Generalisation Guarantees for Continual Learning with Orthogonal Gradient Descent

Anonymous Authors[1]

Abstract

In continual learning settings, deep neural networks are prone to catastrophic forgetting. Orthogonal Gradient Descent (Farajtabar et al., 2019) achieves state-of-the-art results in practice for continual learning, although no theoretical guarantees have been proven yet. We derive the first generalisation guarantees for the algorithm OGD for continual learning, for overparameterized neural networks. We find that OGD is only provably robust to catastrophic forgetting across a single task. We propose OGD+, prove that it is robust to catastrophic forgetting across an arbitrary number of tasks, and that it verifies tighter generalisation bounds. The experiments show that OGD+ outperforms OGD on settings with long range memory dependencies, even though the models are not overparameterized. Also, we derive a closed form expression of the learned models through tasks, as a recursive kernel regression relation, which captures the transferability of knowledge through tasks. Finally, we quantify theoretically the impact of task ordering on the generalisation error, which highlights the importance of the curriculum for lifelong learning.

1. Introduction

Continual learning is a setting in which an agent is exposed to multiples tasks sequentially (Kirkpatrick et al., 2016). The core challenge lies in the ability of the agent to learn the new tasks while retaining the knowledge acquired from previous tasks. Too much plasticity will lead to catastrophic forgetting, which means the degradation of the ability of the agent to perform the past tasks (McCloskey & Cohen 1989, Ratcliff 1990, Goodfellow et al. 2014). On the other hand, too much stability will hinder the agent from adapting to new tasks.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review.

Recent works on the Neural Tangent Kernel (Jacot et al., 2018) and on the convergence of Stochastic Gradient Descent for overparameterized neural networks (Arora et al., 2019) have unlocked powerful tools to analyze the training dynamics of over-parameterized neural networks. We leverage these theoretical findings in order to to prove guarantees on the convergence and the generalisation of the algorithm, Orthogonal Gradient Descent for Continual Learning (Farajtabar et al., 2019).

Our contributions are summarized as follows:

1. We provide closed form expressions of the functions learned across tasks. We find that they can be expressed as a linear combination of kernel regressors, over the previously seen tasks. The relationship also captures task similarity and the transferability of knowledge across tasks through the NTK (Sec. 3, Theorem 1).

2. We prove the first generalisation bound for continual learning with OGD, to our knowledge. We derive bounds for within-task and outside-task generalisation. We find that generalisation through time depends on a task similarity with respect to the NTK, which we quantify rigorously (Sec. 4, Theorem 2).

3. We prove that OGD is robust to forgetting with respect to the previous task only (Sec. 4, Lemma 1).

4. We build-up on this insight to propose OGD+ (Sec. 5, Alg. 1), an extension of OGD, which we prove robust to catastrophic forgetting across an arbitrary number of tasks (Sec. 5, Lemma 3). We also prove tighter generalisation bounds than OGD (Sec. 5, Theorem 3).

5. As a side result, we find that Lemma 2 also quantifies the impact of the learning curriculum on the generalisation error. We find that NTK task dissimilarity impacts negatively generalisation, and that an ordering of tasks that minimises dissimilarity between neighbouring tasks leads to a tighter generalisation bound. (Sec. 2, Lemma 2).

Even though the analysis relies on the assumption that the model is overparametrised, the analysis leads to prac-

tical insights to develop OGD+. Experiments in non-overparametrised settings on the MNIST and CIFAR-100 benchmarks show that OGD+ outperforms OGD on settings with long range memory dependencies (Sec. 6).

2. Preliminaries

Notation We use bold-faced characters for vectors and matrices. We use $\|\cdot\|$ to denote the Euclidian norm of a vector or the spectral norm of a matrix, and $\|\cdot\|_{\mathrm{F}}$ to denote the Frobenius norm of a matrix. We use $\langle\cdot,\cdot\rangle$ for the Euclidian dot product, and $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ the dot product in the Hilbert space $\mathcal{H}$. We index the the task ID by τ. The $\leq$ operator if used with matrices, corresponds to the partial ordering over symetric matrices. We denote $\mathbb{N}$ the set of natural numbers, $\mathbb{R}$ the space of real numbers and $\mathbb{N}^\star$ for the set $\mathbb{N} \setminus \{0\}$. We use $\oplus$ to refer to the direct sum over Euclidian spaces.

2.1. Continual Learning

Continual learning considers a series of tasks $\{\mathcal{T}_1, \mathcal{T}_2, \ldots\}$, where each task can be viewed as a separate supervised learning problem. Similarly to online learning, data from each task is revealed only once. The goal of continual learning is to model each task accurately with a single model. The challenge is to achieve a good performance on the new tasks, while retaining knowledge from the previous tasks (Nguyen et al., 2018).

We assume the data from each task $\mathcal{T}_\tau$, $\tau \in \mathbb{N}^\star$, is drawn from a distribution $\mathcal{D}_\tau$. Individual samples are denoted $(\mathbf{x}_{\tau,i}, y_{\tau,i})$, where $i \in [n_\tau]$. Also, we only consider the binary classification setting for the sake of simplicity: $\mathbf{x}_{\tau,i} \in \mathbb{R}^d$ and $y_{\tau,i} \in \{-1, +1\}$. We note that it does not restrict the scope of the analysis, which can be easily extended to multiclass settings.

2.2. OGD for Continual Learning

Let $\mathcal{T}_T$ the current task, where $T \in \mathbb{N}^\star$. For all $i \in [n_T]$, let $\mathbf{v}_{T,i} = \nabla_\theta f^\star_{T-1}(\mathbf{x}_{T-1,i})$, which is the Jacobian of task $\mathcal{T}_T$. We define $\mathbb{E}_\tau = \mathrm{vec}(\{\mathbf{v}_{\tau,i}, i \in [n_\tau]\})$, which is the subspace induced by the Jacobian. The idea behind OGD is to update the weights along the projection of the gradient on the orthogonal space induced by the Jacobians over the previous tasks $\mathbb{E}_1 \oplus \ldots \oplus \mathbb{E}_{\tau-1}$. The update rule for the task $\mathcal{T}_T$ is as follows (Farajtabar et al., 2019):

$$\mathbf{w}_T(t+1) = \mathbf{w}_T(t) - \eta \Pi_{\mathbb{E}^\perp_{T-1}} \nabla_{\mathbf{w}} \mathcal{L}^T_\lambda(\mathbf{w}_T(t)).$$

The intuition behind OGD is to "preserve the previously acquired knowledge by maintaining a space consisting of the gradient directions of the neural networks predictions on previous tasks" (Farajtabar et al., 2019). Throughout the paper, we only consider the OGD-GTL variant which stores the gradient with respect to the ground truth logit.

To prevent over-fitting and guarantee the uniqueness of the global minimum in the Neural Tangent Kernel (NTK) regime, we apply a ridge regularization with a parameter $\lambda \in \mathbb{R}^+$. For a task $\mathcal{T}_\tau$, we write the corresponding loss as follows:

$$\mathcal{L}^\tau_\lambda(\mathbf{w}) = \sum_{i=1}^{n_\tau} (f_\tau(\mathbf{x}_{\tau,i}) - y_{\tau,i})^2 + \lambda \|\mathbf{w} - \mathbf{w}_{\tau-1}\|^2.$$

2.3. Generalisation for Continual Learning

We define within-task generalisation as the ability of the agent to acquire new knowlege and outside-task generalisation as its ability to preserve the acquired knowledge.

Definition 1 (Within-task and outside-task generalisation) *Consider a loss function $l : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. The population loss over the distribution $\mathcal{D}$, and the empirical loss over n samples $D = \{(\mathbf{x}_i, y_i), i \in [n]\}$ from the same distribution $\mathcal{D}$ are defined as:*

$$L_D(f) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[l(f(\mathbf{x}), y)],$$
$$L_S(f) = \frac{1}{n}\sum_{i=1}^{n} l(f(\mathbf{x}_i), y_i).$$

Let $\mathcal{T}_1, \ldots \mathcal{T}_T$ a sequence of tasks, and $\mathcal{D}_1, \ldots \mathcal{D}_T$ their corresponding distributions.
Let $f^\star_1, \ldots f^\star_T$ the trained models at each task. Let $\tau \in [T]$ fixed.
We define:

- *within-task generalisation of the task $\mathcal{T}_\tau$ as $L_{D_\tau}(f^\star_\tau)$.*
- *outside-task generalisation of the task $\mathcal{T}_\tau$ with respect to a task $\mathcal{T}_{\tau'}$, where $\tau' < \tau$ as $L_{D_\tau}(f^\star_{\tau'})$.*

In practice, several works also tracked these metrics in their experiments (Kirkpatrick et al. 2016, Farajtabar et al. 2019).

2.4. Neural Tangent Kernel

In their seminal paper, Jacot et al. (2018) established the connection between deep networks and kernel methods by introducing the Neural Tangent Kernel (NTK). They showed that at the infinite width limit, the kernel remains constant throughout training. Lee et al. (2019) also showed that a network evolves as a linear model in the infinite width limit when trained on certain losses under gradient descent.

Throughout our analysis, we make the assumption that the neural network is overparameterized, and consider the linear approximation of the neural network around its initialisation:

$$f^{(t)}(\mathbf{x}) \approx f^{(0)}(\mathbf{x}) + \nabla_{\mathbf{w}} f^{(0)}(\mathbf{x})^T (\mathbf{w}(t) - \mathbf{w}(0)).$$

3. Convergence of OGD for Continual Learning

In this section, we derive a closed form expression for the learned models across tasks. We find a recursive kernel ridge regression relationship between the models across tasks. The result is presented in Theorem 1, a stepping stone towards proving the generalisation bound for OGD in Sec. 4.

3.1. Convergence Theorem

Now, we state the main result of this section:

Theorem 1 (Convergence of SGD and OGD for Continual Learning) *Let $\mathcal{T}_1, \dots, \mathcal{T}_T$ be a sequence of tasks. Fix a learning rate sequence $(\eta_\tau)_{\tau \in [T]}$. If, for all τ, the learning rate satisfies*

$$\eta_\tau < \frac{1}{\|k_\tau(\mathbf{X}_\tau, \mathbf{X}_\tau)\| + \lambda_\tau^2},$$

then for all τ, $\mathbf{w}_\tau(t)$ converges linearly to a limit solution $\mathbf{w}_\tau^\star$ such that

$$f_\tau^\star(\mathbf{x}) = f_{\tau-1}^\star(\mathbf{x}) + k_\tau(\mathbf{x}, \mathbf{X}_\tau)^T \mathbf{H}_{\tau,\lambda_\tau}^{-1} \tilde{\mathbf{y}}_\tau,$$

where

$$\begin{aligned} k_\tau(\mathbf{x}, \mathbf{x}') &= \widetilde{\phi}_\tau(\mathbf{x})^T \widetilde{\phi}_\tau(\mathbf{x}'), \\ \tilde{\mathbf{y}}_\tau &= \mathbf{y}_\tau - \mathbf{y}_{\tau-1\to\tau}, \\ \mathbf{y}_{\tau-1\to\tau} &= f_{\tau-1}^\star(\mathbf{X}_\tau), \\ \phi_\tau(\mathbf{x}) &= \nabla_{\mathbf{w}} f_{\tau-1}^\star(\mathbf{x}), \\ \mathbf{H}_{\tau,\lambda_\tau} &= k_\tau(\mathbf{X}_\tau, \mathbf{X}_\tau) + \lambda_\tau^2 \mathbf{I}, \\ \widetilde{\phi}_\tau(\mathbf{x}) &= \begin{cases} \phi_\tau(\mathbf{x}) & \text{for SGD,} \\ \mathbf{T}_\tau \phi_\tau(\mathbf{x}) & \text{for OGD.} \end{cases} \end{aligned}$$

and $\{\mathbf{T}_\tau, \tau \in [T]\}$ are proxy matrices for the analysis.

The theorem describes how the model $f_\tau^\star$ evolves across tasks. The theorem is recursive because the learning is incremental. For a given task $\mathcal{T}_\tau$, $f_{\tau-1}^\star(\mathbf{x})$ is the knowledge acquired by the agent up to the task $\mathcal{T}_{\tau-1}$. At this stage, the model only fits the residual $\tilde{\mathbf{y}}_\tau = \mathbf{y}_\tau - \mathbf{y}_{\tau-1\to\tau}$, which complements the knowledge acquired through previous tasks. This residual is also a proxy for task similarity. If the tasks are identical, the residual is equal to zero. The knowledge increment is captured by the term: $k_\tau(\mathbf{x}, \mathbf{X}_\tau)^T (k_\tau(\mathbf{X}_\tau, \mathbf{X}_\tau) + \lambda_\tau^2 \mathbf{I})^{-1} \tilde{\mathbf{y}}_\tau$. Finally, task similarity is computed with respect to the most recent feature map $\widetilde{\phi}_\tau$, and k_τ is the NTK with respect to the feature map $\widetilde{\phi}_\tau$.

Corollary 1 *The recursive relation from Theorem 1 can also be written as a linear combination of kernel regressors as follows:*

$$f_\tau^\star(\mathbf{x}) = \sum_{k=1}^{\tau} \tilde{f}_k^\star(\mathbf{x}),$$

where

$$\tilde{f}_k^\star(\mathbf{x}) = k_k(\mathbf{x}, \mathbf{X}_k)^T (k_k(\mathbf{X}_k, \mathbf{X}_k) + \lambda_k^2 \mathbf{I})^{-1} \tilde{\mathbf{y}}_k.$$

Proof Sketch: We prove Theorem 1 by induction. We rewrite the loss function as a regression on the residual $\tilde{\mathbf{y}}_\tau$ instead of $\mathbf{y}_\tau$. Then, we rewrite the optimisation objective as an unconstrained strongly convex optimisation problem. Finally, we compute the unique solution in a closed form. The full proof is presented in App. A.1.

3.2. Distance from Initialisation

As described in Sec. 3.1, $\tilde{\mathbf{y}}_\tau$ is a residual. It is equal to zero if the model $f_{\tau-1}^\star$ makes perfect predictions on the next task $\mathcal{T}_\tau$. The more the next task $\mathcal{T}_\tau$ is different, the further the neural network needs to move from its previous state in order to fit it. Corollary 2 tracks the distance from initialisation as a function of task similarity.

Corollary 2 *For SGD, and for OGD under the additional assumption that $\{\mathbf{T}_\tau, \tau \in [T]\}$ are orthonormal,*

$$\left\|\mathbf{w}_{\tau+1}^\star - \mathbf{w}_\tau^\star\right\|_{\mathrm{F}} = \sqrt{\tilde{\mathbf{y}}_\tau^T \mathbf{H}_{\tau,\lambda_\tau}^{-1} \mathbf{H}_{\tau,0} \mathbf{H}_{\tau,\lambda_\tau}^{-1} \tilde{\mathbf{y}}_\tau},$$

where

$$\mathbf{H}_{\tau,\lambda_\tau} = k_\tau(\mathbf{X}_\tau, \mathbf{X}_\tau) + \lambda_\tau^2 \mathbf{I}_\tau.$$

The proof is presented in App. A.2. The orthonormality assumption is not restrictive, since the set $\{\mathbf{T}_\tau, \tau \in [T]\}$ is only a proxy for the analysis; indeed we can choose any convenient basis to work with.

Remark 1 *Corollary 2 can be applied to get a similar result to Theorem 3 by Liu et al. (2019). In this remark, we consider mostly their notations. Their theorem states that under some conditions, for 2-layer neural networks with a RELU activation function, with probability no less than $1 - \delta$ over random initialisation,*

$$\|\mathbf{W}(P) - \mathbf{W}(Q)\|_{\mathrm{F}} \leq \sqrt{\tilde{\mathbf{y}}_{P\to Q}^T H_P^{\infty-1} \tilde{\mathbf{y}}_{P\to Q}} + \epsilon,$$

where, in their work:

$$\begin{aligned} \mathbf{y}_{P\to Q} &= H_{PQ}^{\infty,T} H_P^{\infty-1} \mathbf{y}_P, \\ \tilde{\mathbf{y}}_{P\to Q} &= \mathbf{y}_Q - \mathbf{y}_{P\to Q}. \end{aligned}$$

Note that H_P^∞ is a Gram matrix, which also corresponds to the NTK of the neural network they consider. We see an analogy with our result, where we work directly with the NTK, with no assumptions on the neural network. One important observation is that, to our knowledge, since there are no guarantees for the invertibility of our Gram matrix, we add a ridge regularisation to work with a regularised matrix, which is then invertible. In our setting, by considering $\lambda \to 0$, and with the additional assumption of invertibility of $\mathbf{H}_{\tau,0}$,which is valid in the two-layer overparametrised RELU neural network considered in the setting of Liu et al. (2019), we can recover a similar approximation.

4. Generalisation of OGD for Continual Learning

In this section, we study the generalisation properties of OGD. First, we prove that OGD is robust to catastrophic forgetting with respect to the previous task (Lemma 1). Then, we present the the main generalisation theorem for OGD (Thm. 2). The theorem provides several insights on the relation between task similarity and generalisation. Finally, we present how the Rademacher complexity relates to task similarity across a large number of tasks (Lemma 2). The lemma states that the more dissimilar tasks are, the larger the class of functions explored by the neural network, with high probability. This result highlights the importance of the curriculum for Continual Learning.

4.1. Memorisation property of OGD

The key to obtaining tight generalisation bounds for OGD is Lemma 1.

Lemma 1 (Memorisation Property of OGD) *Given a task $\mathcal{T}_\tau$, for all $\mathbf{x}_{\tau,i} \in D_\tau$, a sample from the training data of the task $\mathcal{T}_\tau$, it holds that*

$$f^\star_{\tau+1}(\mathbf{x}_{\tau,i}) = f^\star_\tau(\mathbf{x}_{\tau,i}). \tag{1}$$

As motivated by Farajtabar et al. 2019, the orthogonality of the gradient updates aims to preserve the acquired knowledge, by not altering the weights along relevant dimensions when learning new tasks. Lemma 1 states that the training error on the previous task is unchanged, when training with OGD. However, there are no guarantees that the knowledge from the tasks before the previous task is preserved.

The proof of Lemma 1 is presented in App. B.1.4

4.2. Generalisation of OGD for Continual Learning

Now, we state the main generalisation theorem for OGD, which provides within-task and outside-task generalisation bounds.

Theorem 2 (Generalisation of OGD for Continual Learning) *Let $\{\mathcal{T}_1, \dots \mathcal{T}_T\}$ be a sequence of tasks. Let be $\{\mathcal{D}_1, \dots, \mathcal{D}_T\}$ the respective distributions over $\mathbb{R}^d \times \{-1, 1\}$. Let $\{(\mathbf{x}_{\tau,i}, y_{\tau,i}), i \in [n_t], \tau \in [T]\}$ be i.i.d. samples from $\mathcal{D}_\tau, \tau \in [T]$. Denote $\mathbf{X}_\tau = (\mathbf{x}_{\tau,1}, \dots, \mathbf{x}_{\tau,n_\tau})$, $\mathbf{y}_\tau = (y_{\tau,1}, \dots, y_{\tau,n_\tau})$. Consider the kernel ridge regression solution $f^\star_T$. Suppose that the kernel matrices satisfy*

$$\mathrm{Tr}(k_\tau(\mathbf{X}_\tau, \mathbf{X}_\tau)) = \mathcal{O}(n_\tau),$$

then, for any loss function $\ell : \mathbb{R} \times \mathbb{R} \to [0, c]$ that is c-Lipchitz in the first argument, with probability at least $1 - \delta$, for within-task generalisation ($\mathcal{T}_T$),

$$\mathcal{L}_{D_T}(f^\star_T) \leq \frac{\lambda}{2}\sqrt{\frac{\tilde{\mathbf{y}}_T^T k_T(\mathbf{X}_T, \mathbf{X}_T)^{-1}\tilde{\mathbf{y}}_T}{n_T}} + \sum_{k=1}^{T}\left(\mathcal{O}\left(\sqrt{\frac{\tilde{\mathbf{y}}_k^T \mathbf{H}_{k,\lambda_k}^{-1}\tilde{\mathbf{y}}_k}{n_k}}\right) + \Delta_k\right),$$

for outside-task generalisation ($\mathcal{T}_{T-1}$),

$$\mathcal{L}_{D_{T-1}}(f^\star_T) \leq \frac{\lambda}{2}\sqrt{\frac{\tilde{\mathbf{y}}_{T-1}^T k_T(\mathbf{X}_T, \mathbf{X}_T)^{-1}\tilde{\mathbf{y}}_{T-1}}{n_T}} + \sum_{k=1}^{T}\left(\mathcal{O}\left(\sqrt{\frac{\tilde{\mathbf{y}}_k^T \mathbf{H}_{k,\lambda_k}^{-1}\tilde{\mathbf{y}}_k}{n_k}}\right) + \Delta_k\right),$$

and for outside-task generalisation ($\mathcal{T}_\tau, \tau < T-1$),

$$\mathcal{L}_{D_\tau}(f^\star_T) \leq \sqrt{A_\tau + \sum_{k=\tau+1}^{T}\frac{1}{4\lambda^2}\frac{\tilde{\mathbf{y}}_k^T \tilde{\mathbf{K}}_{k,\tau}\tilde{\mathbf{y}}_k}{n_\tau}} + \sum_{k=1}^{T}\left(\mathcal{O}\left(\sqrt{\frac{\tilde{\mathbf{y}}_k^T \mathbf{H}_{k,\lambda_k}^{-1}\tilde{\mathbf{y}}_k}{n_k}}\right) + \Delta_k\right),$$

where

$$\begin{aligned}
\Delta_k &= \mathcal{O}\left(\frac{1}{\sqrt{n_k}}\right) + 3c\sqrt{\frac{\log(2/\delta)}{2n_k}},\\
\mathbf{H}_{\tau,\lambda} &= k_\tau(\mathbf{X}_\tau, \mathbf{X}_\tau) + \lambda^2\mathbf{I},\\
A_\tau &= \frac{\lambda_\tau^2}{4}\frac{\tilde{\mathbf{y}}_\tau^T(k_\tau(\mathbf{X}_\tau, \mathbf{X}_\tau))^{-1}\tilde{\mathbf{y}}_\tau}{n_\tau}\\
\phi_\tau(\mathbf{x}) &= \mathbf{T}_\tau \nabla_{\mathbf{W}} f^\star_{\tau-1}(\mathbf{x}),\\
\tilde{\mathbf{y}}_\tau &= \mathbf{y}_\tau - \mathbf{y}_{\tau-1\to\tau},\\
\mathbf{y}_{\tau-1\to\tau} &= f^\star_{\tau-1}(\mathbf{X}_\tau),\\
\mathbf{K}_{i,j,k} &= k_i(\mathbf{X}_j, \mathbf{X}_k),\\
\tilde{\mathbf{K}}_{k,\tau} &= \mathbf{K}_{k,k,k}^{-1}\mathbf{K}_{k,\tau,k}\mathbf{K}_{k,\tau,k}^T\mathbf{K}_{k,k,k}^{-1}.
\end{aligned}$$

The intution behind Theorem 2 is as follows:

- Within-task generalisation: The generalisation error on the most recent task leverages the information learned during training on the previous tasks. The bound is tighter compared to learning from scratch, since it depends on the residual $\tilde{\mathbf{y}}_\tau$, which is a proxy for Forward Transfer. Therefore it captures the transferability of knowledge across tasks.
- Outside-task generalisation ($\mathcal{T}_{T-1}$): The generalisation bound increases only with respect to the Rademacher complexity when training with OGD. The tightness of this bound for OGD is due to Lemma 1. This lemma is valid for OGD and not for SGD, which implies that tighter generalisation is guaranteed compared to SGD.
- Outside-task generalisation ($\mathcal{T}_\tau, \tau \leq T-2$): The upper bound depends on the similarity between the outside task and the latest task; the more dissimilar the subsequent tasks are, the more the upper bound diverges from the initial upper bound. This bound captures catastrophic forgetting as a function of the tasks dissimilarity. This bound is the same for OGD and SGD.

These bounds share some similarities with the bounds derived by Arora et al. (2019), Liu et al. (2019) and Hu et al. (2019), where in these works, the bounds were derived for supervised learning settings, and in some cases for two-layer RELU neural networks. Similarly, the bounds depend on the Gram matrix of the data, with the feature map corresponding to the NTK.

Proof Sketch: The proof is presented in App. B.1. One challenge is that the function class is the set of linear combinations of kernel regressors (Theorem 1). We state Lemma 4 to bound the Rademacher complexity of this function class. Then we derive bounds for the training error for each case in Theorem 2. The first case is straightforward. For the second case, we use Lemma 1, then derive a similar proof to the first case. The third case presents some additional technical challenges. In order to derive the upper bounds, we draw a strong inspiration from Hu et al. (2019), and leverage several of their proof techniques and mathematical tools.

4.3. Distance from Initialisation through Tasks

Now, we state Lemma 2, which tracks the Rademacher complexity through tasks.

Lemma 2 *Keeping the same notations and setting as Theorem 2, the Rademacher Complexity can be bounded as follows:*

$$\hat{\mathcal{R}}(\mathcal{F}_{\mathbf{B}_T}) \leq \sum_{\tau=1}^{T} \mathcal{O}\left(\sqrt{\frac{\tilde{\mathbf{y}}_\tau^T (k_\tau(\mathbf{X}_\tau, \mathbf{X}_\tau) + \lambda_\tau^2 \mathbf{I})^{-1} \tilde{\mathbf{y}}_\tau}{n_\tau}}\right) + \sum_{\tau=1}^{T} \mathcal{O}(\frac{1}{\sqrt{n_\tau}}).$$

The intuition behind Lemma 2 is that the upper bound on the Rademacher complexity increases when the tasks are dissimilar. The **NTK task similarity** between two subsequent tasks τ and $\tau+1$ is $\tilde{\mathbf{y}}_\tau^T (k_t(\mathbf{X}_\tau, \mathbf{X}_\tau) + \lambda^2 \mathbf{I})^{-1} \tilde{\mathbf{y}}_\tau$. The knowledge from the previous tasks is implicitly encoded in the kernel k_τ, which is based on the feature map ϕ_τ. This feature map encodes the acquired knowledge. As an edge case, if two successive tasks are identical, the residual $\tilde{\mathbf{y}}_\tau = \mathbf{y}_\tau - \mathbf{y}_{\tau-1\to\tau} = 0$, therefore, the upper bound on the Rademacher complexity does not increase.

Implications for Curriculum Learning We also observe that the upper bound depends on the task ordering, which may provide a theoretical explanation on the importance of learning with a curriculum (Bengio et al., 2009). In the following, we present an edge case which provided an intuition on how the bound captures the importance of the order. Consider two dissimilar tasks $\mathcal{T}_1$ and $\mathcal{T}_2$. A sequence of tasks alternating between $\mathcal{T}_1$ and $\mathcal{T}_2$ will lead to a large upper bound, as explained in the first paragraph. While, a sequence of tasks concatenating two sequences of $\mathcal{T}_1$ then $\mathcal{T}_2$ will lead to a lower upper bound.

Proof Sketch: The proof techniques for Lemma 2 are exactly the same as the ones for Theorem 2. The full proof is presented in Sec. B.1.6.

5. OGD+: Learning without Forgetting, provably

In the previous section, we demonstrated the limits of OGD, in terms of robustness to catastrophic forgetting on the long run. Now, we present OGD+, an extension of OGD, which we prove robust to catastrophic forgetting, across an arbitrary number of tasks (Lemma 3). Then, we prove tighter generalisation bounds compared to OGD (Theorem 3).

5.1. The OGD+ Algorithm

Algorithm 1 presents the OGD+ algorithm, we highlight the differences with OGD in red. The main difference is that OGD+ stores the feature maps with respect to the samples from previous tasks, in addition to the feature maps with respect to the samples from the current task, as opposed to OGD. This small change unlocks the proof of Lemma 3 given below, which implies tighter bounds for Theorem 3.

The idea behind OGD+ comes from the convergence Theorem (Sec. 3, Thm. 1). After training on a task $\mathcal{T}_\tau$, the learned model is a linear combination of the previous models. For a given sample $\mathbf{x}$ from a task $\mathcal{T}_k$ where $k < \tau$, in order to keep the training error identical, the weights need to be updated along the directions that are orthogonal to *all the subsequent feature maps* of $\mathbf{x}$. OGD only considers the feature map of the source task of the sample. Storing all the feature maps implies that the learned model back from task $\mathcal{T}_k$, can be recovered even after training on an arbitrary number of tasks.

In order to compute the feature maps with respect to the previous samples, OGD+ saves these samples in a dedicated memory, we call this storage the *samples memory*. This memory comes in addition to the orthonormal *feature maps memory*. The only role of the *samples memory* is to compute the feature maps. While the proofs below are under the assumption that the memory size is infinite, in the experiments, we keep a limited size for both memories.

Algorithm 1: OGD+ for Continual Learning

Input : A task sequence $\mathcal{T}_1, \mathcal{T}_2, \ldots$, learning rate η

1. Initialize $S_J \leftarrow \{\}$; $S_D \leftarrow \{\}$; $\mathbf{w} \leftarrow \mathbf{w}_0$
2. **for** *Task ID* $\tau = 1, 2, 3, \ldots$ **do**
   - **repeat**
     - $\mathbf{g} \leftarrow$ Stochastic Batch Gradient for $\mathcal{T}_\tau$ at $\mathbf{w}$;
     - $\tilde{\mathbf{g}} = \mathbf{g} - \sum_{\mathbf{v} \in \mathcal{S}_J} \text{proj}_{\mathbf{v}}(\mathbf{g})$;
     - $\mathbf{w} \leftarrow \mathbf{w} - \eta \tilde{\mathbf{g}}$
   - **until** *convergence*;
   - Sample $\mathcal{S} \subset \mathcal{S}_D$;
   - **for** $(\mathbf{x}, y) \in \mathcal{D}_\tau \bigcup \mathcal{S}$ *and* $k \in [1, c]$ *s.t.* $y_k = 1$ **do**
     - $\mathbf{u} \leftarrow \nabla f_\tau(\mathbf{x}; \mathbf{w}) - \sum_{\mathbf{v} \in \mathcal{S}_J} \text{proj}_{\mathbf{v}}(\nabla f_\tau(\mathbf{x}; \mathbf{w}))$
     - $\mathcal{S}_J \leftarrow \mathcal{S}_J \bigcup \{\mathbf{u}\}$
   - **end**
   - Sample $\mathcal{D} \subset \mathcal{D}_\tau$;
   - Update $\mathcal{S}_D \leftarrow \mathcal{S}_D \bigcup \mathcal{D}$
   - **end**

5.2. Memorisation Property of OGD+

The key to obtaining tight generalisation bounds for OGD+ is the Lemma 3 below. It states that the training error across all previous tasks is unchanged, when training with OGD+.

Lemma 3 (Memorisation Property of OGD+) *Given a task $\mathcal{T}_\tau$, for all $\mathbf{x}_{k,i} \in D_k$, a sample from the training data of a previous task, it holds that:*

$$f^\star_\tau(\mathbf{x}_{k,i}) = f^\star_k(\mathbf{x}_{k,i}).$$

The full proof of Lemma 3 is presented in App . C.1.

5.3. Generalisation Guarantees for OGD+

Now, we state the generalisation theorem for OGD+, which provides tighter generalisation bounds in comparison with Theorem 2, for OGD.

Theorem 3 (Generalisation of OGD+ for Continual Learning) *Under the same conditions as Theorem 2, for OGD+, it holds that, for all tasks $\mathcal{T}_\tau$, within-task and outside-task generalisation error can be bounded as follows*

$$\mathcal{L}_{D_\tau}(f^\star_T) \leq \frac{\lambda}{2}\sqrt{\frac{\tilde{\mathbf{y}}_\tau^T k_T(\mathbf{X}_\tau, \mathbf{X}_\tau)^{-1}\tilde{\mathbf{y}}_\tau}{n_\tau}} + \sum_{k=1}^{T} \mathcal{O}\left(\sqrt{\frac{\tilde{\mathbf{y}}_k^T (k_k(\mathbf{X}_k, \mathbf{X}_k) + \lambda_k^2 \mathbf{I})^{-1}\tilde{\mathbf{y}}_k}{n_k}}\right) + \Delta_k,$$

where

$$\Delta_k = \mathcal{O}(\frac{1}{\sqrt{n_k}}) + 3c\sqrt{\frac{\log(2/\delta)}{2n_k}},$$
$$\phi_\tau(\mathbf{x}) = \mathbf{T}_\tau \nabla_{\mathbf{w}} f^\star_{\tau-1}(\mathbf{x}),$$
$$\tilde{\mathbf{y}}_\tau = \mathbf{y}_\tau - \mathbf{y}_{\tau-1\to\tau},$$
$$\mathbf{y}_{\tau-1\to\tau} = f^\star_{\tau-1}(\mathbf{X}_\tau).$$

The generalisation bounds of Theorem 3 are tighter than the generalisation bounds for OGD. The tightness of the bounds is a consequence of Lemma 3. The term that corresponds to the Rademacher complexity is unchanged, while the term that bounds the training error is tighter. It is also tighter than a standard supervised learning bound, because it captures the transferability of knowledge across tasks through the residual $\tilde{\mathbf{y}}_\tau$, as opposed to the Supervised Learning only bounds, which would depend on $\mathbf{y}_\tau$ instead.

Proof Sketch The full proof is presented in App. C.2. The proof is based on Lemma 3 and Lemma 4. The proof techniques are the same as the ones for Theorem 2.

6. Experiments

We perform experiments on the continual learning benchmarks Permuted MNIST, Rotated MNIST, Split MNIST (Farajtabar et al., 2019), and Split CIFAR-100 benchmark (Chaudhry et al., 2019). In order to assess the robustness to catastrophic forgetting over long tasks sequences, we increase the length of the tasks streams from 5 to 15 for the MNIST benchmarks, and consider all the 20 tasks for the Split CIFAR-100 benchmark.

Baselines We consider SGD and OGD as baselines for comparison. SGD draws a lower bound over the performance as it doesn't explicitly retain information in Continual Learning settings. OGD is an important baseline, as the analysis shows that OGD+ is more robust to Catastrophic Forgetting than OGD in theory and that OGD achieves state-of-art results on Continual Learning benchmarks.

Setup We keep the same neural network architecture and mostly similar hyperparameters as Farajtabar et al. 2019. We use a single-head MLP for Permuted MNIST and Rotated MNIST, a multi-head MLP for Split MNIST and a multi-head LeNet (Lecun et al., 1998) for Split CIFAR-100. We keep a small learning rate of 10^{-2} in order to preserve the locality assumption. Since the tasks streams we consider are large, we set the memory size to 1.000. Reproducibility details are presented in App. E.1.

6.1. Permuted MNIST

Permuted MNIST (Goodfellow et al., 2014) consists of a series of MNIST supervised learning tasks, where the pixels of each task are permuted with respect to a fixed permutation. The permutations are sampled independently in order for the tasks to be equally hard. Therefore, the benchmark is a proxy for the effetiveness of the algorithms to *remember* samples from the previous tasks without any without knowledge transfer between the tasks.

The results are reported in Table 1 and Table 8 (App. E.4.1). We find that OGD+ significantly outperforms OGD and SGD on the oldest tasks. While OGD either outperforms or is equivalent to OGD+ on the most recent tasks.

	Accuracy ± Std. (%)				
	Task 1	**Task 2**	**Task 3**	**Task 4**	**Task 5**
OGD+	**75.5±0.5**	**81.8±2.3**	**81.4±0.6**	**85.4±2.1**	**86.5±1.2**
OGD	37.7±5.6	72.6±4.3	74.2±3.7	81.7±1.9	82.8±1.3
SGD	56.5±4.6	49.5±5.9	57.6±3.9	60.9±2.4	75.1±3.3

Table 1. Permuted MNIST : The test accuracy of models from the indicated task after being trained on all 15 tasks in sequence. The best continual learning results are highlighted in **bold**.

6.2. Rotated MNIST

Rotated MNIST (Farajtabar et al., 2019) consists of a series of MNIST classification tasks, where the images are rotated with respect to a fixed angle, monotonically. We increment the rotation angle by 5 degrees at each new task.

The results are reported in Table 2 and Table 9 (App. E.4.2). We find that OGD+ significantly outperforms OGD and SGD on the oldest tasks, and that OGD and OGD+ become equivalent the more recent the tasks are.

	Accuracy ± Std. (%)				
	Task 1	**Task 2**	**Task 3**	**Task 4**	**Task 5**
OGD+	**50.4±2.5**	**52.5±1.9**	**60.4±1.6**	**67.6±1.9**	**73.1±1.8**
OGD	41.4±2.0	44.3±1.5	51.5±2.2	59.8±1.6	66.9±0.7
SGD	31.4±0.7	34.2±0.7	40.2±0.6	47.2±0.5	55.3±0.5

Table 2. Rotated MNIST : The test accuracy of models from the indicated task after being trained on all 15 tasks in sequence. The best continual learning results are highlighted in **bold**.

6.3. Split MNIST

Split MNIST (Zenke et al., 2017) consists of five binary classification tasks built from MNIST. The labels are split into the sets 0/1, 2/3, 4/5, 6/7 and 8/9. The results are reported in Table 3. We see that OGD+ outperforms OGD and SGD on the oldest task, OGD and OGD+ are equivalent on the two subsequent tasks, then SGD, OGD and OGD+ are almost equivalent on the remaining most recent tasks.

	Accuracy ± Std. (%)				
	Task 1	**Task 2**	**Task 3**	**Task 4**	**Task 5**
OGD+	**97.1±0.3**	**99.0±0.2**	**98.7±0.2**	99.5±0.1	96.9±0.1
OGD	96.5±0.3	**99.1±0.2**	**98.8±0.2**	99.5±0.1	96.8±0.1
SGD	85.7±3.2	98.3±0.5	98.4±0.2	99.4±0.1	**97.0±0.1**

Table 3. Split MNIST : The test accuracy of models from the indicated task after being trained on all 5 tasks in sequence. The best continual learning results are highlighted in **bold**.

6.4. Split CIFAR-100

Split CIFAR-100 (Chaudhry et al., 2019) is constructed by splitting the original CIFAR-100 dataset (Krizhevsky, 2009) into 20 disjoint subsets, where each subset is formed by sampling without replacement 5 classes out of 100.

The results are reported in Table 4 and Table 10 (App. E.4.3). We find that OGD+ outperforms OGD *on average* on the oldest tasks. However, the performance difference is not statistically significant overall. One probable reason is that for the Split CIFAR-100 benchmark, it is likely that independent tasks would share multiple features, therefore the forgetting component is less isolated by the benchmark compared to the benchmarks presented in the previous sections. Therefore, even over long tasks sequences, relevant features may occur multiple times in intermediate tasks, in which case, OGD+ may not have a significant edge over OGD.

In order to verify this hypothesis, we track the dynamics of SGD, OGD and OGD+ on the 8 first tasks (Fig 1, App. E.2). We observe that backward transfer occurs frequently during training, as there are performance improvement spikes on old tasks even though the model is not trained on them specifically. Also, as opposed to the previous benchmarks, the performance margin with SGD is much smaller.

	Accuracy ± Std. (%)				
	Task 1	**Task 2**	**Task 3**	**Task 4**	**Task 5**
OGD+	**62.1±3.2**	69.3±4.3	**75.8±1.4**	**67.9±2.4**	**70.0±3.8**
OGD	**61.6±2.0**	70.0±5.2	**75.4±2.2**	**66.5±1.9**	**70.0±5.1**
SGD	50.7±4.9	66.1±6.0	67.2±5.7	61.6±3.7	59.9±6.0

Table 4. Split CIFAR-100 : The test accuracy of models from the indicated task after being trained on all 20 tasks in sequence. The best continual learning results are highlighted in **bold**.

6.5. Summary

Overall, we find that OGD and OGD+ outperform SGD on all benchmarks. OGD+ outperforms OGD when the task's relevant features' occurrence is the furthest away in the past. Otherwise, OGD and OGD+ are mostly equivalent. These results concur with Lemma 6 which states that OGD is robust to catastrophic forgetting up to a single task ahead, and Lemma 3, which states that OGD+ is robust to catastrophic forgetting across any number of tasks. Also, two probable reasons OGD+ is not perfectly prone to catastrophic forgetting in the experiments are the memory limit and the non-overparameterization of the neural network. We discuss more in detail the limits of OGD+ in Sec. D.

7. Related works

Continual Learning Approaches to Continual Learning can be categorised into: regularization methods, memory based methods, and dynamic architectural methods. We refer the reader to the survey (Parisi et al., 2019) for an extensive overview on the existing methods. The idea behind memory-based methods is to store data from previous tasks in a buffer of fixed size, which can then be reused during training on the current task (Chaudhry et al. 2019, Van de Ven & Tolias 2018). Dynamic architectural methods rely on growing architectures which keep the past knowledge fixed and store new knowledge in new components, such as new nodes, layers ... (Lee et al. 2018, Schwarz et al. 2018) The idea behind regularization methods is to regularize the objective in order to preserve the knowledge acquired from the previous tasks (Kirkpatrick et al. 2016, Aljundi et al. 2018, Farajtabar et al. 2019, Zenke et al. 2017).

Catastrophic Forgetting refers to the tendency of agents to "forget" the previous tasks over the course of training. It has proven to be a challenging problem, several heuristics were developed in order to characterise it (Ans & Rousset 1997, Ans & Rousset 2000, Goodfellow et al. 2014, French 1999, McCloskey & Cohen 1989, Robins 1995, Nguyen et al. 2019).

Several works have also recently started studying Transfer Learning from a theoretical perspective (Yu et al. 2020, Liu et al. 2019, Nguyen et al. 2019, Achille et al. 2019).

Deep Learning Theory Recent works have started to provide explanations about the mechanics of overparametrised Neural Networks. In their seminal work, Du et al. (2018) prove that Gradient Descent on multilayer overparametrised RELU neural networks achieve zero training error at the limit. These works have unlocked the analysis of several properties of Deep Neural Networks, in the context of various applications, such as Transfer Learning (Liu et al., 2019), Noisy Supervision (Hu et al., 2019), Reinforcement Learning (Wang et al., 2020) ... Another line of works provide closed form expressions of the training dynamics of overparameterized neural networks, leveraging tools from statistical physics (Goldt et al. 2019, Goldt et al. 2020).

Statistical Learning Theory Alquier et al. (2017) define a compound regret for lifelong learning, as the regret with respect to the oracle who would have known the best common representation g for all tasks in advance. Another line of works addresses lifelong learning and meta-learning from a statistical learning theory perspective, they define and provide regret bounds for lifelong learning (Alquier et al., 2017), or study the sample complexity and convergence of meta-learning algorithms (Denevi et al. 2018a, Du et al. 2020, Ji et al. 2020, Saunshi et al. 2020, Denevi et al. 2018b).

8. Conclusion

We presented several theoretical results on the convergence and generalisation properties of OGD and OGD+ for Continual Learning, then showed that these results are applicable to practical settings. We also found theoretical connections to Transfer Learning and Curriculum Learning.

Firstly, as opposed to OGD, the memory requirement of OGD+ scales quadratically with the number of samples. Finding a way to prioritize the feature maps to store is critical to scale to larger tasks and model architectures. Another important direction for future investigation is a theoretical analysis of the other Continual Learning training methods and catastrophic forgetting heuristics, such as the ones presented by Nguyen et al. (2019). Even for asymptotic cases, the analysis may provide insights on their properties, limits and directions of improvement. Also, it would be interesting to investigate the applicability of the theory to neighbouring fields such as meta-learning, multi-task learning ... We hope this analysis provides new keys to investigate these directions.

References

Achille, A., Lam, M., Tewari, R., Ravichandran, A., Maji, S., Fowlkes, C., Soatto, S., and Perona, P. Task2vec: Task embedding for meta-learning, 02 2019.

Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., and Tuytelaars, T. Memory aware synapses: Learning what (not) to forget. In *ECCV*, 2018.

Alquier, P., Mai, T. T., and Pontil, M. Regret Bounds for Lifelong Learning. In Singh, A. and Zhu, J. (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 261–269, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR.

Ans, B. and Rousset, S. Avoiding catastrophic forgetting by coupling two reverberating neural networks. *Comptes Rendus de l'Académie des Sciences - Series III - Sciences de la Vie*, 320(12):989 – 997, 1997.

Ans, B. and Rousset, S. Neural networks with a self-refreshing memory: Knowledge transfer in sequential learning tasks without catastrophic forgetting. *Connection Science*, 12:1–19, 04 2000. doi: 10.1080/095400900116177.

Arora, S., Du, S. S., Hu, W., Li, Z., and Wang, R. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. *CoRR*, abs/1901.08584, 2019.

az Rodr í guez, N. D. i., Lomonaco, V., Filliat, D., and Maltoni, D. Don't forget, there is more than forgetting: new metrics for continual learning. *CoRR*, abs/1810.13166, 2018.

Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *J. Mach. Learn. Res.*, 3:463–482, March 2003.

Ben-David, S. and Schuller, R. Exploiting task relatedness for multiple task learning. In Schölkopf, B. and Warmuth, M. K. (eds.), *Learning Theory and Kernel Machines*, pp. 567–580, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. ISBN 978-3-540-45167-9.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 41–48, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553380.

Bietti, A. and Mairal, J. On the inductive bias of neural tangent kernels. In *NeurIPS*, 2019.

Chaudhry, A., Ranzato, M., Rohrbach, M., and Elhoseiny, M. Efficient lifelong learning with a- GEM . In *International Conference on Learning Representations*, 2019.

Denevi, G., Ciliberto, C., Stamos, D., and Pontil, M. Incremental learning-to-learn with statistical guarantees, 2018a.

Denevi, G., Ciliberto, C., Stamos, D., and Pontil, M. Learning to learn around a common mean. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 10169–10179. Curran Associates, Inc., 2018b.

Dicker, L. H., Foster, D. P., and Hsu, D. Kernel ridge vs. principal component regression: Minimax bounds and the qualification of regularization operators. *Electron. J. Statist.*, 11(1):1022–1047, 2017. doi: 10.1214/17-EJS1258.

Du, S. S., Zhai, X., s P ó czos, B. a., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. *CoRR*, abs/1810.02054, 2018.

Du, S. S., Hu, W., Kakade, S. M., Lee, J. D., and Lei, Q. Few-shot learning via learning the representation, provably. *ArXiv*, abs/2002.09434, 2020.

Ebrahimi, S., Elhoseiny, M., Darrell, T., and Rohrbach, M. Uncertainty-guided continual learning with bayesian neural networks. In *International Conference on Learning Representations*, 2020.

Fang, C., Dong, H., and Zhang, T. Over parameterized two-level neural networks can learn near optimal feature representations, 2019.

Farajtabar, M., Azizan, N., Mott, A., and Li, A. Orthogonal gradient descent for continual learning. *ArXiv*, abs/1910.07104, 2019.

French, R. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3:128–135, 05 1999. doi: 10.1016/S1364-6613(99)01294-2.

Goldt, S., Advani, M., Saxe, A. M., Krzakala, F., and Zdeborov´ a , L. Dynamics of stochastic gradient descent for two-layer neural networks in the teacher-student setup. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 6979–6989. Curran Associates, Inc., 2019.

Goldt, S., zard, M. M. e., Krzakala, F., and á , L. Z. Modelling the influence of data structure on learning in neural networks, 2020.

Goodfellow, I. J., Mirza, M., Da, X., Courville, A. C., and Bengio, Y. An empirical investigation of catastrophic forgeting in gradient-based neural networks. *CoRR*, abs/1312.6211, 2014.

Hayou, S., Doucet, A., and Rousseau, J. Mean-field behaviour of neural tangent kernel for deep neural networks, 2019.

Hu, W., Li, Z., and Yu, D. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee, 2019.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. URL http://arxiv.org/abs/1502.03167.

Jacot, A., Gabriel, F., and Hongler, C. e. m. Neural tangent kernel: Convergence and generalization in neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 8580–8589, Red Hook, NY, USA, 2018. Curran Associates Inc.

Ji, K., Yang, J., and Liang, Y. Multi-step model-agnostic meta-learning: Convergence and improved algorithms, 2020.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114, 12 2016. doi: 10.1073/pnas.1611835114.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114.

Krizhevsky, A. Learning multiple layers of features from tiny images. 2009.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pp. 2278–2324, 1998.

Lee, J., Yoon, J., Yang, E., and Hwang, S. J. Lifelong learning with dynamically expandable networks. *ArXiv*, abs/1708.01547, 2018.

Lee, J., Xiao, L., Schoenholz, S. S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent, 2019.

Li, M., Soltanolkotabi, M., and Oymak, S. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. *CoRR*, abs/1903.11680, 2019.

Liu, H., Long, M., Wang, J., and Jordan, M. I. Towards understanding the transferability of deep representations, 2019.

Masse, N., Grant, G., and Freedman, D. Alleviating catastrophic forgetting using context-dependent gating and synaptic stabilization. *Proceedings of the National Academy of Sciences*, 115, 02 2018. doi: 10.1073/pnas.1803839115.

McCloskey, M. and Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24: 109–165, 1989.

Nguyen, C. V., Li, Y., Bui, T. D., and Turner, R. E. Variational continual learning. In *International Conference on Learning Representations*, 2018.

Nguyen, C. V., Achille, A., Lam, M., Hassner, T., Mahadevan, V., and Soatto, S. Toward understanding catastrophic forgetting in continual learning. *ArXiv*, abs/1908.01091, 2019.

Pan, P., Immer, A., Swaroop, S., Eschenhagen, R., Turner, R. E., and Khan, M. E. Continual deep learning by functional regularisation of memorable past, 2020.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54 – 71, 2019.

Ratcliff, R. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97 2:285–308, 1990.

Ritter, H., Botev, A., and Barber, D. Online structured laplace approximations for overcoming catastrophic forgetting. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 3742–3752, Red Hook, NY, USA, 2018. Curran Associates Inc.

Robins, A. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7:123–, 01 1995.

Saunshi, N., Zhang, Y., Khodak, M., and Arora, S. A sample complexity separation between non-convex and convex meta-learning, 2020.

Schwarz, J., Luketina, J., Czarnecki, W., Grabska-Barwinska, A., Teh, Y., Pascanu, R., and Hadsell, R. Progress and compress: A scalable framework for continual learning, 05 2018.

Serra, J., Suris, D., Miron, M., and Karatzoglou, A. Overcoming catastrophic forgetting with hard attention to the task. *CoRR*, abs/1801.01423, 2018.

Smola, A. J., Schö lkopf, B., and Mü ller, K.-R. The connection between regularization operators and support vector kernels. *Neural Netw.*, 11(4):637–649, June 1998. doi: 10.1016/S0893-6080(98)00032-X.

Van de Ven, G. M. and Tolias, A. S. Generative replay with feedback connections as a general strategy for continual learning. *arXiv preprint arXiv:1809.10635*, 2018.

Van de Ven, G. M. and Tolias, A. S. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

Wang, R., Du, S. S., Yang, L. F., and Kakade, S. M. Is long horizon reinforcement learning more difficult than short horizon reinforcement learning?, 2020.

Wei, C., Lee, J. D., Liu, Q., and Ma, T. Regularization matters: Generalization and optimization of neural nets v.s. their induced kernel. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alche Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 9709–9721. Curran Associates, Inc., 2019.

Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., and Finn, C. Gradient surgery for multi-task learning, 2020.

Zenke, F., Poole, B., and Ganguli, S. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 3987–3995. JMLR.org, 2017.

A. Missing proofs of section 3 - Convergence

A.1. Proof of Theorem 1

Orthogonal Gradient Descent Proof

We prove the Theorem 1 by induction. Our induction hypothesis H_τ is the following :

H_τ : For all $k \leq \tau$, Theorem 1 holds.

First, we prove that H_1 holds.

The proof is straightforward. For the first task, since there were no previous tasks $\mathbb{E}_1 = \emptyset$. Therefore, OGD on this task is equivalent to SGD.

Therefore, it is equivalent to minimising the following objective, where $\tau = 1$:

$$\underset{\mathbf{w}\in\mathbb{R}^d}{\arg\min}\left\|\phi_\tau(\mathbf{X}_\tau)^T(\mathbf{w}-\mathbf{w}_\tau^\star)-\tilde{\mathbf{y}}_{\tau+1}\right\|_2^2+\lambda_\tau\|\mathbf{w}-\mathbf{w}_0\|^2$$

The objective is quadratic and the Hessian is positive definite, therefore the minimum exists and is unique :

$$\mathbf{w}_\tau^\star-\mathbf{w}_0=\phi_\tau(\mathbf{X}_\tau)(\phi_\tau(\mathbf{X}_\tau)^T\phi_\tau(\mathbf{X}_\tau)+\lambda_\tau^2 I)^{-1}\tilde{\mathbf{y}}_\tau$$

For $\tau = 1$, since there are no previous tasks $\tilde{\mathbf{y}}_\tau = \mathbf{y}_\tau$. Therefore :

$$\mathbf{w}_\tau^\star-\mathbf{w}_0=k_\tau(\mathbf{x},\mathbf{X}_\tau)(k_\tau(\mathbf{X}_\tau,\mathbf{X}_\tau)+\lambda_\tau^2\mathbf{I})^{-1}\tilde{\mathbf{y}}_\tau$$

Which completes the proof of H_1.

Let $\tau \in \mathbb{N}^\star$, assume H_τ is true, we show $H_{\tau+1}$

On the task $\tau + 1$, we can write the loss $\mathcal{L}^{\tau+1}$ as :

$$\mathcal{L}^{\tau+1}(\mathbf{w}_{\tau+1}(t))=\left\|\phi_\tau(\mathbf{X}_\tau)^T(\mathbf{w}_{\tau+1}(t)-\mathbf{w}_\tau^\star)-\tilde{\mathbf{y}}_{\tau+1}\right\|_2^2+\lambda_{\tau+1}\|\mathbf{w}_{\tau+1}(t)-\mathbf{w}_\tau^\star\|^2$$

We recall that the optimisation problem at time $(\tau + 1)$:

$$\begin{aligned}&\underset{\mathbf{w}\in\mathbb{R}^d}{\arg\min}\left\|\phi_\tau(\mathbf{X}_\tau)^T(\mathbf{w}-\mathbf{w}_\tau^\star)-\tilde{\mathbf{y}}_{\tau+1}\right\|_2^2+\lambda_{\tau+1}\|\mathbf{w}-\mathbf{w}_\tau^\star\|^2\\ &\text{u.c.}\quad \mathbf{V}_{\tau+1}(\mathbf{w}-\mathbf{w}_\tau^\star)=0\end{aligned}$$

Let $\mathbf{T}_{\tau+1} \in \mathbb{R}^{d\times(d-K_{\tau+1})}$ and $\tilde{\mathbf{w}}_{\tau+1} \in \mathbb{R}^{d-K_{\tau+1}}$ such as :

$$\begin{aligned}\mathbf{w}-\mathbf{w}_\tau^\star&=\mathbf{T}_{\tau+1}\tilde{\mathbf{w}}_{\tau+1}\\ K_{\tau+1}&=\dim(\mathbb{E}_{\tau+1})\end{aligned}$$

We rewrite the objective by plugging in the variables we just defined. The two objectives are equivalent :

$$\underset{\tilde{\mathbf{w}}\in\mathbb{R}^{d-K_{\tau+1}}}{\arg\min}\left\|\phi_\tau(\mathbf{X}_\tau)^T\mathbf{T}_{\tau+1}\tilde{\mathbf{w}}-\tilde{\mathbf{y}}_{\tau+1}\right\|_2^2+\lambda_{\tau+1}\|\mathbf{T}_{\tau+1}\tilde{\mathbf{w}}\|_2^2$$

For clarity, we define $\mathbf{Z}_{\tau+1}\mathbb{R}^{n_{\tau+1}\times(d-K_{\tau+1})}$ as :

$$\mathbf{Z}_{\tau+1}=\phi_\tau(\mathbf{X}_\tau)^T\mathbf{T}_{\tau+1}$$

By plugging in $\mathbf{Z}_{\tau+1}$, we rewrite the objective as :

$$\underset{\tilde{\mathbf{w}}\in\mathbb{R}^{d-K_{\tau+1}}}{\arg\min} \left\|\mathbf{Z}_{\tau+1}\tilde{\mathbf{w}} - \tilde{\mathbf{y}}_{\tau+1}\right\|_2^2 + \lambda_{\tau+1}\|\mathbf{T}_{\tau+1}\tilde{\mathbf{w}}\|_2^2$$

The optimisation objective is quadratic, unconstrainted, with a positive definite hessian. Therefore, an optimum exists and is unique :

$$\tilde{\mathbf{w}}^\star_{\tau+1} = \mathbf{Z}^T_{\tau+1}(\mathbf{Z}_{\tau+1}\mathbf{Z}^T_{\tau+1} + \lambda^2_{\tau+1}I)^{-1}\tilde{\mathbf{y}}_{\tau+1}$$

We recover the expression of the optimum in the original space :

$$\mathbf{w}^\star_{\tau+1} - \mathbf{w}^\star_\tau = \mathbf{T}_{\tau+1}\mathbf{Z}^T_{\tau+1}(\mathbf{Z}_{\tau+1}\mathbf{Z}^T_{\tau+1} + \lambda^2_{\tau+1}I)^{-1}\tilde{\mathbf{y}}_{\tau+1}$$

We define the kernel $k_{\tau+1} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ as :

$$k_{\tau+1}(\mathbf{x}, \mathbf{x}') = \phi_\tau(\mathbf{x})^T\mathbf{T}_{\tau+1}\mathbf{T}^T_{\tau+1}\phi_\tau(\mathbf{x}') \quad \text{for all } \mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$$

Now we rewrite $\mathbf{w}^\star_{\tau+1} - \mathbf{w}^\star_\tau$:

$$\mathbf{w}^\star_{\tau+1} - \mathbf{w}^\star_\tau = \mathbf{T}_{\tau+1}\mathbf{Z}^T_{\tau+1}(k_{\tau+1}(\mathbf{X}_\tau, \mathbf{X}_\tau) + \lambda^2_{\tau+1}I)^{-1}\tilde{\mathbf{y}}_{\tau+1}$$

Finally, we recover a closed form expression for $f^\star_{\tau+1}$:

First, we use the induction hypothesis H_τ :

$$\begin{aligned} f^\star_{\tau+1}(\mathbf{x}) &= f^\star_\tau(\mathbf{x}) + \langle \nabla_{\mathbf{w}} f^\star_\tau(\mathbf{x}), \mathbf{w}^\star_{\tau+1} - \mathbf{w}^\star_\tau\rangle \\ &= f^\star_\tau(\mathbf{x}) + \phi_\tau(\mathbf{x})\mathbf{T}_{\tau+1}\mathbf{Z}^T_{\tau+1}(k_{\tau+1}(\mathbf{X}_\tau, \mathbf{X}_\tau) + \lambda^2_{\tau+1}I)^{-1}\tilde{\mathbf{y}}_{\tau+1} \\ &= f^\star_\tau(\mathbf{x}) + k_{\tau+1}(\mathbf{x}, \mathbf{X}_\tau)(k_{\tau+1}(\mathbf{X}_\tau, \mathbf{X}_\tau) + \lambda^2_{\tau+1}I)^{-1}\tilde{\mathbf{y}}_{\tau+1} \end{aligned}$$

At this stage, we have proven H_{t+1}.

We conclude.

Stochastic Gradient Descent The proof is exactly the same as the proof for Orthogonal Gradient Descent, except that there are no equalities constraints.

A.2. Proof of the Corollary 2

Orthogonal Gradient Descent Proof

In the proof of Theorem 1 (App. A.1), we proved that :

$$\mathbf{w}^\star_{\tau+1} - \mathbf{w}^\star_\tau = \mathbf{T}_{\tau+1}\mathbf{Z}^T_{\tau+1}(k_{\tau+1}(\mathbf{X}_\tau, \mathbf{X}_\tau) + \lambda^2_{\tau+1}I)^{-1}\tilde{\mathbf{y}}_{\tau+1}$$

Therefore :

$$\begin{aligned} \left\|\mathbf{w}^\star_{\tau+1} - \mathbf{w}^\star_\tau\right\|^2 &= \tilde{\mathbf{y}}^T_{\tau+1}(k_{\tau+1}(\mathbf{X}_\tau, \mathbf{X}_\tau) + \lambda^2_{\tau+1}\mathbf{I})^{-1}\mathbf{Z}_{\tau+1}\mathbf{T}^T_{\tau+1}\mathbf{T}_{\tau+1}\mathbf{Z}^T_{\tau+1}(k_{\tau+1}(\mathbf{X}_\tau, \mathbf{X}_\tau) + \lambda^2_{\tau+1}I)^{-1}\tilde{\mathbf{y}}_{\tau+1} \\ &= \tilde{\mathbf{y}}^T_{\tau+1}(k_{\tau+1}(\mathbf{X}_\tau, \mathbf{X}_\tau) + \lambda^2_{\tau+1}\mathbf{I})^{-1}k_{\tau+1}(\mathbf{X}_\tau, \mathbf{X}_\tau)(k_{\tau+1}(\mathbf{X}_\tau, \mathbf{X}_\tau) + \lambda^2_{\tau+1}I)^{-1}\tilde{\mathbf{y}}_{\tau+1} \end{aligned}$$

Stochastic Gradient Descent The proof is exactly the same as for Orthogonal Gradient Descent.

B. Missing proofs of section 4 - Generalisation

B.1. Proof of Theorem 2

B.1.1. Notations

We recall that :

$$f_\tau^\star(\mathbf{x}) = \sum_{k=1}^{\tau-1} f_k^\star(\mathbf{x}) + k_\tau(\mathbf{x}, \mathbf{X}_\tau)^T (k_\tau(\mathbf{X}_\tau, \mathbf{X}_\tau) + \lambda_\tau^2 \mathbf{I})^{-1} \tilde{\mathbf{y}}_\tau$$

We define :

$$\tilde{f}_\tau^\star(\mathbf{x}) = k_\tau(\mathbf{x}, \mathbf{X}_\tau)^T \boldsymbol{\alpha}_\tau$$

where :

$$\boldsymbol{\alpha}_\tau = (k_\tau(\mathbf{X}_\tau, \mathbf{X}_\tau) + \lambda_\tau^2 \mathbf{I})^{-1} \tilde{\mathbf{y}}_\tau$$

Then :

$$f_\tau^\star(\mathbf{x}) = \sum_{k=1}^{\tau} \tilde{f}_\tau^\star(\mathbf{x})$$

Reminder on RKHS norm

Let k a kernel, and $\mathcal{H}$ the reproducing kernel Hilbert space (RKHS) corresponding to the kernel k.

Recall that the RKHS norm of a function $f(\mathbf{x}) = \alpha^T k(\mathbf{x}, \mathbf{X})$ is :

$$\|f\|_{\mathcal{H}} = \sqrt{\alpha^T k(\mathbf{X}, \mathbf{X}) \alpha}$$

Reminder on Generalization and Rademacher Complexity Consider a loss function $l : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. The population loss over the distribution $\mathcal{D}$, and the empirical loss over n samples $D = \{(\mathbf{x}_i, y_i), i \in [n]\}$ from the same distribution $\mathcal{D}$ are defined as :

$$L_D(f) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[l(f(\mathbf{x}), y)]$$
$$L_S(f) = \frac{1}{n} \sum_{i=1}^{n} l(f(\mathbf{x}_i), y_i)$$

Theorem 4 *Suppose the loss function is bounded in $[0, c]$ and is $\rho-$Lipchitz in the first argument. Then, with probability at least $1 - \delta$ over sample S of size n :*

$$\sup_{f\in\mathcal{F}} \{L_D(f) - L_S(f)\} \leq 2\rho \hat{\mathcal{R}}(\mathcal{F}) + 3c\sqrt{\frac{\log(2/\delta)}{2n}} \tag{2}$$

B.1.2. Bounding the Rademacher Complexity

Lemma 4 (Rademacher Complexity of a linear combination of kernels) *Let $k_t : \mathcal{X} \times \mathcal{X} \to \mathbb{R}, t \in [T]$ kernels such that :*

$$\sup_{\mathbf{x}\in\mathcal{X}} \|k_t(\mathbf{x}, \mathbf{x})\| < \infty$$

To every kernel k_t, we associate a feature map $\phi_t : \mathcal{X} \to \mathcal{H}_t$, where $\mathcal{H}_t$ is a Hilbert space with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_t}$, and for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, $k_t(\mathbf{x}, \mathbf{x}') = \langle \phi_t(\mathbf{x}), \phi_t(\mathbf{x}') \rangle_{\mathcal{H}_t}$

We define $\mathcal{F}$ as follows :

$$\mathcal{F} = \{\mathbf{x} \to \sum_{t=1}^{T} f_t(\mathbf{x}), \quad f_t(\mathbf{x}) = \alpha_t^T k_t(\mathbf{x}, \mathbf{X}_t) \quad \forall t \in [T], \|f_t\|_{\mathcal{H}_t} \leq B_t\} \tag{3}$$

Let $X_1, ..., X_n$ be random elements of $\mathcal{X}$. Then for the class $\mathcal{F}$, we have :

$$\hat{\mathcal{R}}(\mathcal{F}) \leq \sum_{t=1}^{T} \frac{2B_t}{n_t} (Tr(k_t(\mathbf{X}_t, \mathbf{X}_t)))^{1/2}$$

Proof

Let $f \in \mathcal{F}$, and let $\mathbf{x} \in \mathcal{X}$:

$$f(\mathbf{x}) = \sum_{t=1}^{T} \sum_{i=1}^{n_t} \alpha_i^t k_t(\mathbf{x}, \mathbf{x}_i^t)$$

For all $t \in [T]$, we associate a feature map $\phi_t : \mathbf{X} \to \mathcal{H}_t$

$\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X} \quad k_t(\mathbf{x}, \mathbf{x}') = \langle \phi_t(\mathbf{x}), \phi_t(\mathbf{x}') \rangle_{\mathcal{H}_t}$

Therefore :

$$\begin{aligned} f(\mathbf{x}) &= \sum_{t=1}^{T} \sum_{i=1}^{n_t} \alpha_i^t \langle \phi_t(\mathbf{x}_i^t), \phi_t(\mathbf{x}) \rangle_{\mathcal{H}_t} \\ &= \sum_{t=1}^{T} \langle \sum_{i=1}^{n_t} \alpha_i^t \phi_t(\mathbf{x}_i^t), \phi_t(\mathbf{x}) \rangle_{\mathcal{H}_t} \end{aligned}$$

On the other hand, the following holds $\forall t \in [T]$:

$$\left\| \sum_{i=1}^{n_t} \alpha_i^t \phi_t(\mathbf{x}_i^t) \right\|_{\mathcal{H}_t}^2 = \sum_{i,j} \alpha_i^t \alpha_j^t k_t(\mathbf{x}_i^t, \mathbf{x}_j^t) \leq B_t^2$$

Therefore :

$$\mathcal{F} \subset \{\mathbf{x} \to \sum_{t=1}^{T} \langle \mathbf{w}_t, \phi_t(\mathbf{x}) \rangle_{\mathcal{H}_t}, \|\mathbf{w}_t\|_2 \leq B_t \quad \forall t \in [T]\} := \tilde{\mathcal{F}}$$

Now, we derive an upper bound of the Rademacher complexity of $\mathcal{F}$:

$$\begin{aligned} \hat{\mathcal{R}}(\mathcal{F}) &\leq \hat{\mathcal{R}}(\tilde{\mathcal{F}}) \\ &= \mathbb{E}[\sup_{\|\mathbf{w}_t\|_2 \leq B_t, t \in [T]} \sum_{t=1}^{T} \langle \mathbf{w}_t, \frac{2}{n_t} \sum_{i=1}^{n_t} \epsilon_i \phi_t(\mathbf{x}_i^t) \rangle_{\mathcal{H}_t} | (\mathbf{X}_t)] \\ &= \sum_{t=1}^{T} \mathbb{E}[\sup_{\|\mathbf{w}_t\|_2 \leq B_t} \langle \mathbf{w}_t, \frac{2}{n_t} \sum_{i=1}^{n_t} \epsilon_i \phi_t(\mathbf{x}_i^t) \rangle_{\mathcal{H}_t} | (\mathbf{X}_t)] \\ &\leq \sum_{t=1}^{T} \frac{2B_t}{n_t} (Tr(k_t(\mathbf{X}_t, \mathbf{X}_t)))^{1/2} \end{aligned}$$

The last inequality is obtained by applying the upper bound from Lemma 22 in (Bartlett & Mendelson, 2003), on each function f_t

B.1.3. Bounding $\left\|\tilde{f}_\tau^\star\right\|_{\mathcal{H}_\tau}$:

Lemma 5 *Let $\mathcal{H}_\tau$ the Hilbert space associated to the kernel k_τ .*

We recall that :

$$\tilde{f}_\tau^\star(\mathbf{x}) = k_\tau(\mathbf{x}, \mathbf{X}_\tau)^T \boldsymbol{\alpha}_\tau$$
$$\boldsymbol{\alpha}_\tau = (k_\tau(\mathbf{X}_\tau, \mathbf{X}_\tau) + \lambda_\tau^2 \mathbf{I})^{-1} \tilde{\mathbf{y}}_\tau$$

Then :

$$\left\|\tilde{f}_\tau^\star\right\|_{\mathcal{H}_\tau}^2 \leq \tilde{\mathbf{y}}_\tau^T (k_\tau(\mathbf{X}_\tau, \mathbf{X}_\tau) + \lambda^2 \mathbf{I})^{-1} \tilde{\mathbf{y}}_\tau$$

Proof

$$\begin{aligned}\left\|\tilde{f}_\tau^\star\right\|_{\mathcal{H}_\tau}^2 &= \boldsymbol{\alpha}_\tau^T k_\tau(\mathbf{X}_\tau, \mathbf{X}_\tau) \boldsymbol{\alpha}_\tau \\ &= \mathbf{y}_\tau^T (k_\tau(\mathbf{X}_\tau, \mathbf{X}_\tau) + \lambda^2 \mathbf{I})^{-1} k_\tau(\mathbf{X}_\tau, \mathbf{X}_\tau)(k_\tau(\mathbf{X}_\tau, \mathbf{X}_\tau) + \lambda^2 \mathbf{I})^{-1} \tilde{\mathbf{y}}_\tau\end{aligned}$$

Since $(k_\tau(\mathbf{X}, \mathbf{X}) + \lambda^2 \mathbf{I})^{-1} \leq k_\tau(\mathbf{X}, \mathbf{X})^{-1}$, we get :

$$\left\|\tilde{f}_\tau^\star\right\|_{\mathcal{H}_\tau}^2 \leq \tilde{\mathbf{y}}_\tau^T k_\tau(\mathbf{X}_\tau, \mathbf{X}_\tau)(k_\tau(\mathbf{X}_\tau, \mathbf{X}_\tau) + \lambda^2 \mathbf{I})^{-1} \tilde{\mathbf{y}}_\tau$$

B.1.4. Proof of Lemma 1

The intuition behind the proof is : since the gradient updates were performed orthogonally to the feature maps of the training data of the source task, the parameters in this space are unchanged, while the remaining space, which was changed, is orthogonal to these features maps, therefore, the inference is the same and the training error remanis the same as at the end of training on the source task.

Proof

$$\begin{aligned}f_T^\star(\mathbf{x}_{T-1,i}) &= f_{T-1}^\star(\mathbf{x}_{T-1,i}) + \langle \phi_T(\mathbf{x}_{T-1,i}), \mathbf{w}_T^\star - \mathbf{w}_{T-1}^\star \rangle \\ &= f_{T-1}^\star(\mathbf{x}_{T-1,i}) + \langle \nabla_{\mathbf{W}} f(\mathbf{w}_{T-1}^\star, \mathbf{x}), \mathbf{w}_T^\star - \mathbf{w}_{T-1}^\star \rangle\end{aligned}$$

Since we the training on task $\mathcal{T}_T$ is perfomed with OGD, we have :

$$\Pi_{\mathbb{E}_{T-1}}(\mathbf{w}_T^\star - \mathbf{w}_{T-1}^\star) = 0$$

Since $\nabla_{\mathbf{W}} f(\mathbf{w}_{T-1}^\star, \mathbf{x}) \in \mathbb{E}_{T-1}$ by definition, it follows that :

$$f_T^\star(\mathbf{x}_{T-1,i}) = f_{T-1}^\star(\mathbf{x}_{T-1,i})$$

B.1.5. Bounding the training error

Lemma 6 *The training errors on the source and target tasks can be bounded as follows :*

Let $T \in \mathbb{N}$ fixed. Then, for all $\tau \in [T]$

$$\frac{1}{n_T}\sum_{i=1}^{n_T}(f_T^\star(\mathbf{x}_{T,i}) - y_{T,i})^2 \leq \frac{1}{n_T}\frac{\lambda^2}{4}\tilde{\mathbf{y}}_T^T(k_T(\mathbf{X}_T,\mathbf{X}_T))^{-1}\tilde{\mathbf{y}}_T$$

$$\frac{1}{n_{T-1}}\sum_{i=1}^{n_{T-1}}(f_T^\star(\mathbf{x}_{T-1,i}) - y_{T-1,i})^2 \leq \frac{1}{n_{T-1}}\frac{\lambda^2}{4}\tilde{\mathbf{y}}_{T-1}^T(k_{T-1}(\mathbf{X}_{T-1},\mathbf{X}_{T-1}))^{-1}\tilde{\mathbf{y}}_{T-1}$$

For all $\tau \in [T-2]$

$$\frac{1}{n_\tau}\|f_T^\star(\mathbf{X}_\tau) - \mathbf{y}_\tau\|_2^2 \leq \frac{1}{n_\tau}(\frac{\lambda^2}{4}\tilde{\mathbf{y}}_\tau^T(k_\tau(\mathbf{X}_\tau,\mathbf{X}_\tau))^{-1}\tilde{\mathbf{y}}_\tau + \sum_{k=\tau+1}^{T}\frac{1}{4\lambda^2}\tilde{\mathbf{y}}_k^T k_k(\mathbf{X}_k,\mathbf{X}_k)^{-1}k(\mathbf{X}_\tau,\mathbf{X}_k)k(\mathbf{X}_\tau,\mathbf{X}_k)^T k_k(\mathbf{X}_k,\mathbf{X}_k)^{-1}\tilde{\mathbf{y}}_k)$$

Task $\mathcal{T}_T$ Proof

We start from the definition of the training error :

$$\sum_{i=1}^{n_T}(f_T^\star(\mathbf{x}_{T,i}) - y_{T,i})^2 = \left\|(k_T(\mathbf{X}_T,\mathbf{X}_T)^T(k_T(\mathbf{X}_T,\mathbf{X}_T) + \lambda^2\mathbf{I})^{-1} - \mathbf{I})\tilde{\mathbf{y}}_T\right\|_2^2$$

The expression is very similar to the previous norm, we can derive the same analysis as above to derive the following bound :

$$\sum_{i=1}^{n_T}(f_T^\star(\mathbf{x}_{T,i}) - y_{T,i})^2 \leq \frac{\lambda^2}{4}\tilde{\mathbf{y}}_T^T(k_T(\mathbf{X}_T,\mathbf{X}_T))^{-1}\tilde{\mathbf{y}}_T$$

Therefore :

$$\frac{1}{n_T}\sum_{i=1}^{n_T}(f_T^\star(\mathbf{x}_{T,i}) - y_{T,i})^2 \leq \frac{1}{n_T}\frac{\lambda^2}{4}\tilde{\mathbf{y}}_T^T(k_T(\mathbf{X}_T,\mathbf{X}_T))^{-1}\tilde{\mathbf{y}}_T$$

Task $\mathcal{T}_{T-1}$ Proof

We start with the definition of the training error, then applying Lemma 1 :

$$\begin{aligned}
\sum_{i=1}^{n_{T-1}}(f_T^\star(\mathbf{x}_{T-1,i}) - y_{T-1,i})^2 &= \sum_{i=1}^{n_{T-1}}(f_{T-1}^\star(\mathbf{x}_{T-1,i}) - y_{T-1,i})^2 \\
&= \left\|f_{T-1}^\star(\mathbf{X}_{T-1}) - \mathbf{y}_{T-1}\right\|_2^2 \\
&= \left\|k_{T-1}(\mathbf{X}_{T-1},\mathbf{X}_{T-1})^T(k_{T-1}(\mathbf{X}_{T-1},\mathbf{X}_{T-1}) + \lambda^2\mathbf{I})^{-1} - \mathbf{I})\tilde{\mathbf{y}}_{T-1}\right\|_2^2 \\
&= \left\|-\lambda^2(k_{T-1}(\mathbf{X}_{T-1},\mathbf{X}_{T-1}) + \lambda^2\mathbf{I})^{-1}\tilde{\mathbf{y}}_{T-1}\right\|_2^2 \\
&= \lambda^4\left\|(k_{T-1}(\mathbf{X}_{T-1},\mathbf{X}_{T-1}) + \lambda^2\mathbf{I})^{-1}\tilde{\mathbf{y}}_{T-1}\right\|_2^2 \\
&= \lambda^4\mathbf{y}_{T-1}^T(k_{T-1}(\mathbf{X}_{T-1},\mathbf{X}_{T-1}) + \lambda^2\mathbf{I})^{-2}\tilde{\mathbf{y}}_{T-1}
\end{aligned}$$

Since :

$$(k_{T-1}(\mathbf{X}_{T-1},\mathbf{X}_{T-1})+\lambda^2\mathbf{I})^{-2} \leq \frac{1}{4\lambda^2}k_{T-1}(\mathbf{X}_{T-1},\mathbf{X}_{T-1})^{-1}$$

We get :

$$\sum_{i=1}^{n_{T-1}}(f_T^\star(\mathbf{x}_{T-1,i})-y_{T-1,i})^2 \leq \frac{\lambda^2}{4}\tilde{\mathbf{y}}_{T-1}^T(k_{T-1}(\mathbf{X}_{T-1},\mathbf{X}_{T-1}))^{-1}\tilde{\mathbf{y}}_{T-1}$$

Therefore :

$$\frac{1}{n_{T-1}}\sum_{i=1}^{n_{T-1}}(f_T^\star(\mathbf{x}_{T-1,i})-y_{T-1,i})^2 \leq \frac{1}{n_{T-1}}\frac{\lambda^2}{4}\tilde{\mathbf{y}}_{T-1}^T(k_{T-1}(\mathbf{X}_{T-1},\mathbf{X}_{T-1}))^{-1}\tilde{\mathbf{y}}_{T-1}$$

Task $\mathcal{T}_1, ..., \mathcal{T}_{T-2}$ Proof

Let $\tau \in [T-2]$ fixed.

We recall that :

$$f_T^\star(\mathbf{x}) = f_\tau^\star(\mathbf{x}) + \sum_{k=\tau+1}^{T}\tilde{f}_k^\star(\mathbf{x})$$

Then :

$$\begin{aligned}
\|f_T^\star(\mathbf{X}_\tau)-\mathbf{y}_\tau\|_2^2 &= \left\|f_\tau^\star(\mathbf{X}_\tau)+\sum_{k=\tau+1}^{T}\tilde{f}_k^\star(\mathbf{X}_\tau)-\mathbf{y}_\tau\right\|_2^2 \\
&\leq \|f_\tau^\star(\mathbf{X}_\tau)-\mathbf{y}_\tau\|_2^2 + \left\|\sum_{k=\tau+1}^{T}\tilde{f}_k^\star(\mathbf{X}_\tau)\right\|_2^2 \\
&\leq \underbrace{\|f_\tau^\star(\mathbf{X}_\tau)-\mathbf{y}_\tau\|_2^2}_{\text{A}} + \underbrace{\sum_{k=\tau+1}^{T}\left\|\tilde{f}_k^\star(\mathbf{X}_\tau)\right\|_2^2}_{\text{B}}
\end{aligned}$$

We can upper bound A similarly to the previous paragraphs, therefore we get :

$$\|f_\tau^\star(\mathbf{X}_\tau)-\mathbf{y}_\tau\|_2^2 \leq \frac{\lambda^2}{4}\tilde{\mathbf{y}}_\tau^T(k_\tau(\mathbf{X}_\tau,\mathbf{X}_\tau))^{-1}\tilde{\mathbf{y}}_\tau$$

Now, we upper bound B. Let $k \in [\tau+1, T]$:

$$\begin{aligned}
\left\|\tilde{f}_k^\star(\mathbf{X}_\tau)\right\|_2^2 &= \tilde{\mathbf{y}}_k^T(k_k(\mathbf{X}_k,\mathbf{X}_k)+\lambda_k^2\mathbf{I})^{-1}k(\mathbf{X}_\tau,\mathbf{X}_k)k(\mathbf{X}_\tau,\mathbf{X}_k)^T(k_k(\mathbf{X}_k,\mathbf{X}_k)+\lambda_k^2\mathbf{I})^{-1}\tilde{\mathbf{y}}_k \\
&= \frac{1}{4\lambda^2}\tilde{\mathbf{y}}_k^T k_k(\mathbf{X}_k,\mathbf{X}_k)^{-1} \underbrace{k(\mathbf{X}_\tau,\mathbf{X}_k)k(\mathbf{X}_\tau,\mathbf{X}_k)^T}_{\text{Captures the similarity between the tasks } \mathcal{T}_\tau \text{ and } \mathcal{T}_k} k_k(\mathbf{X}_k,\mathbf{X}_k)^{-1}\tilde{\mathbf{y}}_k
\end{aligned}$$

We conclude by plugging back the upper bounds of A and B

$$\|f_T^\star(\mathbf{X}_\tau)-\mathbf{y}_\tau\|_2^2 \leq \frac{\lambda^2}{4}\tilde{\mathbf{y}}_\tau^T(k_\tau(\mathbf{X}_\tau,\mathbf{X}_\tau))^{-1}\tilde{\mathbf{y}}_\tau + \sum_{k=\tau+1}^{T}\frac{1}{4\lambda^2}\tilde{\mathbf{y}}_k^T k_k(\mathbf{X}_k,\mathbf{X}_k)^{-1}k(\mathbf{X}_\tau,\mathbf{X}_k)k(\mathbf{X}_\tau,\mathbf{X}_k)^T k_k(\mathbf{X}_k,\mathbf{X}_k)^{-1}\tilde{\mathbf{y}}_k$$

Therefore :

$$\frac{1}{n_\tau}\|f_T^\star(\mathbf{X}_\tau)-\mathbf{y}_\tau\|_2^2 \leq \frac{1}{n_\tau}\left(\frac{\lambda^2}{4}\tilde{\mathbf{y}}_\tau^T(k_\tau(\mathbf{X}_\tau,\mathbf{X}_\tau))^{-1}\tilde{\mathbf{y}}_\tau + \sum_{k=\tau+1}^{T}\frac{1}{4\lambda^2}\tilde{\mathbf{y}}_k^T k_k(\mathbf{X}_k,\mathbf{X}_k)^{-1}k(\mathbf{X}_\tau,\mathbf{X}_k)k(\mathbf{X}_\tau,\mathbf{X}_k)^T k_k(\mathbf{X}_k,\mathbf{X}_k)^{-1}\tilde{\mathbf{y}}_k\right)$$

B.1.6. Bounding the Rademacher Complexity

Proof

The proof strategy is exactly the same as sec. B.1. We generalize the previous proof, by applying the same lemmas.

$$\mathcal{F}_{\mathbf{B}_T} = \{\mathbf{x} \to \sum_{\tau=1}^{T} f_\tau(\mathbf{x}), \quad f_\tau(\mathbf{x}) = \boldsymbol{\alpha}_\tau^T k_\tau(\mathbf{x}, \mathbf{X}_\tau) \quad \forall \tau \in [T], \|f_\tau\|_{\mathcal{H}_\tau} \leq B_\tau\}$$

It holds that :

$$f_T^\star \in \mathcal{F}_{\mathbf{B}_T}$$

where, for all $\tau \in [T]$:

$$B_\tau = \sqrt{(\mathbf{y}_\tau - \mathbf{y}_{\tau-1\to\tau})^T (k_\tau(\mathbf{X}_\tau, \mathbf{X}_\tau) + \lambda_\tau^2 \mathbf{I})^{-1} (\mathbf{y}_\tau - \mathbf{y}_{\tau-1\to\tau})}$$

$$\hat{\mathcal{R}}(\mathcal{F}_{\mathbf{B}_T}) \leq \sum_{\tau=1}^{T} \frac{2(B_\tau + \epsilon)}{n_\tau} (Tr(k_\tau(\mathbf{X}_\tau, \mathbf{X}_\tau)))^{1/2}$$

We made the assumption that for all $\tau \in [T]$ $\mathrm{tr}(k_\tau(\mathbf{X}_\tau, \mathbf{X}_\tau)) = \mathcal{O}(n_\tau)$, also, by setting $\epsilon = 1$:

$$\begin{aligned}
\hat{\mathcal{R}}(\mathcal{F}_{\mathbf{B}_T}) &\leq \sum_{\tau=1}^{T} \frac{2(B_\tau + 1)}{n_\tau} \mathcal{O}(\sqrt{n_\tau}) \\
&\leq \sum_{\tau=1}^{T} \mathcal{O}(\frac{B_\tau}{\sqrt{n_\tau}}) + \sum_{\tau=1}^{T} \mathcal{O}(\frac{1}{\sqrt{n_\tau}}) \\
&\leq \sum_{\tau=1}^{T} \mathcal{O}\left(\sqrt{\frac{(\mathbf{y}_\tau - \mathbf{y}_{\tau-1\to\tau})^T (k_\tau(\mathbf{X}_\tau, \mathbf{X}_\tau) + \lambda^2 \mathbf{I})^{-1} (\mathbf{y}_\tau - \mathbf{y}_{\tau-1\to\tau})}{n_\tau}}\right) + \sum_{\tau=1}^{T} \mathcal{O}(\frac{1}{\sqrt{n_\tau}})
\end{aligned}$$

B.1.7. Proof of the Theorem 2

Proof

With probability $1 - \delta$ we have :

$$\sup_{f \in \mathcal{F}_{\mathbf{B}_T}} \{L_D(f) - L_S(f)\} \leq 2\rho \hat{\mathcal{R}}(\mathcal{F}_{\mathbf{B}_T}) + 3c\sqrt{\frac{\log(2/\delta)}{2n}}$$

$$\mathcal{L}_{D_\tau}(f_T^\star) \leq \mathcal{L}_{S_\tau}(f_T^\star) + 2\rho \hat{\mathcal{R}}(\mathcal{F}_{\mathbf{B}_T}) + 3c\sqrt{\frac{\log(2/\delta)}{2n_T}}$$

$$\mathcal{L}_{D_\tau}(f_T^\star) \leq \mathcal{L}_{S_\tau}(f_T^\star) + \sum_{\tau=1}^{T} \mathcal{O}\left(\sqrt{\frac{\tilde{\mathbf{y}}_\tau^T (k_\tau(\mathbf{X}_\tau, \mathbf{X}_\tau) + \lambda^2 \mathbf{I})^{-1} \tilde{\mathbf{y}}_\tau}{n_\tau}}\right) + \sum_{k=1}^{T} \mathcal{O}(\frac{1}{\sqrt{n_k}}) + 3c\sqrt{\frac{\log(2/\delta)}{2n_\tau}}$$

We get Theorem 2 by replacing into $\mathcal{L}_{S_\tau}(f_T^\star)$ using the inequalities from Lemma 6

C. Missing proofs of section 5 - OGD+ : Learning without forgetting

C.1. Memorisation property of OGD+ - Proof

Proof

In the proof of Theorem 1, App. A.1, we showed that, for $\mathcal{T}_\tau$ a fixed task:

$$f^\star_{\tau+1}(\mathbf{x}) = f^\star_\tau(\mathbf{x}) + \langle \nabla_{\mathbf{w}} f^\star_\tau(\mathbf{x}), \mathbf{w}^\star_{\tau+1} - \mathbf{w}^\star_\tau \rangle.$$

We rewrite the recursive relation into a sum:

$$f^\star_{\tau+1}(\mathbf{x}) = \sum_{k=1}^{\tau} \langle \nabla_{\mathbf{w}} f^\star_k(\mathbf{x}), \mathbf{w}^\star_{k+1} - \mathbf{w}^\star_k \rangle.$$

We observe that, for all $k \in [T]$:

$$\mathbf{w}^\star_{k+1} - \mathbf{w}^\star_k \in \mathbb{E}_{k'}.$$

On the other hand, for OGD+, given a sample $\mathbf{x}$ from $\mathcal{D}_\tau$, for all $k' \in [\tau+1, T]$:

$$\nabla_{\mathbf{w}} f^\star_k(\mathbf{x}) \in \mathbb{E}_{k'}$$

Therefore, for all $k' \in [k+1, \tau]$:

$$\langle \nabla_{\mathbf{w}} f^\star_{k'}(\mathbf{x}), \mathbf{w}^\star_{k'+1} - \mathbf{w}^\star_{k'} \rangle = 0$$

Therefore :

$$f^\star_\tau(\mathbf{x}) = f^\star_k(\mathbf{x})$$

We conclude.

C.2. OGD+ Generalisation Theorem - Proof

Proof

The proof is very similar to the proof of Theorem 2.

Let $\mathcal{T}_\tau$ a given task and $T \in \mathbb{N}^\star$ fixed

We start from the following result in Appendix B.1.

$$\mathcal{L}_{D_\tau}(f^\star_T) \leq \mathcal{L}_{S_\tau}(f^\star_T) + \sum_{k=1}^{T} \mathcal{O}\left(\sqrt{\frac{\tilde{\mathbf{y}}_k^T (k_k(\mathbf{X}_k, \mathbf{X}_k) + \lambda^2 \mathbf{I})^{-1} \tilde{\mathbf{y}}_k}{n_k}} \right) + \sum_{k=1}^{T} \mathcal{O}(\frac{1}{\sqrt{n_k}}) + 3c\sqrt{\frac{\log(2/\delta)}{2n_\tau}}$$

We apply Lemma 3 for tasks $\mathcal{T}_\tau$ and $\mathcal{T}_T$:

$$f^\star_T(\mathbf{x}_{\tau,i}) = f^\star_\tau(\mathbf{x}_{\tau,i}).$$

Therefore :

$$\mathcal{L}_{S_\tau}(f^\star_T) = \mathcal{L}_{S_\tau}(f^\star_\tau)$$

We replace into the first inequality :

$$\mathcal{L}_{D_\tau}(f_T^\star) \leq \mathcal{L}_{S_\tau}(f_\tau^\star) + \sum_{k=1}^{T} \mathcal{O}\left(\sqrt{\frac{\tilde{\mathbf{y}}_k^T (k_k(\mathbf{X}_k, \mathbf{X}_k) + \lambda^2 \mathbf{I})^{-1} \tilde{\mathbf{y}}_k}{n_k}}\right) + \sum_{k=1}^{T} \mathcal{O}(\frac{1}{\sqrt{n_k}}) + 3c\sqrt{\frac{\log(2/\delta)}{2n_\tau}}$$

We recall the folowwing result from the proof of Theorem 2 in App. B.1 :

$$\frac{1}{n_\tau} \sum_{i=1}^{n_\tau} (f_\tau^\star(\mathbf{x}_{\tau,i}) - y_{\tau,i})^2 \leq \frac{1}{n_\tau} \frac{\lambda^2}{4} \tilde{\mathbf{y}}_\tau^\tau (k_\tau(\mathbf{X}_\tau, \mathbf{X}_\tau))^{-1} \tilde{\mathbf{y}}_\tau$$

By replacing into the previous inequality, we conclude the proof of Theorem 3.

D. Complementary discussion

Imperfect robustness to Catastrophic Forgetting Two probable reasons OGD+ is not perfectly prone to catastrophic forgetting in the experiments are the memory limit and the non-overparameterization of the neural network. As in Lemma 3, we assumed that the memory is unlimited and that the neural network is overparameterized. We run additional experiments to verify these hypotheses. Table 6 (App. E.3.1) shows that the test accuracy increases uniformly with overparameterization, in which case the linear approximation is more accurate. Table 7 (App. E.3.2) shows that the test accuracy also increases uniformly with the memory size.

OGD and OGD+ on the short run One probable reason OGD+ doens't outperform OGD on the short run is that OGD+ performs a uniform sampling across samples from all past tasks, with the same memory budget as OGD. Also the memory requirements of OGD+ increase quadratically with tasks, We expect an information loss with respect to the most recent tasks, since the corresponding storage is used by OGD+ to "remember" older tasks, while OGD uses the equivalent storage for the most recent task.

E. Experiments :

E.1. Reproducibility

E.1.1. Code details

We forked the initial repository from `https://github.com/GMvandeVen/continual-learning`, then implemented the OGD and OGD+ algorithms. This initial repository is related to Van de Ven & Tolias (2018) and Van de Ven & Tolias (2019).

E.1.2. Hyperparameters

We use the same architecture and mostly the same hyperparameters as Farajtabar et al. (2019). We also keep a small learning rate, in order to preserve the locality assumption of OGD, and in order to verify the conditions of the theorems.

For the MNIST benchmarks, the neural network is a three-layer MLP with 256 hidden units in two layers, each layer uses RELU activation function. The model has either 2 logits or 10 logit outputs, which do not use any activation function. For the Split CIFAR-100 benchmark, the neural network is a multi-head LeNet (Lecun et al., 1998) network with Batch Normalisation (Ioffe & Szegedy, 2015) and 200 hidden units for the penultimate layer. The optimiser is either SGD, OGD or OGD+ and the loss is Softmax cross-entropy. We report the hyperparameters in detail in Table 5.

Hyperparameter	MNIST	CIFAR-100
Epochs	5	50
Architecture	MLP	LeNet
Hidden dimension	256	200
Learning rate	1e-02	
Batch size	256	
Torch seeds	0 to 4	
Memory size	1000	
Activation	RELU	

Table 5. Hyperparameters used across experiments

E.1.3. Experiment setup

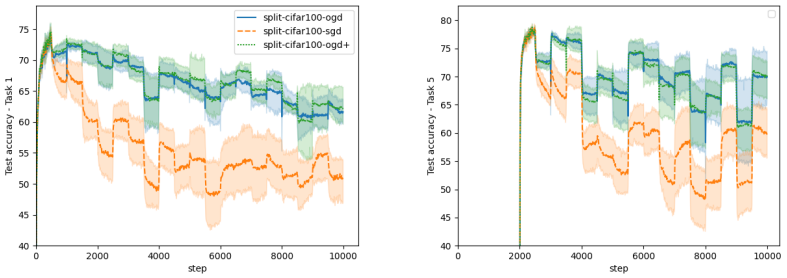
We run each experiment 5 times, the seeds set is the same across all experiments sets. We report the mean and standard deviation of the measurements. The test error is measured every 50 mini-batch interval.

E.1.4. OGD+ Implementation Details

Memory : In practice, we split the memories uniformly across tasks. Also, we construct $\mathcal{S}$ from $\mathcal{S}_D$ by sampling uniformly without replacement. Finally, for the memory reduction step, we truncate the last elements of the storage to free-up the space for the next task's data.

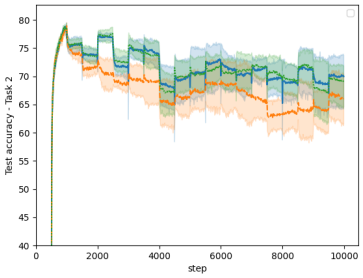
Multi-head models : For the dataset streams Split MNIST and Split MNIST, we consider multi-headed neural networks. We only store the feature maps with respect to the shared weights, therefore the projection step is not performed for the heads' weights.

E.2. Complementary results - Split CIFAR-100

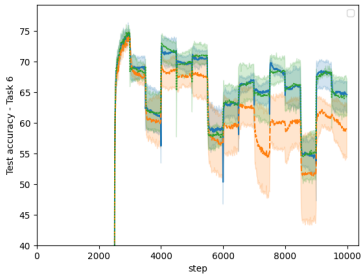


(a) Task 1

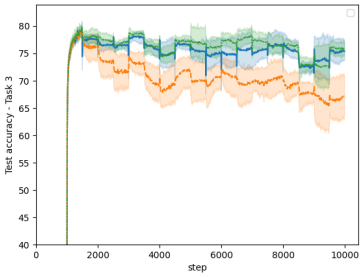
(b) Task 5



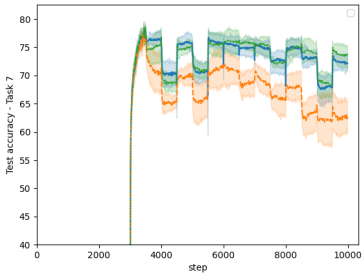
(c) Task 2



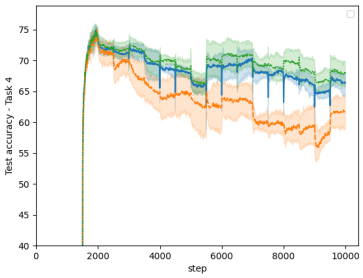
(d) Task 6



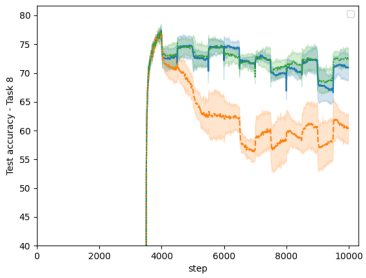
(e) Task 3



(f) Task 7



(g) Task 4



(h) Task 8

Figure 1. Test accuracy on the 8 first tasks of Split CIFAR-100, for SGD, OGD and OGD+. 20 class subsets of size 5 are sampled without replacement from the 100 classes. The model is trained to classify the CIFAR-100 images for 50 epochs for each task. The y-axis is truncated for clarity. We report the mean and standard deviation over 5 independent runs. The test error is measured for every 50 mini-batch interval.

E.3. Complementary experiments - OGD+

E.3.1. OGD+ Overparameterization

In order to assess the hypothesis we made in Sec. D, which states that one reason OGD+ is not perfectly prone to catastrophic forgetting as stated in Thm. 3 is the non-overparameterisation, we track the test accuracy with respect to the MLP hidden layer size, which is a proxy for overparameterisation.

We run the experiments on the Permuted MNIST using OGD+ and vary the hidden size from 100 to 400. The results are presented in Table 6. We measure the accuracy of the models through time after being trained on all 15 tasks. We see that the test accuracy of OGD+ increases uniformly with the hidden size.

We also observe that overparameterisation has a significant impact on the test accuracy the older the task is. The average accuracy margin between the hidden sizes 100 and 400 is 7.7% for the first task, while it is only 0.6% for the latest task.

	Accuracy ± Std. (%)						
	Task 1	**Task 2**	**Task 3**	**Task 4**	**Task 5**	**Task 6**	**Task 7**
100	69.9±5.7	75.9±1.7	76.6±3.1	77.3±3.6	80.7±3.5	83.8±2.5	81.0±3.3
250	73.0±2.6	80.0±1.9	**83.8±2.0**	82.4±2.0	82.5±2.9	83.2±1.7	**87.5±1.7**
400	**76.6±0.9**	**83.7±1.4**	**83.6±5.8**	**85.3±1.2**	**86.3±1.9**	86.4±3.5	**88.2±2.4**

	Accuracy ± Std. (%)							
	Task 8	**Task 9**	**Task 10**	**Task 11**	**Task 12**	**Task 13**	**Task 14**	**Task 15**
100	82.0±2.1	83.2±3.4	88.8±1.6	90.2±1.1	90.7±1.5	93.1±0.8	94.2±0.3	95.4±0.1
250	**88.6±1.0**	89.5±1.9	90.9±1.3	**92.1±0.8**	92.8±0.8	**94.2±0.6**	**95.0±0.1**	95.7±0.1
400	**87.6±2.4**	**91.0±0.7**	91.3±1.2	**92.5±0.9**	**93.4±0.5**	**94.4±0.5**	**95.3±0.4**	**96.0±0.1**

Table 6. Permuted MNIST : The test accuracy of models from the indicated task after being trained on all tasks in sequence for the MLP hidden sizes 100, 250 and 400. The best continual learning results are highlighted in **bold**.

E.3.2. OGD+ Memory size :

In order to assess the hypothesis we made in Sec. D, which states that one reason OGD+ is not perfectly prone to catastrophic forgetting as stated in Thm. 3 is the limited memory, we track the test accuracy with respect to the memory size.

We run the experiments on the Permuted MNIST using OGD+ and vary the memory size from 1.000 to 2 .000. The results are presented in Table 7. We measure the accuracy of the models through time after being trained on all 15 tasks. We see that the *mean* test accuracy of OGD+ increases uniformly with the memory size. The confidence intervals are not tight enough in most cases to conclude that the test accuracy increases *uniformly* with the memory size.

	Accuracy ± Std. (%)						
	Task 1	**Task 2**	**Task 3**	**Task 4**	**Task 5**	**Task 6**	**Task 7**
1000	72.2±2.3	73.7±4.4	77.2±5.2	79.2±1.3	79.3±3.3	83.2±1.9	81.9±3.8
1500	71.9±2.3	76.5±2.0	79.4±2.5	79.4±3.1	79.5±3.4	82.7±1.9	83.8±0.9
2000	**75.5**±2.1	**81.0**±2.0	**80.8**±3.0	**81.9**±1.6	**83.1**±1.5	**83 .7**±1.4	**85.0**±1.3

	Accuracy ± Std. (%)							
	Task 8	**Task 9**	**Task 10**	**Task 11**	**Task 12**	**Task 13**	**Task 14**	**Task 15**
1000	84.9±1.5	**87.7**±2.2	87.9±2.9	89.9±0.7	89.5±2.7	93.1±0.6	94.2±0.6	**95.3**±0.1
1500	85.6±1.9	86.7±1.3	89.1±2.5	90.5±1.7	91.8±1.4	93.1±0.9	94.0±0.4	95.2±0.1
2000	**86.3**±1.6	87.3±1.8	**89.9**±0.4	**90.8**±0.7	**92.1**±0.9	**93.4**±0.4	**94.3**±0.3	95.1±0.1

Table 7. Permuted MNIST : The test accuracy of models from the indicated task after being trained on all tasks in sequence for the memory sizes 1.000, 1.500 and 2.000. The best continual learning results *on average* are highlighted in **bold**.

E.4. Complementary tables - Generalisation

E.4.1. PERMUTED MNIST

We present the results of the Split MNIST experiments in Table 8. We measure the accuracy of the models through time after being trained on all 15 tasks. We can see that OGD+ outperforms OGD and SGD on the 6 initial tasks, and that OGD outperforms OGD+ on the subsequent tasks.

	Accuracy ± Std. (%)						
	Task 1	**Task 2**	**Task 3**	**Task 4**	**Task 5**	**Task 6**	**Task 7**
OGD+	**75.5±0.5**	**81.8±2.3**	**81.4±0.6**	**85.4±2.1**	**86.5±1.2**	83.0±4.9	86.4±1.4
OGD	37.7±5.6	72.6±4.3	74.2±3.7	81.7±1.9	82.8±1.3	**87.2±1.1**	**89.8±0.5**
SGD	56.5±4.6	49.5±5.9	57.6±3.9	60.9±2.4	75.1±3.3	73.6±4.3	74.9±3.6

	Accuracy ± Std. (%)							
	Task 8	**Task 9**	**Task 10**	**Task 11**	**Task 12**	**Task 13**	**Task 14**	**Task 15**
OGD+	**89.3±0.8**	90.5±0.2	92.1±0.2	90.2±0.6	93.4±0.2	**94.6±0.2**	94.8±0.2	95.7±0.1
OGD	**89.4±0.7**	**92.1±0.6**	**92.7±0.5**	**93.7±0.3**	**94.3±0.2**	**94.7±0.4**	**95.6±0.1**	**96.0±0.1**
SGD	73.4±5.6	78.0±3.3	83.3±3.7	90.0±1.2	90.4±0.8	94.1±0.2	94.3±0.4	**96.0±0.2**

Table 8. Permuted MNIST : The test accuracy of models from the indicated task after being trained on all tasks in sequence. The best continual learning results are highlighted in **bold**.

E.4.2. ROTATED MNIST

We present the results of the Rotated MNIST experiments in Table 8. We measure the accuracy of the models through time after being trained on all 15 tasks. We observe that OGD+ outperforms OGD and SGD on the 8 initial tasks, and OGD+ and OGD are equivalent in terms of robustness to catastrophic forgetting on the tasks 9 to 11.

	Accuracy ± Std. (%)						
	Task 1	**Task 2**	**Task 3**	**Task 4**	**Task 5**	**Task 6**	**Task 7**
OGD+	**50.4±2.5**	**52.5±1.9**	**60.4±1.6**	**67.6±1.9**	**73.1±1.8**	**78.0±1.5**	**82.9±1.1**
OGD	41.4±2.0	44.3±1.5	51.5±2.2	59.8±1.6	66.9±0.7	73.5±0.7	79.7±0.6
SGD	31.4±0.7	34.2±0.7	40.2±0.6	47.2±0.5	55.3±0.5	62.0±0.7	70.3±0.9

	Accuracy ± Std. (%)							
	Task 8	**Task 9**	**Task 10**	**Task 11**	**Task 12**	**Task 13**	**Task 14**	**Task 15**
OGD+	**86.8±0.8**	**90.3±0.7**	**93.0±0.3**	**95.1±0.2**	**96.5±0.1**	**97.2±0.1**	97.2±0.0	97.1±0.1
OGD	85.1±0.2	**89.8±0.2**	**92.7±0.2**	**95.1±0.2**	**96.5±0.1**	**97.2±0.1**	97.3±0.1	97.1±0.1
SGD	78.2±0.8	85.0±0.6	89.8±0.5	93.8±0.3	96.0±0.1	97.0±0.1	97.3±0.0	97.2±0.1

Table 9. Rotated MNIST : The test accuracy of models from the indicated task after being trained on all tasks in sequence. The best continual learning results are highlighted in **bold**.

E.4.3. SPLIT CIFAR-100

We present the results of the Split CIFAR-100 experiments in Table 8. We measure the accuracy of the models through time after being trained on all 20 tasks, each task consists of a classes subset of size 5 sampled without repetition from the 100 classes of CIFAR-100.

We see that *on average*, OGD+ outperforms OGD on the long task sequences. The confidence intervals are not tight enough to conclude on the relative performance of OGD and OGD+ on the Split CIFAR-100 benchmark. Also, OGD and OGD+ outperform SGD on all tasks and are equivalent to SGD on the latest task. This result is intuitive as the last task is only a supervised leanring problem which does not involve Continual Learning.

	Accuracy ± Std. (%)									
	Task 1	**Task 2**	**Task 3**	**Task 4**	**Task 5**	**Task 6**	**Task 7**	**Task 8**	**Task 9**	**Task 10**
OGD+	**62.1**±3.2	69.3±4.3	**75.8**±1.4	**67.9**±2.4	**70.0**±3.8	64.2±3.3	**73.6**±2.2	**72.4**±2.0	73.6±4.3	63.0±2.5
OGD	61.6±2.0	**70.0**±5.2	75.4±2.2	66.5±1.9	**70.0**±5.1	**64.6**±2.7	72.1±1.4	71.0±2.6	**74.3**±3.6	**63.9**±3.5
SGD	50.7±4.9	66.1±6.0	67.2±5.7	61.6±3.7	59.9±6.0	58.7±6.3	62.6±4.0	60.5±3.5	64.8±5.7	52.4±5.2

	Accuracy ± Std. (%)									
	Task 11	**Task 12**	**Task 13**	**Task 14**	**Task 15**	**Task 16**	**Task 17**	**Task 18**	**Task 19**	**Task 20**
OGD+	69.4±3.5	76.4±3.0	59.8±3.3	73.9±0.8	65.1±0.8	67.3±2.4	**57.2**±1.4	70.9±2.0	75.8±1.2	73.1±0.8
OGD	**70.0**±3.3	**78.5**±1.8	**60.2**±0.9	**74.8**±1.1	**65.2**±1.3	**67.7**±2.1	56.0±1.9	**72.4**±2.5	**76.6**±1.3	73.1±1.1
SGD	62.5±4.8	68.7±4.7	53.3±4.0	67.0±4.0	58.2±3.1	54.5±4.2	49.8±5.3	67.2±2.0	73.2±3.3	**73.4**±0.9

Table 10. Split CIFAR-100 : The test accuracy of models from the indicated task after being trained on all tasks in sequence. The best continual learning results *on average* are highlighted in **bold**.