std. (%)			
	Vanilla	EWC	SI	Kronecker
1	29.6 ± 0.4	29.4 ± 1.3	29.9 ± 0.9	31.6 ± 0.8
2	32.6 ± 1.1	32.6 ± 0.9	32.8 ± 0.7	37.2 ± 0.6
4	35.3 ± 0.7	34.9 ± 0.8	35.1 ± 1.0	42.0 ± 0.7
8	39.4 ± 1.6	38.4 ± 1.0	38.7 ± 1.3	47.3 ± 0.8
16	43.2 ± 1.4	41.3 ± 1.6	43.2 ± 0.5	54.2 ± 0.7
32	<i>38.0 ± 0.6</i>	44.0 ± 0.9	45.3 ± 0.7	59.9 ± 0.7
64	<i>32.4 ± 0.2</i>	<i>40.9 ± 0.7</i>	<i>45.1 ± 1.1</i>	59.7 ± 0.6
128	<i>30.9 ± 1.6</i>	<i>37.0 ± 0.9</i>	<i>43.8 ± 0.8</i>	53.1 ± 1.1

Another interesting observation is that on Permuted MNIST, for the vanilla SGD and EWC algorithms, the average test accuracy *decreases* as we *increase* the number of epochs to train for each task. Similar phenomenon also happens for vanilla SGD, EWC and SI algorithms on Split CIFAR, when we increase the number of epochs per task beyond 32 (see the numbers in *italic* font in Table 3). This validates our theoretical results that training longer on each individual task may not necessarily be helpful for achieving a good final average accuracy, especially when the approximation error for the Hessian matrix is large. In fact, the behavior of vanilla SGD, EWC, and SI on Split CIFAR exactly matches the prediction from Theorem 2: When we increase the number of epochs per task, the average test accuracy first improves (since the model is trained better for each task), and then becomes worse as the model forgets early tasks.

As a final remark, we note that the strong performance of the Kronecker algorithm comes with a large computation cost. We observe that in our experiments on Permuted/Rotated MNIST, the computation time of Kronecker algorithm is 5 times as much as that of the EWC algorithm, and on Split CIFAR, this ratio is 10. Therefore, for practical applications, we may need to take the trade-off between the accuracy of Hessian approximation and the computational cost into consideration.

8 Conclusions

We studied a unified framework for regularization-based continual learning through the lens of loss function approximation. We showed that existing algorithms such as EWC and Kronecker approximation are special cases of our framework. We further established theoretical guarantees for the algorithms on both optimization and generalization aspects. To the best of our knowledge, our work is among the first ones that try to formulate continual learning from a loss function approximation perspective that enables studying its convergence behavior and finite-sample guarantees. Finally, we validated the theoretical findings with experiments on several continual learning benchmarks.

Acknowledgements

We would like to thank Dilan Gorur, Clara Huiyi Hu, Nevena Lazic, and Michalis Titsias for helpful discussions.

References

- R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–154, 2018.
- P. Alquier, M. Pontil, et al. Regret bounds for lifelong learning. In *Artificial Intelligence and Statistics*, pages 261–269, 2017.
- J. Baxter. Theoretical models of learning to learn. In *Learning to learn*, pages 71–94. Springer, 1998.
- M. A. Bennani and M. Sugiyama. Generalisation guarantees for continual learning with orthogonal gradient descent. *arXiv preprint arXiv:2006.11942*, 2020.
- J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, and S. Wanderman-Milne. JAX: composable transformations of Python+NumPy programs, 2018. URL <http://github.com/google/jax>.
- S. Bubeck. Convex optimization: Algorithms and complexity. *arXiv preprint arXiv:1405.4980*, 2014.
- A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny. Efficient lifelong learning with A-GEM. *arXiv preprint arXiv:1812.00420*, 2018.
- A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun. The loss surfaces of multilayer networks. In *Artificial intelligence and statistics*, pages 192–204, 2015.
- M. Farajtabar, N. Azizan, A. Mott, and A. Li. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, 2020.
- S. Farquhar and Y. Gal. A unifying Bayesian view of continual learning. *arXiv preprint arXiv:1902.06494*, 2019.
- C. Finn, A. Rajeswaran, S. Kakade, and S. Levine. Online meta-learning. *arXiv preprint arXiv:1902.08438*, 2019.
- D. J. Foster, A. Sekhari, and K. Sridharan. Uniform convergence of gradients for non-convex learning and optimization. In *Advances in Neural Information Processing Systems*, pages 8745–8756, 2018.
- I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.
- I. J. Goodfellow, O. Vinyals, and A. M. Saxe. Qualitatively characterizing neural network optimization problems. *arXiv preprint arXiv:1412.6544*, 2014.

- R. Grosse and J. Martens. A Kronecker-factored approximate fisher matrix for convolution layers. In *International Conference on Machine Learning*, pages 573–582, 2016.
- G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.
- A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.
- G. Jerfel, E. Grant, T. L. Griffiths, and K. A. Heller. Reconciling meta-learning and continual learning with online mixtures of tasks. In *Advances in Neural Information Processing Systems*, 2019.
- C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan. How to escape saddle points efficiently. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1724–1732. JMLR. org, 2017.
- N. Kamra, U. Gupta, and Y. Liu. Deep generative dual memory network for continual learning. *arXiv preprint arXiv:1710.10368*, 2017.
- R. W. Keener. *Theoretical statistics: Topics for a core course*. Springer, 2011.
- J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.
- A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Tech Report*, 2009.
- A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.
- Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- X. Li, Y. Zhou, T. Wu, R. Socher, and C. Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. *arXiv preprint arXiv:1904.00310*, 2019.
- D. Lopez-Paz and M. Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476, 2017.
- B. Lüders, M. Schläger, and S. Risi. Continual learning through evolvable neural turing machines. In *NIPS 2016 Workshop on Continual Learning and Deep Networks (CLDL 2016)*, 2016.
- J. Martens and R. Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *International conference on machine learning*, pages 2408–2417, 2015.
- A. Maurer, M. Pontil, and B. Romera-Paredes. The benefit of multitask representation learning. *The Journal of Machine Learning Research*, 17(1):2853–2884, 2016.
- M. McCloskey and N. J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.
- S.-I. Mirzadeh, M. Farajtabar, and H. Ghasemzadeh. Dropout as an implicit gating mechanism for continual learning. *arXiv preprint arXiv:2004.11545*, 2020a.
- S. I. Mirzadeh, M. Farajtabar, R. Pascanu, and H. Ghasemzadeh. Understanding the role of training regimes in continual learning. *arXiv preprint arXiv:2006.06958*, 2020b.
- M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT press, 2018.
- Y. Nesterov and B. T. Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.
- C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner. Variational continual learning. *arXiv preprint arXiv:1710.10628*, 2017.

- C. V. Nguyen, A. Achille, M. Lam, T. Hassner, V. Mahadevan, and S. Soatto. Toward understanding catastrophic forgetting in continual learning. *arXiv preprint arXiv:1908.01091*, 2019.
- D. Park, S. Hong, B. Han, and K. M. Lee. Continual learning by asymmetric loss approximation with single-side overestimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3335–3344, 2019.
- A. Pentina and S. Ben-David. Multi-task and lifelong learning of kernels. In *International Conference on Algorithmic Learning Theory*, pages 194–208. Springer, 2015.
- A. Rios and L. Itti. Closed-loop GAN for continual learning. *arXiv preprint arXiv:1811.01146*, 2018.
- H. Ritter, A. Botev, and D. Barber. Online structured Laplace approximations for overcoming catastrophic forgetting. In *Advances in Neural Information Processing Systems*, pages 3738–3748, 2018.
- A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.
- P. Ruvolo and E. Eaton. ELLA: An efficient lifelong learning algorithm. In *International Conference on Machine Learning*, pages 507–515, 2013.
- A. M. Saxe, J. L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.
- H. Shin, J. K. Lee, J. Kim, and J. Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, pages 2990–2999, 2017.
- M. K. Titsias, J. Schwarz, A. G. d. G. Matthews, R. Pascanu, and Y. W. Teh. Functional regularisation for continual learning using Gaussian Processes. *arXiv preprint arXiv:1901.11356*, 2019.
- Y.-S. Wu, P.-A. Wang, and C.-J. Lu. Lifelong optimization with low regret. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 448–456, 2019.
- T. Xiao, J. Zhang, K. Yang, Y. Peng, and Z. Zhang. Error-driven incremental learning in deep convolutional neural network for large-scale image classification. In *Proceedings of the 22nd ACM International Conference on Multimedia*, pages 177–186. ACM, 2014.
- J. Yoon, E. Yang, J. Lee, and S. J. Hwang. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*. ICLR, 2018.
- F. Zenke, B. Poole, and S. Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3987–3995. JMLR, 2017.
- M. Zhang, T. Wang, J. H. Lim, and J. Feng. Prototype reminding for continual learning. *arXiv preprint arXiv:1905.09447*, 2019.

Appendix

A Proof of Theorem 1

We first provide a bound for the difference between the gradients of $\nabla \tilde{F}(w)$ and $\nabla F(w)$.

Lemma 1. *Let $\Delta(w) = \nabla \tilde{F}(w) - \nabla F(w)$. Then we have*

$$\|\Delta(w)\|_2 \leq \frac{1}{K} \sum_{k=1}^{K-1} \delta \|w - \hat{w}_k\|_2 + \rho \|w - \hat{w}_k\|_2^2.$$

We prove Lemma 1 in Appendix A.1. Since the loss functions for all the tasks $L_k(w)$ are μ -smooth, we know that $F(w)$ is also μ -smooth. Then we have

$$\begin{aligned} F(w_t) &\leq F(w_{t-1}) + \langle \nabla F(w_{t-1}), w_t - w_{t-1} \rangle + \frac{\mu}{2} \|w_t - w_{t-1}\|_2^2 \\ &= F(w_{t-1}) + \langle \nabla \tilde{F}(w_{t-1}) - \Delta(w_{t-1}), -\eta \nabla \tilde{F}(w_{t-1}) \rangle + \frac{\mu\eta^2}{2} \|\nabla \tilde{F}(w_{t-1})\|_2^2 \\ &\leq F(w_{t-1}) - \eta(1 - \frac{\mu\eta}{2}) \|\nabla \tilde{F}(w_{t-1})\|_2^2 + \eta \|\nabla \tilde{F}(w_{t-1})\|_2 \|\Delta(w_{t-1})\|_2. \end{aligned}$$

Therefore, as long as $\|\nabla \tilde{F}(w_{t-1})\|_2 \geq c \|\Delta(w_{t-1})\|_2$ for some $c > 1$, we have

$$F(w_t) \leq F(w_{t-1}) - \eta(1 - \frac{1}{c} - \frac{\mu\eta}{2}) \|\nabla \tilde{F}(w_{t-1})\|_2^2. \quad (4)$$

Then we can complete the proof by combining (4) with Lemma 1.

A.1 Proof of Lemma 1

By the definition of $\tilde{F}(w)$, for some $\xi_k \in [0, 1]$, $k \in [K-1]$, we have

$$\begin{aligned} \Delta(w) &= \frac{1}{K} \sum_{k=1}^{K-1} \nabla L_k(\hat{w}_k) + H_k(w - \hat{w}_k) - \nabla L_k(w) \\ &= \frac{1}{K} \sum_{k=1}^{K-1} (H_k - \nabla^2 L_k(\hat{w}_k + \xi_k(w - \hat{w}_k))) (w - \hat{w}_k), \end{aligned}$$

where the second equality is due to Lagrange's mean value theorem. Then, according to Assumptions 1 and 2, we have

$$\begin{aligned} \|H_k - \nabla^2 L_k(\hat{w}_k + \xi_k(w - \hat{w}_k))\|_2 &\leq \delta + \|\nabla^2 L_k(\hat{w}_k) - \nabla^2 L_k(\hat{w}_k + \xi_k(w - \hat{w}_k))\|_2 \\ &\leq \delta + \rho \|w - \hat{w}_k\|_2. \end{aligned} \quad (5)$$

Then, according to triangle inequality, we obtain

$$\|\Delta(w)\|_2 \leq \frac{1}{K} \sum_{k=1}^{K-1} \delta \|w - \hat{w}_k\|_2 + \rho \|w - \hat{w}_k\|_2^2.$$

B Proof of Proposition 1

We first note that it suffices to construct $F(w)$ and $\tilde{F}(w)$, as one can always choose $L_2(w) \equiv 0$ and then the construction of $F(w)$ and $\tilde{F}(w)$ is equivalent to that of $L_1(w)$ and $\tilde{L}_1(w)$. Let $\hat{w}_1 = 0$,

$$\begin{aligned} F(w) &= (w-1)^2 + \frac{\rho}{6} w^3, w \in [0, 1], \\ \tilde{F}(w) &= (w-1)^2 - \frac{\delta}{4} w^2, w \in [0, 1]. \end{aligned}$$

One can easily check that $F(\hat{w}_1) = \tilde{F}(\hat{w}_1)$, $F'(\hat{w}_1) = \tilde{F}'(\hat{w}_1)$, and $|F''(\hat{w}_1) - \tilde{F}''(\hat{w}_1)| = \frac{\delta}{2}$. In addition, since the second derivative of $F(w)$ is always bounded in $[0, 1]$, we know that $F(w)$ is smooth. Since $F'''(w) \equiv \rho$, we know that $F(w)$ is ρ -Hessian Lipschitz. Therefore, $F(w)$ and $\tilde{F}(w)$ satisfy all of our assumptions.

Since $\tilde{F}'(w) = 2(w - 1) - \frac{\delta}{2}w$, we know that $\tilde{F}'(w) < 0$, $\forall w \in [0, 1]$, and then

$$|\tilde{F}'(w)| < \frac{\delta}{2}w + \frac{\rho}{2}w^2$$

is equivalent to $\frac{\delta}{2}w - 2(w - 1) < \frac{\delta}{2}w + \frac{\rho}{2}w^2$, which implies that $F(w) = 2(w - 1) + \frac{\rho}{2}w^2 > 0$.

C Proof of Theorem 2

Similar to Appendix A, we define $\Delta(w) = \nabla \tilde{F}(w) - \nabla F(w)$. According to Assumptions 1 and 2, we know that both $F(w)$ and \tilde{F} are μ -smooth. By the smoothness of $F(w)$ and using the fact that $\eta = 1/\mu$, we get

$$\begin{aligned} F(w_t) &\leq F(w_{t-1}) + \langle \nabla F(w_{t-1}), w_t - w_{t-1} \rangle + \frac{\mu}{2} \|w_t - w_{t-1}\|_2^2 \\ &= F(w_{t-1}) - \langle \nabla F(w_{t-1}), \eta(\nabla F(w_{t-1}) + \Delta(w_{t-1})) \rangle + \frac{\mu\eta^2}{2} \|\nabla F(w_{t-1}) + \Delta(w_{t-1})\|_2^2 \\ &= F(w_{t-1}) - \frac{1}{2\mu} \|\nabla F(w_{t-1})\|_2^2 + \frac{1}{2\mu} \|\Delta(w_{t-1})\|_2^2, \end{aligned}$$

which implies

$$\|\nabla F(w_{t-1})\|_2^2 \leq 2\mu(F(w_{t-1}) - F(w_t)) + \|\Delta(w_{t-1})\|_2^2. \quad (6)$$

By averaging (6) over $t = 1, \dots, T$, we get

$$\frac{1}{T} \sum_{t=1}^T \|\nabla F(w_{t-1})\|_2^2 \leq \frac{2\mu(F_0 - F^*)}{T} + \frac{1}{T} \sum_{t=1}^T \|\Delta(w_{t-1})\|_2^2.$$

By taking square root on both sides, and using Cauchy-Schwarz inequality as well as the fact that $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$, we get

$$\frac{1}{T} \sum_{t=1}^T \|\nabla F(w_{t-1})\|_2 \leq \frac{\sqrt{2\mu(F_0 - F^*)}}{\sqrt{T}} + \sqrt{\frac{1}{T} \sum_{t=1}^T \|\Delta(w_{t-1})\|_2^2}. \quad (7)$$

We then proceed to bound $\|\Delta(w_{t-1})\|_2^2$. According to Lemma 1, we have

$$\begin{aligned} \|\Delta(w_{t-1})\|_2 &\leq \frac{1}{K} \sum_{k=1}^{K-1} \delta(\|w_{t-1} - w_0\|_2 + \|w_0 - \hat{w}_k\|_2) + 2\rho(\|w_{t-1} - w_0\|_2^2 + \|w_0 - \hat{w}_k\|_2^2) \\ &:= C + \delta\|w_{t-1} - w_0\|_2 + 2\rho\|w_{t-1} - w_0\|_2^2, \end{aligned}$$

where $C := \frac{1}{K} \sum_{k=1}^{K-1} \delta\|w_0 - \hat{w}_k\|_2 + 2\rho\|w_0 - \hat{w}_k\|_2^2$ does not depend on the iteration count t . By Cauchy-Schwarz inequality, we get

$$\|\Delta(w_{t-1})\|_2^2 \leq 3C^2 + 3\delta^2\|w_{t-1} - w_0\|_2^2 + 12\rho^2\|w_{t-1} - w_0\|_2^4. \quad (8)$$

Then we bound $\|w_{t-1} - w_0\|_2^2$. By triangle inequality and Cauchy-Schwarz inequality, we obtain

$$\|w_{t-1} - w_0\|_2 = \left\| -\eta \sum_{\tau=0}^{t-2} \nabla \tilde{F}(w_\tau) \right\|_2 \leq \eta \sum_{\tau=0}^{t-2} \|\nabla \tilde{F}(w_\tau)\|_2 \leq \eta \sqrt{(t-1) \sum_{\tau=0}^{t-2} \|\nabla \tilde{F}(w_\tau)\|_2^2},$$

and therefore

$$\|w_{t-1} - w_0\|_2^2 \leq \eta^2(t-1) \sum_{\tau=0}^{t-2} \|\nabla \tilde{F}(w_\tau)\|_2^2. \quad (9)$$

On the other hand, since $\tilde{F}(w)$ is also μ -smooth, we have for every $t \geq 1$,

$$\begin{aligned}\tilde{F}(w_t) &\leq \tilde{F}(w_{t-1}) + \langle \nabla \tilde{F}(w_{t-1}), w_t - w_{t-1} \rangle + \frac{\mu}{2} \|w_t - w_{t-1}\|_2^2 \\ &= \tilde{F}(w_{t-1}) - \frac{1}{2\mu} \|\nabla \tilde{F}(w_{t-1})\|_2^2,\end{aligned}$$

where in the equality we use the fact that $\eta = 1/\mu$. This implies that

$$\sum_{\tau=0}^{t-2} \|\nabla \tilde{F}(w_{\tau+1})\|_2^2 \leq 2\mu(\tilde{F}_0 - \tilde{F}^*). \quad (10)$$

By combining (9) and (10), we obtain

$$\|w_{t-1} - w_0\|_2^2 \leq \frac{2}{\mu}(\tilde{F}_0 - \tilde{F}^*)(t-1), \quad (11)$$

and combining (8) and (11), we get

$$\|\Delta(w_{t-1})\|_2^2 \leq 3C^2 + \frac{6\delta^2}{\mu}(\tilde{F}_0 - \tilde{F}^*)(t-1) + \frac{48\rho^2}{\mu^2}(\tilde{F}_0 - \tilde{F}^*)^2(t-1)^2. \quad (12)$$

By averaging (12) over $t = 1, \dots, T$ and plugging the result in (7), we obtain

$$\frac{1}{T} \sum_{t=1}^T \|\nabla F(w_{t-1})\|_2 \leq \frac{\sqrt{2\mu(\tilde{F}_0 - \tilde{F}^*)}}{\sqrt{T}} + \sqrt{3}C + \delta \sqrt{\frac{3}{\mu}(\tilde{F}_0 - \tilde{F}^*)\sqrt{T}} + \frac{4\rho}{\mu}(\tilde{F}_0 - \tilde{F}^*)T,$$

which completes the proof.

D Proof of Theorem 3

Let $\tilde{F}^* = \min_{w \in \mathcal{W}} \tilde{F}(w)$. Since we run gradient descent with learning rate $\eta = 1/\mu$ on the convex and μ -smooth function \tilde{F} , according to standard results in convex optimization [Bubeck \(2014\)](#), we have

$$\tilde{F}(w_T) - \tilde{F}^* \leq \frac{2\mu\tilde{D}^2}{T}. \quad (13)$$

We provide the following lemma that bounds the difference between $\tilde{F}(w)$ and $F(w)$.

Lemma 2. *For any $w \in \mathcal{W}$, we have*

$$|F(w) - \tilde{F}(w)| \leq \frac{1}{2K} \sum_{k=1}^{K-1} \delta \|w - \hat{w}_k\|_2^2 + \rho \|w - \hat{w}_k\|_2^3$$

We prove Lemma 2 in Appendix D.1. Here, we proceed to analyze $F(w_T) - F^*$. We have

$$\begin{aligned}F(w_T) - F^* &= F(w_T) - \tilde{F}(w_T) + \tilde{F}(w_T) - \tilde{F}^* + \tilde{F}^* - F^* \\ &\leq F(w_T) - \tilde{F}(w_T) + \tilde{F}^* - F^* + \frac{2\mu\tilde{D}^2}{T}.\end{aligned} \quad (14)$$

To bound $F(w_T) - \tilde{F}(w_T)$, we use the fact that for any convex and smooth functions, when we run gradient descent with learning rate $1/\mu$, the iterates only move closer to the minimum of the function, *i.e.*, $\|w_T - \tilde{w}^*\|_2 \leq \tilde{D}$. Then, by triangle inequality, for any $k \in [K-1]$

$$\|w_T - \hat{w}_k\|_2 \leq \|w_T - \tilde{w}^*\|_2 + \|\tilde{w}^* - \hat{w}_k\|_2 \leq \tilde{D} + \|w_0 - \hat{w}_k\|_2.$$

Using the fact that for any two positive numbers a and b , $(a+b)^2 \leq 2(a^2 + b^2)$ and $(a+b)^3 \leq 4(a^3 + b^3)$, we obtain

$$\|w_T - \hat{w}_k\|_2^2 \leq 2(\tilde{D}^2 + \|\tilde{w}^* - \hat{w}_k\|_2^2), \quad (15)$$

$$\|w_T - \hat{w}_k\|_2^3 \leq 4(\tilde{D}^3 + \|\tilde{w}^* - \hat{w}_k\|_2^3). \quad (16)$$

By combining (15) and (16) with Lemma 2, we obtain

$$F(w_T) - \tilde{F}(w_T) \leq \frac{1}{K} \sum_{k=1}^{K-1} \delta(\tilde{D}^2 + \|\tilde{w}^* - \hat{w}_k\|_2^2) + 2\rho(\tilde{D}^3 + \|\tilde{w}^* - \hat{w}_k\|_2^3). \quad (17)$$

We can use a similar argument to bound $\tilde{F}^* - F^*$. Note that $\tilde{F}^* - F^* \leq \tilde{F}(w^*) - F(w^*)$. Therefore, according to Lemma 2, we have

$$\tilde{F}^* - F^* \leq \frac{1}{2K} \sum_{k=1}^{K-1} \delta\|w^* - \hat{w}_k\|_2^2 + \rho\|w^* - \hat{w}_k\|_2^3. \quad (18)$$

Then we can complete the proof by combining (14), (17), and (18).

D.1 Proof of Lemma 2

By definition of $F(w)$ and $\tilde{F}(w)$, we have

$$\begin{aligned} F(w) - \tilde{F}(w) &= \frac{1}{K} \sum_{k=1}^{K-1} L_k(w) - \left(L_k(\hat{w}_k) + (w - \hat{w}_k)^\top \nabla L_k(\hat{w}_k) + \frac{1}{2}(w - \hat{w}_k)^\top H_k(w - \hat{w}_k) \right) \\ &= \frac{1}{K} \sum_{k=1}^{K-1} \frac{1}{2}(w - \hat{w}_k)^\top (\nabla^2 L_k(\hat{w}_k + \xi_k(w - \hat{w}_k)) - H_k)(w - \hat{w}_k), \end{aligned}$$

for some $\xi_k \in [0, 1]$. Then according to (5), we get

$$\begin{aligned} |F(w) - \tilde{F}(w)| &\leq \frac{1}{2K} \sum_{k=1}^{K-1} \|\nabla^2 L_k(\hat{w}_k + \xi_k(w - \hat{w}_k)) - H_k\|_2 \|w - \hat{w}_k\|_2^2 \\ &\leq \frac{1}{2K} \sum_{k=1}^{K-1} \delta\|w - \hat{w}_k\|_2^2 + \rho\|w - \hat{w}_k\|_2^3. \end{aligned}$$

E Proof of Theorem 4

First, recall the definitions of the population quadratic approximation function $\tilde{L}_{K-1}(w)$ and the empirical quadratic approximation function $L_{K-1}^{\text{prox}}(w)$:

$$\begin{aligned} \tilde{L}_{K-1}(w) &= \sum_{k=1}^{K-1} L_k(\hat{w}_k) + (w - \hat{w}_k)^\top \nabla L_k(\hat{w}_k) + \frac{1}{2}(w - \hat{w}_k)^\top \nabla^2 L_k(\hat{w}_k)(w - \hat{w}_k), \\ L_{K-1}^{\text{prox}}(w) &= \sum_{k=1}^{K-1} \hat{L}_k(\hat{w}_k) + (w - \hat{w}_k)^\top \nabla \hat{L}_k(\hat{w}_k) + \frac{1}{2}(w - \hat{w}_k)^\top \nabla^2 \hat{L}_k(\hat{w}_k)(w - \hat{w}_k). \end{aligned}$$

According to Lemma 2, we know that

$$F(w) \leq \frac{1}{K} (L_K(w) + \tilde{L}_{K-1}(w)) + \frac{\rho}{2K} \sum_{k=1}^{K-1} \|w - \hat{w}_k\|_2^3. \quad (19)$$

Note that here we choose $\delta = 0$ in Lemma 2 since we do not consider the approximation error of the Hessian matrix. The next step is to use $\hat{L}_K(w) + L_{K-1}^{\text{prox}}(w)$ to bound $L_K(w) + \tilde{L}_{K-1}(w)$. For this, we use the standard uniform convergence results with Rademacher complexity (Mohri et al., 2018).

For any fixed $k \in [K]$, with probability at least $1 - \delta$ over the random examples $(x_{k,i}, y_{k,i})$, $i = 1, \dots, n_k$, we have for all $w \in \mathcal{W}$

$$L_k(w) \leq \widehat{L}_k(w) + 2b \underbrace{\mathbb{E}_\sigma \left[\frac{1}{n_k} \sup_{w \in \mathcal{W}} \sum_{i=1}^{n_k} \sigma_i \ell_k(w; x_{k,i}, y_{k,i}) \right]}_{R_k^{(\ell)}} + 3b \sqrt{\frac{\log(2/\delta)}{n_k}}, \quad (20)$$

where σ denotes a sequence of i.i.d. Rademacher random variables. Using normed Rademacher complexity, we can get with probability at least $1 - \delta$,

$$\sup_{w \in \mathcal{W}} \|\nabla L_k(w) - \nabla \widehat{L}_k(w)\|_2 \leq 2g \underbrace{\mathbb{E}_\sigma \left[\frac{1}{n_k} \sup_{w \in \mathcal{W}} \left\| \sum_{i=1}^{n_k} \sigma_i \nabla \ell_k(w; x_{k,i}, y_{k,i}) \right\|_2 \right]}_{R_k^{(g)}} + 3g \sqrt{\frac{\log(4/\delta)}{2n_k}}, \quad (21)$$

and with probability at least $1 - \delta$,

$$\sup_{w \in \mathcal{W}} \|\nabla^2 L_k(w) - \nabla^2 \widehat{L}_k(w)\|_2 \leq 2h \underbrace{\mathbb{E}_\sigma \left[\frac{1}{n_k} \sup_{w \in \mathcal{W}} \left\| \sum_{i=1}^{n_k} \sigma_i \nabla^2 \ell_k(w; x_{k,i}, y_{k,i}) \right\|_2 \right]}_{R_k^{(h)}} + 3h \sqrt{\frac{\log(4/\delta)}{2n_k}}. \quad (22)$$

Then by Cauchy–Schwarz inequality and union bound, we have with probability at least $1 - \delta$,

$$\begin{aligned} & L_K(w) + \widetilde{L}_{K-1}(w) \\ & \leq \widehat{L}_K(w) + L_{K-1}^{\text{prox}}(w) + (L_K(w) - \widehat{L}_K(w)) + \sum_{k=1}^{K-1} L_k(w) - \widehat{L}_k(w) + \|w - \widehat{w}_k\|_2 \|\nabla L_k(w) - \nabla \widehat{L}_k(w)\|_2 \\ & \quad + \frac{1}{2} \|\nabla^2 L_k(w) - \nabla^2 \widehat{L}_k(w)\|_2 \|w - \widehat{w}_k\|_2^2 \\ & \leq \widehat{L}_K(w) + L_{K-1}^{\text{prox}}(w) + 2bR_K^{(\ell)} + 3b \sqrt{\frac{\log(6K/\delta)}{n_K}} + \sum_{k=1}^{K-1} 2bR_k^{(\ell)} + 2gR_k^{(g)} + 2hR_k^{(h)} + 3(b+g+h) \sqrt{\frac{\log(12K/\delta)}{2n_k}}. \end{aligned}$$

The proof is then completed by combining this inequality with (19).

F Experiment Details

We provide the hyper parameters that we used in all the experiments. As mentioned, for EWC, SI, and Kronecker algorithms, we introduce a regularization coefficient λ . In addition, for SI, there is another hyper parameter ξ used in each update step (see Eq. (5) in Section 3 of [Zenke et al. \(2017\)](#) for the definition of ξ). The values of λ and ξ used in our experiments are provided in Table 4.

Table 4: Hyper parameters

	Permuted MNIST	Rotated MNIST	Split CIFAR
EWC λ	1.0	10.0	1.0
SI (λ, ξ)	(1.0, 1.0)	(1.0, 1.0)	(1.0, 0.1)
Kronecker λ	10.0	10.0	1.0