Provable Continual Learning via Sketched Jacobian Approximations

Reinhard Heckel*,†

*Dept. of Electrical and Computer Engineering, Technical University of Munich
†Dept. of Electrical and Computer Engineering, Rice University

December 10, 2021

Abstract

An important problem in machine learning is the ability to learn tasks in a sequential manner. If trained with standard first-order methods most models forget previously learned tasks when trained on a new task, which is often referred to as catastrophic forgetting. A popular approach to overcome forgetting is to regularize the loss function by penalizing models that perform poorly on previous tasks. For example, elastic weight consolidation (EWC) regularizes with a quadratic form involving a diagonal matrix build based on past data. While EWC works very well for some setups, we show that, even under otherwise ideal conditions, it can provably suffer catastrophic forgetting if the diagonal matrix is a poor approximation of the Hessian matrix of previous tasks. We propose a simple approach to overcome this: Regularizing training of a new task with sketches of the Jacobian matrix of past data. This provably enables overcoming catastrophic forgetting for linear models and for wide neural networks, at the cost of memory. The overarching goal of this paper is to provided insights on when regularization-based continual learning algorithms work and under what memory costs.

1 Introduction

Consider the problem of learning a number of tasks sequentially. Even if a neural network has the ability to perform well on all tasks simultaneously, if trained sequentially on different tasks, it tends to perform well only on the task it has most recently been trained on [Fre99; Rob95; Goo+15]. This is known as catastrophic forgetting.

Continual learning algorithms address forgetting and aim to enable sequential learning of several tasks. Continual learning algorithms are often categorized into replay methods, regularization based methods, and approaches that modify a model directly by freezing/masking parts of the model or adding new parts to the model, see for example [Rus+16].

Replay methods [Rob95; Reb+17; Shi+17; LH18] store past task data, or a generative model, and reuse past task data or pseudo-labels generated by the stored generative model when training a new task.

Regularization based methods penalizes models with parameters far from those found important for previous tasks when training a new task. Two widely used regularization based methods are elastic weight consolidation (EWC) [Kir+17] and synaptic intelligence (SI) [ZPG17]. Both penalize the change of individual model coefficients deemed important for previous tasks via a quadratic penalty associated with each weight (or model parameter). This penalty ignores interactions between coefficients. Subsequent works incorporated quadratic penalties on the weights that take interactions between the weights into account, and/or provided generalizations of EWC [RBB18; Liu+18; Cha+18; Sch+18; Pan+21]. This generally tends to improve performance. Versions of

regularization-based methods are also referred to as online-Laplace algorithms, since they can be understood as applying a Laplace-approximation to the posterior.

Regularization based continual learning algorithms are appealing for its simplicity, but—like continual learning algorithms in general—are not well understood theoretically. It is important to study continual learning algorithms theoretically to understand when they work and when they fail. For example, methods perform well on a given continual learning task, but fail on another. Taking the EWC algorithm as an example, it performs almost optimally on the MNIST permutation problem (see Fig. 3B in [Kir+17]), but fails for learning pairs of different digits sequentially (see. Fig. 2a in [Kem+18]).

Motivated this discrepancy, the goal of this paper is to improve the understanding of regularization based continual learning methods by studying a family of algorithms that rely on approximating the loss functions pertaining to different tasks with random projections. The intuition is, that the outer product of the Jacobian is an approximation to the Hessian, and approximate Jacobians can efficiently approximate the Hessian. The family of algorithms uses the Hessian approximation in a quadratic penalty which approximates the loss function pertaining to past data. The EWC algorithm and adding an L2-penalty penalizing the move of coefficients are special cases.

Our contributions are as follows:

- Our main result is to show that a regularization based continual learning algorithm trained with ℓ_2-loss that works with sketched Jacobians provably enables continual learning, both for linear models and for wide neural networks.

- We conduct experiments on the MNIST permutation problem and the incremental MNIST task. The results show that working with a coarse sketch of the Jacobian gives significant improvements over the EWC algorithm, albeit at the cost of storing significantly more data.

- The EWC algorithm and even importance/L2-regularization perform well for continual learning on the MNIST permutation problem. This is surprising, because the penalty used in the EWC algorithm is not a good approximation of the loss function of past data. To understand this, we study a model of the permutation problem theoretically, and show that both EWC and constant importance can work optimally on that task. We complement this result with a statement showing that EWC and importance/L2-regularization can provably fail as well.

Those results contribute to an understanding of when we expect particular regularization based continual learning algorithms to work well and when not.

1.1 Related work

There are relatively few theoretical works on continual learning, compared to the vast literature on algorithms for continual learning and corresponding empirical results. Some recent theoretical developments include [Yin+20; BDS20; AMP17; Doa+21]. Yin et al. [Yin+20] studies optimization and generalization aspects of regularization based algorithms involving the Hessian. Our work differs in that we study a different family of algorithms (including random projections to approximate the Jacobian), and different models of data.

Our work builds on the popular idea of using (an approximation of) the second-order Taylor expansion to approximate the loss of past data. This idea was proposed as early as in [RE13], and of course the regularization based methods mentioned earlier can be viewed as being based on this

idea. Finally, the very recent paper [Li+21] also proposed to use random projections for sketching a regularizer involving the Jacobian, and provided experiments showing that this approach works well. Our works are very complementary, in that we provide explicit theoretical results showing that sketching the Jacobian enables provably continual learning for both linear models and for two-layer neural networks. In addition, we also provide positive and negative results for EWC.

2   Problem statement

We consider the problem of learning a series of regression or classification tasks $A, B, \ldots$ sequentially. For each task T, we are given a set of training examples $\{(\mathbf{x}_{T,1}, y_{T,1}), \ldots, (\mathbf{x}_{T,n}, y_{T,n})\} \in \mathbb{R}^d \times \mathcal{Y}$ drawn iid from an unknown distribution P_T pertaining to task T. Here $\mathcal{Y} = \mathbb{R}$ for a regression task, and $\mathcal{Y} = \{1, \ldots, Q\}$ for a classification task. Our goal is to train a single model $f_{\boldsymbol{\theta}} \colon \mathbb{R}^d \to \mathcal{Y}$ sequentially on the training sets of tasks $A, B, \ldots$, so that after training on task T, the model performs well *on all past tasks*.

More specifically, after training on tasks $A, B, \ldots, T$, the model should perform well in predicting the response y based on the feature vector $\mathbf{x}$, with the unseen example $(\mathbf{x}, y)$ drawn with equal probability from one of the distributions of tasks $A, B, \ldots, T$. Throughout, we assume that the method does not know at test time on which task it is evaluated. This is sometimes called single-head evaluation, as opposed to multi-head evaluation, where the task ID is known [Cha+18], and is considered the more challenging and more practical evaluation mode or setup.

3   A family of regularization based continual learning algorithms

We start with introducing a family of regularization-based continual learning algorithms that rely on approximations of the loss functions pertaining to different tasks. This family incorporates the EWC algorithm and is closely related to other regularization-based methods including [RBB18; Liu+18; Cha+18; Sch+18] in that it approximates the loss associated with past tasks.

The family of algorithms is parameterized by an approximation of the Jacobian of the model $f_{\boldsymbol{\theta}} \colon \mathbb{R}^d \to \mathcal{Y}$ with model parameter $\boldsymbol{\theta} \in \mathbb{R}^p$ applied to the data of a task. Specifically, denote by $\mathbf{K}_T \in \mathbb{R}^{s \times p}$ an approximation of the Jacobian $\mathbf{J}_T \in \mathbb{R}^{n \times p}$ of the predictions of the data of task T:

$$\mathbf{f}_T(\boldsymbol{\theta}) = [f_{\boldsymbol{\theta}}(\mathbf{x}_{T,1}), \ldots, f_{\boldsymbol{\theta}}(\mathbf{x}_{T,n})]^T.$$

Here, n is the number of training examples. The Jacobian $\mathbf{J}_T$ contains the gradients $\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\mathbf{x}_{T,i}), i = 1, \ldots, n$ as rows. The gradients have dimension p for a regression problem, and dimension $(Q-1) \times p$ for a classification problem with Q classes.

We train a model $f_{\boldsymbol{\theta}}$ with a quadratic loss on the data from tasks $A, B, \ldots, T$ as follows. First, we minimize the training loss of task A:

$$\mathcal{L}_A(\boldsymbol{\theta}) = \frac{1}{2} \sum_{i=1}^{n} (f_{\boldsymbol{\theta}}(\mathbf{x}_{A,i}) - y_{A,i})^2. \tag{1}$$

Let $\boldsymbol{\theta}_A$ be a minimizer of this loss function. Next, we learn task B by minimizing the loss

$$\mathcal{L}_{AB}(\boldsymbol{\theta}) = \mathcal{L}_B(\boldsymbol{\theta}) + \frac{\lambda}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_A)^T \mathbf{K}_A^T \mathbf{K}_A (\boldsymbol{\theta} - \boldsymbol{\theta}_A), \tag{2}$$

where $\mathbf{K}_A \in \mathbb{R}^{n \times p}$ is the approximation of the Jacobian of the function $\mathbf{f}_A(\boldsymbol{\theta}) = [f_{\boldsymbol{\theta}}(\mathbf{x}_{A,1}), \dots, f_{\boldsymbol{\theta}}(\mathbf{x}_{A,n})]^T$ at $\boldsymbol{\theta} = \boldsymbol{\theta}_A$, and $\lambda \geq 0$ is a regularization parameter. In order to learn a third task C we minimize the function

$$\mathcal{L}_{ABC}(\boldsymbol{\theta}) = \mathcal{L}_C(\boldsymbol{\theta}) + \frac{\lambda}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_{AB})^T (\mathbf{K}_A^T \mathbf{K}_A + \mathbf{K}_B^T \mathbf{K}_B)(\boldsymbol{\theta} - \boldsymbol{\theta}_{AB}),$$

where $\boldsymbol{\theta}_{AB}$ is a minimum of $\mathcal{L}_{AB}(\boldsymbol{\theta})$ and $\mathbf{K}_B$ is the approximate Jacobian of the predictions $\mathbf{f}_B(\boldsymbol{\theta}) = [f_{\boldsymbol{\theta}}(\mathbf{x}_{B,1}), \dots, f_{\boldsymbol{\theta}}(\mathbf{x}_{B,n})]^T$ at $\boldsymbol{\theta} = \boldsymbol{\theta}_{AB}$. The algorithm proceeds by learning further tasks $D, E, \dots$ analogously.

In the coming sections, we discuss four variants. All variants require at most computation of the Jacobian. The Jacobian is easy to compute for most popular machine learning models, because it simply requires computation of the gradients on the training examples, and first order methods for optimization already compute those in each epoch. We state the memory requirement for K tasks.

i) **Regularization with Original Jacobian:** Take the approximation of the Jacobian as the Jacobian of the predictions $\mathbf{f}_T(\boldsymbol{\theta})$ at the appropriate $\boldsymbol{\theta}$, i.e., $\mathbf{K}_T = \mathbf{J}_T$. Memory requirement: $p(1 + Kn)$.

ii) **Regularization with Sketched Jacobian (RSJ):** Take the approximation of the Jacobian as a random sketch of the Jacobian of the predictions $\mathbf{f}_T(\boldsymbol{\theta})$ at the appropriate $\boldsymbol{\theta}$, obtained by left-multiplying the Jacobian with a Gaussian random projection matrix $\mathbf{S}_T \in \mathbb{R}^{s \times n}$, with iid $\mathcal{N}(0, 1/s)$ entries, i.e., $\mathbf{K}_T = \mathbf{S}_T \mathbf{J}_T \in \mathbb{R}^{s \times p}$. Memory requirement: $p(1 + Ks)$. Note that this algorithm is indexed by s, therefore we refer to it as RSJ-s in the following (e.g., RSJ-50 is RSJ with $s = 50$).

iii) **EWC:** Take the approximation of the Jacobian $\mathbf{K}_T \in \mathbb{R}^{p \times p}$ as the square-root of the diagonal of the outer product of the Jacobians $\mathbf{J}_T^T \mathbf{J}_T$. This algorithm is a variant of the EWC algorithm. Specifically, it corresponds to "online" EWC where instead of the original Fisher matrix, the empirical Fisher matrix is used (see [KBH19] on the relations of the original and empirical Fisher matrices). Memory requirement: $2p$.

iv) **L2:** Take the approximation of the Jacobian $\mathbf{K}_T \in \mathbb{R}^{p \times p}$ as the identity matrix. This amounts to simple constant importance/L2-regularization penalizing the movement of coefficients from one task to the other. Memory requirement: p.

The intuition behind this family of algorithms is as follows. First, consider the variant that takes the exact Jacobian as the matrix $\mathbf{K}_A$. If the model $f_{\boldsymbol{\theta}}(\mathbf{x})$ is linear in the model parameter $\boldsymbol{\theta}$ then, as shown in the next section, for $\lambda = 1$, the algorithm performs optimal continual learning, since the model learned at each step T is equivalent to the model learned when training on all data $A, B, \dots, T$. The model is linear in the model parameter for all kernel methods, and is approximately linear for wide neural networks, as established by the recent theory on the neural tangent kernel [JGH18; Lee+18; Du+18; OS20]. For this linear setup, the importance matrix $\mathbf{J}_A^T \mathbf{J}_A$ is equal to the Hessian matrix.

While working with the original Jacobian gives a continual learning algorithm that provably succeeds for linear models, the corresponding algorithm is impractical for large commonly used models. For small, under-parameterized toy models working with the original Jacobian is a viable

approach, but for large, over-parameterized models used in practice it is infeasible and impractical to store all the Jacobians, because in that case the memory requirement would be larger than that of storing the training data.

The intuition behind regularization with the sketched Jacobian (RSJ algorithm) is that, if we manage to obtain accurate approximation of the Jacobian outer products $\mathbf{J}_T^T \mathbf{J}_T$ for all tasks, then we expect the corresponding algorithm to behave similar to the optimal algorithm that works with the original Jacobian. In Section 5 and 6 we show that the sketched Jacobians can provably enable continual learning for linear models and wide neural networks. This comes at a price: larger values of s give a better approximation at a higher memory cost.

The third variant, the EWC algorithm, can be viewed as taking an extreme approach to approximating the Jacobian outer product $\mathbf{J}_T^T \mathbf{J}_T$ by simply taking its diagonal.

4 Empirical observations

We start by evaluating the methods introduced in the previous section on two popular continual learning problems: The MNIST permutation problem and the incremental MNIST problem. We will make a number of empirical observations, and then explain those empirical observations with theoretical results in the remaining sections. All simulations were run on a single RTX 5000 GPU and are reproducible with the code in the supplement.

There are a variety of interesting new state-of-the-art approaches for continual learning. We do not compare to those, because our goal is not to establish a new state-of-the-art method but rather to understand regularization based continual learning algorithms better. However, we compare to "training on all data" as a reference point, which can be viewed as an upper bound for any continual learning algorithm.

4.1 MNIST permutation Problem

The popular MNIST permutation problem [Goo+15; Kir+17] is as follows. Task A is the original MNIST digit classification problem, and all remaining tasks are obtained by permuting the pixels of each image with a random permutation that is fixed for each task $B, C, \ldots$. This is often called task-incremental learning. We apply the family of algorithms introduced in the previous section to continually learning 10 of such tasks with a two-layer fully connected network with 500 hidden nodes and relu-activation functions. Figure 1 shows the results, including the baseline "all data", which means the network has been trained on all data from all tasks. This baseline is an upper bound on the performance of any continual learning algorithm.

We first observe that, perhaps surprisingly, **EWC performs extremely well on permuted MNIST**: Almost as well as training on all data from all tasks. We find this surprising, because the EWC algorithm works well here even though the Jacobian outer product is not well approximated with its diagonal. To see that the Jacobian outer product, $\mathbf{J}_T^T \mathbf{J}_T$, cannot be well-approximated by its diagonal, note that the number of training examples is $n = 60000$, while the number of model parameters is $p = 397510$, therefore $\mathbf{J}_T^T \mathbf{J}_T$ has rank at most n while its diagonal has rank $p \gg b$.

The experiment also shows that simple constant importance/L2-regularization performs well, it is only 2%-less than optimal on 10 tasks. Note that for L2-regularization to work well it is critical to scale the penalty of each set of parameters (first layer weights, first layer bias, second layer
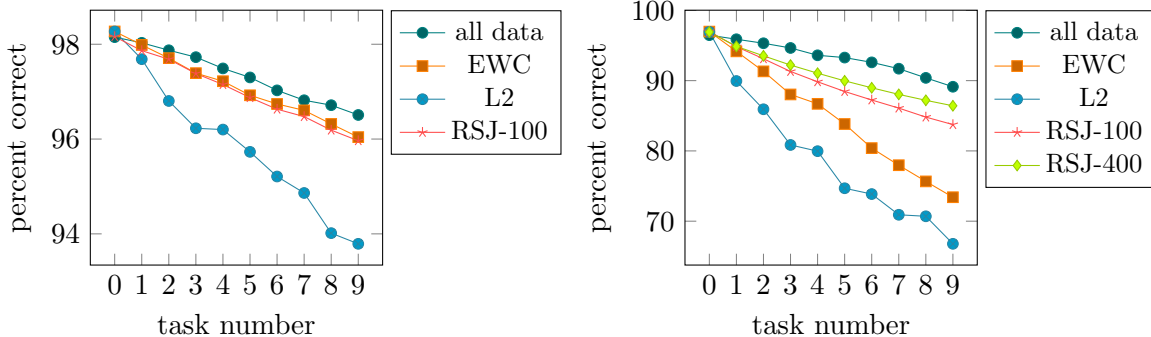
Figure 1: **Left:** Sequential learning on the MNIST permutation problem for a two-layer fully connected relu network. All data refers to training on all the data of all the tasks. The experiment shows that on this task, if correctly tuned, all algorithms perform close to optimal, i.e., close to training sequentially on all data. **Right:** Sequential learning on the MNIST permutation problem for relu-random feature model with $6 \cdot 784$ random features. For this model, which is much smaller than the two-layer model from the past experiment, EWC and L2 regularization do not work well, but the RSJ algorithm which uses a much better approximation of the Jacobian significantly improves performance.

weights, second layer bias) appropriately (through hyperparameter optimization), which we have done here.

In Section 7 we explain theoretically why EWC (and even constant importance/L2-regularization) can perform so well in some situations.

Also note that the random projection based algorithm RSJ-100 performs similar to EWC for this setup, but not better, because there is little room for improvement to training on all data.

Next, we study the performance of the family of continual learning algorithms on a much smaller model, specifically on a random feature model, with $6 \cdot 784$ Gaussian random relu features (i.e., we fit the model $f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathrm{relu}(\boldsymbol{\Theta}\mathbf{x})\boldsymbol{\theta}$, where $\boldsymbol{\Theta}$ is a Gaussian random matrix). This model has much fewer parameters than the neural network considered earlier and is linear in the model parameter $\boldsymbol{\theta}$ (but not in $\mathbf{x}$). Figure 1, right panel, depicts the results.

The results show that for this smaller model, the EWC and L2-regularization algorithms perform significantly worse than the random-projection based RSJ algorithm algorithm with a sufficiently large random projection dimension ($s = 400$). The RSJ-400 algorithm performs almost on par with training on all data.

An important observation from this experiment is that the **the gap between learning on all data and the RP based algorithm increases from task to task**. That is intuitively expected because the error due to the approximate Jacobians accumulates from task to task. We discuss this aspect in Section 5 theoretically.

4.2 Incremental MNIST problem

Finally, we study the problem of incrementally learning to classify digits. This is called class incremental learning by van de Ven and Tolias [vT19] and Hsu et al. [Hsu+19]). Task A is to classify $\{0, 1\}$, task B to classify $\{2, 3\}$, etc, until task E which is to classify $\{8, 9\}$. This problem
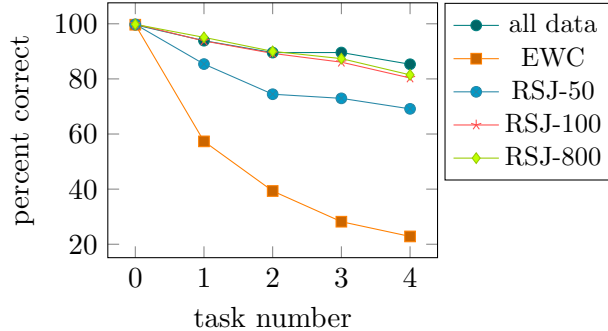
Figure 2: Sequentially learning to classify the digits $\{0,1\}, \{2,3\}, \ldots, \{8,9\}$ with a relu-random feature model with $6 \cdot 784$ random features. For this problem, EWC does not work at all, but the RSJ algorithm with a sufficiently large random projection (i.e., value of s) works almost as well as training on all data.

and variants thereof are a popular continual learning baseline [Kem+18; ZPG17]. Figure 2 shows the performance of the family of algorithms on the incremental learning task. Note that after learning say tasks $A = \{0,1\}$ and $B = \{2,3\}$, the method is evaluated on both tasks simultaneously, i.e., on the test set containing all digits $\{0,1,2,3\}$. This assumes the task on which we test on is unknown, in contrast to the so called multi-head evaluation mode where the task is known. This evaluation mode is considered much harder, as mentioned earlier.

The results, depicted in Figure 2, show that while EWC fails dramatically for this task (which is well known, cf. Fig. 2a in [Kem+18], the RSJ algorithm works almost as well as training on all the data, already for a projection dimension of $s = 100$.

5 Guarantees for linear models

We start by providing guarantees for the Jacobian regularization based learning algorithms (based on the full and sketched Jacobian) for linear models. We consider models $f_{\boldsymbol{\theta}}(\mathbf{x})$ that are linear in $\boldsymbol{\theta}$, i.e., there exists a feature map $\psi \colon \mathbb{R}^d \to \mathbb{R}^p$, so that $f_{\boldsymbol{\theta}}(\mathbf{x}) = \langle \psi(\mathbf{x}), \boldsymbol{\theta} \rangle$. All kernel methods can be written in this form (although for some kernels this feature map is infinite dimensional).

We start with an illustrative result that guarantees that if we work with the original Jacobian, then regularization based continual learning is provably correct:

Proposition 1. *Suppose that the model $f_{\boldsymbol{\theta}}$ is linear in $\boldsymbol{\theta}$, and consider the continual learning algorithm with the original Jacobian and with regularization parameter $\lambda = 1$ trained with ℓ_2-loss. Then the model learned with Jacobian regularization gives exactly the same result as training on the original data.*

We hasten to add that it might be obvious to experts in continual learning that regularization based continual learning with the exact Jacobian provably succeeds for linear models; we neverthe-less state this formally to put the results to come into context.

As mentioned before, for small, under-parameterized toy models working with the original Jacobian is a viable approach, but for large, over-parameterized models used in practice it is

infeasible and impractical, because the memory requirement of storing the Jacobian is larger than that of storing the entire training data.

We next discuss the RSJ algorithm that works with the sketched Jacobian, and for simplicity we focus on learning two tasks A and B. Learning of task A is straightforward and amounts to minimizing the least-squares loss on task A (i.e., $\mathcal{L}_A(\boldsymbol{\theta})$). Learning task B after having learned task A amounts to solving the partially sketched least-squares problem:

$$\mathcal{L}_{AB}(\boldsymbol{\theta}) = \mathcal{L}_B(\boldsymbol{\theta}) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_A)^T \mathbf{J}_A^T \mathbf{S}^T \mathbf{S} \mathbf{J}_A (\boldsymbol{\theta} - \boldsymbol{\theta}_A).$$

This can be viewed as a perturbed version of the least-squares problem

$$\tilde{\mathcal{L}}_{AB}(\boldsymbol{\theta}) = \mathcal{L}_A(\boldsymbol{\theta}) + \mathcal{L}_B(\boldsymbol{\theta}) = \frac{1}{2}\|\mathbf{J}\boldsymbol{\theta} - \mathbf{y}\|_2^2, \tag{3}$$

corresponding to training on the entire data. To see this, note that if the random projection dimension is sufficiently large then the matrix $\mathbf{J}_A^T \mathbf{S}^T \mathbf{S} \mathbf{J}_A$ is a good approximation of the matrix $\mathbf{J}_A^T \mathbf{J}_A$, which in turn means that the loss $\mathcal{L}_{AB}$ is a good approximation of the loss $\tilde{\mathcal{L}}_{AB}$. Here, we defined

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_A \\ \mathbf{J}_B \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \mathbf{J}_A \boldsymbol{\theta}_A \\ \mathbf{y}_B \end{bmatrix}.$$

Note that the sketch of the Jacobian induces a perturbation, and therefore the solution obtained by minimizing the loss $\mathcal{L}_{AB}(\boldsymbol{\theta})$ is in general different than minimizing the loss of the entire data. However, if the Jacobian of $\mathbf{J}_A$ is well approximated by a matrix of rank r, then we expect that a sketch of dimension $s = O(r)$ approximates the Jacobian well, and the estimate obtained by minimizing the sketched least-squares problem is expected to be close to the solution obtained by minimizing the original least-squares problem.

The following result, proven in the supplement, formalizes this intuition by bounding the difference of the gradient descent iterates $\boldsymbol{\theta}_t$ and $\tilde{\boldsymbol{\theta}}_t$ on the least squares loss $\mathcal{L}_{AB}(\boldsymbol{\theta})$ and the un-perturbed least-squares loss in (3).

Theorem 1. *Let $\mathbf{J} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ be the singular value decomposition of $\mathbf{J}$, $\sigma_{\max}$ and $\sigma_{\min}$ are the largest and smallest singular values, and $\mathbf{U}_r$ and $\mathbf{U}_n$ are the left-singular vectors corresponding to the largest r and the remaining singular values. Let $\boldsymbol{\theta}_t$ and $\tilde{\boldsymbol{\theta}}_t$ be the gradient descent iterates with stepsize η starting at $\boldsymbol{\theta}_0 = \tilde{\boldsymbol{\theta}}_0 = \mathbf{0}$ on the aforementioned least-squares problems.*

i) Suppose that t is sufficiently small so that
$(1 - \eta\sigma_{\min}^2)^t \geq 1 - \frac{\sigma_{\min}^2}{\sigma_r^2} \frac{\|\mathbf{U}_r^T \mathbf{y}\|_2}{\|\mathbf{U}_n^T \mathbf{y}\|_2}$. *Then, with probability at least $1 - 4t \exp\left(-\frac{\|\mathbf{J}_A\|_F^2}{2\|\mathbf{J}_A\|^2}\right)$,*

$$\left\|\boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}_t\right\|_2 \leq 8 \frac{\|\mathbf{J}_A\|_F}{\sqrt{s}\sigma_r} \frac{1}{\sigma_r} \left\|\mathbf{U}_r^T \mathbf{y}\right\|_2.$$

ii) Suppose that $\mathbf{J}$ has rank r. Then, with probability at least $1 - 2e^{-r^2}$, for all t,

$$\left\|\boldsymbol{\theta}^t - \tilde{\boldsymbol{\theta}}^t\right\|_2 \leq \frac{\sigma_{\max}}{\sigma_r} \sqrt{c\frac{r}{s}} \frac{1}{\sigma_r} \|\tilde{\mathbf{r}}_0\|_2.$$

The theorem consists of two parts, both guaranteeing closeness of the iterates under slightly different assumptions.

The first part guarantees that if the sketch dimension is sufficiently large relative to the effective rank of the matrix $\mathbf{J}_A$, measured by $\|\mathbf{J}_A\|_F/\sigma_r$, then the solution obtained by applying gradient descent to the original and un-perturbed least-squares problems are very close. If the singular values decay quickly, then the statement ensures that after a certain number of iterations, sufficient to fitting the singular vectors corresponding to large singular values, the iterates are close.

The second part guarantees that the solution obtained by applying gradient descent to the original and un-perturbed least-squares problems are very close, provided that the sketch dimension is sufficiently large relative to $\sigma_{\max}\sqrt{r}/\sigma_r \geq \|\mathbf{J}_A\|_F/\sigma_r$. This is a slightly stronger requirement on the sketch dimension, but the closeness holds for all iterations t.

There is a large body of literature which ensures that the solution of a randomly sketched least-squares problem behaves similarly to the solution obtained for the un-perturbed problem [Sar06; ANW12; PW15]. The proof of Theorem 1 is conceptually similar to those of prior works, but differs as in our setup only part of the least-squares problem is sketched, and more importantly, our results also applies to over-parameterized models, unlike many previous results.

5.1 Continual learning on a sequence of two regression tasks

We next apply Theorem 1 to obtain guarantees for continual learning of a regression task with the RSJ algorithm. We show that a sufficiently large sketch dimension provably enables continual learning.

Suppose the data of task T, for $T \in \{A, B\}$ is generated by a Gaussian linear model

$$y = \langle \mathbf{x}, \boldsymbol{\theta}_T \rangle + z,$$

where $\mathbf{x} \sim \mathcal{N}(0, \mathbf{I})$ and $z \sim \mathcal{N}(0, \sigma^2)$. Consider learning a linear model $f_{\boldsymbol{\theta}}(\mathbf{x}) = \langle \boldsymbol{\theta}, \mathbf{x} \rangle$ with a quadratic loss. The risk if we draw a problem instance from one of the two tasks with equal probability is

$$R(\boldsymbol{\theta}) = \frac{1}{2}\mathbb{E}_{(\mathbf{x},y)\sim P_A}\left[(\langle \boldsymbol{\theta}, \mathbf{x} \rangle - y)^2\right] + \frac{1}{2}\mathbb{E}_{(\mathbf{x},y)\sim P_B}\left[(\langle \boldsymbol{\theta}, \mathbf{x} \rangle - y)^2\right]$$
$$= \|\boldsymbol{\theta} - \boldsymbol{\theta}_A\|_2^2 + \|\boldsymbol{\theta} - \boldsymbol{\theta}_B\|_2^2 + \sigma^2. \tag{4}$$

Thus, the optimal linear model is $\boldsymbol{\theta}^* = \frac{1}{2}(\boldsymbol{\theta}_A + \boldsymbol{\theta}_B)$. Suppose we obtain $n \gg d$ training examples from each of the two tasks and apply gradient descent until convergence on all of the data (i.e., the loss function in (3)). Then, after $t = O(\log(n)/\log(n/d))$ many gradient descent iterations (see supplement), the corresponding estimate $\tilde{\boldsymbol{\theta}}_t$ obeys, with high probability,

$$\left\|\tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*\right\|_2 \leq O\left(\sqrt{\frac{d}{n}}(\|\boldsymbol{\theta}_A\|_2 + \|\boldsymbol{\theta}_B\|_2 + \sigma)\right). \tag{5}$$

If we learn both task sequentially with the RSJ algorithm, we get, by applying Theorem 1, that (see supplement for the details), with high probability,

$$\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|_2 \leq O\left(\left(\sqrt{\frac{d}{n}} + \sqrt{\frac{d}{s}}\right)(\|\boldsymbol{\theta}_A\|_2 + \|\boldsymbol{\theta}_B\|_2 + \sigma)\right). \tag{6}$$

9

Thus, as long as we choose the dimension of the random projection on the order of d (up to log-factors), the continual learning algorithm probably enables obtaining an estimator that has near-optimal risk, since this ensure that the RHS of (6) is less than a small constant times $\|\boldsymbol{\theta}_A\|_2 + \|\boldsymbol{\theta}_B\|_2 + \sigma$, which applied to (4) implies a near-optimal risk (since $R(\boldsymbol{\theta}_t) \leq R(\theta^*) + 2\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|_2 \left(\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|_2 + \|\boldsymbol{\theta}_t - \boldsymbol{\theta}_A\|_2 + \|\boldsymbol{\theta}_t - \boldsymbol{\theta}_B\|_2\right)$).

5.2 Continual learning on more than two regression tasks

The analysis from the previous section shows that if we learn two regression tasks sequentially, we obtain an estimate that is accurate up to two error terms (cf. (6)): The first one is the statistical error due to learning on finite data. The second term is because we are not learning based on the original data of the first task A. If we move from two tasks to three and more the effect of those approximations compounds which makes it difficult to learn a large number of tasks sequentially. This explains why in the simulations shown previously for a linear random feature model (see Fig. 2) the gap between training on all of the data and the RSJ algorithm increases in the number of tasks, and this gap becomes smaller if the approximation becomes better.

6 Guarantees for two-layer neural networks in the NTK regime

We now provide guarantees for wide neural networks in the so called neural-tangent-kernel (NTK) regime, in which the networks behave approximately linearly, as established by recent results [JGH18; Lee+18; Du+18; OS20].

Consider a two-layer neural network with ReLU activation functions and k neurons in the hidden layer:

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{1}{\sqrt{k}}\mathrm{relu}(\mathbf{x}^T\boldsymbol{\Theta})\mathbf{v}. \tag{7}$$

Here, $\mathbf{x} \in \mathbb{R}^d$ is the input of the network, $\boldsymbol{\Theta} \in \mathbb{R}^{d \times k}$ are the trainable weights of the first layer and $\mathbf{v} \in \mathbb{R}^k$ are fixed second-layer weights with the first half equal to 1 and the second half equal to -1. The trainable parameters are the weight-matrix $\boldsymbol{\Theta}$, and we denote with $\boldsymbol{\theta}$ the vectorized version of this matrix.

The training data for task T, for $T \in \{A, B\}$, consists of n points drawn iid from an (unknown) joint distribution $(\mathbf{x}_i, y_i) \sim P_T$, and we assume for convenience that the data points are normalized ($\|\mathbf{x}\|_2 = 1$) and that the labels are bounded ($|y_i| \leq 1$).

Similarly as in the previous section, we consider the composite risk

$$R(\boldsymbol{\theta}) = \frac{1}{2}R_A(\boldsymbol{\theta}) + \frac{1}{2}R_B(\boldsymbol{\theta}) \tag{8}$$

where

$$R_T(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x},y)\sim P_T}\left[\ell(f_{\boldsymbol{\theta}}(\mathbf{x}), y)\right].$$

Here, $\ell: \mathbb{R} \times \mathbb{R} \to [0, 1]$ is a loss function that is 1-Lipschitz in its first argument and obeys $\ell(y, y) = 0$; an example of such a function is $\ell(z, y) = |z - y|$. For the composite risk to be small, both the risks of task A and B have to be small.

The following theorem quantifies the risk of the RSJ algorithm with the loss minimized via t-many iterations of gradient descent. Specifically, we first run gradient descent on the loss of task A, defined in equation (1) until convergence, and then run t iterations on the RSJ-loss (2).

Our risk bound depends on the Gram matrix $\mathbf{K} \in \mathbb{R}^{2n \times 2n}$ with entries defined as

$$[\mathbf{K}]_{ij} = \frac{1}{2} \left(1 - \frac{\cos^{-1}\left(\langle \mathbf{x}_i, \mathbf{x}_j \rangle \right)}{\pi} \right) \langle \mathbf{x}_i, \mathbf{x}_j \rangle,$$

where (i, j) are the pairs of training data points from the two tasks.

Theorem 2. *Let $\alpha > 0$ be the smallest singular value of the Gram matrix $\mathbf{K}$, and consider the network in the NTK regime where $k \to \infty$. Let $f_{\boldsymbol{\theta}_t}$ be the network trained with t iterations of gradient descent applied to the loss of the RSJ algorithm. Then with probability at least $1 - 4t \exp(-\frac{\|\mathbf{J}_A\|_F^2}{2\|\mathbf{J}_A\|^2})$, the risk of the network trained with at least $t \geq \frac{\log(1-\eta\alpha)}{\log(1/n)}$ gradient iterations is bounded by*

$$R(\boldsymbol{\theta}_t) \leq 2\sqrt{\frac{1}{n} \mathbf{y}^T \mathbf{K}^{-1} \mathbf{y}} + \frac{3}{\sqrt{n}} + \frac{1}{\sqrt{s}\alpha^2} \left(10\|\mathbf{J}_A\|_F + \frac{\|\mathbf{K}\|_F}{\sqrt{n}} \right), \tag{9}$$

where $\mathbf{y} = [\mathbf{y}_A, \mathbf{y}_B] \in \mathbb{R}^{2n}$ contains the labels of the training data of the two tasks.

The theorem establishes that the risk is bounded by a complexity measure of the data defined as $\sqrt{\frac{1}{n} \mathbf{y}^T \mathbf{K}^{-1} \mathbf{y}}$, plus two perturbation terms, one of which depends on the quality of the random projection. The complexity measure of the data has been studied by Arora et al. [Aro+19], and measures whether the training set of tasks A and B can be well represented with the kernel associated with the relu-network. If those training sets can be well represented with this kernel, the complexity measure $\sqrt{\frac{1}{n} \mathbf{y}^T \mathbf{K}^{-1} \mathbf{y}}$ is small. See [Aro+19] for a detailed discussion on the interpretation of this complexity measure.

For our purpose, it is important to note that if we were to train the neural network on the data from task A and B simultaneously, we would obtain the same risk bound as above but without the perturbation term in equation (9) starting with $\frac{1}{\sqrt{s}\alpha^2}$. The perturbation term is small if the dimension of the random projection is on the order of the effective rank of the Jacobian $\mathbf{J}_A$; this term can be computed from the data and thus we can verify whether we are choosing the random projection dimension s sufficiently large.

7 Negative results for learning Gaussian mixture models sequentially with EWC

In this section we study the popular data permutation experiment theoretically, and show why EWC and L2-regularization *can* perform well even in a setup where the diagonal of the Jacobian outer product is a poor approximation of the original Jacobian outer product (recall the experiment in Figure 1).

Consider a binary version of the MNIST permutation experiment. Task A is to classify the digits $\{0, 1\}$ from the original feature vectors (i.e., the vectorized 28×28 pixel images), task B to classify the digits $\{0, 1\}$ from the vectorized images permuted with a random perturbation, fixed for task B, and task C is generated in the same way, with a new permutation.
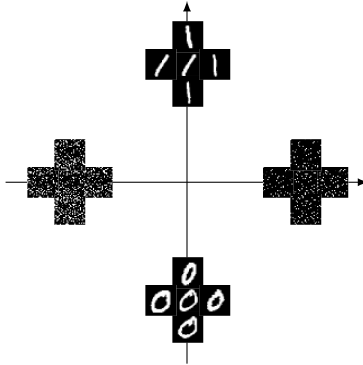
Figure 3: A visualization of the permutation task: The first axis shows task A of distinguishing ones and zeros. The second axis shows task B of distinguishing shuffled ones and zeros (each image is shuffled with the same random permutation). Due to the random shuffling, the two axis are approximately orthogonal. We therefore model this with a Gaussian mixture model where the two tasks are orthogonal to each other.

A simple mathematical abstraction for this is a Gaussian mixture model where the class means of the two classes point in different directions. In the MNIST permutation experiment, task B is obtained by simply shuffling each of the feature vectors (the pixels) with a random permutation.

This can be modeled with a new task B of the Gaussian mixture model with class means that are near-orthogonal to the means of task A. This is illustrated in Figure 3. Motivated by this observation, we study the following Gaussian mixture model for continual learning.

Gaussian mixture model for continual learning: Consider the standard binary classification setting, where the data for each task is distributed as a mixture of two Gaussians. For task $T = A, B, \ldots$, the response $y \in \{-1, 1\}$ is uniformly distributed, and the feature vector $\mathbf{x}$ given the class label y is distributed as $\mathbf{x}|y \sim \mathcal{N}(y\boldsymbol{\mu}_T, \sigma^2\mathbf{I})$. Here, $\boldsymbol{\mu}_T \in \mathbb{R}^d$ is a fixed class mean vector with unit norm, and $\sigma^2 > 0$ is the within-class variance. The Bayes optimal classifier for a given task is $\hat{y}(\mathbf{x}) = \text{sign}(\langle \boldsymbol{\mu}_T, \mathbf{x} \rangle)$.

Consider the linear classifier $\hat{y}_{\boldsymbol{\theta}}(\mathbf{x}) = \text{sign}(\langle \boldsymbol{\theta}, \mathbf{x} \rangle)$. The corresponding Bayes risk on task T is

$$R_T(\boldsymbol{\theta}) = \text{P}_{(\mathbf{x},y)\sim P_T} \left[1\{\hat{y}(\mathbf{x}) \neq y\} \right] = \Phi\left(\frac{\langle \boldsymbol{\theta}, \boldsymbol{\mu}_T \rangle}{\|\boldsymbol{\theta}\|_2 \sigma} \right),$$

where $\Phi(x) = (2\pi)^{-1/2} \int_{-\infty}^{x} e^{-t^2/2} dt$ is the cumulative distribution function of the standard normal distribution. Our goal is to learn the tasks sequentially so that the risk of a task chosen with equal probability is small. Specifically, we wish the risk

$$R_{A:T}(\boldsymbol{\theta}) = \frac{1}{T} \left(R_A(\boldsymbol{\theta}) + R_B(\boldsymbol{\theta}) + \ldots + R_T(\boldsymbol{\theta}) \right)$$

to be small. For simplicity, we consider two tasks A and B. The optimal $\boldsymbol{\theta}$ that minimizes the risk of tasks A and B, i.e., R_{AB} is is a linear combination of the two optimal points for the individual tasks $\boldsymbol{\theta}_{AB}^* = \boldsymbol{\mu}_A + \boldsymbol{\mu}_B$.

We next identify a setup in which the EWC algorithm and L2 regularization provably succeed in learning tasks A and B sequentially. The EWC algorithm in our formulation is trained with

the quadratic loss, and not with another suitable loss such as cross-entropy, and we consider the infinite-data case, i.e., we study EWC and L2 applied to the population risk, so that we do not have to take finite-data effects into account.

Theorem 3. *Suppose that the class means $\boldsymbol{\mu}_A, \boldsymbol{\mu}_B \in \mathbb{R}^d$ lie on a hypercube, i.e., each entry has equal magnitude $1/d$ and that the inner product between the class means is non-negative ($\langle \boldsymbol{\mu}_A, \boldsymbol{\mu}_B \rangle \geq 0$). Then:*

   i) There is a choice of regularization parameter λ, such that EWC is optimal (i.e., its solution minimizes the Bayes risk).

   ii) There is a choice of regularization parameter λ, such that L2 is optimal.

This result allows the diagonal of the Jacobian outer product (i.e., the Hessian for this setup) to be a poor approximation of the Jacobian outer product. Thus, perhaps surprisingly, there are setups where EWC (and L2-regularization) provably succeeds even when the penalty of EWC (or $L2$-regularization) is a poor approximation of the loss on past data.

The MNIST permutation problem approximately corresponds to a setup where the class mean are on a hypercube (because most pixels are one or zero) and therefore the theorem gives a potential justification why EWC works so well on this problem.

At the same time, there exist, as expected, problem instances where EWC and L2 regularization provably fail at continual learning, even if given infinitely many training examples:

Theorem 4. *There are problem instance of the Gaussian mixture model in $\mathbb{R}^d, d \geq 3$ (i.e., a choice of $\boldsymbol{\mu}_A, \boldsymbol{\mu}_B$, and σ^2) such that:*

   i) The risk of EWC for all values of the regularization parameter λ is at least $3/2$-times the optimal risk: $R_{AB}(\boldsymbol{\theta}_{EWC}) \geq 3/2 R_{AB}(\boldsymbol{\theta}_{AB}^*)$.

   ii) The risk of L2-regularization for all values of the regularization parameter λ is $3/2$-times the optimal risk: $R_{AB}(\boldsymbol{\theta}_{L2}) \geq 3/2 R_{AB}(\boldsymbol{\theta}_{AB}^*)$.

8   Conclusion

In this paper we studied a family of algorithms in order to understand regularization based continual learning algorithms. We showed that popular regularization-based learning algorithms, in particular EWC, can provably fail if the regularization does not approximate the loss for past data well. We also showed that this can be fixed by working with better approximations of the Jacobian. However, while the resulting algorithm provably succeeds, it might comes at a high memory cost. Our results indicate that regularization-based continual learning algorithms that provably succeed for a variety of setups might need to have a large memory footprint in return.

Acknowledgements

Code

Code to reproduce the experiments in this paper is at
https://github.com/MLI-lab/regularization_based_continual_learning.

References

[ANW12]   A. Agarwal, S. Negahban, and M. J. Wainwright. "Fast Global Convergence of Gradient Methods for High-Dimensional Statistical Recovery". In: *Annals of Statistics* 40.5 (2012), pp. 2452–2482.

[AMP17]   P. Alquier, T. T. Mai, and M. Pontil. "Regret Bounds for Lifelong Learning". In: *International Conference on Artificial Intelligence and Statistics*. 2017, pp. 261–269.

[Aro+19] S. Arora, S. Du, W. Hu, Z. Li, and R. Wang. "Fine-Grained Analysis of Optimization and Generalization for Overparameterized Two-Layer Neural Networks". In: *International Conference on Machine Learning*. 2019, pp. 322–332.

[BDS20]   M. A. Bennani, T. Doan, and M. Sugiyama. "Generalisation Guarantees for Continual Learning with Orthogonal Gradient Descent". In: *arXiv:2006.11942 [cs, stat]* (2020).

[Cha+18] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. S. Torr. "Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 532–547.

[Doa+21] T. Doan, M. A. Bennani, B. Mazoure, G. Rabusseau, and P. Alquier. "A Theoretical Analysis of Catastrophic Forgetting through the NTK Overlap Matrix". In: *International Conference on Artificial Intelligence and Statistics*. 2021, pp. 1072–1080.

[Du+18]   S. S. Du, X. Zhai, B. Poczos, and A. Singh. "Gradient Descent Provably Optimizes Over-Parameterized Neural Networks". In: *International Conference on Learning Representations*. 2018.

[Fre99]   R. M. French. "Catastrophic Forgetting in Connectionist Networks". In: *Trends in Cognitive Sciences* 3.4 (1999), pp. 128–135.

[Goo+15] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio. "An Empirical Investigation of Catastrophic Forgetting in Gradient-Based Neural Networks". In: *arXiv:1312.6211 [cs, stat]* (2015).

[HY21]   R. Heckel and F. F. Yilmaz. "Early Stopping in Deep Networks: Double Descent and How to Eliminate It". In: *International Conference on Learning Representations*. 2021.

[Hsu+19] Y.-C. Hsu, Y.-C. Liu, A. Ramasamy, and Z. Kira. "Re-Evaluating Continual Learning Scenarios: A Categorization and Case for Strong Baselines". In: *arXiv:1810.12488 [cs]* (2019).

[JGH18]   A. Jacot, F. Gabriel, and C. Hongler. "Neural Tangent Kernel: Convergence and Generalization in Neural Networks". In: *Neural Information Processing Systems*. 2018, pp. 8571–8580.

[Kem+18] R. Kemker, M. McClure, A. Abitino, T. Hayes, and C. Kanan. "Measuring Catastrophic Forgetting in Neural Networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 2018.

[Kir+17]   J. Kirkpatrick et al. "Overcoming Catastrophic Forgetting in Neural Networks". In: *Proceedings of the National Academy of Sciences* 114.13 (2017), pp. 3521–3526.

[KBH19]   F. Kunstner, L. Balles, and P. Hennig. "Limitations of the Empirical Fisher Approximation for Natural Gradient Descent". In: *Neural Information Processing Systems*. 2019.

[Lee+18]   J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein. "Deep Neural Networks as Gaussian Processes". In: *International Conference on Learning Representations*. 2018.

[Li+21]   H. Li, A. Krishnan, J. Wu, S. Kolouri, P. K. Pilly, and V. Braverman. "Lifelong Learning with Sketched Structural Regularization". In: *Asian Conference on Machine Learning*. 2021.

[LH18]   Z. Li and D. Hoiem. "Learning without Forgetting". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.12 (2018), pp. 2935–2947.

[Liu+18]   X. Liu, M. Masana, L. Herranz, J. V. de Weijer, A. M. López, and A. D. Bagdanov. "Rotate Your Networks: Better Weight Consolidation and Less Catastrophic Forgetting". In: *2018 24th International Conference on Pattern Recognition (ICPR)*. 2018, pp. 2262–2268.

[MRT12]   M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. MIT Press, 2012.

[OS20]   S. Oymak and M. Soltanolkotabi. "Towards Moderate Overparameterization: Global Convergence Guarantees for Training Shallow Neural Networks". In: *IEEE Journal on Selected Areas in Information Theory* (2020).

[Pan+21]   P. Pan, S. Swaroop, A. Immer, R. Eschenhagen, R. E. Turner, and M. E. Khan. "Continual Deep Learning by Functional Regularisation of Memorable Past". In: *arXiv:2004.14070 [cs, stat]* (2021).

[PW15]   M. Pilanci and M. J. Wainwright. "Randomized Sketches of Convex Programs With Sharp Guarantees". In: *IEEE Transactions on Information Theory* 61.9 (2015), pp. 5096–5115.

[Reb+17]   S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. "iCaRL: Incremental Classifier and Representation Learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2001–2010.

[RBB18]   H. Ritter, A. Botev, and D. Barber. "Online Structured Laplace Approximations for Overcoming Catastrophic Forgetting". In: *Advances in Neural Information Processing Systems* 31 (2018), pp. 3738–3748.

[Rob95]   A. Robins. "Catastrophic Forgetting, Rehearsal and Pseudorehearsal". In: *Connection Science* 7.2 (1995), pp. 123–146.

[RV10]   M. Rudelson and R. Vershynin. "Non-Asymptotic Theory of Random Matrices: Extreme Singular Values". In: *Proceedings of the International Congress of Mathematicians*. Vol. 3. 2010, pp. 1576–1602.

[Rus+16]   A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell. "Progressive Neural Networks". In: *arXiv:1606.04671 [cs]* (2016).

[RE13]   P. Ruvolo and E. Eaton. "ELLA: An Efficient Lifelong Learning Algorithm". In: *International Conference on Machine Learning*. 2013, pp. 507–515.

[Sar06]   T. Sarlos. "Improved Approximation Algorithms for Large Matrices via Random Projections". In: *IEEE Symposium on Foundations of Computer Science (FOCS'06)*. 2006, pp. 143–152.

[Sch+18]   J. Schwarz, W. Czarnecki, J. Luketina, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu, and R. Hadsell. "Progress & Compress: A Scalable Framework for Continual Learning". In: *International Conference on Machine Learning*. 2018, pp. 4528–4537.

[Shi+17]   H. Shin, J. K. Lee, J. Kim, and J. Kim. "Continual Learning with Deep Generative Replay". In: *Neural Information Processing Systems*. Vol. 30. 2017.

[vT19]   G. M. van de Ven and A. S. Tolias. "Three Scenarios for Continual Learning". In: *arXiv:1904.07734 [cs, stat]* (2019).

[Yin+20]   D. Yin, M. Farajtabar, A. Li, N. Levine, and A. Mott. "Optimization and Generalization of Regularization-Based Continual Learning: A Loss Approximation Viewpoint". In: *arXiv:2006.10974 [cs, stat]* (2020).

[ZPG17]   F. Zenke, B. Poole, and S. Ganguli. "Continual Learning Through Synaptic Intelligence". In: *International Conference on Machine Learning*. 2017, pp. 3987–3995.

A Proof of Proposition 1

For a linear model, the Jacobian $\mathbf{J}_T \in \mathbb{R}^{n \times p}$ contains the feature maps $\psi(\mathbf{x}_i)$ as rows. With this notation, the loss function of task A becomes

$$\mathcal{L}_A(\boldsymbol{\theta}) = \frac{1}{2}\|\mathbf{J}_A\boldsymbol{\theta} - \mathbf{y}_A\|_2^2,$$

where $\mathbf{y}_A \in \mathbb{R}^n$ are the responses of task A. The minimizer $\boldsymbol{\theta}_A$ obeys

$$\mathbf{J}_A^T\mathbf{J}_A\boldsymbol{\theta}_A = \mathbf{J}_A^T\mathbf{y}_A.$$

Next, consider learning B after A and note that

$$\min_{\boldsymbol{\theta}} \mathcal{L}_{AB}(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \frac{1}{2}(\mathbf{J}_B\boldsymbol{\theta} - \mathbf{y}_B)^T(\mathbf{J}_B\boldsymbol{\theta} - \mathbf{y}_B) + \frac{\lambda}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_A)^T\mathbf{J}_A^T\mathbf{J}_A(\boldsymbol{\theta} - \boldsymbol{\theta}_A).$$

$$= \min_{\boldsymbol{\theta}} \frac{1}{2}\|\mathbf{J}_B\boldsymbol{\theta} - \mathbf{y}_B\|_2^2 + \frac{\lambda}{2}\boldsymbol{\theta}^T\mathbf{J}_A^T\mathbf{J}_A\boldsymbol{\theta} - \lambda\boldsymbol{\theta}^T\mathbf{J}_A^T\mathbf{J}_A\boldsymbol{\theta}_A^T$$

$$= \min_{\boldsymbol{\theta}} \frac{1}{2}\|\mathbf{J}_B\boldsymbol{\theta} - \mathbf{y}_B\|_2^2 + \frac{\lambda}{2}\boldsymbol{\theta}^T\mathbf{J}_A^T\mathbf{J}_A\boldsymbol{\theta} - \lambda\boldsymbol{\theta}^T\mathbf{J}_A^T\mathbf{y}_A$$

$$= \min_{\boldsymbol{\theta}} \frac{1}{2}\|\mathbf{J}_B\boldsymbol{\theta} - \mathbf{y}_B\|_2^2 + \frac{\lambda}{2}\|\mathbf{J}_A\boldsymbol{\theta} - \mathbf{y}_A\|_2^2.$$

The analogous argument shows that the minimizer of $\mathcal{L}_{ABC}(\boldsymbol{\theta})$ is equal to the minimizer of $\mathcal{L}_A(\boldsymbol{\theta}) + \mathcal{L}_B(\boldsymbol{\theta}) + \mathcal{L}_C(\boldsymbol{\theta})$, and likewise for more tasks.

B Proof of Theorem 1: Analysis for sketched least-squares

In this section we prove Theorem 1 by showing that under certain conditions, the solution to a sketched least squares is close to that of an associated non-sketched one.

B.1 Proof of Theorem 1, part i

We consider the least-squares objective

$$\mathcal{L}_{AB}(\boldsymbol{\theta}) = \mathcal{L}_B(\boldsymbol{\theta}) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_A)^T\mathbf{J}_A^T\mathbf{S}^T\mathbf{S}\mathbf{J}_A(\boldsymbol{\theta} - \boldsymbol{\theta}_A)$$

$$= \frac{1}{2}\boldsymbol{\theta}^T\mathbf{J}_B^T\mathbf{J}_B\boldsymbol{\theta} - \boldsymbol{\theta}^T\mathbf{J}_B^T\mathbf{y}_B + \frac{1}{2}\boldsymbol{\theta}^T\mathbf{J}_A^T\mathbf{S}^T\mathbf{S}\mathbf{J}_A\boldsymbol{\theta} - \boldsymbol{\theta}^T\mathbf{J}_A^T\mathbf{S}^T\mathbf{S}\mathbf{J}_A\boldsymbol{\theta}_A + c$$

$$= \frac{1}{2}\boldsymbol{\theta}^T \begin{bmatrix} \mathbf{J}_A^T\mathbf{S}^T, \mathbf{J}_B^T \end{bmatrix} \begin{bmatrix} \mathbf{S}\mathbf{J}_A \\ \mathbf{J}_B \end{bmatrix} \boldsymbol{\theta} - \boldsymbol{\theta}^T \begin{bmatrix} \mathbf{J}_A^T\mathbf{S}^T, \mathbf{J}_B^T \end{bmatrix} \begin{bmatrix} \mathbf{S}\mathbf{J}_A\boldsymbol{\theta}_A \\ \mathbf{y}_B \end{bmatrix} + c$$

$$= \frac{1}{2}\boldsymbol{\theta}^T\mathbf{J}^T\mathbf{P}\mathbf{J}\boldsymbol{\theta} - \boldsymbol{\theta}^T\mathbf{J}^T\mathbf{P}\mathbf{y} + c.$$

Here, c is a numerical constant, independent of the optimization parameter $\boldsymbol{\theta}$, and we defined

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_A \\ \mathbf{J}_B \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} \mathbf{S}^T\mathbf{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \mathbf{J}_A\boldsymbol{\theta}_A \\ \mathbf{y}_B \end{bmatrix}, \tag{10}$$

for notational convenience.

The gradient descent iterates with stepsize η for minimizing the loss $\mathcal{L}_{AB}(\boldsymbol{\theta})$ are

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta(\mathbf{J}^T \mathbf{PJ}\boldsymbol{\theta}^t - \mathbf{J}^T \mathbf{Py}).$$

We bound the difference to the gradient iterates for minimizing the least squares objective without the random projection matrix

$$\tilde{\mathcal{L}}_{AB}(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\theta}^T \mathbf{J}^T \mathbf{J}\boldsymbol{\theta} + \boldsymbol{\theta}^T \mathbf{J}^T \mathbf{y}.$$

The gradients descent iterates for minimizing the loss $\tilde{\mathcal{L}}_{AB}(\boldsymbol{\theta})$ are

$$\tilde{\boldsymbol{\theta}}^{t+1} = \tilde{\boldsymbol{\theta}}^t - \eta(\mathbf{J}^T \mathbf{J}\tilde{\boldsymbol{\theta}}^t - \mathbf{J}^T \mathbf{y}).$$

The following lemma bounds the deviation of the two versions of gradient descent.

Lemma 1. *Let $\mathbf{J} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ be the singular value decomposition of $\mathbf{J}$, let $\sigma_{\max}$ and $\sigma_{\min}$ the largest and smallest singular values, and let $\mathbf{U}_r$ and $\mathbf{U}_n$ be the left-singular vectors corresponding to the r-largest and the remaining singular values. Let $\boldsymbol{\theta}_t$ and $\tilde{\boldsymbol{\theta}}_t$ be the gradient descent iterates after t iterations starting at $\boldsymbol{\theta}_0 = \tilde{\boldsymbol{\theta}}_0 = \mathbf{0}$. With probability at least $1 - 4t \exp(-\frac{\|\mathbf{J}_A\|_F^2}{2\|\mathbf{J}_A\|^2})$ over the random sketch $\mathbf{S} \in \mathbb{R}^{s\times n}$, the difference of the iterates of the original and the sketched problem is bounded by*

$$\left\|\boldsymbol{\theta}^t - \tilde{\boldsymbol{\theta}}^t\right\|_2 \leq 5\frac{\|\mathbf{J}_A\|_F}{\sqrt{s}}\left(\frac{1 - (1 - \eta\sigma_r^2)^t}{\sigma_r^2}\left\|\mathbf{U}_r^T\mathbf{y}\right\|_2 + \frac{1 - (1 - \eta\sigma_{\min}^2)^t}{\sigma_{\min}^2}\left\|\mathbf{U}_n^T\mathbf{y}\right\|_2\right).$$

Theorem 1 follows from the lemma, by using the assumption $(1 - \eta\sigma_{\min}^2)^t \geq 1 - \frac{\sigma_{\min}^2}{\sigma_r^2}\frac{\|\mathbf{U}_r^T\mathbf{y}\|_2}{\|\mathbf{U}_n^T\mathbf{y}\|_2}$ to conclude

$$\left\|\boldsymbol{\theta}^t - \tilde{\boldsymbol{\theta}}^t\right\|_2 \leq 5\frac{\|\mathbf{J}\|_F}{\sqrt{s}}\left(\frac{1}{\sigma_r^2}\left\|\mathbf{U}_r^T\mathbf{r}_0\right\|_2 + \frac{1}{\sigma_{\min}^2}\left\|\mathbf{U}_n^T\mathbf{r}_0\right\|_2 \frac{\sigma_{\min}^2}{\sigma_r^2}\frac{\|\mathbf{U}_r^T\mathbf{y}\|_2}{\|\mathbf{U}_n^T\mathbf{y}\|_2}\right)$$

$$= 5\frac{\|\mathbf{J}\|_F}{\sqrt{s}}\frac{1}{\sigma_r^2}\left\|\mathbf{U}_r^T\mathbf{r}_0\right\|_2.$$

In the reminder of this section we prove Lemma 1.

B.2 Proof of Lemma 1

The difference between the two iterates is bounded by

$$\left\|\boldsymbol{\theta}^{t+1} - \tilde{\boldsymbol{\theta}}^{t+1}\right\|_2 = \left\|\boldsymbol{\theta}^t - \eta(\mathbf{J}^T\mathbf{PJ}\boldsymbol{\theta}^t - \mathbf{J}^T\mathbf{Py}) - \left(\tilde{\boldsymbol{\theta}}^t - \eta(\mathbf{J}^T\mathbf{J}\tilde{\boldsymbol{\theta}}^t - \mathbf{J}^T\mathbf{y})\right)\right\|_2$$

$$= \left\|(\mathbf{I} - \eta\mathbf{J}^T\mathbf{PJ})\boldsymbol{\theta}^t - (\mathbf{I} - \eta\mathbf{J}^T\mathbf{J})\tilde{\boldsymbol{\theta}}^t - \eta(\mathbf{J}^T\mathbf{y} - \mathbf{J}^T\mathbf{Py})\right\|_2$$

$$= \left\|(\mathbf{I} - \eta\mathbf{J}^T\mathbf{PJ})\boldsymbol{\theta}^t - (\mathbf{I} - \eta\mathbf{J}^T\mathbf{PJ})\tilde{\boldsymbol{\theta}}^t + (\mathbf{I} - \eta\mathbf{J}^T\mathbf{PJ})\tilde{\boldsymbol{\theta}}^t - (\mathbf{I} - \eta\mathbf{J}^T\mathbf{J})\tilde{\boldsymbol{\theta}}^t - \eta(\mathbf{J}^T\mathbf{y} - \mathbf{J}^T\mathbf{Py})\right\|_2$$

$$= \left\|(\mathbf{I} - \eta\mathbf{J}^T\mathbf{PJ})(\boldsymbol{\theta}^t - \tilde{\boldsymbol{\theta}}^t) + \eta(\mathbf{J}^T\mathbf{J} - \mathbf{J}^T\mathbf{PJ})\tilde{\boldsymbol{\theta}}^t - \eta(\mathbf{J}^T\mathbf{y} - \mathbf{J}^T\mathbf{Py})\right\|_2$$

$$\leq \left\|\mathbf{I} - \eta\mathbf{J}^T\mathbf{PJ}\right\|\left\|\boldsymbol{\theta}^t - \tilde{\boldsymbol{\theta}}^t\right\|_2 + \eta\left\|\mathbf{J}^T(\mathbf{I} - \mathbf{P})(\mathbf{J}\tilde{\boldsymbol{\theta}}^t - \mathbf{y})\right\|_2$$

$$\overset{(i)}{\leq} \left\|\boldsymbol{\theta}^t - \tilde{\boldsymbol{\theta}}^t\right\|_2 + \eta\left\|\mathbf{J}^T(\mathbf{I} - \mathbf{P})(\mathbf{J}\tilde{\boldsymbol{\theta}}^t - \mathbf{y})\right\|_2, \tag{11}$$

where inequality (i) holds for a sufficiently small stepsize, specifically for a stepsize smaller than $\eta \leq \frac{1}{\sigma_{\max}(\mathbf{J}^T\mathbf{PJ})}$. Note that a sufficiently small stepsize is required for gradient descent to converge. We next bound the term on the RHS. Using that

$$\mathbf{I} - \mathbf{P} = \begin{bmatrix} \mathbf{I} - \mathbf{S}^T\mathbf{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

and the definition of $\mathbf{J}$ and $\mathbf{y}$ in equation (10), we get

$$
\begin{aligned}
\left\| \mathbf{J}^T(\mathbf{I}-\mathbf{P})(\mathbf{J}\tilde{\boldsymbol{\theta}}^t - \mathbf{y}) \right\|_2 &= \left\| \mathbf{J}_A^T(\mathbf{I}-\mathbf{S}^T\mathbf{S})\mathbf{J}_A(\tilde{\boldsymbol{\theta}}^t - \boldsymbol{\theta}_A) \right\|_2 \\
&\overset{(i)}{\leq} 8\frac{\|\mathbf{J}\|_F}{\sqrt{s}} \left\| \mathbf{J}_A(\tilde{\boldsymbol{\theta}}^t - \boldsymbol{\theta}_A) \right\|_2 \\
&\overset{(ii)}{\leq} 8\frac{\|\mathbf{J}_A\|_F}{\sqrt{s}} \|\tilde{\mathbf{r}}_t\|_2.
\end{aligned}
\tag{12}
$$

Here, inequality (i) holds with probability at least $1-4e^{-\frac{\|\mathbf{J}_A\|_F^2}{2\|\mathbf{J}_A\|^2}}$, as established by the lemma below. The lemma below applies to a setup where the vector $\mathbf{z} = \mathbf{J}_A(\tilde{\boldsymbol{\theta}}^t - \boldsymbol{\theta}_A)$ is independent of the random matrix $\mathbf{S}$. The vector $\mathbf{z}$ is independent of the random matrix $\mathbf{S}$ since $\tilde{\boldsymbol{\theta}}^t$ are the non-sketched gradient iterations. Moreover, inequality (ii) follows by using that $\left\| \mathbf{J}_A(\tilde{\boldsymbol{\theta}}^t - \boldsymbol{\theta}_A) \right\|_2 \leq \left\| \mathbf{J}\tilde{\boldsymbol{\theta}}^t - \mathbf{y} \right\|_2 = \|\tilde{\mathbf{r}}_t\|_2$, where we defined the residual $\tilde{\mathbf{r}}_t = \mathbf{J}\tilde{\boldsymbol{\theta}}^t - \mathbf{y}$.

Lemma 2. *For any $\mathbf{J}$ and any $\mathbf{z} \in \mathbb{R}^n$, and for $\mathbf{S} \in \mathbb{R}^{s \times n}$ a random projection matrix with iid $\mathcal{N}(0, 1/s)$ entries, we have that*

$$\mathrm{P}\left[\left\| \mathbf{J}^T(\mathbf{I}-\mathbf{S}^T\mathbf{S})\mathbf{z} \right\|_2 \leq 8\|\mathbf{J}\|_F \|\mathbf{z}\|_2 \frac{1}{\sqrt{s}} \right] \geq 1 - 4e^{-\frac{\|\mathbf{J}\|_F^2}{2\|\mathbf{J}\|^2}}$$

Application of the bound (12) to inequality (11), establishes that

$$\left\| \boldsymbol{\theta}^{t+1} - \tilde{\boldsymbol{\theta}}^{t+1} \right\|_2 \leq \left\| \boldsymbol{\theta}^t - \tilde{\boldsymbol{\theta}}^t \right\|_2 + 8\frac{\|\mathbf{J}_A\|_F}{\sqrt{s}} \eta \|\tilde{\mathbf{r}}_t\|_2,$$

with probability at least $1 - 4e^{-\frac{\|\mathbf{J}_A\|_F^2}{2\|\mathbf{J}_A\|^2}}$. Applying the union bound over t iterations, it follows that, with probability at least $1 - 4te^{-\frac{\|\mathbf{J}_A\|_F^2}{2\|\mathbf{J}_A\|^2}}$,

$$\left\| \boldsymbol{\theta}^t - \tilde{\boldsymbol{\theta}}^t \right\|_2 \leq 8\frac{\|\mathbf{J}_A\|_F}{\sqrt{s}} \eta \sum_{\tau=0}^{t-1} \|\tilde{\mathbf{r}}_\tau\|_2.
\tag{13}$$

We next bound the sum of the residuals above. Let $\mathbf{J} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ be the singular value decomposition of $\mathbf{J}$, and note that

$$
\begin{aligned}
\tilde{\mathbf{r}}_t &= \mathbf{J}\boldsymbol{\theta}^t - \mathbf{y} \\
&= (\mathbf{I} - \eta\mathbf{J}^T\mathbf{J})^t\tilde{\mathbf{r}}_0 \\
&= \mathbf{U}(\mathbf{I} - \eta\boldsymbol{\Sigma}^2)^t\mathbf{U}^T\tilde{\mathbf{r}}_0.
\end{aligned}
$$

Let $\mathbf{U}_r$ and $\mathbf{U}_n$ be the singular vectors corresponding to the r-leading and the other singular values. With this notation, we have that

$$\|\tilde{\mathbf{r}}_\tau\|_2 \le (1 - \eta\sigma_r^2)^2 \|\mathbf{U}_r^T\tilde{\mathbf{r}}_0\|_2 + (1 - \eta\sigma_{\min}^2)^2 \|\mathbf{U}_n^T\tilde{\mathbf{r}}_0\|_2$$

We therefore can proceed with bounding the RHS of (13) as

$$\left\|\boldsymbol{\theta}^t - \tilde{\boldsymbol{\theta}}^t\right\|_2 \le 8\frac{\|\mathbf{J}_A\|_F}{\sqrt{s}}\eta \sum_{\tau=0}^{t-1}(1 - \eta\sigma_r^2)^\tau \|\mathbf{U}_r^T\tilde{\mathbf{r}}_0\|_2 + (1 - \eta\sigma_{\min}^2)^2 \|\mathbf{U}_n^T\tilde{\mathbf{r}}_0\|_2$$

$$= 8\frac{\|\mathbf{J}_A\|_F}{\sqrt{s}}\eta \left(\frac{1 - (1 - \eta\sigma_r^2)^t}{\eta\sigma_r^2}\|\mathbf{U}_r^T\tilde{\mathbf{r}}_0\|_2 + \frac{1 - (1 - \eta\sigma_{\min}^2)^t}{\eta\sigma_{\min}^2}\|\mathbf{U}_n^T\tilde{\mathbf{r}}_0\|_2\right),$$

where the last inequality follows from the formula of a geometric series. This concludes the proof of Lemma 1.

B.3   Proof of Lemma 2

It remains to prove Lemma 2. Towards this goal, let $\mathbf{P} \in \mathbb{R}^{n\times n}$ be a orthonormal projection onto $\mathbf{z}$, and let $\mathbf{P}_\perp \in \mathbb{R}^{n\times n}$ be a orthonormal projection on the orthogonal complement. With this notation, we have

$$\left\|\mathbf{J}^T(\mathbf{I} - \mathbf{S}^T\mathbf{S})\mathbf{z}\right\|_2 = \left\|\mathbf{J}^T(\mathbf{P} + \mathbf{P}_\perp)(\mathbf{I} - \mathbf{S}^T\mathbf{S})\mathbf{z}\right\|_2$$

$$\overset{(i)}{\le} \left\|\mathbf{J}^T\mathbf{P}(\mathbf{I} - \mathbf{S}^T\mathbf{S})\mathbf{z}\right\|_2 + \left\|\mathbf{J}^T\mathbf{P}_\perp\mathbf{S}^T\mathbf{S}\mathbf{z}\right\|_2,$$

$$\le \|\mathbf{z}\|_2\epsilon\|\mathbf{J}\| + 4\|\mathbf{J}\|_F\frac{\|\mathbf{z}\|_2}{\sqrt{s}}$$

$$\le 8\|\mathbf{z}\|_2\|\mathbf{J}\|_F\frac{1}{\sqrt{s}},$$

where inequality (i) follows by the triangle inequality, and inequality (ii) holds with probability at least $1 - 4e^{-\frac{\|\mathbf{J}\|_F^2}{2\|\mathbf{J}\|^2}}$ using that

$$\mathrm{P}\left[\left\|\mathbf{J}^T\mathbf{P}(\mathbf{I} - \mathbf{S}^T\mathbf{S})\mathbf{z}\right\|_2 \le \|\mathbf{J}\|\|\mathbf{z}\|_2\epsilon\right] \ge 1 - 2e^{-s\frac{\epsilon^2}{12}} \tag{14}$$

with the choice of $\epsilon = 4\frac{1}{\sqrt{s}}\frac{\|\mathbf{J}\|_F}{\|\mathbf{J}\|}$, and using that

$$\mathrm{P}\left[\left\|\mathbf{J}^T\mathbf{P}_\perp\mathbf{S}^T\mathbf{S}\mathbf{z}\right\|_2 \le 4\|\mathbf{J}\|_F\frac{\|\mathbf{z}\|_2}{\sqrt{s}}\right] \ge 1 - 2e^{-\frac{\|\mathbf{J}\|_F^2}{2\|\mathbf{J}\|^2}}. \tag{15}$$

It remains to prove the bounds (14) and (15). We start with inequality (14). We have that

$$\left\|\mathbf{J}^T\mathbf{P}(\mathbf{I} - \mathbf{S}^T\mathbf{S})\mathbf{z}\right\|_2 \le \|\mathbf{J}\|\left\|\mathbf{P}(\mathbf{I} - \mathbf{S}^T\mathbf{S})\mathbf{z}\right\|_2$$

$$= \|\mathbf{J}\|\left\|\frac{\mathbf{z}}{\|\mathbf{z}\|_2}\frac{\mathbf{z}^T}{\|\mathbf{z}\|_2}(\mathbf{I} - \mathbf{S}^T\mathbf{S})\mathbf{z}\right\|_2$$

$$= \|\mathbf{J}\|\frac{1}{\|\mathbf{z}\|_2}\left|\|\mathbf{z}\|_2^2 - \|\mathbf{S}\mathbf{z}\|_2^2\right|$$

$$\le \|\mathbf{J}\|\|\mathbf{z}\|_2\epsilon,$$

20

where the last inequality holds with probability at least $1 - 2e^{-s\frac{\epsilon^2}{12}}$ for $\epsilon \in (0,1)$, with a standard concentration inequality for Gaussian matrices. This concludes the proof of the bound (14).

We next prove the bound (15). We need to bound the norm of $\mathbf{J}^T \mathbf{P}_\perp \mathbf{S}^T \mathbf{S} \mathbf{z}$. Note that the terms $\mathbf{J}^T \mathbf{P}_\perp \mathbf{S}^T$ and $\mathbf{S} \mathbf{z}$ are independent. Moreover, $\mathbf{S} \mathbf{z}$ is a Gaussian random vector with iid $\mathcal{N}(0, \|\mathbf{z}\|_2^2/s)$ entries. We therefore have that

$$\left\| \mathbf{J}^T \mathbf{P}_\perp \mathbf{S}^T \mathbf{S} \mathbf{z} \right\|_2 = \frac{\|\mathbf{z}\|_2}{\sqrt{s}} \|\mathbf{A} \mathbf{g}\|_2, \tag{16}$$

where $\mathbf{g}$ is a Gaussian vector with iid standard Gaussian entries, independent of $\mathbf{S}$, and $\mathbf{A} = \mathbf{J}^T \mathbf{P}_\perp \mathbf{S}^T$, for notational convenience.

Recall that for a Gaussian vector $\mathbf{g}$ with iid standard Gaussian entries and a L-Lipschitz function, we have

$$\mathrm{P}\left[f(\mathbf{g}) - \mathbb{E}\left[f(\mathbf{g}) \right] \geq t \right] \leq e^{-\frac{t^2}{2L^2}}.$$

Using that $f(\mathbf{g}) = \|\mathbf{A} \mathbf{g}\|_2$ is $\|\mathbf{A}\|$-Lipschitz, we get that

$$\mathrm{P}\left[\|\mathbf{A} \mathbf{g}\|_2 \geq 2\|\mathbf{A}\|_F \right] = \mathrm{P}\left[\|\mathbf{A} \mathbf{g}\|_2 \geq \sqrt{\mathbb{E}\left[\|\mathbf{A} \mathbf{g}\|_2^2 \right]} + \|\mathbf{A}\|_F \right]$$

$$\overset{(i)}{\leq} \mathrm{P}\left[\|\mathbf{A} \mathbf{g}\|_2 \geq \mathbb{E}\left[\|\mathbf{A} \mathbf{g}\|_2 \right] + \|\mathbf{A}\|_F \right]$$

$$\overset{(ii)}{\leq} e^{-\frac{\|\mathbf{A}\|_F^2}{2\|\mathbf{A}\|^2}}.$$

where inequality (i) is by Jensen's inequality (which implies $(\mathbb{E}\left[\|\mathbf{A} \mathbf{g}\|_2 \right])^2 \leq \mathbb{E}\left[\|\mathbf{A} \mathbf{g}\|_2^2 \right]$) and inequality (ii) follows by the Gaussian concentration inequality stated above. Similarly, we obtain

$$\mathrm{P}\left[\|\mathbf{B} \mathbf{S}\|_F \geq 2\|\mathbf{B}\|_F \right] \leq e^{-\frac{\|\mathbf{B}\|_F^2}{2\|\mathbf{B}\|^2}}.$$

Combining those two inequalities, we get that

$$\left\| \mathbf{J}^T \mathbf{P}_\perp \mathbf{S}^T \mathbf{S} \mathbf{z} \right\|_2 = \left\| \mathbf{J}^T \mathbf{P}_\perp \mathbf{S}^T \mathbf{g} \right\|_2 \frac{\|\mathbf{z}\|_2}{\sqrt{s}} \leq 2 \left\| \mathbf{J}^T \mathbf{P}_\perp \mathbf{S}^T \right\|_F \frac{\|\mathbf{z}\|_2}{\sqrt{s}} \leq 4 \|\mathbf{J} \mathbf{P}_\perp\|_F \frac{\|\mathbf{z}\|_2}{\sqrt{s}}, \leq 4 \|\mathbf{J}\|_F \frac{\|\mathbf{z}\|_2}{\sqrt{s}},$$

where the first inequality holds with probability at least $1 - e^{-\frac{\|\mathbf{J}\|_F^2}{2\|\mathbf{J}\|^2}}$, and the second as well, therefore by the union bound the entire inequality holds with probability at least $1 - 2e^{-\frac{\|\mathbf{J}\|_F^2}{2\|\mathbf{J}\|^2}}$. This concludes the proof of bound (15).

B.4 Proof of Theorem 1, part ii

Equation (11) state that

$$\left\| \boldsymbol{\theta}^{t+1} - \tilde{\boldsymbol{\theta}}^{t+1} \right\|_2 \leq \left\| \boldsymbol{\theta}^t - \tilde{\boldsymbol{\theta}}^t \right\|_2 + \eta \left\| \mathbf{J}^T (\mathbf{I} - \mathbf{P})(\mathbf{J} \tilde{\boldsymbol{\theta}}^t - \mathbf{y}) \right\|_2. \tag{17}$$

Since the matrix $\mathbf{J}$ has rank r, the residual $(\mathbf{J}\tilde{\boldsymbol{\theta}}^t - \mathbf{y})$ lies in a $(r+1)$-dimensional subspace, for any $\boldsymbol{\theta}^t$. It follows that, with probability at least $1 - 2e^{-r^2}$,

$$\left\| \mathbf{J}^T(\mathbf{I} - \mathbf{P})(\mathbf{J}\tilde{\boldsymbol{\theta}}^t - \mathbf{y}) \right\|_2 \leq \sigma_{\max} \sqrt{c\frac{r}{d}} \left\| \mathbf{J}\tilde{\boldsymbol{\theta}}^t - \mathbf{y} \right\|_2. \tag{18}$$

This probability bound holds for all t simultaneously. Proceeding analogously as in the proof of Lemma 1, we get

$$\left\| \boldsymbol{\theta}^t - \tilde{\boldsymbol{\theta}}^t \right\|_2 \leq \sigma_{\max} \sqrt{c\frac{r}{d}} \eta \sum_{\tau=0}^{t-1} (1 - \eta\sigma_r^2)^2 \|\tilde{\mathbf{r}}_0\|_2$$

$$= \sigma_{\max} \sqrt{c\frac{r}{d}} \eta \frac{1 - (1 - \eta\sigma_r^2)^t}{\eta\sigma_r^2} \|\tilde{\mathbf{r}}_0\|_2,$$

where the last inequality follows from the formula of a geometric series. This concludes the proof of Theorem 1, part ii.

C   Proof of the results in Section 5.1

In this section, we prove the two claims we made in Section 5.1, specifically that equations (5) and (6) hold with high probability.

Claim 1:   We first show that if we apply gradient descent for $O(\log(n)/\log(n/d))$ iterations to the loss in (3), i.e., to

$$\tilde{\mathcal{L}}_{AB}(\boldsymbol{\theta}) = \mathcal{L}_A(\boldsymbol{\theta}) + \mathcal{L}_B(\boldsymbol{\theta}) = \frac{1}{2}\|\mathbf{J}\boldsymbol{\theta} - \mathbf{y}\|_2^2,$$

then the corresponding estimate $\tilde{\boldsymbol{\theta}}_t$ obeys, with high probability,

$$\left\| \tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^* \right\|_2 \leq O\left(\sqrt{\frac{d}{n}}(\|\boldsymbol{\theta}_A\|_2 + \|\boldsymbol{\theta}_B\|_2 + \sigma) \right).$$

To establish this claim, we first note that the extreme singular values of a Gaussian matrix satisfy, for $t > 0$, [RV10, Equation 2.3]

$$\mathrm{P}\left[\sqrt{n} - \sqrt{d} - t \leq \sigma_{\min} \leq \sigma_{\max} \leq \sqrt{n} + \sqrt{d} + t \right] \geq 1 - 2e^{-t^2/2}. \tag{19}$$

With $t = \sqrt{d}$, we get

$$\mathrm{P}\left[\sqrt{n} - 2\sqrt{d} \leq \sigma_{\min} \leq \sigma_{\max} \leq \sqrt{n} + 2\sqrt{d} \right] \geq 1 - 2e^{-d^2/2}, \tag{20}$$

which we use below to establish the result.

Next, note that the minimizer of $\tilde{\mathcal{L}}_{AB}(\boldsymbol{\theta})$ is given by

$$\hat{\boldsymbol{\theta}} = (\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T \begin{bmatrix} \mathbf{J}_A(\mathbf{J}_A^T\mathbf{J}_A)^{-1}\mathbf{J}_A^T\mathbf{y}_A \\ \mathbf{y}_B \end{bmatrix}$$

$$= (\mathbf{J}^T\mathbf{J})^{-1}(\mathbf{J}_A^T\mathbf{J}_A(\mathbf{J}_A^T\mathbf{J}_A)^{-1}\mathbf{J}_A^T\mathbf{y}_A + \mathbf{J}_B^T\mathbf{y}_B)$$

$$= (\mathbf{J}^T\mathbf{J})^{-1}(\mathbf{J}_A^T\mathbf{y}_A + \mathbf{J}_B^T\mathbf{y}_B).$$

22

It follows that

$$\left\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\right\|_2 = \left\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_A - \boldsymbol{\theta}_B\right\|_2$$

$$= \left\|(\mathbf{J}^T\mathbf{J})^{-1}(\mathbf{J}_A^T\mathbf{J}_A\boldsymbol{\theta}_A + \mathbf{z}_A) + \mathbf{J}_B^T(\mathbf{J}_B\boldsymbol{\theta}_B + \mathbf{z}_B) - \boldsymbol{\theta}_A - \boldsymbol{\theta}_B\right\|_2$$

$$\leq \left\|(\mathbf{J}^T\mathbf{J})^{-1}(\mathbf{J}_A^T\mathbf{J}_A\boldsymbol{\theta}_A + \mathbf{z}_A) - \boldsymbol{\theta}_A\right\|_2 + \left\|(\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}_B^T(\mathbf{J}_B\boldsymbol{\theta}_B + \mathbf{z}_B) - \boldsymbol{\theta}_B\right\|_2$$

$$\leq c\sqrt{\frac{d}{n}}\left(\|\boldsymbol{\theta}_A\|_2 + \|\boldsymbol{\theta}_B\|_2 + \sigma\right), \tag{21}$$

where the last inequality holds with probability at least $1 - 8e^{-d^2/2}$, and follows from

$$\left\|(\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}_A^T(\mathbf{J}_A\boldsymbol{\theta}_A + \mathbf{z}_A) - \boldsymbol{\theta}_A\right\|_2 \leq \left\|\mathbf{I} - (\mathbf{J}^T\mathbf{J})^{-1}(\mathbf{J}_A^T\mathbf{J}_A)\right\|\|\boldsymbol{\theta}_A\|_2 + \left\|(\mathbf{J}^T\mathbf{J})^{-1}\right\|\left\|\mathbf{J}_A^T\mathbf{z}\right\|_2$$

$$\leq c\sqrt{\frac{d}{n}}\|\boldsymbol{\theta}_A\|_2 + c\sqrt{\frac{d}{n}}\sigma,$$

where we used inequality (20), and where c is a numerical constant. Specifically, here we used that, with probability at least $1 - 8e^{-d^2/2}$,

$$\left\|\mathbf{I} - (\mathbf{J}^T\mathbf{J})^{-1}(\mathbf{J}_A^T\mathbf{J}_A)\right\| \leq \left|1 - \frac{(\sqrt{n} + 2\sqrt{d})^2}{(\sqrt{n} - 2\sqrt{d})^2}\right| = \left|\frac{4\sqrt{n}\sqrt{d}}{n + 2d - 2\sqrt{n}\sqrt{d}}\right| \leq \frac{4\sqrt{d}}{\sqrt{n}},$$

and that

$$\left\|(\mathbf{J}^T\mathbf{J})^{-1}\right\|\left\|\mathbf{J}_A^T\mathbf{z}\right\|_2 \leq \frac{1}{(\sqrt{n} - 2\sqrt{d})^2}c\sqrt{n d}\sigma \leq c\frac{\sqrt{d}}{\sqrt{n}},$$

again with high probability. Here, we used that the entries of $\mathbf{J}_A\mathbf{z} \in \mathbb{R}^d$, conditioned on $\mathbf{z}$, are iid Gaussian with norm $\|\mathbf{z}\|_2$, and that $\|\mathbf{z}\|_2$ concentrates around $\sigma\sqrt{n}$.

Next, let $\mathbf{J} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ be the singular value decomposition of $\mathbf{J}$. The gradient descent iterations with stepsize η starting at $\tilde{\boldsymbol{\theta}} = \mathbf{0}$ are

$$\tilde{\boldsymbol{\theta}}^t = \mathbf{V}\mathbf{D}^t\mathbf{U}^T\mathbf{y},$$

where $\mathbf{D}^t$ is a diagonal matrix with i-th entry given by $\frac{1-(1-\eta\sigma_i^2)^t}{\sigma_i}$. With sufficiently small stepsize, gradient descent converges to the minimizer of the loss, i.e., $\hat{\boldsymbol{\theta}} = \tilde{\boldsymbol{\theta}}^\infty$. Thus,

$$\left\|\tilde{\boldsymbol{\theta}}^t - \hat{\boldsymbol{\theta}}\right\|_2 = \left\|\sum_{i=1}^d \mathbf{v}_i(1 - \eta\sigma_i^2)^t \langle\mathbf{u}_i, \mathbf{y}\rangle\right\|_2$$

$$\leq \|\mathbf{y}\|_2(1 - \eta\sigma_{\min}^2)^t$$

$$\leq c\sqrt{n}(\|\boldsymbol{\theta}_A\|_2 + \|\boldsymbol{\theta}_B\|_2 + \sigma)\left(\frac{8\sqrt{d}}{\sqrt{n}}\right)^t,$$

where again the last inequality holds with high probability. Choosing the stepsize as $1/\sigma_{\max}^2$, we get with

$$1 - \frac{\sigma_{\min}^2}{\sigma_{\max}^2} \leq 1 - \frac{(\sqrt{n} + 2\sqrt{d})^2}{(\sqrt{n} - 2\sqrt{d})^2} = \frac{8\sqrt{n}\sqrt{d}}{n + d - 8\sqrt{n}\sqrt{d}} \leq \frac{8\sqrt{d}}{\sqrt{n}}$$

23

that

$$\left\|\tilde{\boldsymbol{\theta}}^t - \hat{\boldsymbol{\theta}}\right\|_2 \leq c(\|\boldsymbol{\theta}_A\|_2 + \|\boldsymbol{\theta}_B\|_2 + \sigma)\left(\frac{8\sqrt{d}}{\sqrt{n}}\right)^t$$

$$\leq c(\|\boldsymbol{\theta}_A\|_2 + \|\boldsymbol{\theta}_B\|_2 + \sigma)\frac{\sqrt{d}}{\sqrt{n}}, \tag{22}$$

provided that $t \geq \frac{2\log(n)}{\log(n) - \log(64d)}$. It follows that for $t \geq O(\log(n)/\log(n/d))$

$$\left\|\tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*\right\|_2 \leq \left\|\tilde{\boldsymbol{\theta}}_t - \hat{\boldsymbol{\theta}}\right\|_2 + \left\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\right\|_2 \leq c(\|\boldsymbol{\theta}_A\|_2 + \|\boldsymbol{\theta}_B\|_2 + \sigma)\frac{\sqrt{d}}{\sqrt{n}},$$

where we used the previously established inequalities (21) and (22). This establishes equation (5) as claimed.

Claim 2: The second claim we made in Section 5.1 is that (cf. equation (6)):

$$\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|_2 \leq O\left(\left(\sqrt{\frac{d}{n}} + \sqrt{\frac{d}{s}}\right)(\|\boldsymbol{\theta}_A\|_2 + \|\boldsymbol{\theta}_B\|_2 + \sigma)\right). \tag{23}$$

This claim follows directly from combining equation (5) with Theorem 1. Specifically, note that for the setup in Section 5.1, Theorem 1 gives, with $r = d$, that

$$\left\|\boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}_t\right\|_2 \leq 8\frac{\|\mathbf{J}_A\|_F}{\sqrt{s}\sigma_{\min}}\frac{1}{\sigma_{\min}}\|\mathbf{y}\|_2$$

$$\leq \frac{\sqrt{dn}}{\sqrt{s}\sqrt{n}}\frac{1}{\sqrt{n}}\sqrt{n}(\|\boldsymbol{\theta}_A\|_2 + \|\boldsymbol{\theta}_B\|_2 + \sigma)$$

$$= \frac{\sqrt{d}}{\sqrt{s}}(\|\boldsymbol{\theta}_A\|_2 + \|\boldsymbol{\theta}_B\|_2 + \sigma)$$

holds with high probability, provided that $n \geq O(d)$. Specifically, we used, $\|\mathbf{J}_A\|_F \leq O(\sqrt{nd})$, $\sigma_{\min} \geq c\sqrt{n}$ (which holds for $n \geq O(d)$), and as established before, $\|\mathbf{y}\|_2 \leq c\sqrt{n}(\|\boldsymbol{\theta}_A\|_2 + \|\boldsymbol{\theta}_B\|_2 + \sigma)$. All three inequalities hold with high probability. Application of this inequality to $\|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|_2 \leq \left\|\boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}_t\right\|_2 + \left\|\tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*\right\|_2$ establishes the bound (23).

D   Proof of Theorem 2: Guarantees for two-layer neural networks

Let $\mathbf{J}_A, \mathbf{J}_B \in \mathbb{R}^{n \times dk}$ be the Jacobians of the network's predictions for the training sets of task A and task B at initialization. At initialization, each entry of the the weight matrix $\boldsymbol{\Theta}$ is initialized by drawing a zero-mean Gaussian with variance ω^2.

The Jacobians depend on the network's parameter, but if the network is sufficiently wide, the Jacobians change very little during gradient descent iterations, and if the network is in the NTK regime (and thus is infinitely wide), the Jacobians are constant and do not change across gradient descent iterations. To simplify exposition, we work in the NTK regime where the Jacobians are

constant throughout gradient descent iterations. We comment on changes that can be made to establish a result where the network is wide, but not infinitely wide, and thus the Jacobians vary little.

We provide a bound on the composite risk in equation (8) by decomposing the risk into the empirical risk and generalization errors of the two tasks

$$
\begin{aligned}
R(f) &= \frac{1}{2}R_A(f) + \frac{1}{2}R_B(f) \\
&= \frac{1}{2}\hat{R}_A(f) + \frac{1}{2}\hat{R}_B(f) + \frac{1}{2}(R_A(f) - \hat{R}_A(f)) + \frac{1}{2}(R_B(f) - \hat{R}_B(f)),
\end{aligned}
\tag{24}
$$

and by bounding the empirical risks and generalization errors separately. Here, the empirical risk of task T is $\hat{R}_T(\boldsymbol{\theta}) = \sum_{i=1}^{n} \ell(f_{\boldsymbol{\theta}}(\mathbf{x}_{T,i}), y_{T,i})$, where the $(\mathbf{x}_{T,i}, y_{T,i})$'s are the training data pertaining to task $T = \{A, B\}$.

Bounding the empirical risk: For bounding the empirical risk, we rely on the following lemma which ensures that the norm of the residual of the sketched problem, which is square-root of the empirical risk, is close to the residual of the norm of the original, non-sketched problem. We use the same notation as in the previous section, specifically we define:

$$
\mathbf{J} = \begin{bmatrix} \mathbf{J}_A \\ \mathbf{J}_B \end{bmatrix} \in \mathbb{R}^{2n \times dk}, \quad \mathbf{P} = \begin{bmatrix} \mathbf{S}^T\mathbf{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \mathbf{J}_A\boldsymbol{\theta}_A \\ \mathbf{y}_B \end{bmatrix}.
$$

Lemma 3. *Let $\mathbf{r}_t = \mathbf{P}^{1/2}\mathbf{J}\boldsymbol{\theta}^t - \mathbf{y}$ be the residual associated with the sketched least-squares problem and let $\tilde{\mathbf{r}}_t = \mathbf{J}\boldsymbol{\theta}^t - \mathbf{y}$ be the residual associated with the original least-squares problem. With probability at least $1 - 4t\exp(-\frac{\|\mathbf{J}_A\|_F^2}{2\|\mathbf{J}_A\|^2})$ over the random sketch $\mathbf{S} \in \mathbb{R}^{s \times n}$ the residuals are close:*

$$
\left\| \mathbf{r}^t - \tilde{\mathbf{r}}^t \right\|_2 \le \frac{\left\| \mathbf{J}\mathbf{J}^T \right\|_F}{\sqrt{s}} \frac{1}{\alpha^2}.
\tag{25}
$$

With this lemma in place, we note that, after t iterations of gradient descent, we have

$$
\begin{aligned}
\sqrt{\sum_{i=1}^{n}(f_{\boldsymbol{\theta}_t}(\mathbf{x}_{A,i}) - y_{A,i})^2 + \sum_{i=1}^{n}(f_{\boldsymbol{\theta}_t}(\mathbf{x}_{B,i}) - y_{B,i})^2} &\overset{(i)}{=} \frac{1}{\sqrt{n}}\|\mathbf{J}\boldsymbol{\theta}_t - \mathbf{y}\|_2 = \frac{1}{\sqrt{n}}\|\mathbf{r}_t\|_2 \\
&\le \frac{1}{\sqrt{n}}\|\tilde{\mathbf{r}}_t\|_2 + \frac{1}{\sqrt{n}}\|\mathbf{r}_t - \tilde{\mathbf{r}}_t\|_2 \\
&\overset{(ii)}{\le} \sqrt{\frac{1}{n}\sum_{i=1}^{2n}\langle \mathbf{u}_i, \mathbf{y}\rangle^2 (1 - \eta\sigma_i^2)^{2t}} + \frac{1}{\sqrt{n}}\frac{\|\mathbf{K}\|_F}{\sqrt{s}}\frac{1}{\alpha^2}.
\end{aligned}
\tag{26}
$$

Here, equation (i) holds if we are in the NTK regime and thus the predictions of the network are its Jacobian at initialization times the model parameter. For finite-width networks the equality holds up to an error that goes to zero as the network's width tends to infinity. Inequality (ii) holds

with probability at least $1 - 4t \exp(-\frac{\|\mathbf{J}_A\|_F^2}{2\|\mathbf{J}_A\|^2})$ by Lemma 3, and by using that $\mathbf{J}^T\mathbf{J} = \mathbf{K}$. With this bound, we obtain

$$
\begin{aligned}
\hat{R}_A(\boldsymbol{\theta}_t) &= \frac{1}{n}\sum_{i=1}^{n}\ell(f_{\boldsymbol{\theta}_t}(\mathbf{x}_{A,i}), y_{A,i}) \\
&\overset{(i)}{\leq} \frac{1}{n}\sum_{i=1}^{n}|f_{\boldsymbol{\theta}_t}(\mathbf{x}_{A,i}) - y_{A,i}| \\
&\leq \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f_{\boldsymbol{\theta}_t}(\mathbf{x}_{A,i}) - y_{A,i})^2} \\
&\overset{(ii)}{\leq} \sqrt{\frac{1}{n}\sum_{i=1}^{2n}\langle\mathbf{u}_i, \mathbf{y}\rangle^2(1 - \eta\sigma_i^2)^{2t} + \frac{1}{\sqrt{n}}\frac{\|\mathbf{K}\|_F}{\sqrt{s}}\frac{1}{\alpha^2}}.
\end{aligned}
$$

Here the sum in the first three equations is over trainings examples from task A, and equation (i) follows from $\ell(z, y) = \ell(z, y) - \ell(y, y) \leq |z - y|$ because the loss is 1-Lipschitz. Equation (ii) follows from equation (26) above.

The same bound holds for the risk of task B, $\hat{R}_B(\boldsymbol{\theta}_t)$.

Bonding the generalization error: We bound the generalization error for task A and task B separately. The derivations for each bound is the same, so we detail the derivations for task A only. We bound the generalization error of task A by bounding the Rademacher complexity of the class of functions that gradient descent can reach with t iterations. This is a common proof technique, see for example the papers [MRT12; Aro+19; HY21].

Let $\mathcal{F}$ be a set of functions mapping a d-dimensional feature vector to a real number, and let $\epsilon_1, \ldots, \epsilon_n$ be iid Rademacher random variables. A Rademacher random variable is chosen uniformly from $\{-1, 1\}$. The *empirical Rademacher complexity* of the function class $\mathcal{F}$ is defined as

$$
\mathcal{R}_{\mathcal{D}}(\mathcal{F}) = \frac{1}{n}\mathbb{E}_{\boldsymbol{\epsilon}}\left[\sup_{f \in \mathcal{F}}\sum_{i=1}^{n}\epsilon_i f(\mathbf{x}_i)\right],
$$

where $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ is a training set containing n examples drawn iid from the distribution pertaining to task A. The following theorem bounds the generalization error uniformly over all functions in the class $\mathcal{F}$ with the empirical Radermacher complexity of the function class $\mathcal{F}$.

Theorem 5 ([MRT12, Thm. 3.1]). *Assume the loss $\ell(\cdot, \cdot)$ is bounded in $[0, 1]$ and 1-Lipschitz in its first argument. With probability at least $1 - \delta$ over the set $\mathcal{D}$ consisting of n-many iid examples the generalization error is bounded by*

$$
\sup_{f \in \mathcal{F}} R(f) - \hat{R}(f) \leq 2\mathcal{R}_{\mathcal{D}}(\mathcal{F}) + 3\sqrt{\frac{\log(2/\delta)}{2n}}.
$$

We consider the class of two-layer neural networks of the form as in equation (7) with weights close to the random initialization $\boldsymbol{\Theta}_0$, defined as:

$$
\mathcal{F}_{Q,M} = \left\{f_{\boldsymbol{\Theta}} \colon \|\boldsymbol{\Theta} - \boldsymbol{\Theta}_0\|_F \leq Q, \|\boldsymbol{\theta}_r - \boldsymbol{\theta}_{0,r}\|_2 \leq \omega M\right\}, \tag{27}
$$

where $\boldsymbol{\theta}_r$ is the r-th column of the weight matrix $\boldsymbol{\Theta} \in \mathbb{R}^{d \times k}$.

The Rademacher complexity of this class of functions is bounded in the following lemma, which is a version of Lemma 5.4 in the paper [Aro+19] and a version of Lemma 4 in the paper [HY21].

Lemma 4. *Let $\boldsymbol{\Theta}_0$ be drawn from a Gaussian distribution with $\mathcal{N}(0, \omega^2)$ entries, and suppose half of the entries of $\mathbf{v}_0$ are 1 and the other half is -1. Assume the examples $(\mathbf{x}_i, y_i)$ are drawn iid from some distribution with $\|\mathbf{x}_i\|_2 = 1$ and $|y_i| \leq 1$. With probability at least $1 - \delta$ over the random training set, the empirical Rademacher complexity of $\mathcal{F}_{Q,M}$ is, simultaneously for all Q, bounded by*

$$\mathcal{R}_{\mathcal{D}}(\mathcal{F}_{Q,M}) \leq \frac{Q}{\sqrt{n}} + 4\omega M \left(\sqrt{k}M + \sqrt{\log(2/\delta)/2} \right). \tag{28}$$

We set $M = O(\frac{\xi}{\alpha} k^{-1/4})$, where ξ is an error tolerance parameter that goes to zero . With this choice, the term on the right hand side above is bounded by

$$\omega(4M^2\sqrt{k} + 4M\sqrt{\log(2/\delta)/2}) \leq O(\xi/\alpha),$$

where we used $\omega \leq 1$ and $\frac{\sqrt{\log(2/\delta)/2}}{k^{1/4}} \leq 1$ by the network being sufficiently wide. Recall that we consider the regime where $k \to \infty$, so this condition is satisfied.

Let $Q_i = i$ for $i = 1, 2, \ldots$. Simultaneously for all i, by the lemma above, for this choice of M, the function class $\mathcal{F}_{Q_i,M}$ has Rademacher complexity bounded by

$$\mathcal{R}_{\mathcal{D}}(\mathcal{F}_{Q_i,M}) \leq \frac{Q_i}{\sqrt{n}} + O(\xi/\alpha). \tag{29}$$

We next choose the radius Q as $Q = \sqrt{\sum_{i=1}^{2n} \left(\langle \mathbf{u}_i, \mathbf{y} \rangle \frac{1-(1-\eta\sigma_i^2)^t}{\sigma_i} \right)^2} + 5 \frac{\|\mathbf{J}_A\|_F}{\sqrt{s}} \frac{1}{\alpha^2} \sqrt{n} + \frac{\xi}{\alpha} \sqrt{n}$, where ξ is an approximation parameter that we can choose arbitrarily small for $k \to \infty$.

This choice is motivated as follows. We have that

$$\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_0\|_2 \leq \left\| \tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_0 \right\|_2 + \left\| \boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}_t \right\|_2$$

$$\leq \sqrt{\sum_{i=1}^{2n} \left(\langle \mathbf{u}_i, \mathbf{y} \rangle \frac{1-(1-\eta\sigma_i^2)^t}{\sigma_i} \right)^2} + 5 \frac{\|\mathbf{J}_A\|_F}{\sqrt{s}} \frac{1}{\alpha^2} \|\mathbf{y}\|_2,$$

where the last inequality holds by Lemma 1 with probability at least $1 - 4t \exp(-\frac{\|\mathbf{J}_A\|_F^2}{2\|\mathbf{J}_A\|^2})$. The extra term $\frac{\xi}{\alpha}\sqrt{n}$ is due to the error of the Jacobian varying slightly over iterations, and goes to zero as the width $k \to \infty$.

Let i^* be the smallest integer such that $Q \leq Q_{i^*}$, so that $Q_{i^*} \leq Q + 1$. We have that $i^* \leq O(\sqrt{n}/\alpha)$ and

$$\mathcal{R}_{\mathcal{D}}(\mathcal{F}_{Q_{i^*},M}) \leq \frac{(Q+1)}{\sqrt{n}} + O(\xi/\alpha)$$

$$\leq \sqrt{\frac{1}{n} \sum_{i=1}^{2n} \left(\langle \mathbf{u}_i, \mathbf{y} \rangle \frac{1-(1-\eta\sigma_i^2)^t}{\sigma_i} \right)^2} + \frac{1}{\sqrt{n}} + 5 \frac{\|\mathbf{J}_A\|_F}{\sqrt{s}} \frac{1}{\alpha^2} + O(\xi/\alpha). \tag{30}$$

Next, from a union bound over the finite set of integers $i = 1, \ldots, i^*$, we obtain

$$\max_{i=1,\ldots,i^*} \sup_{f \in \mathcal{F}_{Q_i,M}} R_A(f) - \hat{R}_A(f) \leq \sqrt{\frac{1}{n} \sum_{i=1}^{2n} \left(\langle \mathbf{u}_i, \mathbf{y} \rangle \frac{1 - (1 - \eta \sigma_i^2)^t}{\sigma_i} \right)^2} + \frac{1}{\sqrt{n}} + 5 \frac{\|\mathbf{J}_A\|_F}{\sqrt{s}} \frac{1}{\alpha^2} + O(\xi/\alpha)$$

$$\leq \sqrt{\frac{1}{n} \mathbf{y}^T (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{y}} + 5 \frac{\|\mathbf{J}_A\|_F}{\sqrt{s}} \frac{1}{\alpha^2} + \frac{1}{\sqrt{n}} + O(\xi/\alpha).$$

Assembling the bounds: Combining the bounds on the empirical risk and on the generalization errors by inserting them in the right-hand-side of equation (24) yields

$$R(f_{\boldsymbol{\theta}_t}) \leq \sqrt{\frac{1}{n} \sum_{i=1}^{2n} \langle \mathbf{u}_i, \mathbf{y} \rangle^2 (1 - \eta \sigma_i^2)^{2t}} + \frac{1}{\sqrt{n}} \frac{\|\mathbf{K}\|_F}{\sqrt{s}} \frac{1}{\alpha^2}$$

$$+ 2\sqrt{\frac{1}{n} \mathbf{y}^T \mathbf{K}^{-1} \mathbf{y}} + 10 \frac{\|\mathbf{J}_A\|_F}{\sqrt{s}} \frac{1}{\alpha^2} + \frac{2}{\sqrt{n}} + O(\xi/\alpha)$$

$$\leq 2\sqrt{\frac{1}{n} \mathbf{y}^T \mathbf{K}^{-1} \mathbf{y}} + \frac{3}{\sqrt{n}} + \frac{1}{\sqrt{s}\alpha^2} \left(10\|\mathbf{J}_A\|_F + \frac{\|\mathbf{K}\|_F}{\sqrt{n}} \right),$$

where we upper-bounded the first term by using the assumption $t \geq \frac{\log(1 - \eta\alpha)}{\log(1/n)}$, and where we also again used the assumption that the network is infinitely wide and thus $\xi \to 0$. This concludes the proof of the theorem.

D.1 Proof of Lemma 3

With similar steps as in equation (11) we get

$$\left\| \mathbf{r}^{t+1} - \tilde{\mathbf{r}}^{t+1} \right\|_2 = \left\| (\mathbf{I} - \eta \mathbf{J} \mathbf{J}^T \mathbf{P})(\mathbf{r}^t - \tilde{\mathbf{r}}^t) \right\|_2 + \eta \left\| \mathbf{J} \mathbf{J}^T (\mathbf{I} - \mathbf{P}) \tilde{\mathbf{r}}^t \right\|_2$$

$$\leq \left\| \mathbf{r}^t - \tilde{\mathbf{r}}^t \right\|_2 + \eta \left\| \mathbf{J} \mathbf{J}^T (\mathbf{I} - \mathbf{P}) \tilde{\mathbf{r}}^t \right\|_2$$

$$\leq \left\| \mathbf{r}^t - \tilde{\mathbf{r}}^t \right\|_2 + \eta \frac{\left\| \mathbf{J} \mathbf{J}^T \right\|_F}{\sqrt{s}} \left\| \tilde{\mathbf{r}}^t \right\|_2,$$

where the last inequality holds with probability at least $1 - 4e^{-\frac{\|\mathbf{J}_A\|_F^2}{2\|\mathbf{J}_A\|^2}}$ as established by Lemma 2. Applying this inequality recursively, we obtain, by the union bound, that with probability at least $1 - 4te^{-\frac{\|\mathbf{J}_A\|_F^2}{2\|\mathbf{J}_A\|^2}}$

$$\left\| \mathbf{r}^t - \tilde{\mathbf{r}}^t \right\|_2 \leq \eta \frac{\left\| \mathbf{J} \mathbf{J}^T \right\|_F}{\sqrt{s}} \sum_{i=0}^{n-1} \left\| \tilde{\mathbf{r}}^t \right\|_2$$

$$\leq \frac{\left\| \mathbf{J} \mathbf{J}^T \right\|_F}{\sqrt{s}} \frac{1}{\alpha^2}.$$

This concludes the proof.

E Proof of the results in Section 7

E.1 Proof of Theorem 3

The population loss for task T is

$$\mathcal{L}_T(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x},y)\sim P_T}\left[(\langle \mathbf{x}, \boldsymbol{\theta}\rangle - y)^2\right]$$
$$= (\langle \boldsymbol{\theta}, \boldsymbol{\mu}_T\rangle - 1)^2 + \sigma^2\|\boldsymbol{\theta}\|_2^2.$$

Thus training on task A yields

$$\boldsymbol{\theta}_A = (\sigma^2\mathbf{I} + \boldsymbol{\mu}_A\boldsymbol{\mu}_A^T)^{-1}\boldsymbol{\mu}_A = \frac{1}{1+\sigma^2}\boldsymbol{\mu}_A,$$

where the second equality follows from the Sherman-Morrison-Woodbury formula.

Now consider the loss associated with EWC, given by

$$\mathcal{L}_{AB}(\boldsymbol{\theta}) = \mathbb{E}_B\left[(\langle \mathbf{x}, \boldsymbol{\theta}\rangle - y)^2\right] + \lambda(\boldsymbol{\theta} - \boldsymbol{\theta}_A)^T\mathbf{D}_A(\boldsymbol{\theta} - \boldsymbol{\theta}_A),$$

where $\mathbf{D}_A$ is the diagonal of the Hessian (or Jacobian outer product) of the loss associated with task A, i.e., $\mathbf{D}_A = \text{diag}(\sigma^2\mathbf{I} + \boldsymbol{\mu}_A\boldsymbol{\mu}_A^T)$. The minimizer of $\mathcal{L}_{AB}(\boldsymbol{\theta})$ denoted $\boldsymbol{\theta}_{AB}^{EWC}$, is, again by using the Sherman-Morrison formula

$$\boldsymbol{\theta}_{AB}^{EWC} = (\sigma^2\mathbf{I} + \boldsymbol{\mu}_B\boldsymbol{\mu}_B^T + \lambda\mathbf{D}_A)^{-1}(\boldsymbol{\mu}_B + \lambda\mathbf{D}_A\boldsymbol{\mu}_A)$$
$$= (\mathbf{D} + \boldsymbol{\mu}_B\boldsymbol{\mu}_B^T)^{-1}(\boldsymbol{\mu}_B + \lambda\mathbf{D}_A\boldsymbol{\mu}_A)$$
$$= \left(\mathbf{D}^{-1} - \frac{1}{1+\boldsymbol{\mu}_B^T\mathbf{D}^{-1}\boldsymbol{\mu}_B}\mathbf{D}^{-1}\boldsymbol{\mu}_B\boldsymbol{\mu}_B^T\mathbf{D}^{-1}\right)(\boldsymbol{\mu}_B + \lambda\mathbf{D}_A\boldsymbol{\mu}_A)$$
$$= \frac{1}{1+\boldsymbol{\mu}_B^T\mathbf{D}^{-1}\boldsymbol{\mu}_B}\mathbf{D}^{-1}\boldsymbol{\mu}_B + \lambda\left(\mathbf{I} - \frac{1}{1+\boldsymbol{\mu}_B^T\mathbf{D}^{-1}\boldsymbol{\mu}_B}\mathbf{D}^{-1}\boldsymbol{\mu}_B\boldsymbol{\mu}_B^T\right)\mathbf{D}^{-1}\mathbf{D}_A\boldsymbol{\mu}_A,$$

where we defined $\mathbf{D} = \lambda\mathbf{D}_A + \sigma^2\mathbf{I}$ for notational convenience. Next, we use the assumption that all entries of $\boldsymbol{\mu}_T$ have absolute value $\sqrt{1/d}$. This assumption implies that $\mathbf{D}_A = (\sigma^2 + 1/d)\mathbf{I}$ and $\mathbf{D} = q\mathbf{I}$ with $q = \sigma^2 + \lambda(\sigma^2 + 1/d)$, and yields

$$\boldsymbol{\theta}_{AB}^{EWC} = \frac{1}{1+q}\boldsymbol{\mu}_B + \frac{\lambda(\sigma^2 + \frac{1}{d})}{q}\boldsymbol{\mu}_A - \frac{\lambda(\sigma^2 + \frac{1}{d})}{(1+q)q}\boldsymbol{\mu}_B\langle\boldsymbol{\mu}_A, \boldsymbol{\mu}_B\rangle$$
$$= \boldsymbol{\mu}_B\frac{1}{1+q}\left(1 - \frac{\lambda(\sigma^2 + 1/d)}{q}\langle\boldsymbol{\mu}_A, \boldsymbol{\mu}_B\rangle\right) + \boldsymbol{\mu}_A\frac{\lambda(\sigma^2 + \frac{1}{d})}{\lambda(\sigma^2 + \frac{1}{d} + \sigma^2)}$$

Next, note that the term associated with μ_A is increasing from 0 to 1 in λ. Contrary, the term associated with μ_B is decreasing in λ. Thus, λ interpolates between linear combinations of $\boldsymbol{\mu}_A$ and $\boldsymbol{\mu}_B$ and therefore there exists a regularization parameter λ such that $\boldsymbol{\theta}_{AB}^{EWC}$ points in the same direction as the optimal parameter $\boldsymbol{\theta}_{AB}^* = \boldsymbol{\mu}_A + \boldsymbol{\mu}_B$, and is therefore Bayes optimal. This concludes the first part of the theorem. The proof of the second part, given below, is analogous.

E.2 Proof of Theorem 3, part two

The proof is analogous to the proof of part one of the theorem. The loss associated with L2 regularization is given by

$$\mathcal{L}_{AB}(\boldsymbol{\theta}) = \mathbb{E}_B\left[(\langle \mathbf{x}, \boldsymbol{\theta}\rangle - y)^2\right] + \lambda(\boldsymbol{\theta} - \boldsymbol{\theta}_A)^T(\boldsymbol{\theta} - \boldsymbol{\theta}_A).$$

Application of the Sherman-Morrison formula yields

$$
\begin{aligned}
\boldsymbol{\theta}_{AB}^{L2} &= (\sigma^2\mathbf{I} + \boldsymbol{\mu}_B\boldsymbol{\mu}_B^T + \lambda\mathbf{I})^{-1}(\boldsymbol{\mu}_B + \lambda\boldsymbol{\mu}_A) \\
&= \left(\frac{1}{\sigma^2 + \lambda}\mathbf{I} - \frac{1}{1 + 1/(\sigma^2 + \lambda)}\frac{1}{(\sigma^2 + \lambda)^2}\boldsymbol{\mu}_B\boldsymbol{\mu}_B^T\right)(\boldsymbol{\mu}_B + \lambda\boldsymbol{\mu}_A) \\
&= \frac{1}{\sigma^2 + \lambda}\left(\mathbf{I} - \frac{1}{1 + \sigma^2 + \lambda}\boldsymbol{\mu}_B\boldsymbol{\mu}_B^T\right)(\boldsymbol{\mu}_B + \lambda\boldsymbol{\mu}_A) \\
&= \frac{1}{\sigma^2 + \lambda}\left(\frac{\sigma^2 + \lambda}{1 + \sigma^2 + \lambda}\boldsymbol{\mu}_B + \lambda\boldsymbol{\mu}_A - \frac{\lambda\langle\boldsymbol{\mu}_A, \boldsymbol{\mu}_B\rangle}{1 + \sigma^2 + \lambda}\boldsymbol{\mu}_B\right) \\
&= \frac{1}{\sigma^2 + \lambda}\left(\frac{\sigma^2 + \lambda - \lambda\langle\boldsymbol{\mu}_A, \boldsymbol{\mu}_B\rangle}{1 + \sigma^2 + \lambda}\boldsymbol{\mu}_B + \lambda\boldsymbol{\mu}_A\right).
\end{aligned}
$$

The term in front of $\boldsymbol{\mu}_B$ is decreasing in λ and varies from $\frac{\sigma^2}{1+\sigma^2}$ to 0 as λ increases. The term associated with $\boldsymbol{\mu}_A$ varies from 0 to 1. Thus, again, there is a parameter λ such the solution is optimal.

E.3 Proof of Theorem 4

To prove this statement, it is sufficient to construct a problem instance consisting of task means $\mathbf{m}_A$ and $\boldsymbol{\mu}_B$ and a variance σ^2 for which the risk of $\boldsymbol{\theta}_{AB}^{EWC}(\lambda)$ is large for all λ. Such an problem instance is $\mu_A = [1, -0.8, 0.8]$, $\mu_B = [-1, 0.5, -0.8]$, and σ sufficiently small, as illustrated in the code supplement. The proof of the second part is analogous, by constructing a similar problem instance for L2.