On Continual Learning using Deep Linear Networks

by

Alexandre Michael John Kaiser

A thesis submitted in partial fulfillment

of the requirements for the degree of

Master of Science

Courant Institute of Mathematical Sciences

New York University

May, 2024

Professor Arthur Jacot

To my loving parents, Katherine Philips-Kaiser and Kevin Kaiser, with affection.

Acknowledgements

To my advisor Arthur Jacot, I have had the great privilege of working with you for the past year. In that time there was rarely a week that we did not meet for an hour if not more to discuss our work and I am extremely grateful for the time you generously lent and the thoughts you shared. Thank you for embodying a "yes, and" approach to being an advisor. I sincerely hope that we have the opportunity to continue our collaboration past the conclusion of this Masters degree.

I would also like to thank my 2nd reader Saining Xie, whom I first met in his course Large Language and Vision Models. The format of your course which likened a conference presentation really put me in the right path to be able to think about the importance of the impact of a work. Particularly in theory, it can be easy to ignore the real problems when you are deep in the math, so I would like to thank you for the reminder that a great result must have real impact. Though this thesis studies a simplified version of practical deep learning problems, I hope that the analysis is aligned to the objectives that are set in practice.

Other notable professors who have helped shape my understanding of machine learning and theory during my time at Courant include (in order of having met them) professors Mehryar Mohri, Matus Telgarsky and Joan Bruna. Your respective courses covering mathematics and machine learning have each had significant impact on my formation and my time in Courant. Thank you all.

Abstract

In the burgeoning field of deep learning, fueled by recent advancements in generative technologies, there is a significant increase in model and data scales, leading to escalated training costs. Yet, understanding the intricate training dynamics of neural networks remains a challenging theoretical endeavor. This thesis explores these dynamics through the lens of continual learning with deep linear networks, focusing on the stability of training with respect to critical failure modes such as Catastrophic Forgetting and rank underestimation, which we term *Catastrophic Weight Loss*.

We employ projections to formalize the continual learning process, enabling a detailed examination of the effects of common training practices such as stochastic gradient descent (SGD) with small batch sizes, and lifelong learning in environments with non-stationary input distributions. Our findings reveal that standard techniques such as weight decay, small batch sizes, and 1-pass training exacerbate the likelihood of catastrophic events. However, our analysis suggests that Catastrophic Forgetting does not unequivocally hinder learning progress. In fact, if the training dynamics are stable, models continue to evolve towards a more accurate representation of the set of previously seen tasks, despite apparent setbacks. This insight challenges the traditional view of Catastrophic Forgetting as purely detrimental, suggesting it can be effectively managed with targeted interventions such as data replay. This reevaluation provides a fresh perspective on enhancing the resilience and effectiveness of training protocols in large-scale neural networks.

Contents

List of Figures

1 Introduction

1.1 Continual Learning

Continual learning, also known as lifelong learning, represents a significant challenge and a vibrant area of interest in artificial intelligence. This paradigm enables a model to learn continually from a stream of data, assimilating new knowledge while preserving previously acquired information. The concept of catastrophic forgetting in multitask sequential learning, pivotal to continual learning, emerged in the late 1980s and early 1990s with foundational papers by McCloskey and Cohen (1989) [McCloskey and Cohen 1989] and Ratcliff (1990) [Ratcliff 1990]. Subsequent research, often under the broader umbrella of transfer learning, has further explored these dynamics ([Pratt 1993],[Suddarth and Holden 1991],[Caruana 1993]). Initial strategies to mitigate catastrophic forgetting, such as replay methods ([Robins 1995]), have evolved with the resurgence of deep learning, leading to sophisticated techniques like Elastic Weight Consolidation (EWC) [Kirkpatrick et al. 2017] and Synaptic Intelligence [Zenke et al. 2017].

Despite advancements, the theoretical exploration of catastrophic forgetting primarily focuses on model losses rather than structural changes within the model that precipitate forgetting. Viewed through the lens of online learning, the issue of forgetting is often analyzed through regret minimization frameworks ([Abernethy et al. 2011], [Shimkin 2016]), with techniques such as AdaGrad [Duchi et al. 2011] and RMSProp [Tieleman 2012] providing foundational methods in online convex optimization. In continual learning, however, the assumption that the underlying function remains stationary adds another constraint to the optimization that these online convex optimization algorithms are not quick to take advantage of, thereby explaining the occurrence and challenge of forgetting. This dilemma typically necessitates re-acquaintance with previous tasks, allowing the model to refine its approach toward a higher-rank, more stable solution.

Recent studies ([Zhang et al. 2020], [Zhang et al. 2022]) have begun to challenge the traditional narrative of catastrophic forgetting, suggesting that significant forgetting can occur with minimal changes in the features themselves, particularly in contexts related to transfer and feature learning.

1.2 Deep Linear Networks

In parallel to continual learning, the theory of deep linear networks has developed, albeit more subtly compared to nonlinear models. Deep linear networks—neural networks characterized by linear activation functions—offer a simplified yet powerful framework for analyzing deeper, more complex architectures. These networks provide critical insights into neural behavior, learning dynamics, loss landscapes, and capacity constraints. Despite their simplicity, they reveal complex learning behaviors and theoretical nuances applicable to broader model classes. Therefore, this thesis employs deep linear networks to investigate continual learning dynamics, inspired by significant findings such as those by Kawaguchi (2016), which demonstrated that the loss surface of these networks comprises a single global minimum amidst a landscape of saddle points [Kawaguchi 2016].

Research into the low-rank bias of these networks ([Wang and Jacot 2023], [Li et al. 2021]), along with studies on their task alignment ([Ji and Telgarsky 2019]) and dynamic properties ([Jacot et al. 2022]), further underscores their relevance and potential for elucidating complex learning phenomena.

1.3 Projections

Projection techniques have long been instrumental across various disciplines, aiding in the simplification of high-dimensional spaces and the optimization of complex functions. In continual learning, projections help manage the discrete and iterative nature of task learning by segmenting the function space into task-specific subspaces. This partitioning, crucial for minimizing interference among tasks, aligns closely with observed forgetting behaviors. Notably, projections used in this context are not strictly orthogonal, highlighting potential overlaps and interferences between tasks.

This thesis also reflects on the training methodologies of foundational models like Large Language Models (e.g., LLaMA 2 [Touvron et al. 2023]), emphasizing similarities with SGD applications that incorporate weight decay but not more advanced techniques like EWC.

By intertwining continual learning with deep linear network theory and projection methods, this work aims to forge a deeper understanding of both domains while addressing the challenges inherent in learning within dynamic environments. This synthesis promises not only enhanced theoretical insight but also practical strategies for developing robust models capable of adapting over time without compromising on previously mastered capabilities.

2 Preliminaries

Let us solve a learning problem using gradient descent. We are given a model $f(\theta)$ to optimize using the cost function C for a dataset $(X, Y) \in \mathbb{R}^{N\times d} \times \mathbb{R}^{N\times d}$, where N is the number of datapoints and d is the dimensionality of the data. Let's start by making the simplifying assumptions that Y is generated by a linear function of the data $Y = XA_*$ for some unknown A_*. We are interested in using a deep linear network as our model

$$f(\theta) = W_1 \cdot W_2 \cdots W_{L-1} \cdot W_L = A_\theta \tag{2.1}$$

where L is the depth of the network. Lastly, for our cost function, we will study the most common loss function - the squared loss - with L2 regularization. The squared loss is usually expressed as follows, where $||\cdot||$ denotes the Frobenius norm.

$$\mathcal{L}(f) = \frac{1}{2}||f(X, \theta) - Y||^2$$

Since Y is generated by some linear mapping A_*, we can simplify the cost function.

$$C(X, \theta, \lambda) = \frac{1}{2}||X(A_\theta - A_*)||^2 + \lambda \sum_{\ell=1}^{L} ||W_\ell||^2$$

The problem of continual learning stems from sequentially learning different subsets of the input space. To reflect that in our analysis, we will project our full-batch of data, which is assumed to be full rank, into a subspace using a projection matrix P. The resulting data PX is a mini-batch and is assumed to be rank $r < d$. In our analysis, we will exclusively use orthogonal projection matrices, thus the image of P is orthogonal to its kernel (see Appendix A for more details about projection matrices).

The term *episodes* will refer to an uninterrupted period of training using projection P_i, for episode i. Consider P_i to be the task of episode i. Note the case of training on a single datapoint x_i is the same as training on a rank-1 projection $P = \frac{x_i x_i^T}{||x_i||^2}$. This case becomes relevant when studying the effect of stochastic gradient descent (SGD) with small batch sizes.

In the context of our optimization problem, we are trying to minimize the cost $C(P, \theta, \lambda)$. Since X is assumed to be full rank, our optimization is independent of the data. We can define our optimization problem as follows:

$$\min_{\theta} C(P, \theta, \lambda) = \min_{\theta} \frac{1}{2}||P(A_\theta - A_*)||^2 + \lambda \sum_{\ell=1}^{L} ||W_\ell||^2 \tag{2.2}$$

This thesis will focus on the dynamics of our model A_θ when going from one training episode to the next. Ideally, A_θ should approach A_* if the union of training episodes span the entire input space. Catastrophic forgetting occurs when the loss associated to prior tasks $C(P_i, \theta, 0)$ has increases significantly due to having trained on other tasks more recently. Note that the loss in this context can be expressed as the unregularized cost, where $\lambda = 0$. Formally, we can express the forgetting $\mathcal{F}$ exhibited for task P_i as the difference between the loss of task P_i using the most current model A_θ and the loss achieved at the end of the original episode i, namely C_i, that the task was being trained on.

$$\mathcal{F}(P_i, C_i, \theta) = C(P_i, \theta, 0) - C_i \tag{2.3}$$

3 | Single-Layer Case

To form reasonable expectations for the behavior of deep linear networks, let's start by analyzing the problem in its simplest case, where $L = 1$. In the single-layer case the unregularized cost function is convex, allowing gradient descent to converge to the global minimum for each episode.

Proof. The unregularized cost function is the Frobenius norm squared of a linear combination of matrix A_θ. As a Frobenius norm, it can expressed as the following sum.

$$C(P, \theta, \lambda = 0) = \frac{1}{2} \sum_{(i,j)} \left(\sum_{k} P_{ik} (A_{\theta,kj} - A_{*,kj}) \right)^2$$

As the sum of convex functions, the cost is convex. □

3.1 Single episode dynamics

Although the training dynamics of the model follow gradient descent, the gradient function $\nabla C(\theta)$ changes discontinuously from one episode to the next due to the sudden change in task P_i. To analyze the dynamics of the model for the entire training period, let's start by analyzing the dynamics for a given episode. Within a given episode i, the dynamics of the model are as follows.

$$\begin{aligned}\frac{dA_\theta}{dt} &= -\nabla_\theta C(P_i, \theta, \lambda) \\ &= P(A_* - A_\theta) - 2\lambda A\end{aligned}$$

Let A_0 be the initialization of A_θ at time $t = 0$. Note that in the unregularized case where $\lambda = 0$ the dynamics exist only along its projection in the current task. As an important baseline, we are interested in whether the model gets closer to the true function A_*.

Proposition 3.1. *Each unregularized gradient descent step can only get the model $A_{\theta,t}$ closer to the true function A_*.*

Proof. Induction

$$\begin{aligned}||A_{\theta,t+1} - A_*||^2 &= ||A_{\theta,t} - \eta P(A_* - A_{\theta,t} - A_*)||^2 \\ &= ||(I - \eta P)(A_{\theta,t} - A_*)||^2 \\ &= \mathrm{Tr}((A_{\theta,t} - A_*)^T (I - \eta P)^T (I - \eta P)(A_{\theta,t} - A_*)) \\ &= ||A_{\theta,t} - A_*||^2 - \eta(2 - \eta)||P(A_{\theta,t} - A_*)||^2\end{aligned}$$

For $\eta \in [0, 2]$,

$$||A_{\theta,t+1} - A_*||^2 \leq ||A_{\theta,t} - A_*||^2$$

□

Although we prove that each unregularized gradient descent step makes progress towards the true function, it doesn't prove convergence. Let's start by finding the convergence point of a single episode by solving for $A_\theta(t)$. We can solve for $A_\theta(t)$ analytically by separating the problem along the image and kernel of the task respectively.

Proposition 3.2. *In a given episode, the weights along the kernel of the task only experience decay.*

Proof. The task defined by its projection P, has a kernel characterized by $(I - P)$

$$\begin{aligned}
\frac{d(1-P)A_\theta}{dt} &= (I-P)P(A_* - A_\theta) - 2\lambda(I-P)A \\
&= -2\lambda(I-P)A_\theta \\
(1-P)A_\theta(t) &= (I-P)A_0 e^{-2\lambda}
\end{aligned}$$

□

Proposition 3.3. *In a given episode, the weights along the image of the task converge exponentially to* $P\frac{A_*}{1+2\lambda}$.

Proof. This proof follows similarly the proof for Proposition 3.2, where now we are interested in the time varying dynamics of PA_θ.

$$\begin{aligned}
\frac{dPA_\theta}{dt} &= PP(A_* - A_\theta) - 2\lambda PA \\
&= PA_* - (1+2\lambda)PA_\theta \\
PA_\theta(t) &= P\frac{A_*}{1+2\lambda} + P\left(A_0 - \frac{A_*}{1+2\lambda}\right)e^{-(1+2\lambda)t}
\end{aligned}$$

□

The full parameter space can be described by the union of the dynamics within the image space, given in Proposition 3.3, and that of the kernel space provided in Propositions 3.2. Due to the orthogonality between these two spaces, the full dynamics for a single episode can be expressed as the sum of the dynamics above.

$$A_{\theta,t} = P\frac{A_*}{1+2\lambda} + P\left(A_0 - \frac{A_*}{1+2\lambda}\right)e^{-(1+2\lambda)t} + (I-P)A_0 e^{-2\lambda t} \tag{3.1}$$

In the regularized setting, where $\lambda > 0$, the first term of the full dynamics is constant, and all the other terms are decaying exponentially to zero. Thus, in the limit as t goes to infinity, the model will converge to $P\frac{A_*}{1+2\lambda}$. Remark first that the use of weight decay applies a shrinking factor $\frac{1}{1+2\lambda}$ to the unregularized target PA_*. This shrink is typical of weight decay in general as it attracts any solution radially towards the origin. Additionally, remark that in this case the model converges to a matrix which is independent of the model's initialization. That proves that single-layer, single-episode, regularized problem has a unique solution.

Unfortunately, the unique solution is not a good one. As a reminder, we generally assume that A_* is full rank, and the projection P is rank deficient, which means that the unique solution is rank deficient compared to the true task A_*. Moreover, the unique solution being independent of the model's initialization means that if each episode was trained to convergence, the final model would only be a function of the final task that it was trained on.

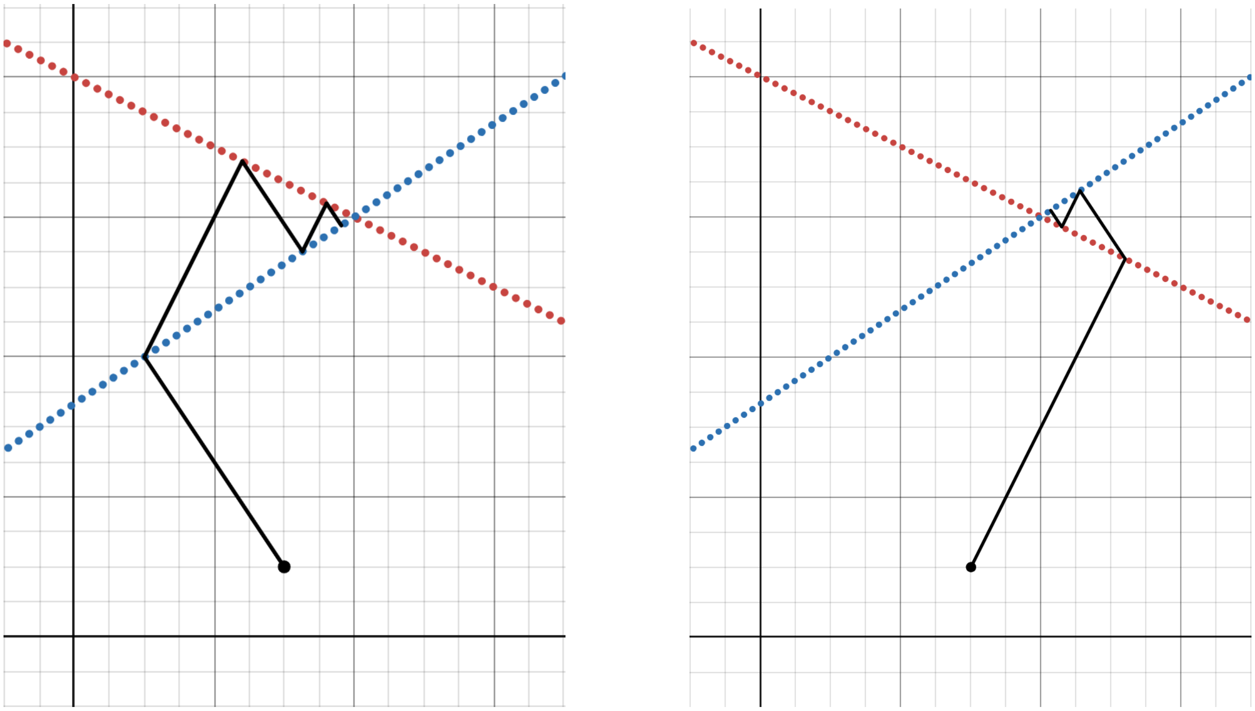
However, note that this issue of uniqueness was created in part because of the use of weight decay as a regularizer. Without it, when $\lambda = 0$, the single episode dynamics converge instead to $PA_* + (I - P)A_0$. This is good news because we are maintaining the information from previous episodes. In the next section, we will unpack how this preservation translates into overall performance.

3.2 Consecutive episode dynamics

When training a model for multiple episodes, assuming that each episode is trained to convergence, we can express the state of the model after k episodes of $\lambda = 0$ as

$$A_{\theta,k} = P_k A_* + (I - P_k)A_{\theta,k-1}$$

The unregularized dynamics visualized in Figure 3.1 shows the iterative convergence to the true function A_* by alternating between the kernel of the two tasks being represented. We can see that the speed to converge depends on the angle between the two kernels. As the angle gets



(a) Dynamics starting with the blue task

(b) Dynamics starting with the red task

Figure 3.1: Unregularized dynamics for alternating pair of tasks in matrix space

smaller, it is as if the two projections are approximately equivalent which makes it difficult to learn. Whereas two orthogonal kernels would achieve convergence immediately upon training both once. Nevertheless, as noted previously, if there is any weight decay then the state of the model will be $A_{\theta,k} = P_k A_*$, and there will be no convergence too the full rank A_*.

Proposition 3.4. *The state of the model after k episodes of unregularized training can be expressed explicitly as*

$$A_{\theta,k} = \left(\prod_{i=1}^{k} (I - P_k) \right) (A_0 - A_*) + A_*$$

where $\prod$ is the recursive left-multiply operator.

Proof. We have that the state of the model can be written as the sequence

$$A_{\theta,k} = P_k A_* + (I - P_k) A_{\theta,k-1}$$

By grouping up terms, we can reformulate the sequence explicitly as

$$A_{\theta,k} = \left(\prod_{i=1}^{k}(I - P_i)\right) A_0 + \sum_{i=1}^{k}\left(\prod_{j=i+1}^{k}(I - P_j)\right) P_i A_*$$

From here, we can prove by induction that

$$I - \sum_{i=1}^{k}\left(\prod_{j=i+1}^{k}(I - P_j)\right) P_i = \prod_{i=1}^{k}(I - P_i)$$

At initialization, for $k = 1$, we trivially have $I - P_1 = I - P_1$.

For the inductive step, we need to show that the sequence grows by the factor $(I - P_{k+1})$.

$$\prod_{i=1}^{k+1}(I - P_i) = (I - P_{k+1})\prod_{i=1}^{k}(I - P_i)$$

$$\begin{aligned} I - \sum_{i=1}^{k+1}\left(\prod_{j=i+1}^{k+1}(I - P_j)\right) P_i &= I - P_{k+1} - (I - P_{k+1})\sum_{i=1}^{k}\left(\prod_{j=i+1}^{k}(I - P_j)\right) P_i \\ &= (I - P_{k+1})\left(I - \left(\sum_{i=1}^{k}\left(\prod_{j=i+1}^{k}(I - P_j)\right) P_i\right)\right) \end{aligned}$$

Due to the equivalence of the two sequences, we can simplify the expression for the state of the model $A_{\theta,k}$

$$A_{\theta,k} = \left(\prod_{i=1}^{k}(I - P_k)\right)(A_0 - A_*) + A_*$$

□

Across multiple training episodes, let's analyze the interactions between consecutive episodes, given their projections P_1 and P_2 respectively. Ideally, the model would not experience catastrophic forgetting, however, we previously explained that in the case of training each episode with weight decay until convergence, the model would forget everything that isnt involved with the most recent task, here P_2.

Due to the inherent forgetting of training to convergence with weight decay, the rest of our discussion in this section will focus on the unregularized setting where $\lambda = 0$. Let's now measure the forgetting in the unregularized case.

Proposition 3.5. *In general, when training each episode to convergence without the use of weight decay, the single-layer model will experience forgetting unless all pairs of tasks commute, that is $P_iP_j = P_jP_i$ for any i, j.*

Proof. By induction, let's start by analyzing the forgetting of the first two tasks. Let P_2 be the projection for the most recent episode of training, and P_1 is the projection of the previous episode. If both episodes were trained to convergence then the current state of the model is

$$A_\theta = P_2A_* + (I - P_2)(P_1A_* + (I - P_1)A_0$$

As needed to evaluate the forgetting exhibited for task P_1, the original loss of the first task was

$$C_1 = \frac{1}{2}||P_1((P_1A_* - (I - P_1)A_0) - A_*)||^2 = 0$$

The forgetting of the first task is measured as follows.

$$\begin{aligned}\mathcal{F}(P_1, C_1, \theta) &= \frac{1}{2}||P_1(A_\theta - A_*)||^2 - C_1\\ &= \frac{1}{2}||P_1((P_2A_* + (I - P_2)(P_1A_* + (I - P_1)A_0) - A_*)||^2\\ &= \frac{1}{2}||(P_1P_2 - P_1P_2P_1)(A_* - A_0)||^2\end{aligned}$$

In general, for the forgetting to be zero for any A_0 after the second episode, then it must be that $P_1P_2 = P_1P_2P_1$. For this to be true, two projection matrices must commute. See the Appendix for further discussion on when projection matrices commute.

Similarly, we can measure the forgetting for any task after k episodes. Let's use Proposition 3.4 as the current state of the model after k episodes.

$$A_{\theta,k} = \left(\prod_{i=1}^{k}(I - P_k)\right)(A_0 - A_*) + A_*$$

Now lets measure the forgetting for some task P_n for $n < k$.

$$\mathcal{F}(P_n, 0, \theta) = \frac{1}{2}||P_n(A_{\theta,k} - A_*)||^2$$

For the forgetting to be zero for any A_0, we need

$$P_n\left(\left(\prod_{i=1}^{k}(I - P_k)\right)(A_0 - A_*) + A_*\right) - P_n A_* = 0$$

$$P_n \prod_{i=1}^{k}(I - P_k) = 0$$

If all pairs of tasks commute then we can simplify the expression by shifting P_i to the left enough. By shifting along the product operator, we eventually find that P_n will be multiplied by $(I - P_n)$ which will produce the intended result of zero. □

In general, it is unreasonable to assume all pairs of tasks commute, that means that the single-layer model will experience forgetting. Nevertheless, we can we can still prove convergence towards A_* even if it will experience forgetting on its path.

Similarly to the proof for Proposition 3.1, let's prove that the point of convergence in each episode gets closer to the true function A_*.

Proposition 3.6. *The state of the model after each episode can not grow further from the true function A_**

Proof.

$$
\begin{aligned}
||A_{\theta,k} - A_*||^2 &= ||\left(\prod_{i=1}^{k}(I - P_k)\right)(A_0 - A_*) + A_* - A_*||^2 \\
&= ||\left(\prod_{i=1}^{k}(I - P_k)\right)(A_0 - A_*)||^2 \\
&\leq ||\left(\prod_{i=1}^{k-1}(I - P_k)\right)(A_0 - A_*)||^2
\end{aligned}
$$

Given the norm reducing property of projections, we find that with every new episode the squared distance between the model and the true function is bounded by the distance at the previous episode.

$$
||A_{\theta,k+1} - A_*||^2 \leq ||A_{\theta,k} - A_*||^2
$$

□

Although the result from Proposition 3.6 is promising, it still doesn't prove convergence. In general, it may be possible that each task is identical, in which case we see that we won't necessarily converge to the true function after an infinite number of projections.

Nevertheless, we can talk about the probability of converging given random projections.

Proposition 3.7. *If each projection P_i is sampled uniformly from a set of rank-1 orthogonal tasks, then the probability $\mathbb{P}(A_{\theta,k} = A_*)$ of the model matching the true function is*

$$
\sum_{i=1}^{d}(-1)^{d+1}\binom{d}{i}\left(1 - \frac{i}{d}\right)^k
$$

Proof. As a reminder, our problem maps $\mathbb{R}^d \rightarrow \mathbb{R}^d$. Since the set of tasks is orthogonal and rank-1, there are d tasks in the set to span the entire space.

From Proposition 3.5 we know that if the tasks are orthogonal then there will be no forgetting. Thus, to match the true function, we only need to run at least one episode for each tasks in the set.

Given k independent uniform draws of the set of d tasks, let C_i denote the event that task i is selected within the k draws.

$$\mathbb{P}(C_i) = \left(\frac{d-1}{d}\right)^k$$

Using the inclusion-exclusion principle which states

$$\bigcup_{i=1}^{d} C_i = \sum_{i=1}^{d} C_i - \sum_{i<j}^{d}(C_i \cap C_j) + \sum_{i<j<k}^{d}(C_i \cap C_j \cap C_k) + \cdots + (-1)^{d+1} \bigcap_{i=1}^{d} C_i$$

For some set of n excluded tasks S, where $|S| = n$, the probability that all of them are unselected is

$$\mathbb{P}(\bigcap_{i \notin S} C_i) = \left(1 - \frac{n}{d}\right)^k$$

Thus by the inclusion-exclusion principle, we find that the probability of all tasks being trained at least once to be

$$\mathbb{P}\left(\bigcap_{i=1}^{d} C_i\right) = \sum_{i=1}^{d}(-1)^{d+1}\binom{d}{i}\left(1 - \frac{i}{d}\right)^k$$

where $\binom{d}{i}$ denotes the combinatorial operation d choose i. □

To be more realistic, we can consider the more general problem of sampling our tasks from the set of rank-1 tasks.

Proposition 3.8. *If each projection P_i is sampled uniformly from a set of rank-1 tasks, then the expected distance between $A_{\theta,k}$ and A_* is bounded by*

$$\mathbb{E}||A_{\theta,k} - A_*|| \leq d^{-\frac{k}{2}}(A_0 - A_*)$$

Proof. To bound the probability of $A_{\theta,k}$ matching A_*, we use the fact that, for some rank-1 task,

$(I - P_i)$ is a projection onto the hyperplane normal to the image of P_i, which is a line.

From Proposition 3.4 we have the state of the model.

$$A_{\theta,k} = \left(\prod_{i=1}^{k} (I - P_k) \right) (A_0 - A_*) + A_*$$

The model will only match the true function when $\prod_{i=1}^{k}(I - P_k) = 0$. Nevertheless, we can bound this product using the principal angles between the hyperplanes.

Note, a rank-1 projection can be written as the outerproduct of a unit vector $P = vv^T$. In this case, $||(I - P_2)(I - P_1)X|| \leq |\cos(\theta)| \cdot ||(I - P_1)A||$ where θ is the principal angle between the two tasks. Because the projections are rank-1, $\cos(\theta) = v_2^T v_1$.

By sampling the projections uniformly and independently, we can produce the following bound.

$$||A_{\theta,k} - A_*|| \leq |v_2^T v_1|^k (A_0 - A_*)$$

By rotational symmetry, the distribution of their inner-product of two uniformly random unit vectors v_1 and v_2 is the same as if we were to fix one of them. Let $v_1 = e_1$, the inner-product $v_2^T v_1$ can be written by the following sum.

$$v_2^T v_1 = v_2^T e_1 \sum_{i=1}^{d} v_{2,i} 1_{i=1} = v_{2,1}$$

By rotational symmetry, $\mathbb{E}[v_{2,1}] = 0$. To solve for the variance, note that v_2 is a unit vector, so $\sum_{i=1}^{d} v_{2,i}^2 = 1$. Since v_2 is drawn uniformly, it is has coordinate symmetry, thus $\text{Var}[v_{2,1}] = \frac{1}{d}$.

By the definition of the standard deviation, and since the mean of $v_2^T v_1$ is zero, we find that $|v_2^T v_1|$ is of the order of $\frac{1}{\sqrt{d}}$, which leads to the simplification

$$\mathbb{E}||A_{\theta,k} - A_*|| \leq d^{-\frac{k}{2}} (A_0 - A_*)$$

□

3.3 Role of weight decay

So far we have only considered training each episode to convergence, which makes weight decay undesirable. However, in practice we are training each episode for finite time. From Equation 3.1, the two decaying terms in the single episode dynamics are decaying at very different rates. The convergence towards the target PA_* occurs in $e^{-t} + \mathcal{O}(\lambda t)$ whereas the convergence caused by weight decay along the kernel of the task occurs in $e^{-\lambda t}$. Due to the difference in scales, for finite time training there exists a λ small enough such that the regularized dynamics are approximately unregularized.

Proposition 3.9. *If the model is trained on each episode for at most time T, then the dynamics of A_θ are approximately unregularized for $\lambda \ll \frac{1}{T}$*

Proof. The Taylor series expansion of $A_{\theta,T}$ recovers the unregularized dynamics to within the order of $\mathcal{O}(\lambda T)$

$$\begin{aligned}
A_{\theta,T} &= P\frac{A_*}{1+2\lambda} + P\left(A_0 - \frac{A_*}{1+2\lambda}\right)e^{-(1+2\lambda)T} + (I-P)A_0 e^{-2\lambda T} \\
&= PA_* + P(A_0 - A_*)e^{-T} + (I-P)A_0 - 2\lambda T(P(A_0 - A_*)e^{-T} + (I-P)A_0) \\
&= PA_* + (1-2\lambda T)(P(A_0 - A_*)e^{-T} + (I-P)A_0)
\end{aligned}$$

In practice we have $\lambda > 0$ small and $T \gg 1$ large. Nevertheless, for $2\lambda T \ll 1$, the dynamics are approximated unregularized. □

Although it's good news that small regularization wont significantly affect performance, if the goal is to be approximately unregularized the question becomes why use weight decay at all.

The reason it has not seemed fitting to use weight decay has been the assumption that A_* is full rank. If A_* is full rank then there exists a unique solution $A_\theta = A_*$ for the problem. However, if the true function was rank deficient, we can think in terms of A_* being some projected matrix of a super target function A_{**}, where $A_* = P_* A_{**}$. In this way it becomes clear from our previous analysis that there are an infinite number of solutions to the problem along the kernel of P_*. Thus, to assume a unique solution to the continual learning problem, we can use small weight decay.

Proposition 3.10. *With weight decay, the weights of the model along the kernel of the true function will decay exponentially to zero across each episode, unaffected by the choice of task.*

Proof. Let P_i be the projection for the current task, and the true function will be written as $P_* A_{**}$. We can separate the model A_θ in terms of its projection in P_* and its projection in $(I - P_*)$.

$$\begin{aligned}
A_{\theta,t} &= P\frac{P_* A_{**}}{1+2\lambda} + P\left(A_0 - \frac{P_* A_{**}}{1+2\lambda}\right)e^{-(1+2\lambda)t} + (I-P)A_0 e^{-2\lambda t} \\
&= PP_*\frac{A_{**}}{1+2\lambda} + PP_*\left(A_0 - \frac{A_{**}}{1+2\lambda}\right)e^{-(1+2\lambda)t} + P(I-P_*)A_0 e^{-(1+2\lambda)t} + (I-P)A_0 e^{-2\lambda t} \\
&= PP_*\frac{A_{**}}{1+2\lambda} + PP_*\left(A_0 - \frac{A_{**}}{1+2\lambda}\right)e^{-(1+2\lambda)t} + \left(e^{-t}P + (I-P)\right)(I-P_*)A_0 e^{-2\lambda t} + (I-P)P_* A_0 e^{-2\lambda t} \\
&= P_*\frac{A_{**}}{1+2\lambda} + PP_*\left(A_0 - \frac{A_{**}}{1+2\lambda}\right)e^{-(1+2\lambda)t} + (I-P_*)A_0 e^{-2\lambda t} - (1-e^{-t})P(I-P_*)A_0 e^{-2\lambda t} + (I-P)P_* A_0 e^{-2\lambda t}
\end{aligned}$$

By isolating the term that only depends on the kernel of P_*, we see that it decays independently of the choice of task. By contrast, the weights along the kernel of P_* will only decay in the unregularized case if the range of P is included in the range of P_*. □

In practice, we are interested in the case where the rank of each task is small compared to the rank of the true function. In that case, it is reasonable to assume that many tasks could be

4 | Deep Diagonal Case

In the previous chapter we explored the single-layer linear network in a continual learning problem. We found many positive results such as convergence to the true function, as well as the role, albeit mild, of weight decay. Now we will explore the deep case where $L > 1$, and find that the dynamics become nonlinear with respect to the parameters θ of the model.

As a reminder, the deep model can be written as

$$A_\theta = \prod_{\ell=1}^{L} W_\ell$$

where $\prod$ denotes the recursive right-multiply operator. Note that the variable ℓ is reserved for designating a particular layer ℓ of the model.

First and foremost, all of our analysis in the deep case will rely on the standard balancedness assumption used in much of the literature on deep linear networks. We state and prove it generally in the following theorem.

Theorem 4.1 (Balancedness)**.** *With weight decay, deep linear networks converge exponentially to being balanced, i.e.*

$$W_{\ell+1}W_{\ell+1}^T = W_\ell^T W_\ell$$

Proof. To simplify notation, if a layer subscript finishes with an unresolved + or − sign then it denotes the following product.

$$\begin{cases} W_{\ell-} = \prod_{i=1}^{\ell-1} W_i, \\ W_{\ell+} = \prod_{i=\ell+1}^{L} W_i. \end{cases}$$

Thus A_θ can be written as $W_{\ell-}W_\ell W_{\ell+}$ for any ℓ.

Since the weights are trained with gradient descent, we find that the dynamics to be as follows.

$$\begin{aligned} \frac{dW_\ell}{dt} &= -\nabla_{W_\ell} C(P, \theta, \lambda) \\ &= -\nabla_{W_\ell} \frac{1}{2} ||P(W_{\ell-}W_\ell W_{\ell+} - A_*)||^2 + \lambda \sum_{i=1}^{L} ||W_i||^2 \\ &= W_{\ell-}^T P(A_* - A_\theta) W_{\ell+}^T - 2\lambda W_\ell \end{aligned}$$

Now we can solve for the evolution of the balancedness of the model.

$$\begin{aligned} \frac{d}{dt}\left(W_{\ell+1}W_{\ell+1}^T - W_\ell^T W_\ell\right) &= \frac{dW_{\ell+1}}{dt} W_{\ell+1}^T + W_{\ell+1}\frac{dW_{\ell+1}^T}{dt} - \frac{dW_\ell^T}{dt} W_\ell - W_\ell^T \frac{dW_\ell}{dt} \\ &= W_{\ell+1-}^T P(A_\theta - A_*) W_{\ell+1+}^T W_{\ell+1}^T + W_{\ell+1} W_{\ell+1+}(A_\theta^T - A_*^T) P W_{\ell+1-} \\ &\quad - W_{\ell+}(A_\theta^T - A_*^T) P W_{\ell-} W_\ell - W_\ell^T W_{\ell+}^T P(A_\theta - A_*) W_{\ell-}^T \\ &\quad - 4\lambda W_{\ell+1} W_{\ell+1}^T + 4\lambda W_\ell^T W_\ell \\ &= W_{\ell+1-}^T P(A_\theta - A_*) W_{\ell+}^T + W_{\ell+}(A_\theta^T - A_*^T) P W_{\ell+1-} \\ &\quad - W_{\ell+}(A_\theta^T - A_*^T) P W_{\ell+1-} - W_{\ell+1+}^T P(A_\theta - A_*) W_{\ell-}^T \\ &\quad - 4\lambda \left(W_{\ell+1}W_{\ell+1}^T - W_\ell^T W_\ell\right) \\ &= -4\lambda \left(W_{\ell+1}W_{\ell+1}^T - W_\ell^T W_\ell\right) \end{aligned}$$

$$\left(W_{\ell+1}W_{\ell+1}^T - W_\ell^T W_\ell\right)(t) = \left(W_{\ell+1}W_{\ell+1}^T - W_\ell^T W_\ell\right)(0)\ e^{-4\lambda}$$

□

Theorem 4.1 highlights the very important role of weight decay that is new in the deep case. For chapters 4 and 5, this property will be systematically assumed to be true. Previously, we introduced the importance of upper bounding λ by the inverse of the length T of a given episode, $\lambda \ll \frac{1}{T}$. Here, note that the convergence to balancedness occurs independently from each episode, so there will be a lower bound on λ as a function of the total training time kT, to ensure approximate balancedness. Alternatively, consider the case where the model is initialized as balanced, then even without regularization, it will remain balanced throughout training.

4.1 Alignment equivalence

In this chapter, we will begin by focusing on a reduction of the deep case that occurs when the model aligns itself with a given episode's local true function PA_*.

Theorem 4.2 (Diagonal Reduction). *If the model A_θ is balanced and can be decomposed into $U_* D_\theta V_*^T$ for some diagonal matrix D_θ, where U_* and V_* are the left and right singular vector matrices of PA_*, then the problem in this given episode can be reduced to the diagonal problem*

$$\min_\theta \frac{1}{2}||D_{\theta,r} - \Sigma_*||^2 + \lambda L \sum_{i=1}^{d} D_{\theta,ii}^{\frac{2}{L}}$$

where Σ_* is the singular value matrix of PA_* and $D_{\theta,r}$ denotes the truncated D_θ where all diagonal elements $D_\theta, ii \longleftarrow 0$ are set to zero for all indices $i > r$ where $r = Rank(P)$.

Proof. Let's start by developing the original minimization problem.

$$\min_\theta \frac{1}{2}||P(A_\theta - A_*)||^2 + \lambda \sum_{\ell=1}^{L} ||W_\ell||^2$$

$$= \min_\theta \frac{1}{2}||PU_* D_\theta V_*^T - U_* \Sigma_* V_*^T)||^2 + \lambda \sum_{\ell=1}^{L} ||W_\ell||^2$$

To start, considering that $U_*\Sigma_* V_*^T$ is the singular value decomposition of PA_*, and the projection P is idempotent, we have that $PU_*\Sigma_* V_*^T = U_*\Sigma_* V_*^T$. Since P is rank deficient, we know that Σ_* has some zero singular values. Let r denote the rank of PA_*, we can the singular vector matrices as a concatenation between the eigenvectors associated to non-zero singular values, and those associated to singular values of zero. Let's denote $U_* = [U_{*,\leq r}|U_{*,>r}]$ and $V_* = [V_{*,\leq r}|V_{*,>r}]$ as the separation.

For $U_*\Sigma_* V_*^T$ to be unchanged by the projection P, we have that $PU_{*,\leq r} = U_{*,\leq r}$, thus $U_{*,\leq r}$ is in the image of P. With the added assumption that A_* is full rank, we get that r is the rank of P and $U_{*,>r}$ is in the kernel of P.

Thus $PA_\theta = U_* D_{\theta,r} V_*^T$, where all diagonal elements of $D_{\theta,r,ii} = 1_{i\leq r} D_{\theta,ii}$ are either unchanged or masked.

Now we can factor out the eigenvectors from the minimization problem using the property that the Frobenius norm square can be written as $||X||^2 = \text{Tr}(X^T X)$ and the two following properties of trace operators, $\text{Tr}(ABC) = \text{Tr}(CAB)$ and $\text{Tr}(A+B) = \text{Tr}(A) + \text{Tr}(B)$. Let's start by focusing on the first term of the minimization problem.

$$
\begin{aligned}
&\frac{1}{2}||PU_* D_\theta V_*^T - U_*\Sigma_* V_*^T)||^2 \\
&= \frac{1}{2}||U_* D_{\theta,r} V_*^T - U_*\Sigma_* V_*^T)||^2 \\
&= \frac{1}{2}\,\text{Tr}\left((V_* D_{\theta,r}^T U_*^T - V_*\Sigma_*^T U_*^T)(U_* D_{\theta,r} V_*^T - U_*\Sigma_* V_*^T)\right) \\
&= \frac{1}{2}\left(\text{Tr}(D_{\theta,r}^T D_{\theta,r}) - \text{Tr}(\Sigma_*^T D_{\theta,r}) - \text{Tr}(D_{\theta,r}^T \Sigma_*) + \text{Tr}(\Sigma_*^T\Sigma_*)\right) \\
&= \frac{1}{2}||D_{\theta,r} - \Sigma_*||^2
\end{aligned}
$$

Note that for any singular value decomposition, we get that $||U\Sigma V^T||^2 = \text{Tr}(\Sigma^T\Sigma) = \sum_{i=1}^d \sigma_i^2$. For the second term, we will rely on the this property of the Frobenius norm and solve for the singular values at each layer using the balancedness condition of A_θ. Let $A_\theta = U_\theta \Sigma_\theta V_\theta^T$ be a singular value decomposition of A_θ.

$$\begin{aligned}
A_\theta A_\theta^T &= W_1 W_2 \cdots W_L W_L^T \cdots W_2^T W_1^T \\
U_\theta \Sigma_\theta^2 U_\theta &= (W_1 W_1^T)^L \\
U_\theta \Sigma_\theta^2 U_\theta &= U_1 \Sigma_1^{2L} U_1^T
\end{aligned}$$

By equivalence, we find that each singular value of the first layer $\sigma_{1,i}^L = \sigma_{\theta,i}$.

Due to balancedness, we also have that all layers have the same singular values

$$\begin{aligned}
W_{\ell+1} W_{\ell+1}^T &= W_\ell^T W_\ell \\
U_{\ell+1} \Sigma_{\ell+1}^2 U_{\ell+1}^T &= V_\ell \Sigma_\ell^2 V_\ell^T \\
\Sigma_{\ell+1} &= \Sigma_\ell
\end{aligned}$$

Now to evaluate the second term of the minimization problem, we need to prove that the singular values of A_θ are the absolute values of the diagonal D_θ. The proof relies on the fact that an unordered singular value decomposition (i.e. one in which the diagonal of the singular value matrix is unordered), consists of any decomposition of a matrix into a product ABC, where A and C are orthogonal and B is diagonal and positive. The current decomposition of A_θ almost satisfies the criteria for being a singular value decomposition except its values may be negative. To fix that, we can pass on the negative sign of any negative elements in D_θ to the corresponding vector in V_*. Since V_* is orthogonal, then changing the sign of any of its columns ensures it is still orthogonal. With this transformation, we have transformed D_θ to be non-negative diagonal matrix, which is left and right multiplied by orthogonal matrices to produce A_θ. Thus it is a valid singular value decomposition.

$$\begin{aligned}\lambda\sum_{\ell=1}^{L}||W_\ell||^2 &= \lambda\sum_{\ell=1}^{L}\sum_{i=1}^{d}\sigma_{\ell,i}^2\\ &= \lambda L\sum_{i=1}^{d}\sigma_{\theta,i}^{\frac{2}{L}}\\ &= \lambda L\sum_{i=1}^{d}D_{\theta,ii}^{\frac{2}{L}}\end{aligned}$$

Putting both terms together, we recover the reduced minimization problem, which is fully diagonalized. □

With the reduction to a diagonal problem, this chapter will be focused on the diagonal deep case. That is, where P, A_* and every layer W_ℓ is a diagonal matrix and A_* is even positive semi-definite.

4.2 Single episode dynamics

The reduction from the general case to the diagonal case turns the matrix problem into a scalar problem.

Lemma 4.3. *The diagonalized and balanced formulation can be expressed as a scalar minimization problem.*

$$\min_\theta \frac{1}{2}\sum_{i=1}^{d}P_i\left(W_i^L - A_{*,i}\right)^2 + \lambda L\sum_{i=1}^{d}W_i^2$$

Proof. The matrix product of two diagonal matrices is the same as the element-wise product. This enables us to solve for each element of the problem.

$$(P(A_\theta - A_*))_i = P_i\left(\prod_{\ell=1}^{L}W_{\ell,i} - A_{*,i}\right)$$

We also have that the Frobnius norm squared is just the sum of squared elements.

$$\frac{1}{2}||P(A_\theta - A_*)||^2 + \lambda \sum_{\ell=1}^{L} ||W_\ell||^2 = \frac{1}{2}\sum_{i=1}^{d} P_i \left(\prod_{\ell=1}^{L} W_{\ell,i} - A_{*,i}\right)^2 + \lambda \sum_{\ell=1}^{L}\sum_{i=1}^{d} W_{\ell,i}^2$$

Lastly, from the balancedness assumption we have that $W_i = W_j$ for all i, j, which leads to the following simplification.

$$\frac{1}{2}\sum_{i=1}^{d} P_i \left(W_i^L - A_{*,i}\right)^2 + \lambda L \sum_{i=1}^{d} W_i^2$$

□

Let's start by solving for the dynamics of the model using gradient descent.

Proposition 4.4. *The dynamics of each diagonal element $A_{\theta,i}$ solve the scalar ordinary differential equation*

$$\frac{dA_{\theta,i}}{dt} = LP_i(A_{*,i} - A_{\theta,i})A_{\theta,i}^{2-\frac{1}{L}} - 2\lambda L A_{\theta,i}$$

Proof. From Lemma 4.3 we have that the cost is

$$C(P,\theta,\lambda) = \frac{1}{2}\sum_{i=1}^{d} P_i \left(W_i^L - A_{*,i}\right)^2 + \lambda L \sum_{i=1}^{d} W_i^2$$

Although it's tempting to solve for the dynamics of W_i, we are not able to simply take the gradient of the cost with respect to W_i because we need to take the gradient with respect to each parameter $W_{\ell,i}$.

$$\begin{aligned}\frac{dW_{\ell,i}}{dt} &= -\nabla_{W_{\ell,i}} C(P,\theta,\lambda)\\ &= -\nabla_{W_{\ell,i}}\left(\frac{1}{2}\sum_{i=1}^{d} P_i\left(W_i^{L-1}W_{\ell,i} - A_{*,i}\right)^2 + \lambda\sum_{\ell=1}^{L}\sum_{i=1}^{d} W_{\ell,i}^2\right)\\ &= P_i(A_{*,i} - W_i^L)W_{\ell,i}^{L-1} - 2\lambda W_i\end{aligned}$$

From there we find the dynamics of the model.

$$
\begin{aligned}
\frac{dA_{\theta,i}}{dt} &= \sum_{\ell=1}^{L} W_{\ell-,i} \frac{dW_{\ell,i}}{dt} W_{\ell+,i} \\
&= W_i^{L-1} \sum_{\ell=1}^{L} \left(P_i(A_{*,i} - W_i^L) W_{\ell,i}^{L-1} - 2\lambda W_i \right) \\
&= LP_i(A_{*,i} - A_{\theta,i}) A_{\theta,i}^{2-\frac{1}{L}} - 2\lambda L A_{\theta,i}
\end{aligned}
$$

□

4.3 Fixed point analysis

From Proposition 4.4, the dynamics of the model are characterized by a first degree homogeneous nonlinear ODE. The ODE doesn't have an explicit solution mainly due to the factor $A_{\theta,i}^{\frac{2(L-1)}{L}}$ making it a non-polynomial ODE. Nevertheless, we can study its general behavior using a fixed point analysis.

Corollary 4.5. *As a diagonal projection matrix,* $P_i \in \{0, 1\}$

Proof. P is an orthogonal projection matrix, thus it must be idempotent and symmetric. As a diagonal matrix, we get that it is symmetric. However for it to be idempotent, its diagonal elements must satisfy $P_i^2 = P_i$ thus, $P_i \in \{0, 1\}$. □

Since P_i is either 1 or 0, we recognize the idea from Proposition 3.2 that for $P_i = 0$, the dynamics are simply decaying to zero.

Proposition 4.6. *If* $P_i = 1$, *then the element* $A_{\theta,i}$ *exhibits three fixed points, two of which are named*

h and $A_{\lambda *}$ respectively for clarity and convenience.

$$A_{\theta,i} = \begin{cases} 0, \\ h = \left(\dfrac{2\lambda}{A_{*,i}}\right)^{\frac{L}{L-2}}, \\ A_{\lambda *} = A_{*,i} - 2\lambda A_{*,i}^{\frac{L}{L-2}}. \end{cases}$$

Proof. Since the ODE is homogeneous, we can factor out $A_{\theta,i}$, trivially proving that zero is a fixed point. The other two fixed points come from the solutions of the remaining equation.

$$(A_{*,i} - A_{\theta,i}) A_{\theta,i}^{\frac{L-2}{L}} = 2\lambda$$

Since λ is small, we know that the solutions are $0 + \mathcal{O}(\lambda)$ and $A_{*,i} + \mathcal{O}(\lambda)$. For more accuracy, we can approximate the fixed points by solving for a linear perturbation around those two points.

First near 0, let $A_{\theta,i} = \epsilon$

$$\begin{aligned}(A_{*,i} - \epsilon)\epsilon^{\frac{L-2}{L}} \approx A_{*,i}\epsilon^{\frac{L-2}{L}} &= 2\lambda \\ \Rightarrow \epsilon \approx \left(\frac{2\lambda}{A_{*,i}}\right)^{\frac{L}{L-2}}\end{aligned}$$

Similarly near $A_{*,i}$, let $A_{\theta,i} = A_{*,i} + \epsilon$

$$\begin{aligned}(A_{*,i} - A_{*,i} - \epsilon)(A_{*,i} + \epsilon)^{\frac{L-2}{L}} \approx -\epsilon A_{*,i}^{\frac{L-2}{L}} &= 2\lambda \\ \Rightarrow \epsilon \approx -2\lambda A_{*,i}^{\frac{L}{L-2}}\end{aligned}$$

□

Proposition 4.6 proves the existence of the three fixed points, now we can study the stability at each point.

Proposition 4.7. *The fixed points* 0 *and* $A_{\lambda*}$ *are stable, and the fixed point* h *is unstable.*

Proof. To find the stability of each point, we must solve for the sign of $\nabla\frac{dA_{\theta,i}}{dt}$ at each fixed point.

$$\nabla\frac{dA_{\theta,i}}{dt} = (2-3L)A_{\theta,i}^{1-\frac{2}{L}} + (2L-2)A_{*,i}A_{\theta,i}^{1-\frac{2}{L}} - 2\lambda L$$

At the origin, the gradient is negative, thus the origin is stable.

Although we don't have the exact form of h and $A_{\lambda*}$, we can indirectly solve for their stabilities using the intermediate value theorem.

We have that

$$\frac{dA_{\theta,i}}{dt}\Big|_{\frac{A_{*,i}}{2}} = \left(\frac{1}{2}\left(\frac{A_{*,i}}{2}\right)^{2+\frac{2}{L}} - 2\lambda\right)LA_{*,i} > 0$$

We also have that

$$\frac{dA_{\theta,i}}{dt}\Big|_{2A_{*,i}} = -LA_{*,i}\left(2A_{*,i}\right)^{2+\frac{2}{L}} - 2\lambda LA_{*,i} < 0$$

By the intermediate value theorem, since h is the only fixed point between the origin and $\frac{A_{*,i}}{2}$, it's gradient must be positive, thus it is unstable. In the same way, since $\frac{dA_{\theta,i}}{dt}$ goes to $-\infty$ and the only fixed point greater than h is $A_{\lambda*}$ then it must be stable. □

Proposition 4.7 implies that for $A_{\theta,i}$ initialized outside of the interval $[0, A_{\lambda*}]$ it will converge to the their nearest fixed points (either 0 or $A_{\lambda*}$), for an initialization in the interval $[0, h)$, it will converge to 0 and for an initialization in the interval $(h, A_{\lambda*}$ it will converge to $A_{\lambda*}$.

This is bad news because any initialization of the opposite sign of A_* will go to zero. In the next chapter, we will show that the freedom to rotate largely mitigates this catastrophic behavior. Nonetheless, there is more bad news for points inside the interval $[0, h)$. Due to the decaying nature of the model when $P_i = 0$, then training on a task for too long might lead to a phenomena we will call *catastrophic weight loss*. Unlike catastrophic forgetting, catastrophic weight loss is irreversible using gradient descent. For this reason, we will refer to fixed point h as the event horizon. All weights that decay past the event horizon are unrecoverable.

4.4 Consecutive episode dynamics

In the previous section, we found that the dynamics of the model are driven by d independent ODEs where the weights either converge to the regularized target $A_{\lambda *}$ or to zero. Since all the tasks in the diagonal setting are orthogonal to one-another, the only question for consecutive episode dynamics is whether any of the weights ever pass the below the event horizon h. To adequately answer this question, we can frame it in turns of the time to reach the event horizon.

Proposition 4.8. *If $P_i = 0$, the time it takes to reach the event horizon using gradient descent is*

$$t = \frac{\ln(A_{\theta,i}(0)) - \frac{L}{L-2}\left(\ln(2\lambda) - \ln(A_{*,i})\right)}{2\eta\lambda L}$$

Proof. For $P_i = 0$, we can solve for $A_{\theta,i}(t)$ with a learning rate η.

$$\frac{dA_{\theta,i}}{dt} = -2\eta\lambda L A_{\theta,i}$$
$$A_{\theta,i}(t) = A_{\theta,i}(0)e^{-2\eta\lambda L t}$$

Using gradient descent, the number of steps to pass the event horizon is

$$t = \frac{\ln(A_{\theta,i}(0)) - \ln(h)}{2\eta\lambda L}$$

Using our previous approximation for h, we get

$$t = \frac{\ln(A_{\theta,i}(0)) - \frac{L}{L-2}\left(\ln(2\lambda) - \ln(A_{*,i})\right)}{2\eta\lambda L}$$

□

For $\lambda \ll 1$, the time it takes to reach the event horizon is of the order $O\left(\ln(\frac{1}{\lambda})\right)$, which is very large, but can be achieved reasonably if training for exponential time.

4.5 Stochastic gradient descent

If we were to sample a new projection at each training step, the learning dynamics can be reinterpreted as the dynamics of stochastic gradient descent. Let the rank of P match the number of unique tasks selected by a batch size of b, can we measure the probability of catastrophic weight loss?

Proposition 4.9. *The dynamics of stochastic gradient descent in the regularized deep diagonal case can be translated into a gambler's ruin problem.*

Proof. The gambler's ruin problem is a binomial random walk problem where binomial probability and step size at each discrete time t is variable. In the case of using SGD with weight decay to train a deep diagonal network, the game becomes how many steps are remaining for each singular value before the fall below the event horizon. By using time instead of distance, it becomes a standard version of gamblers ruin problem because each step has the same "cost" of one weight decay step, and has a chance to hit the lottery and increase overall, extending the time it would take to reach the event horizon.

Let's formalize the concept of cost and lottery in the context of the deep diagonal model. Let t_i be the time it would take singular value σ_i to decay past the event horizon if P_i keeps getting sampled as zero. Finally, let η be the learning rate for SGD, the gradient step of $A_{\theta,i}$ is as follows

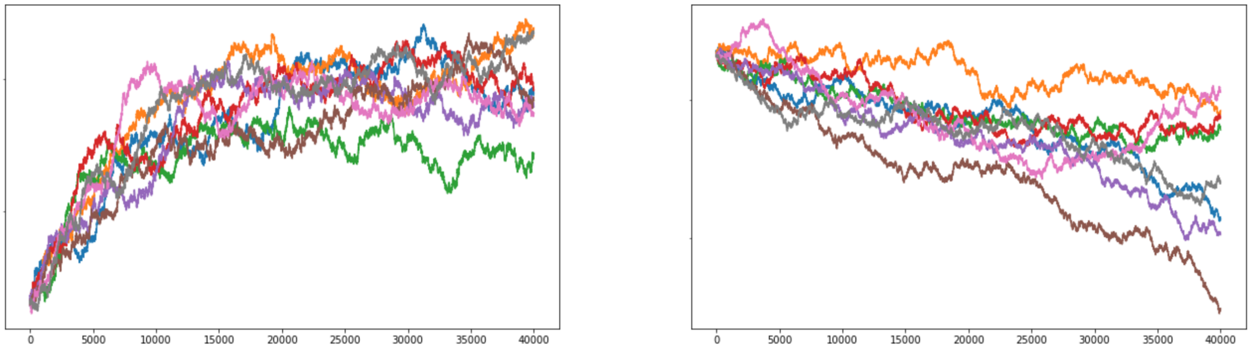
$$A_{\theta,i,t+1} = A_{\theta,i,t} - 2L\eta \left(P_i \cdot A_{\theta,i,t}^{\frac{2(L-1)}{L}} (A_{\theta,i,t} - A_{*,i}) + \lambda A_{\theta,i,t} \right)$$

$$
\begin{aligned}
A_{\theta,i,t+1} &= A_{\theta,i,t} - 2L\eta\lambda A_{\theta,i,t} \\
&= A_{\theta,i,t}\left(1 - 2L\eta\lambda\right) \\
&= A_{*,i}\left(1 - 2L\eta\lambda\right)^{t+1} \\
A_{\theta,i,t} &\leq h \\
\left(1 - 2L\eta\lambda\right)^{t} &\leq \frac{h}{A_{*,i}} \\
t &\geq \frac{\ln(h) - \ln(A_{*,i})}{\ln(1 - 2L\eta\lambda)}
\end{aligned}
$$

To measure the lottery, the amount by which t increases when $P_i = 1$, we need to solve for the difference in time between $P_i = 0$ and $P_i = 1$.

$$
\begin{aligned}
A_{\theta,i,t}\left(1 - 2L\eta\lambda\right) &= A_{\theta,i,t+1+s} \\
&= A_{\theta,i,t+1}\left(1 - 2L\eta\lambda\right)^{s} \\
&= \left(A_{\theta,i,t}\left(1 - 2L\eta\lambda\right) - 2L\eta P_i \cdot A_{\theta,i,t}^{\frac{2(L-1)}{L}}\left(A_{\theta,i,t} - A_{*,i}\right)\right)\left(1 - 2L\eta\lambda\right)^{s} \\
s &= \frac{\ln\left(A_{\theta,i,t}\left(1 - 2L\eta\lambda\right)\right) - \ln\left(A_{\theta,i,t}\left(1 - 2L\eta\lambda\right) - 2L\eta P_i \cdot A_{\theta,i,t}^{\frac{2(L-1)}{L}}\left(A_{\theta,i,t} - A_{*,i}\right)\right)}{\ln(1 - 2L\eta\lambda)} \\
&= 1 - \frac{\ln\left(\left(1 - 2L\eta\lambda\right) + 2L\eta P_i \cdot A_{\theta,i,t}^{\frac{L-2}{L}}\left(A_{*,i} - A_{\theta,i,t}\right)\right)}{\ln(1 - 2L\eta\lambda)} \\
&\leq 1 - \frac{\ln\left(\left(1 - 2L\eta\lambda\right) + 2L\eta P_i \cdot A_{*,i}\right)}{\ln(1 - 2L\eta\lambda)}
\end{aligned}
$$

By bounding the positive effect of the $P_i = 1$, we can study the dynamics of forgetting by analyzing the binomial walk in the following algorithm. Let's assume a uniform distribution of



(a) $\lambda = 5 \times 10^{-2}$

(b) $\lambda = 6 \times 10^{-2}$

Figure 4.1: SGD simulation on deep diagonal network ($b = 1$, $d = 8$)

$1/d$ for each sample point. Thus, for a batch size of b, the probability $\mathbb{P}(P_i = 1) = \frac{1}{d^b}$

```
INITIALIZE $X_i = \frac{\ln(h) - \ln(A_{\theta,i}(t=0))}{\ln(1 - 2L\eta\lambda)}$
At each time t:
IF    $X_i \leq 0$ THEN HALT
ELSE
    $X_i = X_i - 1$                                                        with probability $1 - \frac{1}{d^b}$
    $X_i = X_i - \frac{\ln\left(1 + 2L\eta(A_{*,i} - \lambda)\right)}{\ln(1 - 2L\eta\lambda)}$        with probability $\frac{1}{d^b}$
```

It is a binomial walk with a halting condition, thus it is a gambler's ruin problem. This also reveals an inverse relationship between batch size and dimensionality. □

In Figure 4.1 we see how small changes in λ, b and d could have dramatic consequences when it comes to catastrophic weight loss.

5 | General Deep Case

Unlike the previous chapters, the general case does not have dynamics that are simple to characterize due to two competing non-linear dynamics: the rotation of singular vectors, and the change in singular values. Although Chapter 3 experienced both of these phenomena, its dynamics were linear. In the general case, the dynamics are characterized by the following equation.

$$\frac{dA_\theta}{dt} = \sum_{\ell=1}^{L} (A_\theta A_\theta^T)^{\frac{\ell-1}{L}} P(A_* - A_\theta)(A_\theta^T A_\theta)^{\frac{L-\ell}{L}} - 2\lambda L A_\theta \tag{5.1}$$

5.1 Single episode dynamics

Lemma 5.1. *All the fixed points of the regularized cost satisfy that A_θ is in the image of P.*

Proof. The training dynamics are defined as gradient flow with respect to the parameters, so lets start with the dynamics of each layer W_ℓ.

$$\begin{aligned}
\frac{dW_\ell}{dt} &= -\nabla_{W_\ell} C(P, \theta, \lambda) \\
&= -\nabla \left(||P(W_{\ell-} W_\ell W_{\ell+} - A_*)||^2 + \sum_{i=1}^{L} ||W_i||^2 \right) \\
&= W_{\ell-}^T P(A_* - A_\theta) W_{\ell+}^T - 2\lambda W_\ell
\end{aligned}$$

Using the chain rule and the balancedness assumption, we are able to solve for the dynamics of the model A_θ as a function of just P, A_* and A_θ itself (with the additional constants λ and L).

$$\begin{aligned}
\frac{dA_\theta}{dt} &= \sum_{\ell=1}^{L} W_{\ell-}\frac{dW_\ell}{dt}W_{\ell+} \\
&= \sum_{\ell=1}^{L}(W_1W_1^T)^{\ell-1}P(A_* - A_\theta)(W_L^TW_L)^{L-\ell} - 2\lambda LA \\
&= \sum_{\ell=1}^{L}(A_\theta A_\theta^T)^{\frac{\ell-1}{L}}P(A_* - A_\theta)(A_\theta^TA_\theta)^{\frac{L-\ell}{L}} - 2\lambda LA_\theta
\end{aligned}$$

Let the converged solution be written as the sum $A_\theta = PA + (I-P)A$ for some full rank A. Here we show that weight decay drives $(I-P)A$ to zero.

$$\begin{aligned}
\frac{dA_\theta}{dt} &= \sum_{\ell=1}^{L}\left(P(AA^T)^{\frac{\ell-1}{L}}P + (I-P)(AA^T)^{\frac{\ell-1}{L}}(I-P)\right)P(A_* - A_\theta)(A_\theta^TA_\theta)^{\frac{L-\ell}{L}} - 2\lambda LA_\theta \\
&= \sum_{\ell=1}^{L}P(AA^T)^{\frac{\ell-1}{L}}P(A_* - A)(A^TPA + A^T(I-P)A)^{\frac{L-\ell}{L}} - 2\lambda LPA - 2\lambda L(I-P)A
\end{aligned}$$

Since P is orthogonal to $(I-P)$, then for the dynamics to converge to a fixed point the model must be in the image of P.

□

Lemma 5.1 proves even in the general case, weight decay still exhibits an explicit low rank bias.

Theorem 5.2. *The event horizon exists in the general deep case as a function of A_**

Proof. In the general case, the event horizon is the point past which the SGD dynamics are unable to recover from the rank underestimate of the model. To study the rank, we can analyze the dynamics of the singular values of A_θ.

$$\begin{aligned}\sigma_i &= u_i^T A_\theta v_i \\ \frac{d\sigma_i}{dt} &= \frac{du_i^T}{dt} A_\theta v_i + u_i^T \frac{dA_\theta}{dt} v_i + u_i^T A_\theta \frac{dv_i}{dt}\end{aligned}$$

The vectors u_i and v_i exist on a sphere of radius 1. Their dynamics exist on the tangent plane at the point u_i and v_i respectively. This is because

$$\frac{d}{dt}\langle u, u\rangle = \frac{d}{dt} 1 = 0$$

By definition of their tangent plane, the dynamics are orthogonal u_i and v_i respectively. Since u_i and v_i are columns of a orthogonal basis for their space, the dynamics $\frac{du_i}{dt}$ and $\frac{dv_i}{dt}$ can be written as a linear combination of the columns of U and V respectively.

$$\begin{aligned}\frac{du_i^T}{dt} &= \sum_{j=1}^{d} \langle \frac{du_i^T}{dt}, u_j^T \rangle u_j^T \\ &= \sum_{j \neq i}^{d} \langle \frac{du_i^T}{dt}, u_j^T \rangle u_j^T\end{aligned}$$

Substituting into our equation for $\frac{d\sigma_i}{dt}$, we get the following simplifications.

$$\begin{aligned}
\frac{du_i^T}{dt}A_\theta v_i &= \sum_{j\neq i}^{d}\langle \frac{du_i^T}{dt}, u_j^T\rangle u_j^T A_\theta v_i \\
&= \sum_{j\neq i}^{d}\langle \frac{du_i^T}{dt}, u_j^T\rangle \sigma_j v_j^T v_i \\
&= 0
\end{aligned}$$

Similarly, $\frac{dv_i}{dt} = \sum_{j\neq i}^{d}\langle \frac{dv_i}{dt}, v_j\rangle v_j$ which leads to $u_i^T A_\theta \frac{dv_i}{dt} = 0$. This leaves us with one term to study.

$$\begin{aligned}
\frac{d\sigma_i}{dt} &= u_i^T \frac{dA_\theta}{dt} v_i \\
&= u_i^T \left(\sum_{\ell=1}^{L} (W_1 W_1^T)^{\ell-1} P(A_* - A_\theta)(W_L^T W_L)^{L-\ell} - 2\lambda L A_\theta \right) v_i \\
&= \sum_{\ell=1}^{L} u_i^T U_1 S_1^{\frac{2}{L}(\ell-1)} U_1^T P(A_* - A_\theta) V_L S_L^{\frac{2}{L}(L-\ell)} V_L^T v_i - 2\lambda L \sigma_i \\
&= \sum_{\ell=1}^{L} u_i^T U_1 S_1^{\frac{2}{L}(\ell-1)} U_1^T P(A_* - A_\theta) V_L S_L^{\frac{2}{L}(L-\ell)} V_L^T v_i - 2\lambda L \sigma_i \\
&= \sum_{\ell=1}^{L} \sigma_i^{\frac{2}{L}(\ell-1)} u_i^T P(A_* - A_\theta) v_i \sigma_i^{\frac{2}{L}(L-\ell)} - 2\lambda L \sigma_i \\
&= L\sigma_i^{\frac{2(L-1)}{L}} (u_i^T P A_* v_i - u_i P A_\theta v_i) - 2\lambda L \sigma_i \\
&\leq \hat{\sigma}_* L \sigma_i^{\frac{2(L-1)}{L}} - 2\lambda L \sigma_i
\end{aligned}$$

From Property A.7 we get that the rate of change for each of the singular values can be bounded by the expression in terms of the largest singular value $\hat{\sigma}_*$ of A_*. This enables us to define a region in which no rank expansion is possible (a region of the event horizon).

For $L > 2$ and $\sigma_i > 0$

$$\begin{aligned} \frac{d\sigma_i}{dt} &< 0 \\ \hat{\sigma}_* L \sigma_i^{\frac{L-2}{L}} - 2\lambda L &< 0 \\ \sigma_i^{\frac{L-2}{L}} &< 2\frac{\lambda}{\hat{\sigma}_*} \end{aligned}$$

□

Unfortunately, since the general case still has an event horizon, the low rank bias is a bias towards the irrecoverable catastrophic weight loss. To be able to find the singular values most likely to cross the event horizon, we can rely on the von Neumann-Wigner's non crossing rule [von Neuman and Wigner 1929], which states that singular values can not cross each other. In other words, the smallest singular value will always remain the smallest, and thus it will be the first singular value to cross the event horizon.

Conjecture 5.3 (von Neumann-Wigner Non-crossing rule). *The distinct singular values of deep linear networks can not cross each other.*

Although it is difficult to prove for deep linear models. The following analysis strongly suggests its existence.

Analysis. We can observe the non-crossing behavior of the deep network by analyzing the dynamics of the singular-vectors.

First let's find a way to solve for eigenvectors of time-varying matrix M

$$\begin{aligned}
Mv &= \lambda v \\
0 &= (M-\lambda I)v \\
0 &= \frac{d}{dt}((M-\lambda I)v) \\
(M-\lambda I)\frac{dv}{dt} &= \left(\frac{d}{dt}(M-\lambda I)\right)v
\end{aligned}$$

Let $M = AA^T$ or $M = A^TA$ depending on whether we are interested in the right singular-vectors or left singular-vectors. Thus the matrix is positive semi-definite symmetric, the eigenvalues can be converted to singular values $\lambda = \sigma^2$ and the eigenvectors are singular-vectors. From the previous proof for Theorem 5.2, we can write $\frac{dv_i}{dt}$ as follows.

$$\frac{dv_i}{dt} = \sum_{j\neq i}\langle\frac{dv_i}{dt}, v_j\rangle v_j$$

To solve for $\frac{dv_i}{dt}$, all we need is to solve for the set of inner-products $\langle\frac{dv_i}{dt}, v_j\rangle$, for all $j \neq i$.

$$\begin{aligned}
(M-\sigma_i^2 I)\frac{dv_i}{dt} &= \left(\frac{d}{dt}(\sigma_i^2 I - M)\right)v_i \\
\sum_{j\neq i}\langle\frac{dv_i}{dt}, v_j\rangle(M-\sigma_i^2 I)v_j &= v_i\frac{d}{dt}(v_i^T M v_i) - \frac{dM}{dt}v_i \\
\sum_{j\neq i}\langle\frac{dv_i}{dt}, v_j\rangle(\sigma_j^2-\sigma_i^2)v_j &= v_i v_i^T\frac{dM}{dt}v_i - \frac{dM}{dt}v_i \\
\langle\frac{dv_i}{dt}, v_j\rangle(\sigma_j^2-\sigma_i^2) &= v_j^T\left(I - v_i v_i^T\right)\frac{dM}{dt}v_i \\
(\sigma_j^2-\sigma_i^2)\langle\frac{dv_i}{dt}, v_j\rangle &= v_j^T\frac{dM}{dt}v_i
\end{aligned}$$

Note the symmetry $\langle\frac{dv_i}{dt}, v_j\rangle = -\langle\frac{dv_j}{dt}, v_i\rangle$. The symmetrical nature of these dynamics enable us to study them in pairs.

Let's begin by analyzing the dynamics of the left singular-vectors. Due to the balancedness assumption, we have that the left singular vectors of A_θ are equivalent to the left singular vectors of W_1. Thus we can substitute $M = W_1 W_1^T$ to find the dynamics.

$$
\begin{aligned}
\frac{dW_1}{dt} &= -\nabla_{W_1} C(P, \theta, \lambda) \\
&= P(A_* - A_\theta) W_{1+}^T - 2\lambda W_1 \\
\frac{d}{dt} W_1 W_1^T &= \frac{dW_1}{dt} W_1^T + W_1 \frac{dW_1^T}{dt} \\
&= P(A_* - A_\theta) A^T + A(A_*^T - A_\theta^T) P - 4\lambda W_1 W_1^T
\end{aligned}
$$

Substituting back into the equation for the left singular-vectors u_i.

$$
\begin{aligned}
(\sigma_j^2 - \sigma_i^2) \langle \frac{du_i}{dt}, u_j \rangle &= u_j^T \frac{dM}{dt} u_i \\
&= u_j^T \left(P(A_* - A_\theta) A^T + A(A_*^T - A_\theta^T) P - 4\lambda W_1 W_1^T \right) u_i \\
&= (\sigma_i^{2L} + \sigma_j^{2L}) u_j^T P u_i - \sigma_i^L u_j^T P A_* v_i - \sigma_j^L v_j^T A_*^T P u_i
\end{aligned}
$$

Notice the singularity as σ_j is close to σ_i. For σ_i and σ_j bounded, then as their gap gets smaller, for $|\sigma_j - \sigma_i| \leq \epsilon$, we find $\langle \frac{du_i}{dt}, u_j \rangle = O(\frac{1}{\epsilon})$. By symmetry, $\langle \frac{du_j}{dt}, u_i \rangle = O(\frac{1}{\epsilon})$ also. This signifies that when the gap gets small between two singular values, their respective singular-vectors rotate approximately in their plane.

Now, let's similarly analyze the right singular-vectors. Due to balancedness, the right singular vectors of A_θ are equivalent to that of W_L. Thus we can substitute $M = W_L^T W_L$.

$$
\begin{aligned}
\frac{dW_L}{dt} &= -\nabla_{W_L} C(P, \theta, \lambda) \\
&= W_{L-}^T P(A_* - A_\theta) - 2\lambda W_L \\
\frac{d}{dt} W_L^T W_L &= A^T P(A_* - A_\theta) + (A_*^T - A_\theta^T) PA - 4\lambda W_L^T W_L
\end{aligned}
$$

Substituting back into the equation for the right singular-vectors v_i.

$$
\begin{aligned}
(\sigma_j^2 - \sigma_i^2)\langle \frac{dv_i}{dt}, v_j \rangle &= v_j^T \frac{dM}{dt} v_i \\
&= v_j^T \left(A^T P(A_* - A_\theta) + (A_*^T - A_\theta^T) PA - 4\lambda W_L^T W_L \right) v_i \\
&= 2\sigma_i^L \sigma_j^L u_j^T P u_i - \sigma_i^L u_j^T P A_* v_i - \sigma_j^L v_j^T A_*^T P u_i
\end{aligned}
$$

Here we observe the same near-crossing acceleration that was noted above. To achieve non-crossing between two singular values that are on course to cross, these rotations along both the left and right singular vectors would need to halt, and/or reverse, the dynamics of σ_i and σ_j.

Let $\sigma_i < \sigma_j$ and $\frac{d\sigma_i}{dt} > \frac{d\sigma_j}{dt}$.

$$
\begin{gathered}
L\sigma_i^{\frac{2(L-1)}{L}} (u_i^T P A_* v_i - u_i P A_\theta v_i) - 2\lambda L \sigma_i > L\sigma_j^{\frac{2(L-1)}{L}} (u_j^T P A_* v_j - u_j P A_\theta v_j) - 2\lambda L \sigma_j \\
u_i^T P(A_* - A_\theta) v_i > u_j^T P(A_* - A_\theta) v_j
\end{gathered}
$$

If we analyze a single gradient step, we get

$$
\begin{cases}
u_i^T P(A_* - A_\theta) v_i \to (u_i + \frac{1}{\epsilon} u_j)^T P(A_* - A_\theta)(v_i + \frac{1}{\epsilon} v_j) & \approx u_j^T P(A_* - A_\theta) v_j, \\
u_j^T P(A_* - A_\theta) v_j \to (u_j + \frac{1}{\epsilon} u_i)^T P(A_* - A_\theta)(v_j + \frac{1}{\epsilon} v_i) & \approx u_i^T P(A_* - A_\theta) v_i.
\end{cases}
$$

Thus, as they get closer, instead of crossing they swap momentum. This behavior is also observed empirically by tracking the singular vectors for two close singular values. □

The consequences of having non-crossing is that the large singular values will always remain the large ones. This creates the idea of dominant singular values, the top k matching the rank of the current task, and the suppressed ones, the ones required to match the rank of the true function, yet are not being actively promoted during training. Nonetheless, to fit the true function, both the dominant and suppressed singular values need to match the distribution of singular values of A_*. The question then becomes, if the suppressed singular values are close to the event horizon, can they be increased? Figure 5.1 shows empirically that if the new task is particularly aligned to a suppressed singular value, the increase it can experience is at least of the same order as itself. This is good news because it can increase exponentially if the new tasks are orthogonal to previous tasks (see more in Chapter 6).

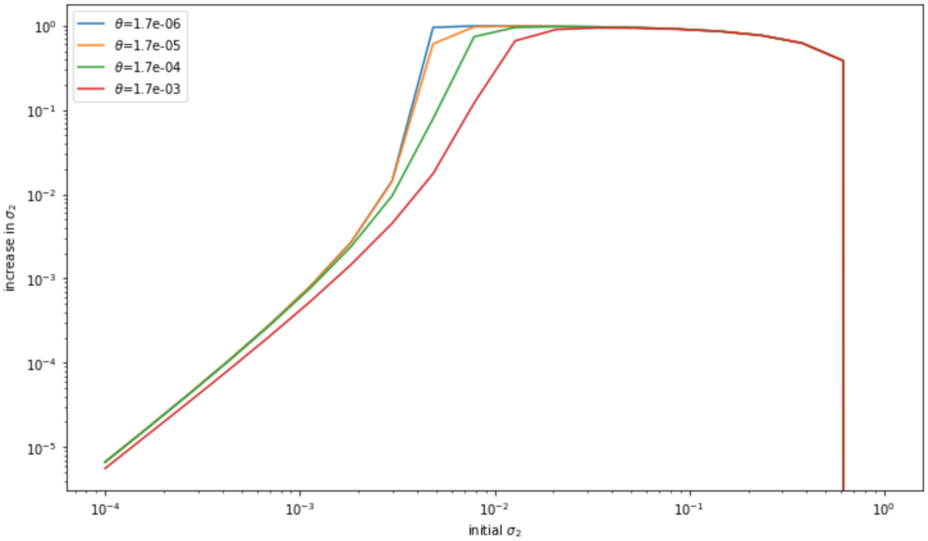


Figure 5.1: The single-episode increase observed of the small singular value which is initially very aligned with the task (for different principal angles)

However there is one caveat to learning a full rank solution.

Theorem 5.4 (Determinant lock)**.** *The determinant of the model can not change sign during training. If the true function has a determinant of the opposite sign then the model can not fit the true* *function.*

Proof. The model A_θ varies smoothly in time, thus so does its determinant. By the midpoint theorem, for the determinant to change sign it must pass zero. The determinant can only be zero if at least one of its eigenvalues is zero. That would place at least one singular value inside the event horizon, thus the model can not find the full rank solution. □

Although Theorem 5.4 finds that the model can not fit the true function for half the space of possible initializations, the model is able to fit the best rank $\text{Rank}(A_*) - 1$ approximation of the true function, since both would have a determinant of zero.

5.2 2-dimensional system

The nature of two sets of singular values, dominant and suppressed, implies that the problem of keeping the rank high is similar to a 2-dimensional problem. Although the behavior is not simplified, lowering the dimensionality of the problem allows for more interpretable empirical results, as we saw in Figure 5.1, and as is explored in more depth in Chapter 6.

In this way, the dynamics of the model is a combination of its extreme regimes, when the new task is perfectly aligned to dominant singular values, and when it is aligned to suppressed singular values. Those cases reduce to the diagonal case which suggest that the more aligned it is to dominant singular values, the less the model will change, and thus the it will not experience forgetting, however that could quickly lead to the suppressed singular values passing the event horizon. In the other case where the new task is nearly orthogonal to the previous one, the suppressed singular values can increase. Between these two extreme regimes, the added freedom to rotate the singular vectors leads the model to do a mix of both. It is this rotation that is the source of Catastrophic Forgetting. The further the suppressed singular values are from their matching true singular value the more rotation the model will experience, and thus the more forgetting there will be.

5.3 Catastrophic weight loss

Due to the existence of the event horizon from Theorem 5.2, it is worth questioning whether it is reasonable for catastrophic weight loss to occur. In the previous chapter, we were able to reduce this question to a gambler's ruin problem due to the orthogonality of the problem. However, the general case is more difficult because the projection is no longer binary, we instead need to consider the angles between P, A_* and A_θ.

Proposition 5.5. *Training high dimensional problems with low batch sizes over exponential training time has a large probability of catastrophic weight loss.*

Proof. For batch size of 1, a projection can be viewed as Pvv^T for some unit vector v. In the naive case where each v_i is sampled i.i.d, you would ideally want to uniformly cover the input space which lies on a d-dimensional sphere $\mathbb{S}^{d-1}$.

The idea is that we have

$$\frac{d\sigma_i}{dt} = L\sigma_i^{\frac{2(L-1)}{L}}(u_i^T P A_* v_i - u_i P A_\theta v_i) - 2\lambda L\sigma_i$$

We can bound the largest singular value, and perhaps the smallest. By sampling $P = vv^T$ uniformly, the dynamics should be really slow.

$$\frac{d\sigma_i}{dt} = \langle u_i, v\rangle L\sigma_i^{2-\frac{1}{L}} v^T A_* v_i - \sigma_i^{3-\frac{1}{L}}\langle u_i, v\rangle^2 - 2\lambda L\sigma_i$$

To find a bound for the probability that $\mathbb{P}(\frac{d\sigma_i}{dt} < 0)$, we can relax the constraint.

$$\begin{aligned}\mathbb{P}(\frac{d\sigma_i}{dt} < 0) &= \mathbb{P}\left(\langle u_i, v\rangle L\sigma_i^{2-\frac{1}{L}} v^T A_* v_i - \sigma_i^{3-\frac{1}{L}}\langle u_i, v\rangle^2 < 2\lambda L\sigma_i\right) \\ &\geq \mathbb{P}\left(\langle u_i, v\rangle \sigma_i^{1-\frac{1}{L}} v^T A_* v_i < 2\lambda\right) \\ &\geq \mathbb{P}\left(|\langle u_i, v\rangle| \sigma_i^{1-\frac{1}{L}} \sigma_{*,1} < 2\lambda\right) \\ &\geq \mathbb{P}\left(|\langle u_i, v\rangle| < \frac{2\lambda}{\sigma_i^{1-\frac{1}{L}}\sigma_{*,1}}\right)\end{aligned}$$

The mass of $|\langle u_i, v\rangle| \sim \frac{1}{\sqrt{d}}$ goes to zero as the number of input dimensions increases. Thus the probability that $|\langle u_i, v\rangle|$ is less than some constant of order $O(\frac{\lambda}{\sigma_{*,1}})$ is non-negligible, especially for high-dimensional problems.

For larger batch sizes, the projection can be decomposed into $P = VV^T$ where V is an orthogonal basis of its image. In this case we can similarly bound the probability that a singular value decreases using principal angles.

$$\begin{aligned}\mathbb{P}(\frac{d\sigma_i}{dt} < 0) &= \mathbb{P}\left(u_i^T V L\sigma_i^{2-\frac{1}{L}} v^T A_* v_i - \sigma_i^{3-\frac{1}{L}} ||Vu_i^T||^2 < 2\lambda L\sigma_i\right) \\ &\geq \mathbb{P}\left(||V^T u_i|| < \frac{2\lambda}{\sigma_i^{1-\frac{1}{L}}\sigma_{*,1}}\right)\end{aligned}$$

By symmetry, the distribution of the principal angle is similar to the distribution of a dimensionally reduced inner product, $||V^T u_i|| \approx \frac{1}{\sqrt{d - \text{Rank}(V)}}$. For a batch size b, we have $\text{Rank}(V) \leq b$, so if the batch size does not grow proportionally with the dimensionality of the problem, the same issue of decreasing singular values persists.

To achieve catastrophic weight loss, these decreases need to occur consecutively. Since batches are usually drawn independently, the probability that these singular values decrease over consecutive steps is exponential.

□

Although Proposition 5.5 is pessimistic, it only applies to lifelong learners that train for an unbounded amount of time. In most other real scenarios, as long as the batch size is not 1, and the training time is not exponential in the order of $\ln\left(\frac{1}{\lambda}\right)$, then catastrophic weight loss is not a reasonable concern.

6 | Experiments

6.1 Synthetic data

The following graphs typically refer to diagnostics, performance and forgetting. These are defined as follows:

- Diagnostics: the singular values of the model over the course of training
- Performance: the loss, a.k.a the unregularized cost of the model over the course of training
- Forgetting: for each task, it is the difference in performance between the current model and that of the model at the end of the episode in which the task was trained.

6.1.1 2-dimensional

One of the conveniences of 2-dimensional matrices is that each singular vector can be characterized by a single angle. In that way, the following figures will plot the dynamics of those angles, named *U angles* and *V angles* respectively, over the entire training period to get a full picture of how these rotations relate to the singular values (plotted under "diagnostics").

6.1.1.1 Single projection

Figure 6.1 reflects a typical episode in which the small singular value is very aligned with the task. We see that the small singular value increases all while its eigenvectors rotate away from the task

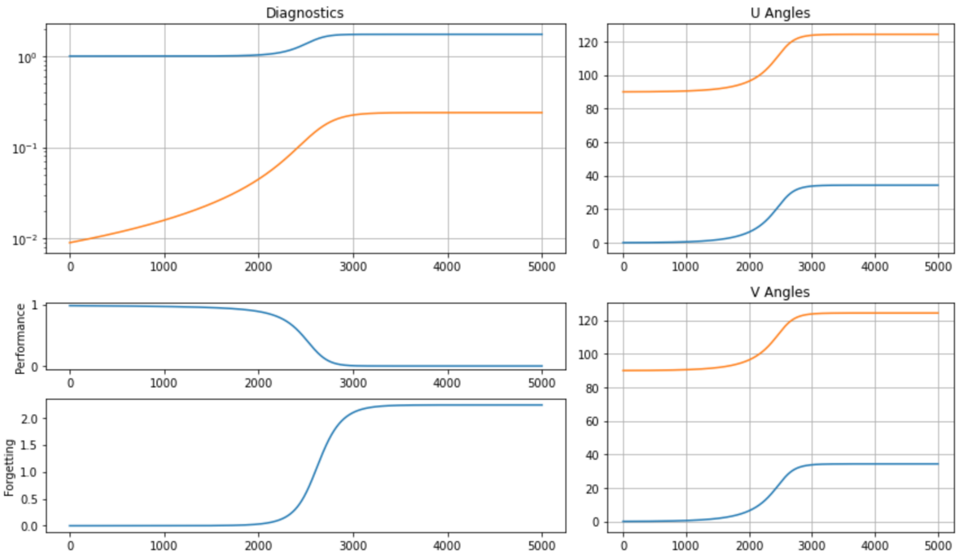


Figure 6.1: Single episode unregularized dynamics

it was aligned so well to. This phenomena is what produces forgetting. The large singular value still reflects most of the mass of the matrix, and thus there is a balancing act between increasing the mass of the small singular value with the correct orientation, and rotating the mass of the large singular value to align with the task. Importantly, neither singular value will perfectly align with the current task, primarily due to the nature of projections. Both the rotation of the large singular value and the increase in the small singular value are positively affecting the projection of the model PA_θ. As that projection approaches the current task PA_* both the rotation and the increase in singular values will cease.

6.1.1.2 Replay

When repeating the same experiment for multiple episodes, Figure 6.2 shows that as the small singular value remains negligible, the large singular value must rotate to each new task, leading to near-perfect forgetting after every episode. However, with each new episode the small singular value continues to increase, to the point that eventually it finds its correct value, which puts an end to the significant rotation from the initial period. You can clearly see that the rotations

(U angles, and V angles), the forgetting, and the peak in performance at the start of every new episode behave in the same way until the exact moment that the small singular value finds its true value.

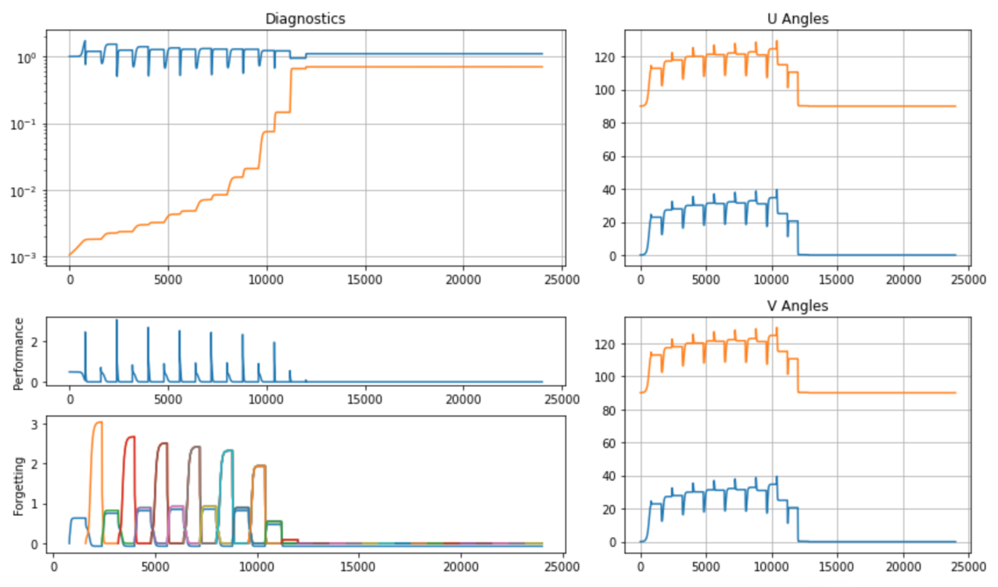


Figure 6.2: Consecutive episode unregularized dynamics with positive determinant

However, in Figure 6.3 we find that the small singular value is decreasing and tending to zero. This demonstrates Theorem 5.4 which explains that the if the determinant is the wrong side, the best approximation for the true function is to fit its best rank $\mathrm{Rank}(A_*)$ solution. We can see from the plots of "U angles" and "V angles" that the determinant can not change sign. Rotating the matrix would flip the sign of both the eigenvalues so the determinant would remain the same sign. Alternatively, for the eigenvalue to change its sign alone, it would need to pass through the origin, which it can not. Since the associated singular value detracts from the performance, all it can do is it make it go to zero, leading to catastrophic weight loss. The remaining singular value is left to fully align with each new task, leading to full forgetting forever.

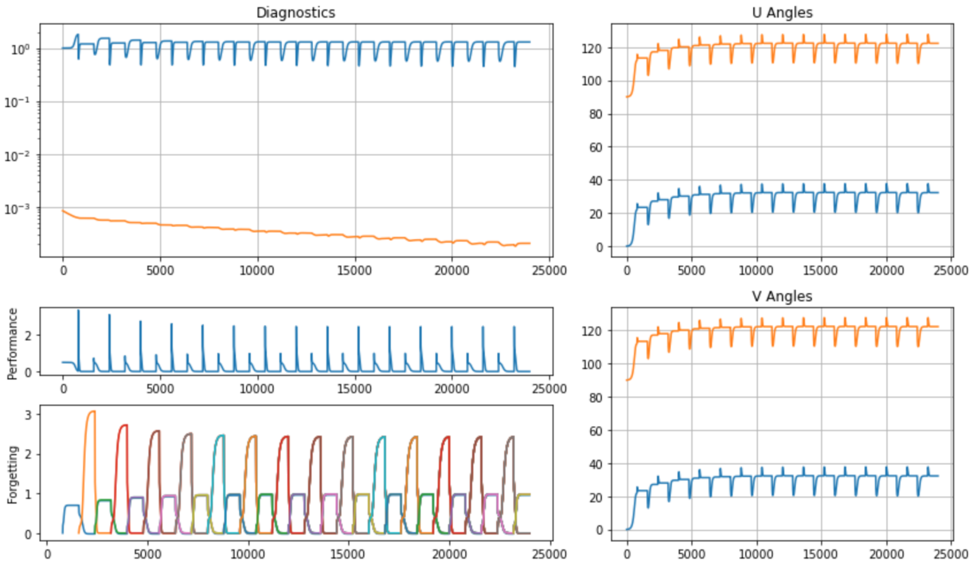


Figure 6.3: Consecutive episode unregularized dynamics with negative determinant

6.1.2 High-dimensional

6.1.2.1 Single projection

In high dimensions, the initial transient rotational dynamics of the model become more visible. To capture the rotation of each singular vector in this high dimensional setting, "U angles" and "V angles" refers to the principal angles between each singular vector and the task. Figure 6.4 reflects the dynamics of a typical episode in high dimensions. The plot for forgetting is specifically measuring the forgetting with respect to the kernel of the episode's task, and we see that even in high dimensions forgetting remains significant.

6.1.2.2 Replay

As in the 2-dimensional case, Figure 6.5 simulates the set of tasks is orthogonal. There are ten tasks which are trained in order, for 4 repetitions. We see that the forgetting typically decreases across the board with every full repetition. Additionally, the plots for forgetting show that Figure 6.5(a) converges towards the true function and the other is not able to, as found in Theorem

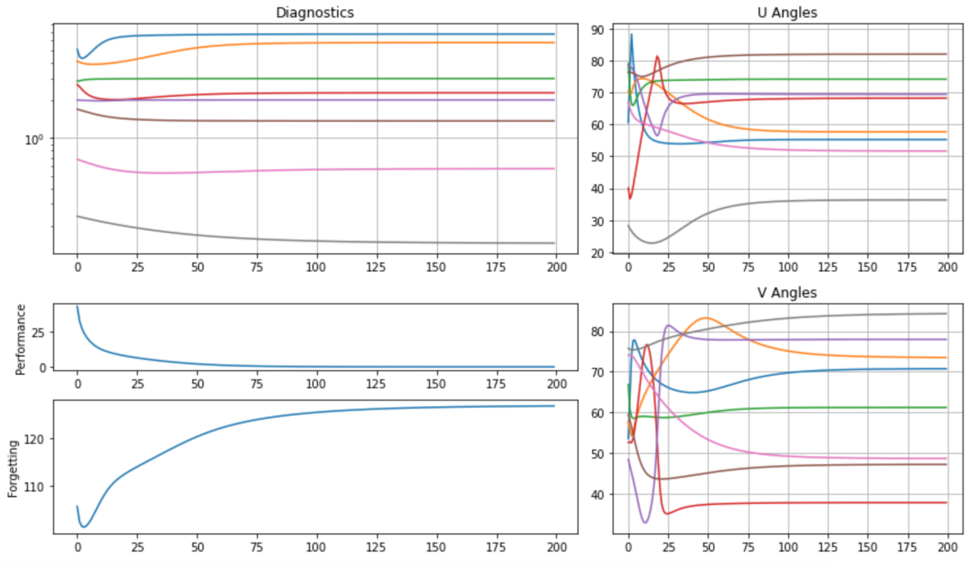
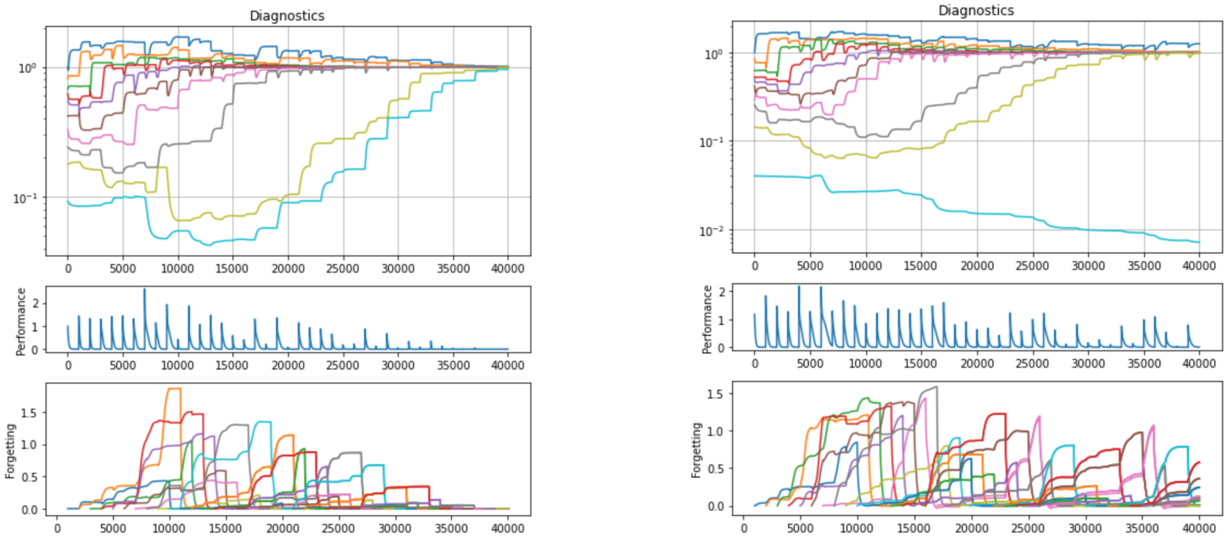


Figure 6.4: Single-episode unregularized dynamics with matching determinant



(a) Matching determinant sign

(b) Mismatching determinant sign

Figure 6.5: High dimensional consecutive episode unregularized dynamics with replay

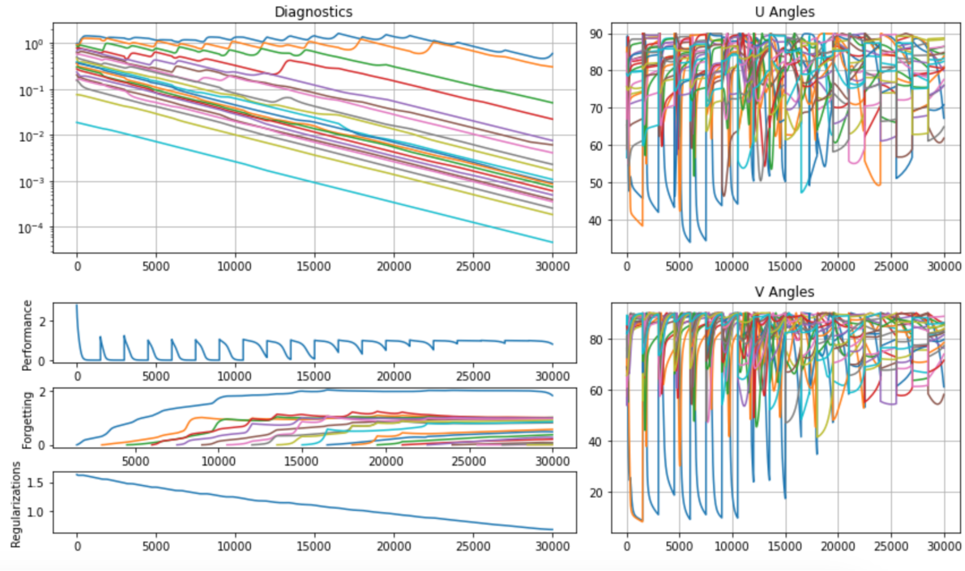
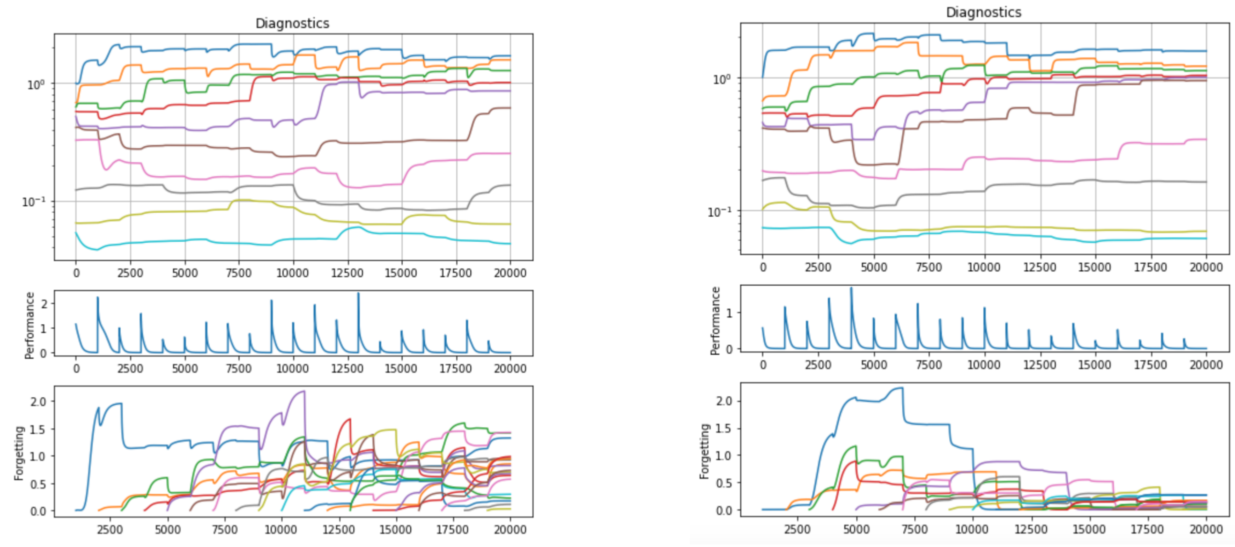


Figure 6.6: Consecutive episode overregularized dynamics

5.4. Figure 6.5(b) does show that forgetting becomes more sparse, this is due to the fact that the model can still realistically fit all but one task perfectly. By the end of the simulation we do see that roughly only one projection at a time is forgotten. We see a similar behavior in the per-episode performance which becomes increasingly sparse with replay.

In Figure 6.6, we see the clear decay that occurs when there is too much weight decay in high dimensions. It behaves worse than we predicted in Figure 4.1(b) with the gambler's ruin problem. The strong weight decay starts by overcoming the smallest singular values. This leads to a more sparse model which increases the probability of catastrophic weight loss, and this continues until the whole model is zeroed out. We can see the matrix tend to zero through the performance which is leveling off at the same order of magnitude as it did for unseen tasks. Since weight decay removes the weights in the kernel of a given episode's task, the spike in forgetting with each new episode roughly equals the loss of the zero matrix, i.e. $||PA_*||^2$.



(a) Unregularized dynamics from 1-pass training

(b) Unregularized dynamics from 2-pass training

Figure 6.7: Comparison between 1-pass and 2-pass training

6.1.2.3 1-pass

In this section we consider sampling projections uniformly at random to compare the use of 1-pass vs 2-pass. Figure 6.7 compares the training dynamics for identical initial conditions. Both 1-pass and 2-pass were trained using 20 projections total, however 1-pass was trained on 20 i.i.d uniform projections whereas 2-pass was trained first on 10 i.i.d uniform projections before replaying each projection once more. Figure 6.7(a) does not show any clear signs of convergence towards the true task. On the other hand, Figure 6.7(b) clearly shows signs of convergence once the replays begin. Overall, even with trading off data for replays, 2-pass vastly outperforms 1-pass for the same training time, calling into question the use of 1-pass at all.

6.2 MNIST

Both Figure 6.8 and 6.9 show that linear models can achieve better than random performance on MNIST. We do see that in the case of initializing with small weights led to a model that converged

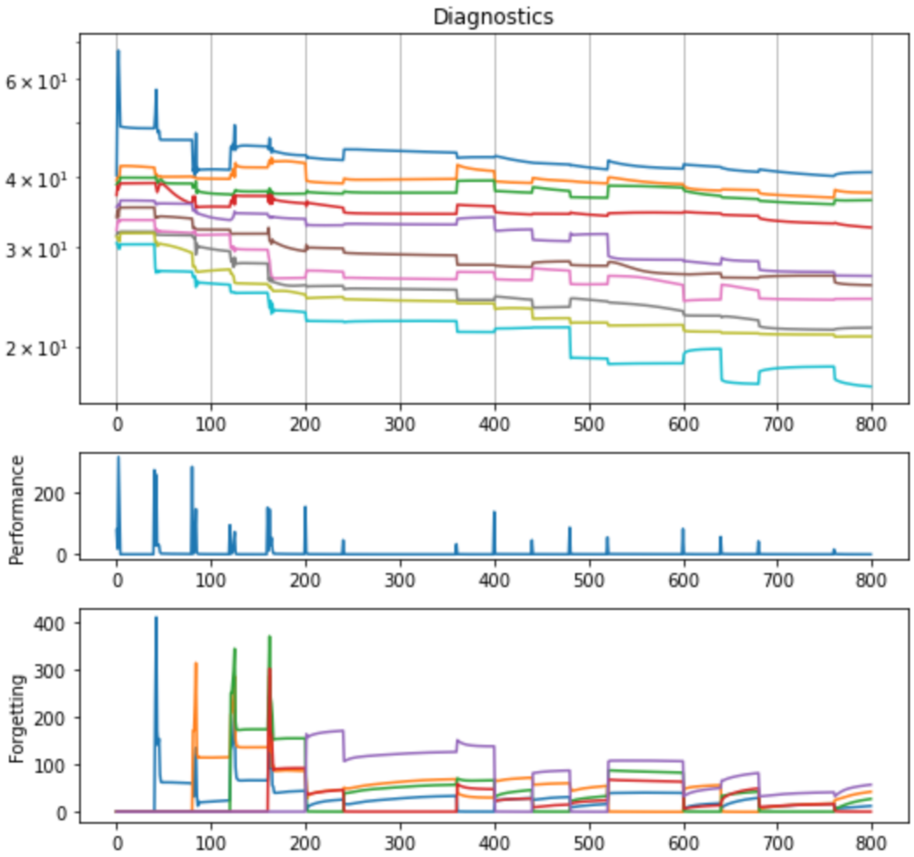


Figure 6.8: Learning MNIST by pairing digits 0-1, 2-3, etc.

faster and performs slightly better, even though the diagnostic trend of 6.9 is towads being less sparse.

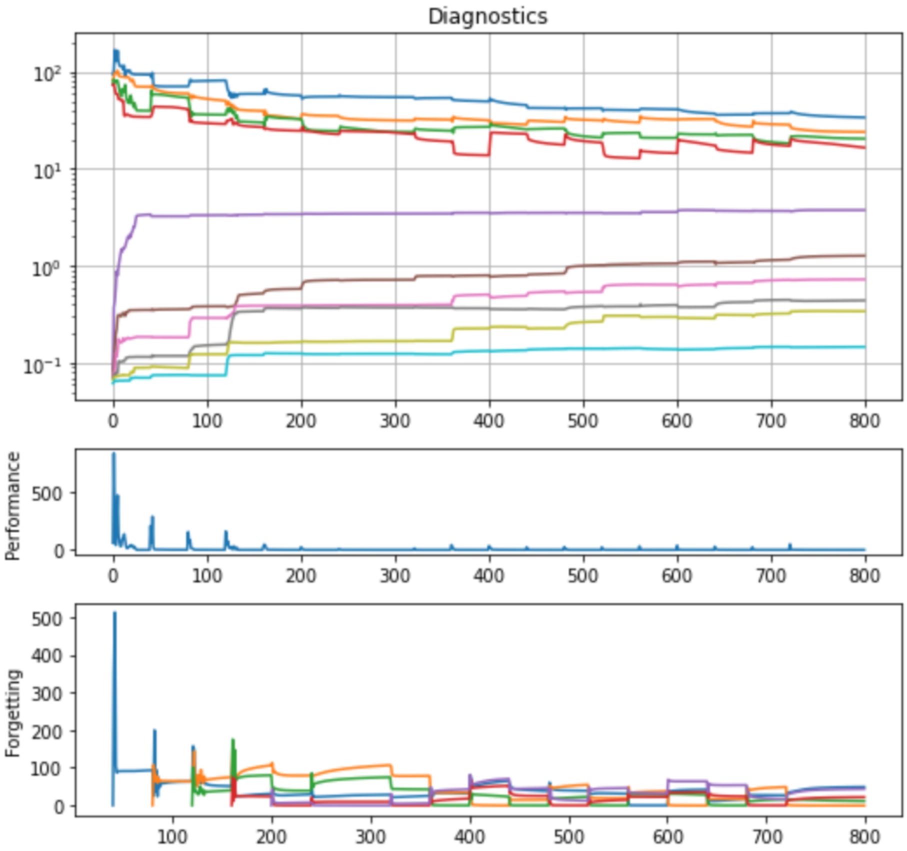


Figure 6.9: Learning MNIST with some small initial singular values (pairing digits 0-1, 2-3, etc.)

7 | Conclusion

The region of stability for safely training neural network gets smaller as they increase in scale. The common attitude to always use weight-decay may be overused as we find that its bias to low-rank representation can often lead to underestimating the rank of the problem. As well, we find that Catastophic Forgetting usually occurs when the singular value decomposition is not distributed similarly to that of the true task. Nevertheless, when training on new tasks, which increases forgetting, the model continues to make progress towards approaching the singular value distribution of true function. Thus Catastrophic Forgetting is not so catastrophic, all it takes to correct is a disproportionately small amount of replay. In addition to reinforcing the importance of replay, our work finds that the use of small batch sizes and 1-pass training could lead to similar catastrophic rank underestimating results as we discussed with weight decay.

A | Appendix: Projections

Projection matrices are a very useful way to characterize a linear subspace. In this Appendix, we will enumerate and prove various properties of orthogonal projection matrices that were used throughout this thesis.

Definition A.1 (Idempotence)**.** By definition, a projection matrix P is any idempotent matrix, i.e. $PP = P$. The projection of some vector x will be its projection onto the $\mathrm{Im}(P)$ along the $\mathrm{Ker}(P)$, i.e. Px.

Property A.2 (Symmetry)**.** A projection is an orthogonal projection matrix if and only if it is symmetric, i.e. $P^T = P$.

Proof. For a projection to be orthogonal, its image and kernel must be orthogonal. In that case, P can be written as the outerproduct of an orthogonal basis for its image, denoted as $P = VV^T$. Thus P is symmetric. □

Property A.3 (Kernel Projection)**.** If P is an orthogonal projection, then $(I - P)$ is an orthogonal projection onto the kernel of P.

Proof. For $(I - P)$ to be an orthogonal projection, it must be idempotent and symmetric. As the difference of two symmetric matrices, it is clearly symmetric. To show idempotence, we can take its square.

$$(I - P)(I - P) = I - P - P + P^2 = I - P$$

To prove that it's image is the kernel of P, we show that any vector x in the kernel of P satisfies $(I - P)x = x$. This is clearly true since Px is defined to be 0. □

Property A.4 (Complimentarity)**.** For an orthogonal projection P, any matrix A can be decomposed into the sum of its projections onto the the image of P and the kernel of P respectively.

Proof. The projection of A on the image of P is PA. By Property A.3, the projection of A on the kernel of P is $(I - P)A$. The sum returns the original matrix A. □

Property A.5 (Principal angle)**.** The principal angle between a vector x and a subspace is defined by the smallest angle between the two. For a subspace characterized by an orthonormal basis V, the principal angle θ can be solved for using $\cos\theta = \frac{||Px||}{||x||}$

Proof. The principal angle between a vector x and basis V is defined as $||u|| \cdot ||VV^T u|| \cos\theta = \langle u, VV^T u\rangle$. Since VV^T is an orthogonal projection, the inner product simplifies to $||VV^T u||^2$. Lastly, since V is normalized, it is the same as $\frac{||V^T u||}{||u||}$ □

Property A.6 (Norm reduction)**.** Projecting a matrix A into a subspace $\text{Im}(P)$ can only reduce its norm, or remain unchanged, i.e., $||PA||^2 \leq ||A||^2$

Proof.

$$\begin{aligned}
||A||^2 &= ||(I + P - P)A||^2 \\
&= ||PA + (I - P)A||^2 \\
&= \text{Tr}(A^T PA) + \text{Tr}(A^T(I - P)A) \\
&= ||PA||^2 + ||(I - P)A||^2 \\
&\geq ||PA||^2
\end{aligned}$$

□

Property A.7 (Singular value reduction)**.** Prove that $|u^T PAv| \leq \sigma_1$

Proof. Let $A = U\Sigma V^T$ be the singular value decomposition of $A \in \mathbb{R}^{d\times d}$.

$$\begin{aligned}
|u^T PAv| &= |u^T PU\Sigma V^T v| \\
&= |\langle \sqrt{\Sigma} U^T Pu, \sqrt{\Sigma} V^T v\rangle| \\
&\leq ||\sqrt{\Sigma} U^T Pu|| \cdot ||\sqrt{\Sigma} V^T v|| && \text{(Cauchy-Schwartz)} \\
&= \sqrt{\sum_{i=1}^{d} \sigma_i \langle Pu, u_i\rangle^2} \cdot \sqrt{\sum_{i=1}^{d} \sigma_i \langle v, v_i\rangle^2}
\end{aligned}$$

If we normalize each term by $||U^T Pu||$ and $||V^T v||$ respectively, then the sums above can be considered expectations, which are bounded by the largest element. In this case, the largest element is first singular value σ_1. Note that since U and V are orthonormal matrices, we have $||U^T Pu|| = ||Pu|| \leq ||u|| = 1$ and $||V^T v|| = ||v|| = 1$.

$$u^T PAv \leq \sigma_1 ||Pu|| \leq \sigma_1$$

□

Property A.8 (Commuting projections)**.** Two orthogonal projection matrices P_1 and P_2 will commute if and only if they have the same eigenvectors.

Proof. By definition, an orthogonal projection is symmetric and idempotent so it can be decomposed using the singular value decomposition into $P = Q\Sigma Q^T$, where Σ has all singular values either 0 or 1. We can alternatively write this as

$$P = \begin{bmatrix} V_{\text{Im}} & V_{\text{Ker}} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_{\text{Im}} \\ V_{\text{Ker}} \end{bmatrix},$$

where V_{Im} is a basis for the for the range of P and V_{Ker} is an orthogonal basis for the kernel of

P. Since the eigenvalues of P are either 0 or 1, we have that x is an eigenvector of P if either $x \in \mathrm{Im}(P)$ or $x \in \mathrm{Ker}(P)$. Remark that $\mathrm{Im}(P)$ is orthogonal to $\mathrm{Ker}(P)$.

We have that $Px \in \mathrm{Im}(P)$, so it can be decomposed into a linear combination of its eigenvectors that have a eigenvalue of 1. Thus when multiplying two projections with the same eigenvalues, the resulting P_1P_2x will be a linear combination of the eigenvectors that have a singular value of 1 for both P_1 and P_2.

In this way, the product $P_1P_2 = P_3$ also has eigenvalues $\lambda \in \{0, 1\}$. The eigenvectors with an eigenvalue of 1 are $\{v : v \in (\mathrm{Im}(P_1) \cap \mathrm{Im}(P_2)\}$. The eigenvectors with an eigenvalue of 0 are $\{v : v \in (\mathrm{Ker}(P_1) \cup \mathrm{Ker}(P_2)\}$. Thus the product is also an orthogonal projection. We then have

$$P_1P_2 = P_3 = P_3^T = P_2P_1$$

□

Bibliography

Abernethy, J., Bartlett, P. L., and Hazan, E. (2011). Blackwell approachability and no-regret learning are equivalent. In Kakade, S. M. and von Luxburg, U., editors, *Proceedings of the 24th Annual Conference on Learning Theory*, volume 19 of *Proceedings of Machine Learning Research*, pages 27–46, Budapest, Hungary. PMLR.

Caruana, R. (1993). Multitask learning: A knowledge-based source of inductive bias. In *International Conference on Machine Learning*.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159.

Jacot, A., Ged, F., Şimşek, B., Hongler, C., and Gabriel, F. (2022). Saddle-to-saddle dynamics in deep linear networks: Small initialization training, symmetry, and sparsity.

Ji, Z. and Telgarsky, M. (2019). Gradient descent aligns the layers of deep linear networks.

Kawaguchi, K. (2016). Deep learning without poor local minima.

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

Li, Z., Luo, Y., and Lyu, K. (2021). Towards resolving the implicit bias of gradient descent for matrix factorization: Greedy low-rank learning.

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

Pratt, L. Y. (1993). Transferring previously learned back-propagation neural networks to new learning tasks.

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

Robins, A. (1995). Catastrophic forgetting, rehearsal and pseudorehearsal. In *Connection Science, Vol. 7, No 2*, pages 123–146.

Shimkin, N. (2016). An online convex optimization approach to blackwell's approachability. *Journal of Machine Learning Research*, 17(129):1–23.

Suddarth, S. C. and Holden, A. D. (1991). Symbolic-neural systems and the use of hints for developing complex systems. *International Journal of Man-Machine Studies*, 35(3):291–311.

Tieleman, T. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models.

von Neuman, J. and Wigner, E. (1929). Uber merkwürdige diskrete Eigenwerte. Uber das Verhalten von Eigenwerten bei adiabatischen Prozessen. *Physikalische Zeitschrift*, 30:467–470.

Wang, Z. and Jacot, A. (2023). Implicit bias of sgd in l_2-regularized linear dnns: One-way jumps from high to low rank.

Zenke, F., Poole, B., and Ganguli, S. (2017). Continual learning through synaptic intelligence.

Zhang, X., Dou, D., and Wu, J. (2022). Feature forgetting in continual representation learning.

Zhang, X., Li, X., Dou, D., and Wu, J. (2020). Measuring information transfer in neural networks.