CONTINUAL LEARNING WITH SKETCHED STRUCTURAL REGULARIZATION

by

Haoran Li

A thesis submitted to Johns Hopkins University

in conformity with the requirements for the degree of

Master of Science in Engineering

Baltimore, Maryland

May, 2021

Abstract

Preventing catastrophic forgetting while continually learning new tasks is an essential problem in continual learning. Structural regularization (SR) refers to a family of algorithms that mitigate catastrophic forgetting by penalizing the network for changing its “critical parameters" from previous tasks while learning a new one. The penalty is often induced via a quadratic regularizer defined by an *importance matrix*, e.g., the (empirical) Fisher information matrix in the Elastic Weight Consolidation framework. In practice and due to computational constraints, most SR methods crudely approximate the importance matrix by its diagonal. In this paper, we propose *Sketched Structural Regularization* (Sketched SR) as an alternative approach to compress the importance matrices used for regularizing in SR methods. Specifically, we apply *linear sketching methods* to better approximate the importance matrices in SR algorithms. We show that sketched SR: (i) is computationally efficient and straightforward to implement, (ii) provides an approximation error that is justified in theory, and (iii) is method oblivious by construction and can be adapted to any method that belongs to the structural regularization class. We show that our proposed approach consistently improves various SR algorithms' performance on both synthetic experiments and benchmark continual learning tasks, including permuted-MNIST and CIFAR-100.

Acknowledgments

Thanks to Prof. Vladimir Braverman for kindly giving me the opportunity to conduct this project, for directing me into this field, and also for the instructions throughout the research process.

Thanks to Aditya Krishnan, Jingfeng Wu, Soheil Kolouri and Praveen K. Pilly for their enormous help, valuable instruction and fruitful discussion in pursuing the experimental and theoretical aspects of this project.

And as always, thanks to all my friends and all those who have given me help and compassion - without you I might have not been able to keep the perseverance throughout conducting and finishing this project.

This report is of the same project and in conformity with the paper preprint from the same author (Li et al., 2021). The author has the full right to use the preprint content in this report.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

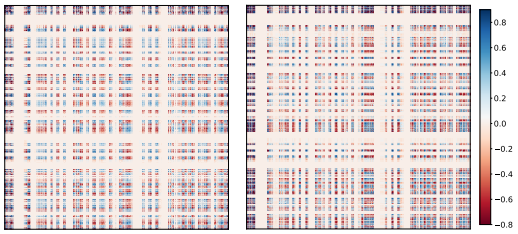
Continual learning, also referred to as *lifelong learning* or *incremental learning*, is the ability to continuously learn in a varying environment through integrating the newly acquired knowledge while preserving the previously learned experiences (Parisi et al., 2019). A key issue that prevents the state-of-the-art machine learning models (e.g. deep neural networks) from achieving continual learning is *catastrophic forgetting*, i.e. learning a new task may severely modify the model parameters, including those that are critical to the previous tasks (Parisi et al., 2019).

Structural regularization (SR), or *selective synaptic plasticity*, is a general and widely-adopted framework to alleviate catastrophic forgetting in continual learning (Kolouri et al., 2020; Aljundi et al., 2018; Kirkpatrick et al., 2017; Chaudhry et al., 2018; Zenke, Poole, and Ganguli, 2017). From a geometric perspective (Kolouri et al., 2020; Chaudhry et al., 2018), SR methods construct an (positive semi-definite) *importance matrix* (IM) that measures the relative importance of the model parameters to the old tasks (which are aimed be preserved in continual learning), and add a quadratic regularizer defined

by the importance matrix when training on new tasks. The intuition behind structural regularization is clear: the quadratic regularizer adaptively penalizes parameters from changing according to their criticality measured by the importance matrix. As a result, structural regularization encourages the model to learn the new task using non-important parameters, so that it is able to maintain the important information from old tasks. For example, Kirkpatrick et al. (2017) choose the (diagonal) *empirical Fisher information matrix*[1] (empirical Fisher, EF) as the importance matrix in their seminal algorithm, *Elastic Weight Consolidation* (EWC) (Kirkpatrick et al., 2017; Kolouri et al., 2020; Chaudhry et al., 2018). However, a full IM (e.g. empirical Fisher) scales as $\mathcal{O}(m^2)$ for a model with m parameters and can be prohibitively big to use for modern neural networks. Often in practice, the diagonal, which scales as $\mathcal{O}(m)$, is used as a crude approximation to the full IM (Kirkpatrick et al., 2017; Kolouri et al., 2020; Aljundi et al., 2018). We refer to structural regularization with a diagonal-approximated importance matrix as *diagonal SR*.

While developing new and effective importance matrices has been a hot direction for structural regularization (Kolouri et al., 2020; Aljundi et al., 2018; Kirkpatrick et al., 2017; Chaudhry et al., 2018; Zenke, Poole, and Ganguli, 2017), little effort has been spent on examining the effectiveness of the crude diagonal approximation (a few exceptions, e.g. (Liu et al., 2018; Ritter, Botev, and Barber, 2018), are discussed later in Chapter 2). Intuitively speaking, a

[1] In their original paper (Kirkpatrick et al., 2017) (and follow-up papers, e.g., (Kolouri et al., 2020)), the importance matrix in EWC is termed as the "Fisher information matrix", but precisely, it should be called the "empirical Fisher" — the two terms are often interchangeably used in the community, though they are not identical. See (Kunstner, Balles, and Hennig, 2019) for a detailed clarification.



(a) Full EF (b) Sketched EF

Frobenius-Norm Approx. Error $\|\widetilde{\Omega} - \Omega\|_F^2 / \|\Omega\|_F^2$	
Diagonal	Block-Diagonal
$94.7\% \pm 0.6\%$	$82.8\% \pm 1.3\%$
Sketched	Low-Rank
$\mathbf{8.1}\% \pm \mathbf{10.2}\%$	$0.0\% \pm 0.0\%$

Figure 1.1: Illustration of the (sketched) empirical Fisher on a synthetic 2D binary classification task, and the approximation error of each methods to the full empirical Fisher.

diagonal IM assumes independence between parameters, which is far from reality (Liu et al., 2018; Ritter, Botev, and Barber, 2018). In mathematics, a positive semi-definite matrix can rarely be well-approximated by its diagonal — the only non-trivial exception to our knowledge is when the matrix is diagonally dominant (Horn and Johnson, 2012). Unfortunately, for the importance matrices considered in SR methods this is not likely to be the case, especially when training using neural networks. As an illustration, we examine the empirical Fisher as the importance matrix (Kirkpatrick et al., 2017) of a synthetic experiment adopted from Pan et al. (2020); the full empirical Fisher is shown in Figure 1.1(a). The plot shows that the full empirical Fisher is far from diagonal; in fact the diagonal only contributes to less than 5.3% of the Frobenius norm of the empirical Fisher matrix (see table in Figure 1.1). Hence, approximating importance matrix with its diagonal might be problematic. A natural question then is:

Is there a computational and memory efficient method to approximate the importance matrix without losing critical information in the matrix?

In this report, we answer the above question by providing a *linear sketching method* (Charikar, Chen, and Farach-Colton, 2002) as a *provable*, *ubiquitous*, *efficient* and *effective* approach to approximate the importance matrix in SR methods. Specifically, in one pass of the data (which is also required for diagonal approximation), a $\mathcal{O}(tm)$ size sketched matrix can be produced that approximately recovers the quadratic regularizer defined by the $\mathcal{O}(m^2)$ size importance matrix. Here, $t \ll m$ is a tuneable hyperparameter that balances the computation cost and matrix size with the quality of approximation, and can be chosen as a small number in practice. Our method, called *sketched SR*, has the following notable advantages:

1. Has a *theoretically guaranteed* small approximation error, providing that the importance matrix has a well-behaved spectrum, e.g. has low effective rank. Fortunately, for deep neural network and commonly used SR methods, the importance matrix (e.g. empirical Fisher) does indeed have low (effective) rank (Sagun et al., 2017; Chaudhari and Soatto, 2018), but is not diagonal (see Figure 1.1).

2. Is *algorithm oblivious* by construction, i.e. for any algorithm that belongs to the structural regularization paradigm (defined in Section 3.1), a sketched version can be readily established without additional, algorithm specific considerations.

3. Is *computationally efficient* and *easy to implement*. Both sketched SR and diagonal SR make only one pass of the data (of the old task) to obtain the approximation. Though sketched SR saves $\mathcal{O}(tm)$ parameters, which is slightly larger than the $\mathcal{O}(m)$ parameters in diagonal SR. This additional

cost is easily affordable as setting $t \leq 50$ is sufficient for sketched SR to outperform diagonal SR in our experiments.

4. *Consistently outperforms* its diagonal counterpart on overcoming catastrophic forgetting, in both synthetic experiments and benchmark continual-learning tasks, including permuted-MNIST and CIFAR-100.

The remaining part of this report is organized as follows: the related literature is reviewed in Chapter 2; in Chapter 3, we formally introduce our sketched structural regularization, and also present its practical implementation and theoretical properties; then in Chapter 4, we experimentally compare our methods with the diagonal counterparts, which verifies the effectiveness of our methods; finally, we draw the conclusion of this report in Chapter 5.

Chapter 2

Related Work

In this chapter, we present a review of literature related to our sketched structural regularization algorithm.

Functional Regularization. Apart from structural regularization, another widely-used category of approaches to overcome catastrophic forgetting is *functional regularization* (Jung et al., 2016; Li and Hoiem, 2017; Rannen et al., 2017; Shin et al., 2017; Hu et al., 2018; Rozantsev, Salzmann, and Fua, 2018; Wu et al., 2018; Li et al., 2019). Similar to structural regularization, functional regularization also adds a regularizer (when training new tasks) as penalty to mitigate the forgetting of useful old knowledge; however, functional regularization may use very general (hence, functional) regularizers, in addition to quadratic ones. For example, Jung et al. (2016) and Li and Hoiem (2017) snapshot a teacher model that learned from old tasks, and use it to regularize a student model that fits new tasks. Moreover, generative models are applied to generate pseudo-data (*memory*) of old tasks, and the pseudo-data is mixed to the new data distribution as a regularization (*replay*) for learning new tasks (Rannen et al., 2017; Rostami, Kolouri, and Pilly, 2019; Shin et al., 2017; Wu

et al., 2018; Hu et al., 2018). This is also known as *memory replay*. Finally, we remark that functional regularization can be used together with structural regularization (Shin et al., 2017; Rozantsev, Salzmann, and Fua, 2018). Our focus of this report is to use linear sketching methods to improve SR methods; an interesting future work is to apply similar ideas (e.g., coresets (Feldman and Langberg, 2011; Har-Peled and Mazumdar, 2004)) to improve functional regularization methods, especially for those based on memory replay.

Non-Diagonal Importance Matrix. Diagonal approximation is a crude, but de facto approach to compress the full IM in most existing SR algorithms (Kolouri et al., 2020; Aljundi et al., 2018; Kirkpatrick et al., 2017; Chaudhry et al., 2018; Zenke, Poole, and Ganguli, 2017). Before this work, there exists a few studies that investigate structural regularization with non-diagonal IM (Liu et al., 2018; Ritter, Botev, and Barber, 2018), which we discuss in sequence. Ritter, Botev, and Barber (2018) adopt the layer-wise block-diagonal approximation as a better replacement to the commonly used diagonal version for the importance matrix: even so, the cross-layer weight dependence is being ignored; moreover, block-diagonal empirical Fisher is not a good approximation to empirical Fisher matrix either (see Figure 1.1). Liu et al. (2018) propose layer-wise rotation of the empirical Fisher such that the new matrix can be more diagonal-alike; this procedure not only assumes cross-layer independence (of weights), but even further assumes the independence between layer inputs and layer gradients (see Eq. (7) in (Liu et al., 2018)). In comparison, the sketching methods adopted in this report only require a very weak assumption, i.e., the importance matrix has a low effective rank.

Linear Sketching. Linear sketching is a widely studied technique for dimensionality reduction. We rely on the popular sketching method *CountSketch* (Charikar, Chen, and Farach-Colton, 2002) that has its roots in the Johnson-Lindenstrauss transform. Randomized linear sketching methods, such as CountSketch, draw a random matrix $S \in \mathbb{R}^{t \times m}$ and embed the columns of the input matrix $W \in \mathbb{R}^{n \times m}$ into a smaller dimension $t \ll n$ by outputting SW. By carefully constructing the random distribution, it can be shown that the sketch SW *preserves the norms of the vectors in the subspace spanned by the columns of* W up to some error. Such sketching techniques are known as oblivious subspace embeddings (OSEs). This property of OSEs makes them a natural tool for approximating the quadratic regularizer in SR methods.

Sparse OSE methods (Nelson and Nguyên, 2013; Cohen, 2016) such as CountSketch have a two-wise advantage: i) they're *oblivious*, which means that the random distribution is defined independent of the input matrix W and ii) the sketch SW can be computed in time that is linear in the input size (e.g. proportional to the number of non-zero entries in W). These methods have been widely used, giving fast algorithms for various problems such as low-rank approximation, linear regression (Sarlos, 2006; Clarkson and Woodruff, 2017; Meng and Mahoney, 2013), k-means clustering (Cohen et al., 2015), leverage score estimation (Drineas et al., 2012) and numerous other problems (Lee, Song, and Zhang, 2019; Ahle et al., 2020; Brand et al., 2021).

Chapter 3

Methodology

In this chapter, we formally introduce the details and theoretical properties of our sketched structural regularization algorithm.

3.1 Preliminaries

We use $(x, y) \in \mathbb{R}^s \times \mathbb{R}^k$ to denote a feature-label pair, and $\theta \in \mathbb{R}^m$ to denote the model parameter. A parametric model is denoted by $\phi(\cdot\,; \theta) : \mathbb{R}^s \to \mathbb{R}^k$. Given a distance measure of two distributions, $d(\cdot, \cdot)$, the individual loss over data point (x, y) can be formulated as

$$\ell(x, y; \theta) := d(\phi(x; \theta), y).$$

For example, in deep neural networks, $\phi(\cdot\,; \theta)$ is the network output, and $d(\cdot, \cdot)$ is usually chosen to be the cross entropy loss (Goodfellow et al., 2016).

3.1.1 Structural Regularization

Let task A with data distribution $(x, y) \sim \mathcal{D}_A$ be an already well-learned task on network ϕ with learnt parameters θ_A^*. In order to overcome catastrophic forgetting when learning a new task B, with data distribution $(x, y) \sim \mathcal{D}_B$,

structural regularization algorithms apply an extra regularizer $\mathcal{R}(\theta)$ to the main loss and optimize the following total loss:

$$\arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}_B} \left[\ell(x, y; \theta)\right] + \lambda \cdot \mathcal{R}(\theta).$$

Here, the expectation should be understood as the empirical expectation over the training set. As for the regularization term, λ is a hyper-parameter, and $\mathcal{R}(\theta)$ is a quadratic regularizer that penalizes the weight for being deviated from θ^*_A, the learnt weight from the previous task A:

$$\mathcal{R}(\theta) := \frac{1}{2}(\theta - \theta^*_A)^\top \Omega (\theta - \theta^*_A),$$

where $\Omega \in \mathbb{R}^{m\times m}$ is an importance matrix and is positive semi-definite (PSD). As we will see shortly, the PSD matrix Ω usually has a natural decomposition as (Kirkpatrick et al., 2017; Aljundi et al., 2018):

$$\Omega = \frac{1}{n} W^\top W, \tag{3.1}$$

where each row of $W \in \mathbb{R}^{n\times m}$ is a Jacobian matrix of a certain individual loss (which might not be the one used for the main loss) of data x from task A, and n is the number of training data in task A. Then, the structural regularizer $\mathcal{R}(\theta)$ can be written as

$$\mathcal{R}(\theta) = \frac{1}{2n} \|W \cdot (\theta - \theta^*_A)\|_2^2, \quad W \in \mathbb{R}^{n\times m}. \tag{3.2}$$

Two Examples. Table 3.1 summarizes two examples for the importance matrices in: *Elastic Weight Consolidation* (EWC) (Kirkpatrick et al., 2017) and *Memory Aware Synapses* (MAS) (Aljundi et al., 2018). It is worth noting that the importance matrix used in EWC is the *empirical Fisher* evaluated at the optimal weight for task A.

Table 3.1: The construction and factorization of the importance matrices in EWC and MAS.

	Matrix Ω	Row-Vector $(W)_x$
EWC	$\mathbb{E}_{(x,y)\sim\mathcal{D}_A} \nabla_\theta \ell(x, y; \theta_A^*) \cdot \nabla_\theta \ell(x, y; \theta_A^*)^\top$	$\nabla_\theta \ell(x, y; \theta_A^*)$
MAS	$\mathbb{E}_{x\sim\mathcal{D}_A} \left(\nabla_\theta \left\|\phi(x; \theta_A^*)\right\|_2^2\right) \cdot \left(\nabla_\theta \left\|\phi(x; \theta_A^*)\right\|_2^2\right)^\top$	$\nabla_\theta \left\|\phi(x; \theta_A^*)\right\|_2^2$

3.1.2 Diagonal Approximation

Unfortunately, both matrices Ω and W have m^2 and mn entries respectively, which makes them prohibitively large to compute and store for big models like deep neural networks. As a compromise, practitioners often take the diagonal of Ω as an approximation. This leads to the presented version of EWC (Kirkpatrick et al., 2017) and MAS (Aljundi et al., 2018) in their original paper. These are called *diagonal EWC* and *diagonal MAS* respectively in this report to be distinguishing with our variants. However, as we have discussed and demonstrated in Figure 1.1, such a treatment ignores the dependence between weights and exacerbates performance degeneration for overcoming catastrophic forgetting. In the following we present our sketched version of the above algorithms, which can make use of the off-diagonal entries of Ω to improve the diagonal approximated version.

3.2 Sketched Structural Regularization

In this section we propose our framework of sketching the regularizer from (3.2) and describe the specific sketch construction along with some theoretical guarantees. We describe our construction in terms of the general framework of structural regularization for continual learning from Section 3.1. Then we

contrast our approximation method with other compression methods like PCA. Finally we describe how we go from the two-task settings to an online version of the algorithm in a way that is standard in works on structural regularization (Kolouri et al., 2020; Chaudhry et al., 2018; Schwarz et al., 2018).

3.2.1 Algorithm

We propose a method to sketch the matrix Ω from (3.1) by reducing the dimensionality of each of the matrix W from n dimensions to t dimensions for a $t \ll \min\{n, m\}$. Specifically, we draw a random matrix $S \in \mathbb{R}^{t \times n}$ from a carefully chosen distribution and approximate the regularizer (3.2) in SR methods with

$$\widetilde{\mathcal{R}}(\theta) = \frac{1}{2n} \|\widetilde{W} \cdot (\theta - \theta_A^*)\|_2^2, \quad \widetilde{W} := SW \in \mathbb{R}^{t \times m}. \tag{3.3}$$

We use *CountSketch* (Charikar, Chen, and Farach-Colton, 2002) to construct the sketched matrix $\widetilde{W} = SW$, which is formally presented in Algorithm 1. CountSketch reduces the number of rows (aka, the dimension of the columns) of W by the following: first the rows of W are randomly partitioned into t groups (Algorithm 1, line 5), then rows in each group are randomly, linearly combined (with random signs as weights) into a single new row (Algorithm 1, line 7).

Two remarks are in order for the practical implementation of Algorithm 1: (i) note that Algorithm 1 only makes one pass of the data from task A, which is as required for computing diagonal approximation; (ii) note that Algorithm 1 requires $\mathcal{O}(t)$ times auto-differentiation, but since t is small and the sketch construction only needs to done once per new task, the cost is affordable in

Algorithm 1 Sketch Construction in Sketched SR

Input: Data from task A and optimized neural network $\phi(\cdot\,; \theta_A^*)$ for task A

Parameters: Size of sketch $t \in \mathbb{N}^+$

Initialize 2-wise and 4-wise independent hash functions $h : [n] \to [t]$ and $\sigma : [n] \to \{-1, 1\}$ respectively

for $k = 1, \ldots, t$ **do**

  Group data $G_k := \{x \in A : h(x) = k\}$

  Compute $\sum_{x \in G_k} \sigma(x)(W)_x$ as per Table 3.1 by auto-differentiation

  Set $(\widetilde{W})_k \leftarrow \sum_{x \in G_k} \sigma(x)(W)_x$

end for

return $\widetilde{W} \in \mathbb{R}^{t \times m}$

practice (see more in Chapter 4).

Comparison with Low-Rank Approximation Methods. The main advantage of using CountSketch over more complicated low-rank approximation methods (e.g. PCA) to compress the importance matrix in SR methods, is that it can be computed with only a small amount of additional computation and only a modest blow-up in memory compared to the diagonal approximation. However PCA is usually computational intractable for big models such as deep neural networks. Moreover, in below, we show CountSketch achieves provable small approximation error (for matrix with low stable-rank), as can be guaranteed by PCA.

3.2.2 Theoretical Properties

The following theorem from Cohen, Nelson, and Woodruff (2016) builds on several results on CountSketch matrices, giving theoretical guarantees for sketching quadratic forms of matrices.

Theorem 1 (Theorem 6, Cohen, Nelson, and Woodruff (2016))**.** *Let* $W \in \mathbb{R}^{n \times m}$

be a matrix, $k \in \mathbb{N}^+$ *be a parameter and let* $\epsilon, \delta > 0$ *be constants. There exists a constant* $C > 0$ *such that a CountSketch matrix* $S \in \mathbb{R}^{t \times n}$ *with* $t = \frac{Ck^2}{\epsilon^2 \delta}$ *has the property that for all* $\theta \in \mathbb{R}^m$*,*

$$\left| \|SW\theta\|_2^2 - \|W\theta\|_2^2 \right| \leq \epsilon \|\theta\|_2^2 \left(\|W\|_2^2 + \frac{\|W\|_F^2}{k} \right) \tag{3.4}$$

with probability at least $1 - \delta$ *and where the probability is taken over the randomness of the CountSketch matrix* S*.*

This theorem is re-phrased for our purposes as in Corollary 1.1, showing the quality of approximation by the sketch in preserving ℓ_2-norms of vectors in the subspace spanned by the columns of W, the matrix that is being sketched. There is a trade-off in the quality of approximation by the sketch and its size, given by the dimension of the columns t. In particular, the error in preserving the ℓ_2-norm of any $W\theta$ depends on the spectrum of W; when $t \geq \|W\|_F^4/(\epsilon^2 \|W\|_2^4)$ the error is *additive* and scales with $\epsilon \|W\|_2^2 \|\theta\|_2^2$, which we detail in the following theorem.

Corollary 1.1. *For a matrix* $W \in \mathbb{R}^{n \times m}$ *with stable rank*[1] r*, a CountSketch matrix* $S \in \mathbb{R}^{t \times n}$ *with* $t = \mathcal{O}(r^2/\epsilon^2)$ *has the property that with probability at least* 0.99*,*

$$\left| \|SW\theta\|_2^2 - \|W\theta\|_2^2 \right| \leq \epsilon \cdot \|W\|_2^2 \cdot \|\theta\|_2^2$$

for all vectors $\theta \in \mathbb{R}^m$*.*

Corollary 1.1 directly follows Theorem 1 by noting that when $t \geq \|W\|_F^4/(\epsilon^2 \|W\|_2^4)$, the error scales with $\epsilon \|W\|_2^2 \|\theta\|_2^2$.

[1]The stable rank of a matrix W is $\|W\|_F^2/\|W\|_2^2$.

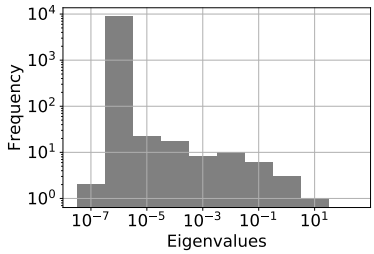


Figure 3.1: The spectrum of the empirical Fisher studied in Figure 1.1.

Notice that stable rank never exceeds the usual rank, and can be significantly smaller when the matrix has a decaying spectrum. The importance matrix considered in SR methods usually have fast decaying spectrum (see Figure 3.1), i.e. small stable rank, making it effective to use CountSketch to approximate quadratic forms with the matrices. For instance, in the synthetic experiment we considered, the stable rank of the empirical Fisher shown in Figure 1.1(a) is 1.26 with standard deviation 0.13, measured over 5 trials. Note that the empirical Fisher is $8,770 \times 8,770$.

3.2.3 Online Extension of Sketched SR

Continual learning often requires learning more than two tasks sequentially. One method of extending the Sketched SR method to learn on multiple tasks to maintain separate sketches for each task and compute the regularizer $\widetilde{\mathcal{R}}(\theta)$ in (3.3) from each of the previous tasks when learning the current one. This approach would cause the memory requirement to grow linearly in the number of tasks and can become a bottleneck in scaling the method. A standard way to tackle this is in works on structural regularization is to apply the *moving average* method to aggregate the histories (Chaudhry et al., 2018; Schwarz et al., 2018). Specifically, let $\widetilde{\Omega}_{\tau-1}$ be the importance matrix maintained after training on the

$(\tau - 1)$-th task, then, given the (approximate) importance matrix $\widetilde{\Omega}$ outputted on the data from task τ, the histories are updated as

$$\widetilde{\Omega}_\tau \leftarrow \alpha\widetilde{\Omega} + (1-\alpha)\widetilde{\Omega}_{\tau-1} \tag{3.5}$$

where $\alpha \in (0, 1]$ is a hyperparameter.

Since the matrix $\widetilde{\Omega}$ is a diagonal matrix for each task in the aforementioned methods, computing the sum from (3.5) is straightforward. Sketched SR, however, doesn't explicitly compute the matrix $\widetilde{\Omega} = \widetilde{W}^\top\widetilde{W}$, hence we cannot hope to compute the matrix $\widetilde{\Omega}_\tau$ defined by the sum in (3.5). We propose the following method: let $\widetilde{W}_{\tau-1}$ be the maintained sketch after training on the $(\tau-1)$-th task, then, given the weight θ^* and the sketch $\widetilde{W}$ outputted on the data from task τ, we update the importance matrix as

$$\widetilde{W}_\tau \leftarrow \sqrt{\alpha}\widetilde{W} + \sqrt{1-\alpha}\widetilde{W}_{\tau-1}. \tag{3.6}$$

When learning on task $\tau + 1$ we use the regularizer

$$\widetilde{\mathcal{R}}_\tau(\theta) := \frac{1}{2n}\|\widetilde{W}_\tau(\theta - \theta^*)\|_2^2. \tag{3.7}$$

A priori, it is not clear why the regularizer from (3.7) is a good approximation to that induced by the importance matrix from (3.5). We give a theorem along with its proof, that implies that for any fixed $\theta \in \mathbb{R}^m$ the regularizer given by (3.7) is close to that induced by the importance matrix $\widetilde{\Omega}_\tau$ from (3.5).

Theorem 2. *Let $W_1, \ldots, W_\tau \in \mathbb{R}^{n\times m}$ be a sequence of matrices, $\alpha_1, \ldots, \alpha_\tau \geq 0$ be a sequence of weights , and $S_1, \ldots, S_\tau \in \mathbb{R}^{t\times n}$ be a sequence of independent CountSketch matrices with sketch size $t \in \mathbb{N}^+$. There exists a constant $C > 0$*

such that for any fixed $\theta \in \mathbb{R}^m$,

$$\left| \left\| \left(\sum_{i=1}^{\tau} \sqrt{\alpha_i} S_i W_i \right) \theta \right\|_2^2 - \sum_{i=1}^{\tau} \alpha_i \|W_i \theta\|_2^2 \right| \leq \frac{C}{\sqrt{t}} \cdot \sum_{i=1}^{\tau} \alpha_i \|W_i \theta\|_2^2$$

with probability at least 0.99.

Proof of Theorem 2. Throughout this proof, we let $(a)_i$ denote the i-th entry of a vector $a \in \mathbb{R}^m$ and let $(A)_j$ denote the j-th row of a matrix $A \in \mathbb{R}^{n \times m}$.

We start with a lemma on the properties of matrix S which we use in our proof of the theorem.

Lemma 3. *The CountSketch matrix* $S \in \mathbb{R}^{t \times n}$ *with sketch size* $t \in \mathbb{N}^+$ *has the property that for any vector* $y \in \mathbb{R}^n$ *and index* $i \in [t]$ *and* $j \neq i$, *i)* $\mathbb{E}(Sy)_i = 0$, *ii)* $\mathbb{E}[(Sy)_i (Sy)_j] = 0$, *and iii)* $\mathbb{E}(Sy)_i^2 = \|y\|^2/t$ *where the expectation is taken over the randomness of the CountSketch matrix.*

Proof. Let $S \in \mathbb{R}^{t \times n}$ be the CountSketch matrix with sketch size t resulting from the 2-wise independent hash function $h : [n] \to [t]$ and the 4-wise independent hash function $\sigma : [n] \times \{1, -1\}$ (see Algorithm 1 for descriptions of h and σ). Let $y \in \mathbb{R}^n$ be a vector and let $i, j \in [t]$ be indices such that $i \neq j$.

To prove i), notice that $\mathbb{E}\left[(Sy)_i\right] = \sum_{k=1}^{n} \mathbb{P}\left(h(k) = i\right) \cdot \mathbb{E}\left[\sigma(k)\right] \cdot (y)_i = 0$ since $\mathbb{E}\left[\sigma(k)\right] = 0$.

To prove ii), we notice that by the definition of h, the random variable $\mathbb{1}\left[h(k) = i\right] \mathbb{1}\left[h(k) = j\right] = 0$ for any $i \neq j$. Then we can expand $\mathbb{E}(Sy)_i (Sy)_j$ as follows:

$$\begin{aligned} \mathbb{E}\left[(Sy)_i (Sy)_j\right] &= 2 \sum_{k=2}^{n} \sum_{l=1}^{k-1} \mathbb{E}\left[\mathbb{1}\left[h(k) = i\right] \mathbb{1}\left[h(l) = j\right] \cdot \sigma(k)\sigma(l) \cdot (y)_i (y)_j\right] \\ &= 2 \sum_{k=2}^{n} \sum_{l=1}^{k-1} \mathbb{E}\left[\sigma(k)\sigma(l)\right] \cdot \mathbb{E}\left[\mathbb{1}\left[h(k) = i\right] \mathbb{1}\left[h(l) = j\right] (y)_i (y)_j\right] = 0 \end{aligned}$$

where the last equality follows from the fact that σ is a 4-wise independent hash function and $i \neq j$.

Finally, we show property iii) as follows:

$$\begin{aligned}
\mathbb{E}(Sy)_i^2 &= \sum_{k=1}^{n} \mathbb{E}\left[\mathbb{1}\left[h(k)=i\right](y)_i^2\right] \\
&\quad + 2\sum_{k=2}^{n}\sum_{l=1}^{k-1} \mathbb{E}\left[\sigma(k)\sigma(l)\right]\mathbb{E}\left[\mathbb{1}\left[h(k)=i\right]\mathbb{1}\left[h(l)=i\right](y)_i^2\right] \\
&= \sum_{k=1}^{n} \mathbb{E}\left[\mathbb{1}\left[h(k)=i\right](y)_i^2\right] + 0 = \sum_{k=1}^{n} \mathbb{P}\left(\mathbb{1}\left[h(k)=i\right]\right)(y)_i^2 \\
&= \sum_{k=1}^{n} \frac{1}{t}\cdot(y)_i^2 = \frac{\|y\|_2^2}{t}.
\end{aligned}$$

In the second equality we used the fact that σ is a 4-wise independent hash function. □

We are now ready to prove Theorem 2.

Proof of Theorem 2. Fix an arbitrary $\theta \in \mathbb{R}^m$ and let $y_1, \ldots, y_\tau \in \mathbb{R}^n$ be the vectors such that $y_i = \sqrt{\alpha_i} W_i \theta$. We then have that:

$$\begin{aligned}
\mathbb{E}\left[\left\|\sum_{k=1}^{\tau} S_k y_k\right\|^2 - \left(\sum_{k=1}^{\tau}\|S_k y_k\|^2\right)\right] &= \mathbb{E}\left[2\sum_{k=2}^{\tau}\sum_{k<l}\sum_{i=1}^{t}(S_l y_l)_i (S_k y_k)_i\right] \\
&= 2\sum_{k<l}\sum_{i=1}^{t}\mathbb{E}(S_l y_l)_i \mathbb{E}(S_k y_k)_i = 0.
\end{aligned}$$

In the second equality we use the fact that S_k and S_l are independent random matrices for $l \neq k$ and property i) from Lemma 3. Next, we bound the variance:

$$\mathrm{Var}\left[\left\|\sum_{k=1}^{\tau} S_k y_k\right\|^2 - \left(\sum_{k=1}^{\tau}\|S_k y_k\|^2\right)\right] = \mathbb{E}\left[\left(\left\|\sum_{k=1}^{\tau} S_k y_k\right\|^2 - \left(\sum_{k=1}^{\tau}\|S_k y_k\|^2\right)\right)^2\right]$$

$$= \mathbb{E}\left[\left(2\sum_{k=2}^{\tau}\sum_{l<k}\sum_{i=1}^{t}(S_k y_k)_i(S_l y_l)_i\right)^2\right]$$

$$= 4\,\underbrace{\mathbb{E}\left[\sum_{k<l}\sum_{i}(S_k y_k)_i^2(S_l y_l)_i^2\right]}_{z_1} + 4\,\underbrace{\mathbb{E}\left[\sum_{k<l}\sum_{i\neq j}(S_k y_k)_i(S_k y_k)_j(S_l y_l)_i(S_l y_l)_j\right]}_{z_2}$$

$$+ 4\,\underbrace{\mathbb{E}\left[\sum_{\substack{k<l,r<s \\ \text{s.t. } \{k\neq r \text{ or } l\neq s\}}}\sum_{i,j}(S_k y_k)_i(S_l y_l)_i(S_r y_r)_j(S_s y_s)_j\right]}_{z_3}.$$

We first argue that $z_3 = 0$; since either $k \neq r$ or $l \neq s$, without loss of generality let $k < r$. As a result, $k < l$ and $k < r < s$. We then have that $\mathbb{E}[(S_k y_k)_i(S_l y_l)_i(S_r y_r)_j(S_s y_s)_j] = \mathbb{E}[(S_k y_k)_i] \cdot \mathbb{E}[(S_l y_l)_i(S_r y_r)_j(S_s y_s)_j] = 0$ since $\mathbb{E}[(S_k y_k)_i] = 0$ using property i) from Lemma 3. Next we argue that $z_2 = 0$; since $\mathbb{E}[(S_k y_k)_i(S_k y_k)_j] = 0$ using property ii) from Lemma 3, for any $k \in [\tau]$ and any $i \neq j$ we have that $\mathbb{E}[(S_k y_k)_i(S_k y_k)_j(S_l y_l)_i(S_l y_l)_j] = \mathbb{E}[(S_k y_k)_i(S_k y_k)_j] \cdot \mathbb{E}[(S_l y_l)_i(S_l y_l)_j] = 0$.

Finally, we can bound z_1 and hence the variance:

$$z_1 = 4\sum_{k<l}\sum_{i}\mathbb{E}(S_k y_k)_i^2\mathbb{E}(S_l y_l)_i^2 = 4\sum_{k<l} t\frac{\|y_k\|^2}{t}\cdot\frac{\|y_l\|^2}{t}$$

$$= \frac{4}{t}\sum_{k<l}\|y_k\|^2\cdot\|y_l\|^2 \leq \frac{2}{t}\left(\sum_{k=1}^{\tau}\|y_k\|^2\right)^2.$$

In the second equality we use the property iii) from Lemma 3 for $\mathbb{E}(S_k y_k)_i^2$ and $\mathbb{E}(S_l y_l)_i^2$.

The theorem follows by applying Chebyshev's inequality on $\|\sum_{k=1}^{\tau} S_k y_k\|_2^2 - \sum_{k=1}^{\tau}\|S_k y_k\|_2^2$ and the definition of $y_1, \ldots, y_\tau$. □

As a corollary to Theorem 2, we show the approximation error of the regularizer from (3.7).

Corollary 3.1. *For any fixed $\theta \in \mathbb{R}^m$, the regularizer $\widetilde{\mathcal{R}}\tau(\theta)$ given by* (3.7) *has the property that*

$$\widetilde{\mathcal{R}}\tau(\theta) = \left(1 \pm \mathcal{O}\left(\frac{1}{\sqrt{t}}\right)\right) \cdot (\theta - \theta^*)^\top \widetilde{\Omega}_\tau (\theta - \theta^*)$$

with probability 0.99 and where $\widetilde{\Omega}_\tau$ is the matrix given by the recurrence in (3.5) *with $\widetilde{\Omega} = \widetilde{W}^\top \widetilde{W}$.*

Remark. Note that Corollary 1.1 enjoys a stronger guarantee than that of Theorem 2, i.e., while the approximation guarantee in Theorem 2 holds for *any fixed* vector $\theta \in \mathbb{R}^m$, the guarantee in Corollary 1.1 holds for all $\theta \in \mathbb{R}^m$ *simultaneously*. We expect that the stronger guarantee of Corollary 1.1 can be achieved in the setting of Theorem 2 by computing $\mathcal{O}(\log(t))$ independent copies of the aggregated sketch $\widetilde{W}_\tau$ from (3.6). The regularizer used when learning on task $\tau+1$ is simply the average of the regularizer $\widetilde{\mathcal{R}}_\tau(\theta)$ from (3.7) outputted by each copy of $\widetilde{W}_\tau$. We leave it to future work to analyze this extension of Sketched SR in order to obtain the stronger guarantee for the setting in Theorem 2.

Chapter 4

Experiments

In this chapter, we present empirical evidence that verifies the effectiveness of our proposed Sketched SR methods. The experiments are conducted with variants of two representative SR algorithms, EWC (Kirkpatrick et al., 2017) and MAS (Aljundi et al., 2018). All the reported numerical results are averaged over 5 runs with different random seeds.

4.1 Synthetic Experiments

We start with a series of synthetic experiments.

Setup. We first consider a synthetic 2D binary classification task from Pan et al. (2020). The experiment consists of 5 classification tasks learnt sequentially using the regularization induced by each of EWC and MAS with a small multi-layer perceptron. The network has $8,770$ parameters. For the regularization matrix induced by EWC and MAS, we compare the performance of various approaches to approximating the matrix including:

(i) a diagonal approximation;

(ii) a block-diagonal approximation, with a sequence of 50×50 non-zero blocks

along the diagonal;

(iii) sketched SR with sketch size $t = 50$;

(iv) a rank-50 SVD;

(v) and the full importance matrix.

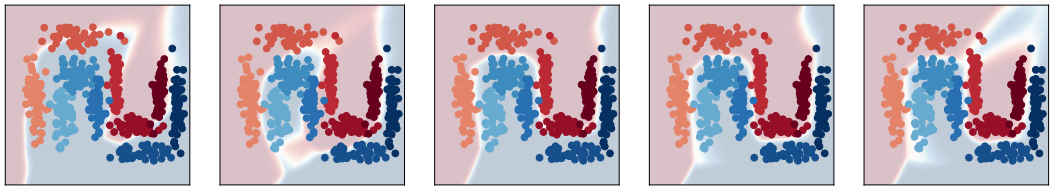
We use a small multi-layer perceptron with the architecture $2 \to 128 \to 64 \to 2$ and with ReLU activation function. For all algorithms, we use ADAM as the optimizer with learning rate 10^{-3}. The minibatch size is 100, and we use the importance parameter $\lambda = 10^3$ and the online learning parameter $\alpha = 0.5$ for all experiments. We repeat all toy example experiments 5 times with different fixed seeds, and report the average accuracy on all tasks. These toy example experiments are conducted on one RTX2080Ti GPU.

Online Learning in Synthetic Experiments. For non-sketched approaches, the regularizer (3.2) in SR methods is approximated by

$$\widetilde{\mathcal{R}}(\theta) := \frac{1}{2}(\theta - \theta_A^*)^\top \widetilde{\Omega}(\theta - \theta_A^*) \tag{4.1}$$

where $\widetilde{\Omega}$ approximates the importance matrix Ω. The online extension of Sketched SR (see Section 3.2) applies moving average on the sketch $\widetilde{W}$, and cannot be directly applied on the regularizer in Equation 4.1. To ensure faithful comparison, moving average is applied on the importance matrix $\widetilde{\Omega}$ in synthetic experiments according to Equation (3.5).

Approximation vs Full Matrix Comparison. We first plot the empirical Fisher (the importance matrix in EWC methods) and the sketched empirical Fisher in Figure 1.1. The empirical Fisher is obtained with the optimal weight that fits the first four tasks and the sketched empirical Fisher uses sketch size



(a) Diagonal (b) Block-Diag. (c) Sketched (d) Low-Rank (e) Full EWC

Acc.	Diagonal	Block-Diag.	Sketched	Low-Rank	Full IM
EWC	$88.0 \pm 7.4\%$	$79.3 \pm 9.8\%$	$92.1 \pm 6.8\%$	$93.9 \pm 3.9\%$	$95.7 \pm 5.4\%$
MAS	$83.1 \pm 5.6\%$	$76.6 \pm 12.0\%$	$85.9 \pm 9.6\%$	$84.2 \pm 9.8\%$	$89.9 \pm 7.2\%$

Figure 4.1: Variants of EWC and MAS on a synthetic 2D binary classification task.

$t = 50$. From the figure we observe that the empirical Fisher cannot be well-approximated by its diagonal or block-diagonal; moreover, the sketched empirical Fisher can utilize the off-diagonal entries to generate a better approximation. This is further supported by the numerical approximation error shown in the table within Figure 1.1. Note that while the low-rank method can offer a better approximation, it is not computationally efficient in practice.

Performance of the Compared Algorithms. We then compare the performance of each algorithms in Figures 4.1. The plots consistently indicate that sketched SR methods are more effective than diagonal SR methods for overcoming catastrophic forgetting. Additionally, while low-rank SR and full SR perform better than sketched SR, they are not computationally feasible in practical settings with large models.

4.2 Permuted-MNIST

Next we demonstrate the effectiveness of our methods with experiments on permuted-MNIST.

Setup. In this benchmark experiment for continual learning (Kirkpatrick et al., 2017; Zenke, Poole, and Ganguli, 2017; Rostami, Kolouri, and Pilly, 2019; Ritter, Botev, and Barber, 2018; Ramasesh, Dyer, and Raghu, 2021), there are 10 sequential tasks, each of them is a 10-classes classification task based on a permuted MNIST dataset, where the pixels in each figure are permuted according to certain rule (to be more specific, the permutation rule is same within a task but random across different tasks). We use a multi-layer perceptron with the architecture $784 \rightarrow 1024 \rightarrow 512 \rightarrow 256 \rightarrow 10$ with ReLU activation function and no bias to learn this classification task. We use ADAM as the optimizer with learning rate 10^{-4} and the online learning parameter $\alpha = 0.25$ for all algorithms. The minibatch size is 100. For each algorithm, a grid search on the regularization coefficient $\lambda \in \{10^i \mid i = 2, 3, \ldots, 6\}$ is used to determine the optimal hyperparameter for the reported results. We uses 50 sketches in Sketched SR to approximate the full importance matrix. All permuted-MNIST experiments are repeated 5 times with different fixed seeds, and we report average accuracy on all tasks. We run permuted-MNIST experiments on a Tesla K80.

Performance of the Compared Algorithms. Figure 4.2 shows the average accuracy across previously learned tasks after each epoch of training for the compared methods. Table 4.1 reports the averaged accuracy (across all tasks) of the compared algorithms. From the figures and the table, we consistently

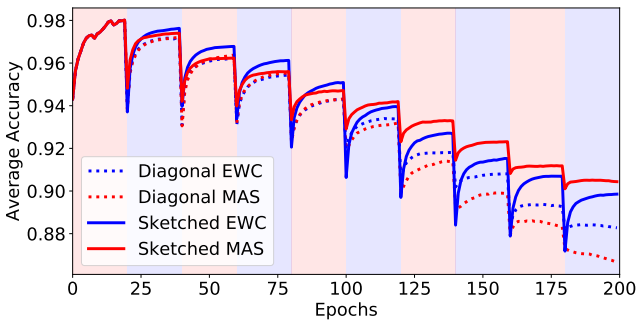


Figure 4.2: The average accuracy across previously learned tasks after each epoch of training for both diagonal and sketched methods on permuted-MNIST.

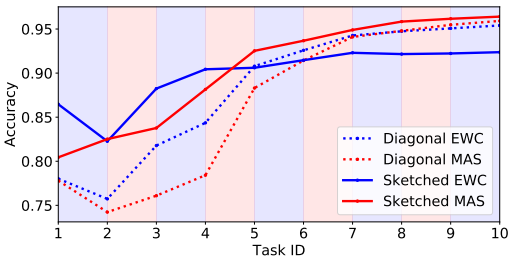


Figure 4.3: The accuracy of each task (after training on all tasks) of sketched methods vs. diagonal methods on permuted-MNIST.

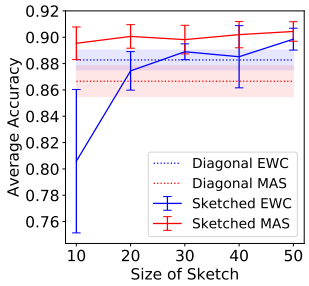
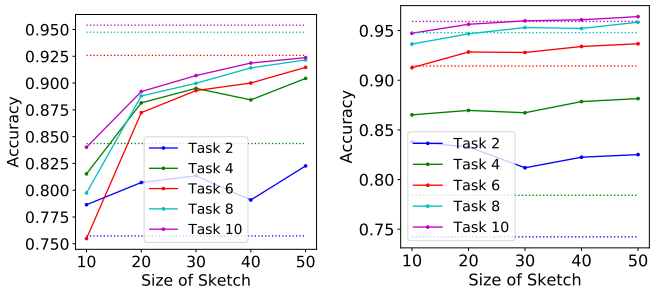


Figure 4.4: Effect of the sketch size (t) on the average accuracy of sketched methods for learning permuted-MNIST tasks.

see that sketched SR methods outperform their diagonal counterparts, in both EWC and MAS regimes, in terms of overcoming catastrophic forgetting. This is explored deeper in Figure 4.3, where we show the accuracy on each task after training on all the tasks for the compared algorithms. According to Figure 4.3, sketched SR methods forget less about the early tasks, which directly demonstrate its advantage for overcoming catastrophic forgetting. This is consistent to our finding from the synthetic experiments.

Effects of the Sketch Size. We then study the effects of the size of the sketch, i.e. t in (3.3), on the performance of sketched SR. The results are shown



(a) Sketched EWC (b) Sketched MAS

Figure 4.5: Effect of the sketch size (t) on task accuracy of sketched methods for learning 10 permuted-MNIST tasks.

in Figure 4.4. From the plot we see a clear trade-off between the size of the sketch and the average accuracy, where the average accuracy generally grows as the size of sketches increases — however using more sketches costs more computation resources. Fortunately, even with a very small sketch size, e.g. $t \geq 30$, which is easily affordable in practice, sketched SR methods already significantly outperform diagonal SR methods. This demonstrates the practical effectiveness of the proposed sketched SR framework.

Effects of the Sketch Size per Task. We further study the effects of the size of the sketch t (See Equation 3.3) on the performance of sketched SR on each task. The results are shown in Figure 4.5. From the plot we see a clear trade-off between the size of the sketch and the accuracy on later tasks, where the accuracy consistently increases as the size of sketches grows. This directly shows that increasing of the size of sketches improves learning capability for new tasks (known in the literature as *intransigence*) with only little trade-off in catastrophic forgetting, with the expense of more computation resources.

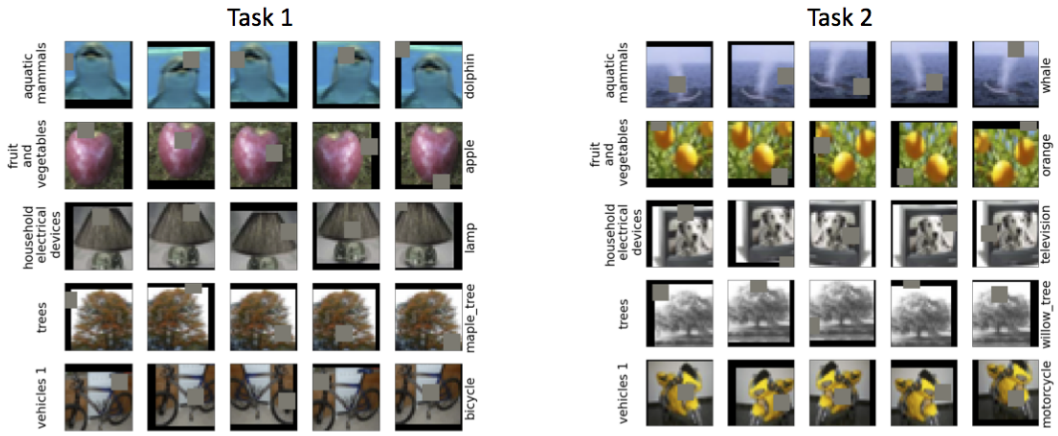


Figure 4.6: Sample images with 5 random augmentations for Task 1 and Task 2 in our CIFAR-100 experiment.

4.3 CIFAR-100

Finally, we provide further verification for the effectiveness of our methods with CIFAR-100 experiments.

Setup. For our CIFAR-100 experiment, we follow the 2-task *CIFAR-100 Distribution Shift* dataset introduced in Ramasesh, Dyer, and Raghu, 2021. The main difference from the split CIFAR experiment commonly used in the literature (see, e.g., (Zenke, Poole, and Ganguli, 2017)) is that the *CIFAR-100 Distribution Shift* does not require task-specific neural network heads for classifying classes of each task. Such a setting is consistent with our previous experiments, in which the same network is used to learn all tasks. In our experiment, similar to Ramasesh, Dyer, and Raghu, 2021, both tasks are 5-class classification problems where each class is one of the 20 superclasses of the CIFAR-100 dataset. For instance, we take the five superclasses *aquatic mammals*, *fruits and vegetables*, *household electrical devices*, *trees*, and *vehicles-1*. The corresponding subclasses for Task 1 are (1) *dolphin*, (2) *apple*, (3) *lamp*,

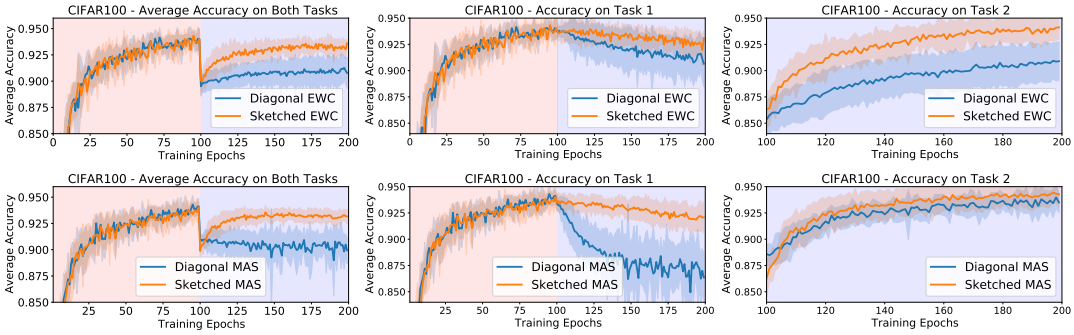


Figure 4.7: The average accuracy (over both tasks) of sketched SR and diagonal SR methods on CIFAR-100.

(4) *maple tree*, and (5) *bicycle*, while for Task 2, they are (1) *whale*, (2) *orange*, (3) *television*, (4) *willow*, and (5) *motorcycle*. Figure 4.6 shows sample images and five random augmentations for the classes in both tasks.

In all experiments, we used a Wide-ResNet (Zagoruyko and Komodakis, 2016) as our backbone. The network has 16 layers, a widening factor of 4, and a dropout rate of 0.2. We leveraged random flip, translation, and cutout (DeVries and Taylor, 2017) as augmentation. We use ADAM as our optimizer for all experiments, with learning rate 10^{-3} and momentum 0.9. The importance parameter λ for each algorithm is: 10^5 for EWC, 10^2 for Sketched EWC, 10^5 for MAS, and 10^3 for Sketched MAS. The minibatch size is 64. The online learning parameter is $\alpha = 0.25$ for all experiments. In Sketched SR algorithms, we uses 50 sketches to approximate the full importance matrix. All reported results are averaged over 10 runs with different random seeds.

Performance of the Compared Algorithms. Figure 4.7 shows the performance comparison between the sketched and diagonal variations of EWC and MAS methods. The plots suggest that sketched variants are significantly more effective than the diagonal versions in terms of overcoming catastrophic

forgetting. The results are consistent with those in synthetic experiments and permuted-MNIST experiments.

Table 4.1: The average accuracy (over all tasks) of sketched SR and diagonal SR methods on Permuted-MNIST and CIFAR-100.

Dataset	Regime	Diagonal	Sketched
Permuted-MNIST	EWC	88.3±0.8%	**89.8±0.9%**
	MAS	86.7±1.2%	**90.4±0.8%**
CIFAR-100	EWC	90.8±1.5%	**93.6±0.4%**
	MAS	89.9±1.2%	**93.2±0.6%**

Chapter 5

Conclusion

In this report we present sketched structural regularization as a general framework for overcoming catastrophic forgetting in continual learning. Compared with the widely-used diagonal version of structural regularization approaches, our methods achieve better performance for overcoming catastrophic forgetting, since an improved approximation to the large importance matrix is adopted. In contrast to the inefficient low-rank approximation methods (e.g., PCA), the proposed sketched structural regularization is computational affordable for practical continual learning models. Finally, the effectiveness of the proposed methods are verified in multiple benchmark continual learning tasks.

References

Li, Haoran, Aditya Krishnan, Jingfeng Wu, Soheil Kolouri, Praveen K Pilly, and Vladimir Braverman (2021). "Lifelong Learning with Sketched Structural Regularization". In: *arXiv preprint arXiv:2104.08604*.

Parisi, German I, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter (2019). "Continual lifelong learning with neural networks: A review". In: *Neural Networks* 113, pp. 54–71.

Kolouri, Soheil, Nicholas A. Ketz, Andrea Soltoggio, and Praveen K. Pilly (2020). "Sliced Cramer Synaptic Consolidation for Preserving Deeply Learned Representations". In: *International Conference on Learning Representations*. URL: `https://openreview.net/forum?id=BJge3TNKwH`.

Aljundi, Rahaf, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars (2018). "Memory aware synapses: Learning what (not) to forget". In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 139–154.

Kirkpatrick, James, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. (2017). "Overcoming catastrophic forgetting in neural networks". In: *Proceedings of the national academy of sciences* 114.13, pp. 3521–3526.

Chaudhry, Arslan, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr (2018). "Riemannian walk for incremental learning: Understanding forgetting and intransigence". In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 532–547.

Zenke, Friedemann, Ben Poole, and Surya Ganguli (2017). "Continual learning through synaptic intelligence". In: *International Conference on Machine Learning*. PMLR, pp. 3987–3995.

Kunstner, Frederik, Lukas Balles, and Philipp Hennig (2019). "Limitations of the empirical fisher approximation for natural gradient descent". In: *arXiv preprint arXiv:1905.12558*.

Liu, Xialei, Marc Masana, Luis Herranz, Joost Van de Weijer, Antonio M Lopez, and Andrew D Bagdanov (2018). "Rotate your networks: Better weight consolidation and less catastrophic forgetting". In: *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, pp. 2262–2268.

Ritter, Hippolyt, Aleksandar Botev, and David Barber (2018). "Online structured laplace approximations for overcoming catastrophic forgetting". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 3742–3752.

Horn, Roger A and Charles R Johnson (2012). *Matrix analysis*. Cambridge university press.

Pan, Pingbo, Siddharth Swaroop, Alexander Immer, Runa Eschenhagen, Richard E Turner, and Mohammad Emtiyaz Khan (2020). "Continual deep learning by functional regularisation of memorable past". In: *arXiv preprint arXiv:2004.14070*.

Charikar, Moses, Kevin Chen, and Martin Farach-Colton (2002). "Finding frequent items in data streams". In: *International Colloquium on Automata, Languages, and Programming*. Springer, pp. 693–703.

Sagun, Levent, Utku Evci, V Ugur Guney, Yann Dauphin, and Leon Bottou (2017). "Empirical analysis of the hessian of over-parametrized neural networks". In: *arXiv preprint arXiv:1706.04454*.

Chaudhari, Pratik and Stefano Soatto (2018). "Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks". In: *2018 Information Theory and Applications Workshop (ITA)*. IEEE, pp. 1–10.

Jung, Heechul, Jeongwoo Ju, Minju Jung, and Junmo Kim (2016). "Less-forgetting learning in deep neural networks". In: *arXiv preprint arXiv:1607.00122*.

Li, Zhizhong and Derek Hoiem (2017). "Learning without forgetting". In: *IEEE transactions on pattern analysis and machine intelligence* 40.12, pp. 2935–2947.

Rannen, Amal, Rahaf Aljundi, Matthew B Blaschko, and Tinne Tuytelaars (2017). "Encoder based lifelong learning". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1320–1328.

Shin, Hanul, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim (2017). "Continual learning with deep generative replay". In: *arXiv preprint arXiv:1705.08690*.

Hu, Wenpeng, Zhou Lin, Bing Liu, Chongyang Tao, Zhengwei Tao, Jinwen Ma, Dongyan Zhao, and Rui Yan (2018). "Overcoming catastrophic forgetting for continual learning via model adaptation". In: *International Conference on Learning Representations*.

Rozantsev, Artem, Mathieu Salzmann, and Pascal Fua (2018). "Beyond sharing weights for deep domain adaptation". In: *IEEE transactions on pattern analysis and machine intelligence* 41.4, pp. 801–814.

Wu, Chenshen, Luis Herranz, Xialei Liu, Joost van de Weijer, Bogdan Raducanu, et al. (2018). "Memory replay gans: Learning to generate new categories without forgetting". In: *Advances in Neural Information Processing Systems* 31, pp. 5962–5972.

Li, Xilai, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong (2019). "Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting". In: *International Conference on Machine Learning*. PMLR, pp. 3925–3934.

Rostami, Mohammad, Soheil Kolouri, and Praveen K Pilly (2019). "Complementary learning for overcoming catastrophic forgetting using experience replay". In: *arXiv preprint arXiv:1903.04566*.

Feldman, Dan and Michael Langberg (2011). "A Unified Framework for Approximating and Clustering Data". In: *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*. STOC '11. San Jose, California, USA: Association for Computing Machinery, 569–578. ISBN: 9781450306911. DOI: 10.1145/1993636.1993712. URL: https://doi.org/10.1145/1993636.1993712.

Har-Peled, Sariel and Soham Mazumdar (2004). "On Coresets for K-Means and k-Median Clustering". In: *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*. STOC '04. Chicago, IL, USA: Association for Computing Machinery, 291–300. ISBN: 1581138520. DOI: 10.1145/1007352.1007400. URL: https://doi.org/10.1145/1007352.1007400.

Nelson, Jelani and Huy L Nguyên (2013). "OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings". In: *2013 ieee 54th annual symposium on foundations of computer science*. IEEE, pp. 117–126.

Cohen, Michael B (2016). "Simpler and tighter analysis of sparse oblivious subspace embeddings". In: *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), to appear*.

Sarlos, Tamas (2006). "Improved approximation algorithms for large matrices via random projections". In: *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*. IEEE, pp. 143–152.

Clarkson, Kenneth L and David P Woodruff (2017). "Low-rank approximation and regression in input sparsity time". In: *Journal of the ACM (JACM)* 63.6, pp. 1–45.

Meng, Xiangrui and Michael W Mahoney (2013). "Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression".

In: *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pp. 91–100.

Cohen, Michael B, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu (2015). "Dimensionality reduction for k-means clustering and low rank approximation". In: *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pp. 163–172.

Drineas, Petros, Malik Magdon-Ismail, Michael W Mahoney, and David P Woodruff (2012). "Fast approximation of matrix coherence and statistical leverage". In: *The Journal of Machine Learning Research* 13.1, pp. 3475–3506.

Lee, Yin Tat, Zhao Song, and Qiuyi Zhang (2019). "Solving empirical risk minimization in the current matrix multiplication time". In: *Conference on Learning Theory*. PMLR, pp. 2140–2157.

Ahle, Thomas D, Michael Kapralov, Jakob BT Knudsen, Rasmus Pagh, Ameya Velingker, David P Woodruff, and Amir Zandieh (2020). "Oblivious sketching of high-degree polynomial kernels". In: *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, pp. 141–160.

Brand, Jan van den, Binghui Peng, Zhao Song, and Omri Weinstein (2021). "Training (Overparametrized) Neural Networks in Near-Linear Time". In: *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*. Ed. by James R. Lee. Vol. 185. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 63:1–63:15. ISBN: 978-3-95977-177-1. URL: https://drops.dagstuhl.de/opus/volltexte/2021/13602.

Goodfellow, Ian, Yoshua Bengio, Aaron Courville, and Yoshua Bengio (2016). *Deep learning*. Vol. 1. MIT press Cambridge.

Schwarz, Jonathan, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell (2018). "Progress & compress: A scalable framework for continual learning". In: *International Conference on Machine Learning*. PMLR, pp. 4528–4537.

Cohen, Michael B., Jelani Nelson, and David P. Woodruff (2016). "Optimal Approximate Matrix Product in Terms of Stable Rank". In: *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*. Ed. by Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi. Vol. 55. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 11:1–11:14. ISBN: 978-3-95977-013-2. DOI: 10.4230/LIPIcs.ICALP.2016.11. URL: http://drops.dagstuhl.de/opus/volltexte/2016/6278.

Ramasesh, Vinay Venkatesh, Ethan Dyer, and Maithra Raghu (2021). "Anatomy of Catastrophic Forgetting: Hidden Representations and Task Semantics". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=LhY8QdUGSuw.

Zagoruyko, Sergey and Nikos Komodakis (2016). "Wide Residual Networks". In: *British Machine Vision Conference 2016*. British Machine Vision Association.

DeVries, Terrance and Graham W Taylor (2017). "Improved regularization of convolutional neural networks with cutout". In: *arXiv preprint arXiv:1708.04552*.

Haoran Li

DOB: 07 May 2001 *Phone:* 410-949-6154 *E-mail:* hli143@jhu.edu

Education

Johns Hopkins University, Baltimore, MD 2019 – Present

- M.S.E. in Computer Science, GPA: 3.76/4.0
- **Courses**: Combinatorics and Graph Theory (A+), Randomized Algorithms (A), High-Dimensional Approximation & Statistical Learning, Probabilistic Models of the Visual Cortex (A); Machine Learning, Computer Vision, Object-Oriented Software Engineering, Causal Inference

Tsinghua University, Beijing, China 2015 – 2019

- B.Eng. in Automation, GPA: 3.52/4.0
- **Courses**: Calculus (A-), Linear Algebra (A), Probability & Statistics (A-), Applied Stochastic Processes, Optimization Algorithms, Operation Research; Artificial Intelligence, Digital Image Processing, Pattern Recognition

Publication

Li, H., Krishnan, A., Wu, J., Kolouri, S., Pilly, P. K., & Braverman, V. (2021). Lifelong Learning with Sketched Structural Regularization. *arXiv preprint arXiv:2104.08604*. *pdf arxiv*

Li, H., & Xu, H. (2019). Video-Based Sentiment Analysis with hvnLBP-TOP Feature and bi-LSTM. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 33, pp. 9963-9964). *pdf doi*

Selected Research Experience

Lifelong Learning via Sketched Structural Regularization | RA 2020.9 – Present

Advisor: Prof. Vladimir Braverman, Associate Professor of Computer Science, JHU

Perform lifelong learning on multiple continuous tasks with different data distributions. Paper submitted to ICML.

- Conducted lifelong learning by adding a structural regularization (SR) penalty, which is the quadratic form of the diagonal approximation to a huge ($\sim 10^6 \times 10^6$) matrix.
- Proposed Sketched Structural Regularization (Sketched SR), which efficiently compresses and better approximates the matrix than diagonal approximation with theoretical guarantee.
- Applied the algorithm on SR methods (EWC, MAS). Sketched SR outperforms diagonal approximation on all methods in multiple benchmark datasets.

Neural Network Pruning via Coreset | RA 2020.9 – 2020.12

Advisor: Prof. Vladimir Braverman, Associate Professor of Computer Science, JHU

Compress deep neural network by neural pruning, providing a theoretical bound of error.

- Conduct data-independent compression on general deep networks including MLP, CNN.
- Proposed a new neural pruning algorithm, providing the first provable approximation error of data-independent neural pruning algorithm for multi-layer neural network.

AlexNet Neuron Geometry | RA 2020.6 – 2020.8

Advisor: Prof. Ed Connor, Professor of Neuroscience, JHU

Analyze the geometric structure of AlexNet convolutional layer responses on computer-graphics generated stimuli.

- Applied genetic algorithms to generate stimuli images with high response on each neuron of Alexnet.
- Performed Laplacian embedding on samples of responses on different AlexNet layers, finding low-dimensional structure (protruding lines) of high-response samples out of low-response sample pool.

Lung Histopathological Image Analysis | RA 2018.10 – 2019.6

Advisor: Prof. Rui Jiang, Associate Professor in Bioinformatics, Tsinghua University

Extract medical information of lung slice images based on cell features and cell location graph.

- Dealt with whole-slide pathological images from lung cancer patients and healthy individuals.
- Introduced attributed graph representation of pathological images based on cell features and locations, which consists of geometric information of cells, thus achieving better biomedical interpretability.
- Applied GNN on attributed graph of lung images tiles for cancer classification, which outperforms all feature-based methods.

Sentiment Analysis on Video and Multimodal Data | RA 2017.12 – 2018.9

Advisor: Prof. Hua Xu, Associate Professor of Computer Science, Tsinghua University

Evaluate the emotional status of people based on their facial expression features, and conduct multimodal sentiment analysis with visual, audio and textual features. Paper accepted by AAAI 2019.

- Dealt with multiple video and multimodal datasets with sentimental labels.
- Organized multimodal sentiment dataset, which was extracted from product reviews collected from Chinese video site *bilibili.com*; Different accents of Mandarin, Cantonese and Szechuanese were included.
- Proposed a novel feature extraction method for video-based sentiment analysis that outperforms the state-of-the-art video sentiment classification model.

Scholarships & Awards

Academic Excellence Scholarship (24/146) 2016

2^{nd} *Prize* National Collegiate Physics Competition 2016

2^{nd} *Prize* National Mathematics Olympiad(Beijing Division) 2013

Skills

- Programming Languages: Python, MATLAB, C/C++
  - Python packages: pytorch, tensorflow, numpy, scipy, sklearn