Theoretical Understanding of the Information Flow on Continual Learning Performance

Joshua Andle and Salimeh Yasaei Sekeh

University of Maine, Orono ME 04469, USA
salimeh.yasaei@maine.edu

Abstract. Continual learning (CL) requires a model to continually learn new tasks with incremental available information while retaining previous knowledge. Despite the numerous previous approaches to CL, most of them still suffer forgetting, expensive memory cost, or lack sufficient theoretical understanding. While different CL training regimes have been extensively studied empirically, insufficient attention has been paid to the underlying theory. In this paper, we establish a probabilistic framework to analyze information flow through layers in networks for sequential tasks and its impact on learning performance. Our objective is to optimize the information preservation between layers while learning new tasks. This manages task-specific knowledge passing throughout the layers while maintaining model performance on previous tasks. Our analysis provides novel insights into information adaptation within the layers during incremental task learning. We provide empirical evidence and practically highlight the performance improvement across multiple tasks. Code is available at https://github.com/Sekeh-Lab/InformationFlow-CL.

Keywords: Continual Learning; Information Flow; Forgetting

1 Introduction

Humans are continual learning systems that have been very successful at adapting to new situations while not forgetting about their past experiences. Similar to the human brain, continual learning (CL) tackles the setting of learning new tasks sequentially without forgetting information learned from the previous tasks [3,15,20]. A wide variety of CL methods mainly either minimize a loss function which is a combination of forgetting and generalization loss to reduce catastrophic forgetting [11,13,18,30,31] or improve quick generalization [7,27]. While these approaches have demonstrated state-of-the-art performance and achieve some degree of continual learning in deep neural networks, there has been limited prior work extensively and analytically investigating the impact that different training regimes can have on learning a sequence of tasks. Although major advances have been made in the field, one recurring problem that still remains not completely solved is that of catastrophic forgetting (CF). An approach to address this goal is to gradually extend acquired knowledge learned within layers in the network and use it for future learning. While the CF issue

has been extensively studied empirically, little attention has been paid from a theoretical angle [10,18,21]. To the best of our knowledge, there are no works which explain what occurs when certain portions of a network are more important than others for passing information of a given task downstream to the end of the network. In this paper, we explore the CL performance and CF problem from a probabilistic perspective. We seek to understand the connection of the passing of information downstream through layers in the network and learning performance at a more in-depth and fundamental theoretical level. We integrate these studies into two central questions:

(1) *Given a sequence of joint random variables and tasks, how much does information flow between layers affect learning performance and alleviate CF?*
(2) *Given a sequence of tasks, how much does the sparsity level of layers on task-specific training influence the forgetting?*

The answers to these questions are theoretically and practically important for continual learning research because: (1) despite the tangible improvements in task learning, the core problem of deep network efficiency on performance assists selective knowledge sharing through downstream information within layers; (2) a systematic understanding of learning tasks provides schemes to accommodate more tasks to learn; and (3) monitoring information flow in the network for each task alleviates forgetting.
Toward our analysis, we measure the information flow between layers given a task by using dependency measures between filters in consecutive layers conditioned on tasks. Given a sequence of joint random variables and tasks, we compute the forgetting by the correlation between task and trained model's loss on the tasks in the sequence. To summarize, our contributions in this paper are,

- Introducing the new concept of *task-sensitivity*, which targets task-specific knowledge passing through layers in the network.
- Providing a theoretical and in-depth analysis of information flow through layers in networks for task sequences and its impact on learning performance.
- Optimizing the information preservation between layers while learning new tasks by freezing task-specific important filters.
- Developing a new bound on expected forgetting using optimal freezing mask.
- Providing experimental evidence and practical observations of layer connectivities in the network and their impact on accuracy.

Organization: The paper is organized as follows. In Section 2 we briefly review the continual learning problem formulation and fundamental definitions of the performance. In addition, a set of new concepts including task sensitivity and task usefulness of layers in the network is introduced. In Section 3 we establish a series of foundational and theoretical findings that focus on performance and forgetting analysis in terms of the sensitivity property of layers. A new bound on expected forgetting based on the optimal freezing mask is given in this section. Finally, in Section 5 we provide experimental evidence of our analysis using the CIFAR-10/100 and Permuted MNIST datasets. The main proofs of the theorems

in the paper are given in the supplementary materials, although in Section 3.4 we provide the key components and techniques that we use for the proofs.

2 Problem Formulation

In supervised continual learning, we are given a sequence of joint random variables $(\mathbf{X}_t, T_t)$, with realization space $\mathcal{X}_t \times \mathcal{T}_t$ where $(\mathbf{x}_t, y_t)$ is an instance of the $\mathcal{X}_t \times \mathcal{T}_t$ space. We use $\|.\|$ to denote the Euclidean norm for vectors and $\|.\|_F$ to denote the Frobenius norm for matrices. In this section, we begin by presenting a brief list of notations and then provide the key definitions.
Notations: We assume that a given DNN has a total of L layers where,

- $F^{(L)}$: A function mapping the input space $\mathcal{X}$ to a set of classes $\mathcal{T}$, i.e. $F^{(L)} : \mathcal{X} \mapsto \mathcal{T}$.
- $f^{(l)}$: The l-th layer of $F^{(L)}$ with M_l as number of filters in layer l.
- $f_i^{(l)}$: i-th filter in layer l.
- $F^{(i,j)} := f^{(j)} \circ \ldots \circ f^{(i)}$: A subnetwork which is a group of consecutive layers.
- $F^{(j)} := F^{(1,j)} = f^{(j)} \circ \ldots \circ f^{(1)}$: First part of the network up to layer j.
- $\sigma^{(l)}$: The activation function in layer l.
- $\widetilde{f}_t^{(l)}$: Sensitive layer for task t.
- $\widetilde{F}_t^{(L)} := F_t^{(L)}/\widetilde{f}_t^{(l)}$: The network with L layers when l-th sensitive layer $\widetilde{f}^{(l)}$ is frozen while training on task t.
- $\pi(T_t)$: The prior probability of class label $T_t \in \mathcal{T}_t$.
- η_{tl}, γ_{tl}: Thresholds for sensitivity and usefulness of l-th layer $f^{(l)}$ for task t.

In this section, we revisit the standard definition of training performance and forgetting and define the new concepts *task-sensitive layer* and *task-useful layer*.

Definition 1. (Task-Sensitive Layer) *The l-th layer, $f^{(l)}$, is called a t-task-sensitive layer if the average information flow between filters in consecutive layers l and $l+1$ is high i.e.*

$$\Delta_t(f^{(l)}, f^{(l+1)}) := \frac{1}{M_l\, M_{l+1}} \sum_{i=1}^{M_l} \sum_{j=1}^{M_{l+1}} \rho\left(f_i^{(l)}, f_j^{(l+1)} | T_t\right) \geq \eta_{lt}, \tag{1}$$

where ρ is a connectivity measure given task T_t such as conditional Pearson correlation or conditional Mutual Information [4,5]. In this work we focus on only Pearson correlation as the connectivity measure between layers l and $l+1$.

Without loss of generality, in this work we assume that filters $f_i^{(l)}$, $i = 1, \ldots, M_l$, are normalized such that

$$\mathbb{E}_{(\mathbf{X}_t, T_t)\sim D_t}\left[f_i^{(l)}(\mathbf{X}_t)|T_t\right] = 0 \;\text{ and }\; \mathbb{V}\left[f_i^{(l)}(\mathbf{X}_t)|T_t\right] = 1, \quad l = 1, \ldots, L,$$

Therefore the Pearson correlation between the i-th filter in layer l and the j-th filter in layer $l+1$ becomes

$$\rho(f_i^{(l)}, f_j^{(l+1)}|T_t) := \mathbb{E}_{(\mathbf{X}_t,T_t)\sim D_t}\left[f_i^{(l)}(\mathbf{X}_t)f_j^{(l+1)}(\mathbf{X}_t)|T_t\right]. \tag{2}$$

Note that in this paper we consider the absolute value of ρ in the range $[0,1]$.

Definition 2. (Task-Useful Layer) *Suppose input $\mathbf{X}_t$ and task T_t have joint distribution $\mathcal{D}_t$. For a given distribution $\mathcal{D}_t$, the l-layer $f^{(l)}$ is called t-task-useful if there exist two mapping functions $G_l : \mathcal{L}_l \mapsto \mathcal{T}_t$ and $K_l : \mathcal{X}_t \mapsto \mathcal{L}_l$ such that*

$$\mathbb{E}_{(\mathbf{X}_t,T_t)\sim\mathcal{D}_t}\left[T_t \cdot G_l \circ f^{(l)}(K_{l-1} \circ \mathbf{X}_t)\right] \geq \gamma_{tl}. \tag{3}$$

Note that here $f^{(l)}$ is a map function $f^{(l)} : \mathcal{L}_{l-1} \mapsto \mathcal{L}_l$.

Within this formulation, two parameters determine the contributions of the l-th layer of network $F^{(l)}$ on task T_t: η_{tl} the contribution of passing forward the information flow to the next consecutive layer, and γ_{tl}, the contribution of the l-th layer in learning task T_t. Training a neural network $F_t^{(L)} \in \mathcal{F}$ is performed by minimizing a loss function (empirical risk) that decreases with the correlation between the weighted combination of the networks and the label:

$$\mathbb{E}_{(\mathbf{X}_t,T_t)\sim D_t}\left\{L_t(F_t^{(L)}(\mathbf{X}_t), T_t)\right\} = -\mathbb{E}_{(\mathbf{X}_t,T_t)\sim D_t}\left\{T_t \cdot \left(b + \sum_{F_t\in\mathcal{F}} w_{F_t} \cdot F_t^{(L)}(\mathbf{X}_t)\right)\right\}. \tag{4}$$

We remove offset b without loss of generality. Define

$$\ell_t(\omega) := -\sum_{F_t\in\mathcal{F}} w_{F_t} \cdot F_t^{(L)}(\mathbf{X}_t), \tag{5}$$

therefore the loss function in (4) becomes $\mathbb{E}_{(\mathbf{X}_t,T_t)\sim D_t}\left\{T_t \cdot \ell_t(\omega)\right\}$. Let ω_t^* be the set of parameters when the network is trained on task T_t that minimizes (4):

$$\omega_t^* := argmin_{\omega_t}\mathbb{E}_{(\mathbf{X}_t,T_t)\sim D_t}\left\{T_t \cdot (\ell_t(\omega_t))\right\}, \tag{6}$$

where ℓ_t is defined in (5). The total risk of all seen tasks $t < \tau$ is given by

$$\sum_{t=1}^{\tau}\mathbb{E}_{(\mathbf{X}_t,T_t)\sim D_t}\left\{T_t \cdot \ell_t(\omega_\tau)\right\}. \tag{7}$$

The set of parameters when the network $F^{(l)}$ is trained after seeing all tasks is the solution of minimizing the risk in (7) and is denoted by ω_τ^*.

Definition 3. (Performance Difference) *Suppose input $\mathbf{X}_t$ and task T_t have joint distribution $\mathcal{D}_t$. Let $\widetilde{F}_t^{(L)} := F_t^{(L)}/\widetilde{f}_t^{(l)} \in \mathcal{F}$ be the network with L layers when l-layer $f^{(l)}$ is frozen while training on task t. The performance difference between training $F_t^{(L)}$ and $\widetilde{F}_t^{(L)}$ is defined as*

$$d(F_t^{(L)}, \widetilde{F}_t^{(L)}) := \mathbb{E}_{(\mathbf{X}_t,T_t)\sim D_t}\left\{L_t(F_t^{(L)}(\mathbf{X}_t), T_t) - L_t(\widetilde{F}_t^{(L)}(\mathbf{X}_t), T_t)\right\}. \tag{8}$$

Let ω_t^* and $\widetilde{\omega}_t^*$ be the convergent parameters after training $F_t^{(L)}$ and $\widetilde{F}_t^{(L)}$ has been finished for task T_t, respectively. Define the training deviation for T_t as:

$$\delta_t(\omega_t^*|\widetilde{\omega}_t^*) := \ell_t(\omega_t^*) - \ell_t(\widetilde{\omega}_t^*). \tag{9}$$

The optimal performance difference in Definition 3 is the average of δ_t in (9):

$$d(F_t^{(L)}, \widetilde{F}_t^{(L)}) = \mathbb{E}_{(\mathbf{X}_t,T_t)\sim D_t}\left[T_t \cdot \delta_t(\omega_t^*|\widetilde{\omega}_t^*)\right] = \mathbb{E}_{(\mathbf{X}_t,T_t)\sim D_t}\left[T_t \cdot \left(\ell_t(\omega_t^*) - \ell_t(\widetilde{\omega}_t^*)\right)\right].$$

3 Continual Learning Performance Study

Our goal is to decide which filters trained for intermediate task T_t to prune/freeze when training the network on task T_{t+1}, given the sensitivity scores of layers introduced in (1), so that the predictive power of the network is maximally retained and not only does forgetting not degrade performance but we also gain a performance improvement. In this section, we first take an in-depth look at the layers and show the relationship between task sensitive and task useful layers. Second we provide an analysis in which we show that sensitive layers affect performance if they get frozen while training the network on the new task.

3.1 Performance Analysis

The motivation of our objective in this section is that the difference between the loss functions produced by the original network $F^{(L)}$ and the frozen network $\widetilde{F}_t^{(L)}$ should be maximized with respect to sensitive and important filters. We begin by showing that sensitive layers are useful in improving network performance.

Theorem 1. *For a given sequence of joint random variables $(\mathbf{X}_t, T_t) \sim \mathcal{D}_t$ and network $F^{(L)}$, if the l-th layer, $f^{(l)}$ is t-task-sensitive then it is t-task-useful.*

Theorem 2. *Suppose input $\mathbf{x}_t$ and label y_t are samples from $(\mathbf{X}_t, T_t)$ with joint distribution $\mathcal{D}_t$. For a given distribution $\mathcal{D}_t$, if the layer l is a t-task-useful layer,*

$$\mathbb{E}_{(\mathbf{X}_t,T_t)\sim\mathcal{D}_t}\left[T_t \cdot G_l \circ f^{(l)}(K_{l-1} \circ \mathbf{X}_t)\right] \geq \gamma_{tl}, \tag{10}$$

where $G_l : \mathcal{L}_l \mapsto \mathcal{T}_t$ and $K_l : \mathcal{X}_t \mapsto \mathcal{L}_l$ are map functions. Then removing layer l decreases the performance i.e.

$$d(F_t^{(L)}, \widetilde{F}_t^{(L)}) := \mathbb{E}_{(\mathbf{X}_t,T_t)\sim D_t}\left\{L_t(F_t^{(L)}(\mathbf{X}_t), T_t) - L_t(\widetilde{F}_t^{(L)}(\mathbf{X}_t), T_t)\right\} > K(\gamma_{tl}). \tag{11}$$

Here $\widetilde{F}_t^{(L)} := F_t^{(L)}/\widetilde{f}_t^{(l)} \in \mathcal{F}$ is the network with L layers when layer l is frozen while training on task t. The function $K(\gamma_{tl})$ is increasing in γ_{tl}.

An immediate result from the combination of Theorems 1 and 2 is stated below:

Theorem 3. *Suppose input $\mathbf{x}_t$ and label y_t are samples from joint random variables $(\mathbf{X}_t, T_t)$ with distribution $\mathcal{D}_t$. For a given distribution $\mathcal{D}_t$, if the layer l is a t-task-sensitive layer i.e. $\Delta_t(f^{(l)}, f^{(l+1)}) \geq \eta_{tl}$, then the performance difference between $d(F_t^{(L)}, \widetilde{F}_t^{(L)})$ is bounded as*

$$d(F_t^{(L)}, \widetilde{F}_t^{(L)}) := \mathbb{E}_{(\mathbf{X}_t, T_t)\sim D_t}\left\{L_t(F_t^{(L)}(\mathbf{X}_t), T_t) - L_t(\widetilde{F}_t^{(L)}(\mathbf{X}_t), T_t)\right\} \geq g(\eta_{tl}), \tag{12}$$

where g is an increasing function of η_{tl}. Here $\widetilde{F}_t^{(L)} := F_t^{(L)}/\widetilde{f}_{t-1}^{(l)} \in \mathcal{F}$ is the network with L layers when layer l is frozen while training on task t.

One important takeaway from this theorem is that as sensitivity between layers η_{tl} increases the performance gap between the original and frozen network's loss functions increases. An important property of filter importance is that it is a probabilistic measure and can be computed empirically along the network. The total loss (empirical risk) on the training set for task T_t is approximated by $\frac{1}{|T_t|}\sum_{(\mathbf{x}_t, y_t)} y_t \ell_t(\omega_t; \mathbf{x}_t, y_t)$, where ℓ_t is a differentiable loss function (5) associated with data point $(\mathbf{x}_t, y_t)$ for task T_t or we use cross entropy loss.

3.2 Forgetting Analysis

When sequentially learning new tasks, due to restrictions on access to examples of previously seen tasks, managing the forgetting becomes a prominent challenge. In this section we focus on measuring the forgetting in CL with two tasks. It is potentially possible to extend these findings to more tasks.
Let ω_t^* and ω_{t+1}^* be the convergent parameters after training has been finished for the tasks T_t and T_{t+1} sequentially. Forgetting of the t task is defined as

$$O_t := \ell_t(\omega_{t+1}^*) - \ell_t(\omega_t^*) \tag{13}$$

In this work, we propose the expected forgetting measure based on correlation between task T_t and forgetting (13) given distribution $\mathcal{D}_t$:

Definition 4. (Expected Forgetting) *Let ω_t^* and ω_{t+1}^* be the convergent or optimum parameters after training has been finished for the t and $t+1$ task sequentially. The expected forgetting denoted by EO_t is defined as*

$$EO_t := \mathbb{E}_{(\mathbf{X}_t, T_t)\sim\mathcal{D}_t}\left[T_t \cdot \left|\left(\ell_t(\omega_{t+1}^*) - \ell_t(\omega_t^*)\right)\right|\right]. \tag{14}$$

Theorem 4. *Suppose input $\mathbf{x}_t$ and label y_t are samples from joint distribution $\mathcal{D}_t$. For a given distribution $\mathcal{D}_t$, if the layer l is a t-task-useful layer,*

$$\mathbb{E}_{(\mathbf{X}_t, T_t)\sim\mathcal{D}_t}\left[T_t \cdot G_l \circ f^{(l)}(K_{l-1} \circ \mathbf{X}_t)\right] \geq \gamma_{tl}, \tag{15}$$

then expected forgetting EO_t defined in (14) is bounded by $\epsilon(\gamma_{tl})$, a decreasing function of γ_{tl} i.e.

$$\widetilde{EO}_t := \mathbb{E}_{(\mathbf{X}_t, T_t)\sim D_t}\left\{L_t(\widetilde{F}_{t+1}^{(L)}(\mathbf{X}_t), T_t) - L_t(F_t^{(L)}(\mathbf{X}_t), T_t)\right\} < \epsilon(\gamma_{tl}), \tag{16}$$

where $\widetilde{F}_{t+1}^{(L)} := F_{t+1}^{(L)} / \widetilde{f}_{t+1}^{(l)} \in \mathcal{F}$ *is the network with* L *layers when layer* l *is frozen while training on task* $t+1$.

A few notes on this bound: (i) based on our finding in (16), we analytically show that under the assumption that the l-th layer is highly t-task-useful i.e. when the hyperparameter γ_{tl} is increasing then average forgetting is decreasing if we freeze the layer l during training the network on new task T_{t+1}. This is achieved because $\epsilon(\gamma_{tl})$ is a decreasing function with respect to γ_{tl}; (ii) by a combination of Theorems 1 and 2 we achieve an immediate result that if layer l is t-task-sensitive then forgetting is bounded by a decreasing function of threshold η_{tl}, $\epsilon(\eta_{tl})$; (iii) We prove that the amount of forgetting that a network exhibits from learning the tasks sequentially correlates with the connectivity properties of the filters in consecutive layers. In particular, the larger these connections are, the less forgetting happens. We empirically verify the relationship between expected forgetting and average connectivity in Section 5.

3.3 A Bound on EO_t Using Optimal Freezing Mask

Let ω_t^* be the set of parameters when the network is trained on task T_t, the optimal sparsity for layer $f^{(l)}$ with optimal mask ${m^*}_{t+1}^{(l)}$ while training on task T_{t+1} is achieved by

$$(\omega_{t+1}^*, {m^*}_{t+1}^{(l)}) := \underset{\omega_{t+1}, m}{\arg\min}\, \mathbb{E}_{(\mathbf{X}_t, T_t)\sim D_t} \Big\{ \big| T_t \cdot \big(\ell_t(m_{t+1}^{(l)} \odot \omega_{t+1}) - \ell_t(\omega_t^*) \big) \big| \Big\}, \quad (17)$$

where ${m^*}_{t+1}^{(l)}$ is the binary mask matrix created after freezing filters in the l-th layer after training on task T_t (masks are applied to the past weights) and before training on task T_{t+1}. Denote ${P^*}_m^{(l)} = \frac{\|{m^*}_{t+1}^{(l)}\|_0}{|{\omega^*}_{t+1}^{(l)}|}$ the optimal sparsity of frozen filters in layer l in the original network $F^{(L)}$.

Definition 5. (Task-Fully-Sensitive Layer) *The* l*-th layer,* $f^{(l)}$*, is called a* t*-task-fully-sensitive layer if the average information flow between filters in layers* l *and* $l+1$ *is maximum i.e.* $\Delta_t(f^{(l)}, f^{(l+1)}) \to 1$ *(a.s.). Note that here* ρ *in (1) is a connectivity measure which varies in* $[0, 1]$.

Theorem 5. *Suppose input* $\mathbf{x}_t$ *and label* y_t *in space* $\mathcal{X}_t \times \mathcal{T}_t$ *are samples from random variables* $(\mathbf{X}_t, T_t)$ *with joint distribution* $\mathcal{D}_t$*. For a given distribution* $\mathcal{D}_t$*, if layer* l *is* t*-task-fully-sensitive and* ${P^*}_m^{(l)} = \frac{\|{m^*}_{t+1}^{(l)}\|_0}{|{\omega^*}_{t+1}^{(l)}|} \to 1$ *(a.s.), this means that the entire layer* l *is frozen when training on task* T_{t+1}*. Let* $\widetilde{\omega}_{t+1}^{*(l)}$ *be the optimal weight set for layer* l*, masked and trained on task* T_{t+1}*,* $\widetilde{\omega}_{t+1}^{*(l)} = {m^*}_{t+1}^{(l)} \odot {\omega^*}_{t+1}^{(l)}$*, Then the expected forgetting* $\widetilde{EO}_t$ *defined in*

$$\widetilde{EO}_t = \mathbb{E}_{(\mathbf{X}_t, T_t)\sim D_t} \left\{ \left| T_t \cdot \left(\ell_t(\widetilde{\omega}_{t+1}^*) - \ell_t(\omega_t^*) \right) \right| \right\}, \text{ is bounded by}$$

$$\widetilde{EO}_t \leq \frac{1}{2} \mathbb{E}_{(\mathbf{X}_t, T_t)\sim D_t} \left\{ T_t \cdot \lambda_t^{max} \left(C + \frac{C_\epsilon}{\lambda_t^{max}} \right)^2 \right\}, \; C \;\&\; C_\epsilon \text{ are constants}, \quad (18)$$

and λ_t^{max} *is the maximum eigenvalue of Hessian* $\nabla^2 \ell_t(\omega_t^*)$.

Based on the argumentation of this section, we believe the bound found in (18) can provide a supportive study in how freezing rate affects forgetting explicitly. In [18], it has been shown that lower λ_t^{max} or equivalently wider loss function L_t leads to less forgetting however, our bound in (18) is not a monotonic function of maximum eigenvalue of Hessian. Therefore we infer that when a layer has highest connectivity, freezing the entire layer and blocking it for a specific task does not necessarily control the forgetting. Our inference is not only tied to the reduction of λ_t^{max} which describes the width of a local minima [12], but we also need to rely on other hidden factors that is undiscovered for us up to this time. Although we believe that to reduce forgetting, each task should push its learning towards information preservation by protecting sensitive filters and can possibly employ the same techniques used to widen the minima to improve generalization.

3.4 Key Components to Prove Theorems

The main proofs of Theorems 1-5 are provided in supplementary materials, however in this section, we describe a set of widely used key strategies and components that are used to prove findings in Section 3.

Theorem 1 To prove that a task-sensitive layer is a task-useful layer, we use key components: **(I)** Set $\overline{\sigma}_j(s) = s.\sigma_j(s)$ where σ_j is activation function:

$$\Delta_t(f^{(l)}, f^{(l+1)}) \propto \sum_{i=1}^{M_l} \sum_{y_t \in T_t} \pi(y_t) \mathbb{E}\left[\sum_{j=1}^{M_{l+1}} \overline{\sigma}_j \left(f_i^{(l)}(\mathbf{X}_t) \right) | T_t = y_t \right]. \tag{19}$$

(II) There exist a constant C_t such that

$$\begin{aligned} & C_t \sum_{i=1}^{M_l} \sum_{y_t \in T_t} y_t \pi(y_t) \mathbb{E}_{\mathbf{X}_t | y_t} \left[f_i^{(l)}(\mathbf{X}_t) | T_t = y_t \right] \\ \geq & \sum_{i=1}^{M_l} \sum_{y_t \in T_t} \pi(y_t) \mathbb{E}\left[\sum_{j=1}^{M_{l+1}} \overline{\sigma}_j \left(f_i^{(l)}(\mathbf{X}_t) \right) | T_t = y_t \right]. \end{aligned} \tag{20}$$

Theorem 2 Let ω_t^* and $\widetilde{\omega}_t^*$ be the convergent or optimum parameters after training $F_t^{(L)}$ and $\widetilde{F}_t^{(L)}$ has been finished for task t, respectively. Here we establish three important components:

(I) Using Taylor approximation of ℓ_t around $\widetilde{\omega}_t^*$:

$$\ell_t(\omega_t^*) - \ell_t(\widetilde{\omega}_t^*) \approx \frac{1}{2} (\omega_t^* - \widetilde{\omega}_t^*)^T \nabla^2 \ell_t(\widetilde{\omega}_t^*) (\omega_t^* - \widetilde{\omega}_t^*). \tag{21}$$

(II) Let $\widetilde{\lambda}_t^{min}$ be the minimum eigenvalue of $\nabla^2 \ell_t(\widetilde{\omega}_t^*)$, we show

$$\begin{aligned} & \frac{1}{2} \mathbb{E}_{(\mathbf{X}_t, T_t) \sim D_t} \left[T_t \cdot \left((\omega_t^* - \widetilde{\omega}_t^*)^T \nabla^2 \ell_t(\widetilde{\omega}_t^*) (\omega_t^* - \widetilde{\omega}_t^*) \right) \right] \\ & \quad \geq \frac{1}{2} \mathbb{E}_{(\mathbf{X}_t, T_t) \sim D_t} \left[T_t \cdot \left(\widetilde{\lambda}_t^{min} \| \omega_t^* - \widetilde{\omega}_t^* \|^2 \right) \right]. \end{aligned} \tag{22}$$

(III) There exist a constant $C^{(l)}$ and a map function $G_l : \mathcal{L}_l \mapsto \mathcal{T}_t$ such that

$$\mathbb{E}_{(\mathbf{X}_t,T_t)\sim D_t}\left[T_t \cdot G_l \circ \sigma_t^{(l)}\big((\omega_t^* - \widetilde{\omega}_t^*)\mathbf{X}_t\big)\right] \leq C^{(l)}\,\mathbb{E}_{(\mathbf{X}_t,T_t)\sim D_t}\left[T_t \cdot G_l \circ \left|\sigma_t^{(l)}(\omega_t^*\mathbf{X}_t) - \sigma_t^{(l)}(\widetilde{\omega}_t^*\mathbf{X}_t)\right|\right]. \tag{23}$$

Theorem 4 Let $\widetilde{\omega}_{t+1}^*$ be the optimal weight after training $\widetilde{F}_{t+1}^{(L)}$ on task $t+1$. Here are the key components we need to use to prove the theorem: **(I)** we show

$$\mathbb{E}_{(\mathbf{X}_t,T_t)\sim D_t}\left\{T_t \cdot \big(\ell_t(\widetilde{\omega}_{t+1}^*) - \ell_t(\widetilde{\omega}_t^*)\big)\right\} \leq \frac{1}{2}\mathbb{E}_{(\mathbf{X}_t,T_t)\sim D_t}\big\{T_t \cdot \widetilde{\lambda}_t^{max}\|\widetilde{\omega}_{t+1}^* - \widetilde{\omega}_t^*\|^2\big\}, \tag{24}$$

(II) Let $\widetilde{w}_t'$ be the convergent or (near-) optimum parameters after training $\widetilde{F}_t^{(L)}$ and $\widetilde{\lambda}_t^{max}$ be the maximum eigenvalue of $\nabla^2\ell_t(\widetilde{\omega}_t^*)$:

$$\nabla\ell_t(\widetilde{\omega}_t') - \nabla\ell_t(\widetilde{\omega}_t^*) \approx \nabla^2\ell_t(\widetilde{\omega}_t^*)(\widetilde{\omega}_t' - \widetilde{\omega}_t^*) \leq \widetilde{\lambda}_t^{max}\|\widetilde{\omega}_t' - \widetilde{\omega}_t^*\|, \tag{25}$$

(III) If the convergence criterion is satisfied in the ϵ-neighborhood of $\widetilde{\omega}_t^*$, then

$$\|\widetilde{\omega}_{t+1}^* - \widetilde{\omega}_t^*\| \leq \frac{C_\epsilon}{\widetilde{\lambda}_t^{max}}, \qquad C_\epsilon = \max\{\epsilon, 2\sqrt{\epsilon}\}.$$

Theorem 5 Denote $\widetilde{\omega}_{t+1}^{*(l)} = {m^*}_{t+1}^{(l)} \odot {\omega^*}_{t+1}^{(l)}$ where ${m^*}_{t+1}^{(l)}$ is the binary freezing mask for layer l. For the optimal weight matrix $\widetilde{\omega}_{t+1}^*$ with mask ${m^*}_{t+1}$, define

$$\widetilde{EO}_t = \mathbb{E}_{(\mathbf{X}_t,T_t)\sim D_t}\left\{|T_t \cdot \big(\ell_t(\widetilde{\omega}_{t+1}^*) - \ell_t(\omega_t^*)\big)|\right\}.$$

(I) Once we assume that only one connection is frozen in the training process, we can use the following upper bound of the model [14]:

$$|\ell_t(\widetilde{\omega}_{t+1}^*) - \ell_t(\omega_{t+1}^*)| \leq \frac{\|{\omega^*}_{t+1}^{(l)} - \widetilde{\omega}_{t+1}^{*(l)})\|_F}{\|{\omega^*}_{t+1}^{(l)}\|_F}\prod_{j=1}^{L}\|{\omega^*}_{t+1}^{(l)}\|_F, \tag{26}$$

(II) Under the assumption ${P^*}_m^{(l)} = \frac{\|{m^*}_{t+1}^{(l)}\|_0}{|{\omega^*}^{(l)}{}_{t+1}|} \to 1$, we show

$$\widetilde{EO}_t \leq \mathbb{E}_{(\mathbf{X}_t,T_t)\sim D_t}\left\{T_t \cdot |\left(\ell_t(\omega_{t+1}^*) - \ell_t(\omega_t^*)\right)|\right\}. \tag{27}$$

4 Related Work

In recent years significant interest has been given to methods for the sequential training of a single neural network on multiple tasks. One of the primary obstacles to achieving this is catastrophic forgetting (CF), the decrease in performance observed on previously trained tasks after learning a new task. As such, overcoming CF is a primary desiderata of CL methods. Several approaches have been taken to address this problem, including various algorithms which mitigate forgetting, as well as investigation into the properties of CF itself.

Catastrophic Forgetting: The issue of catastrophic forgetting isn't new [1,17], however the popularity of deep learning methods has brought it renewed attention. Catastrophic forgetting occurs in neural networks due to the alterations of weights during the training of new tasks. This changes the network's parameters from the optimized state achieved by training on the previous task. Recent works have aimed to better understand the causes and behavior of forgetting [22,6], as well as to learn how the specific tasks being trained influence it and to empirically study its effects [9,19]. Such theoretical research into CF provides solutions to mitigate catastrophic forgetting beyond the design of the algorithm. Similarly, our investigation into the relationship between information flow and CF provides a useful tool for reducing forgetting independent of a specific algorithm.

Continual Learning: Several methods have been applied to the problem of CL. These generally fall into four categories: Regularization [13,32], Pruning-Based [16,28,26], Replay [25,29], and Dynamic Architecture approaches [23,31]. Regularization approaches attempt to reduce the amount of forgetting by implementing a regularization term on previously optimized weights based on their importance for performance. Replay methods instead store or generate samples of past tasks in order to limit forgetting when training for a new task. Dynamic architectures expand the network to accommodate new tasks. Lastly, Pruning-based methods aim to freeze the most important partition of weights in the network for a given task before pruning any unfrozen weights.

While pruning-based methods are able to remove forgetting by freezing and masking weights, they are often implemented to make simple pruning decisions, either using fixed pruning percents for the full network or relying on magnitude-based pruning instead of approaches which utilize available structural information of the network. Other recent works have demonstrated the importance of structured pruning [2,8], suggesting that pruning-based CL methods would benefit from taking advantage of measures of information such as connectivity. While these methods commonly use fixed pruning percentages across the full network, some works outside of the domain of CL investigate different strategies for selecting layer-wise pruning percents, and together they demonstrate the importance of a less homogeneous approach to pruning [14,24].

5 Experimental Evidence

To evaluate the influence of considering knowledge of information flow when training on sequential tasks, we perform multiple experiments demonstrating improved performance when reducing pruning on the task-sensitive layers defined in 1. The experimental results section is divided into two main parts aligning with the overall goal of analyzing downstream information across layers. The first part discusses the performance of CL in the context of protecting highly task-sensitive layers during pruning when adding multiple tasks in a single neural network as in [16]. The second part focuses on the connectivity across layers given tasks and how connectivity varies across the layers and between tasks.

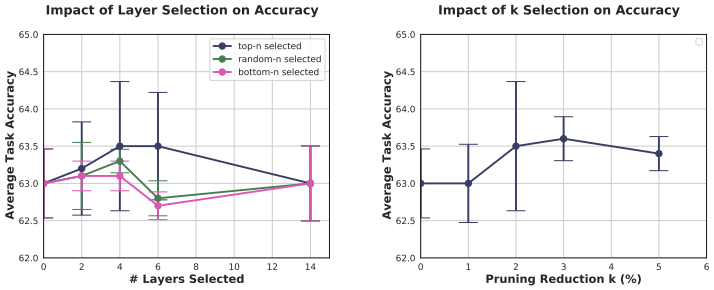

Fig. 1: The average accuracy across tasks is reported for varying values of n when $k = 2\%$(left) and k when $n = 4$(right), where n is the number of layers selected for reduced pruning and k is the hyper-parameter dictating how much the pruning on selected layers is reduced by. We compare the performance when the n layers are selected as the most (top-n), least (bottom-n), or randomly chosen (random-n) connected layers.

Setting: We carry out training with a VGG16 model on a split CIFAR-10/100 dataset, where task 1 is CIFAR-10 and tasks 2-6 are each 20 consecutive classes of CIFAR-100. We perform experiments on the Permuted MNIST dataset to determine how the characteristics of information flow differ between datasets (supporting experiments on MNIST are included in the supplementary materials). Three trials were run per experiment. After training on a given task T_t, and prior to pruning, we calculate $\Delta_t(f^{(l)}, f^{(l+1)})$ between each adjacent pair of convolutional or linear layers as in 1. Connectivity figures are plotted by layer index, which includes all VGG16 layers (ReLu, pooling, conv2D, etc), however only trainable layers are plotted. As a baseline we prune 80% of the lowest-magnitude, unfrozen weights in each layer (freezing the remaining 20%).

5.1 How Do Task Sensitive Layers Affect Performance?

Top-Connectivity Layer Freezing: For this experiment we select the n layers with the highest value of Δ_t and prune $k\%$ fewer weights in those layers for Task T_t, where both n and k are hyper-parameters. This reduction is determined individually for each task, and only applies to the given task. By reducing pruning on the most task-sensitive layer, information flow through the network is better maintained, preserving performance on the current task. This is demonstrated in Fig. 1, in which selecting the most connected layers for reduced pruning outperforms selecting the least connected or random layers. Although n and k have the same values in each case, by selecting the top-n layers we better maintain the flow of information by avoiding pruning highly-connected weights. By taking values of $n > 1$, we can account for cases where reducing the pruning on a single layer doesn't sufficiently maintain the flow of information above the baseline. Fig. 1 also shows that the performance increase for pruning the top-n connected layers varies depending on the reduction in pruning k.

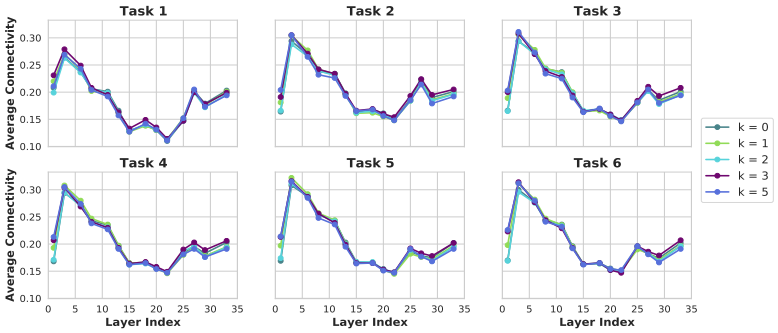

Fig. 2: For each layer the average connectivity value with the subsequent layer is reported. The connectivities are plotted for each task in CIFAR-10/100, for various k when $n = 4$ most connected layers are selected for reduced pruning.

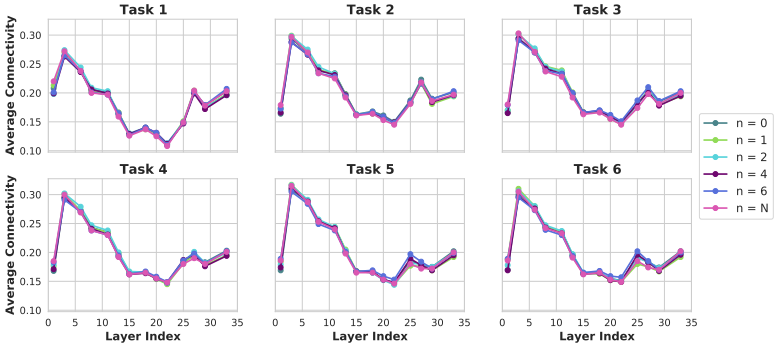

Fig. 3: The average connectivities across layers with the subsequent layer is reported. The scores are plotted for each task in CIFAR-10/100, when the n most-connected layers are selected to have their pruning percent reduced by $k = 2\%$.

Connectivity Analysis: To better characterize our measure of information flow and determine which layers are most task-sensitive, we plot the values of Δ_t for each convolutional or linear layer, as in Figs. 2, 3, and 4. These figures show how connectivity varies over several experimental setups as we change n and k during the freezing of the top-n connected layers. We compare these trends to those seen when performing the baseline $(n, k = 0)$ on Permuted MNIST.

6 Discussion

In-depth Analysis of Bounds: The bound established in Theorem 3 shows that the performance gap between original and adapted networks with task-specific

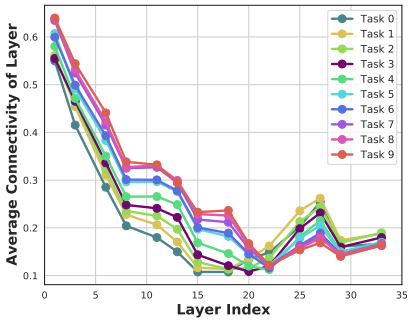

Fig. 4: The average connectivity for each layer is reported for training on Permuted MNIST. Training was done with the baseline setting when $n, k = 0$. Each of the 10 tasks in the Permuted MNIST dataset are plotted.

frozen layers grows as the layers contribute more in passing the information to the next layer given tasks. This gap has a direct relationship with the activations' lipschitz property and the minimum eigenvalues of the Hessian at optimal weights for the pruned network. From the forgetting bound in Theorem 4, we infer that as a layer is more useful for a task then freezing it reduces the forgetting more. In addition from Theorem 5, we establish that the average forgetting is a non-linear function of width of a local minima and when the entire filters of a fully sensitive layer is frozen the forgetting tends to a tighter bound.

Information Flow: The connectivities plotted in Figs. 2, 3, and 4 display patterns which remain generally consistent for a given dataset, but have noticeable differences between each dataset. For Figs. 2 and 3 tasks 2-6, which correspond to CIFAR-100, show larger connectivities across most of the network compared to CIFAR-10, particularly in the early and middle layers. Meanwhile, for MNIST we observe connectivities which are much different from those of CIFAR-10 and CIFAR-100. These observations suggest that when applied to different datasets, the task sensitivity of the layers in a network (VGG16 in this case) differ, indicating that the optimal freezing masks and pruning decisions differ as well. Further, Fig. 4 prominently shows that as subsequent tasks are trained, the connectivity of the early layers increases while the later layers' connectivity values decrease. This can also be seen to a lesser extent in Figs. 2 and 3, where the peak in the last four layers decreases, while the first three layers take larger values for later tasks. This indicates that not only is the data important for determining which layers are task-sensitive, but the position of a given task in the training order is as well. For the data shown here, we would suspect the optimal freezing mask to more readily freeze the earlier, more highly connected layers, in the network.

Top-n Layer Freezing: The selection of the most connected layers in Fig. 1 demonstrated an improvement over the baseline, least connected, or randomly chosen layers, showing that the improved performance isn't simply due to freezing more weights. While the performance improves for top-n freezing, the standard deviation also noticeably increases and overlaps the bottom-n results. This was observed for all top-n experiments, and may be linked to the observations in Figs. 2 and 3 that the top-n connected layers are found at the beginning of the network, as perhaps repeated freezing of early layers has a more destabilizing effect. While further work is needed to see if these results can be further improved upon, these observations lend support to the idea that making pruning decisions by utilizing knowledge of information flow in the network is an available tool to retain performance in pruning-based continual learning applications.

7 Conclusion

We've theoretically established a relationship between information flow and catastrophic forgetting and introduced new bounds on the expected forgetting. We've shown empirically how the information flow (measured by the connectivity between layers) varies between the layers of a network, as well as between tasks. Looking ahead these results highlight future possible directions of research in investigating differences in connectivity trends between various datasets, using a probabilistic connectivity measure like mutual information, and investigation on which portions of a network would be most important for passing information.

Finally, we have also empirically demonstrated that utilizing the knowledge of information flow when implementing a pruning-based CL method can improve overall performance. While these core experiments would benefit from further supporting investigations, such as the effects of different networks or tuning hyper-parameters beyond n and k, the reported results nonetheless show promising support for the utility of information flow. Here we limited our investigation to using connectivity when determining the extent of pruning/freezing within a layer, however it would be of significant interest to see possible applications in determining which weights are pruned (as an alternative to magnitude-based pruning), or even the use of information flow in CL methods which don't utilize pruning. These are left as a very interesting future work.

While this paper uses common CL datasets for validation of our theoretical work and focuses on pruning-based methods, applying the methods to a number of larger/more complex datasets will be the focus of more empirical future work, and may help further assess our method's capabilities as well as whether or not the connectivity trends seen here also reflect other, more complex datasets. The core theory and measure of information flow are independent of the scale of the data, so the method is expected to still work with larger datasets.

Acknowledgements: This work has been partially supported by NSF 2053480 and NSF 5409260; the findings are those of the authors only and do not represent any position of these funding bodies.

References

1. Ans, B., Rousset, S.: Neural networks with a self-refreshing memory: knowledge transfer in sequential learning tasks without catastrophic forgetting. Connection science **12**(1), 1–19 (2000)
2. Chen, T., Zhang, Z., Liu, S., Chang, S., Wang, Z.: Long live the lottery: The existence of winning tickets in lifelong learning. In: International Conference on Learning Representations (2020)
3. Chen, Z., Liu, B.: Lifelong machine learning. Synthesis Lectures on Artificial Intelligence and Machine Learning **12**(3), 1–207 (2018)
4. Cover, T., Thomas, J.A.: Elements of information theory. 1st edn. John Wiley & Sons, Chichester (1991)
5. Csiszár, I., Shields, P.C.: Information theory and statistics: A tutorial. J. Royal Statist. Soc. Ser. B (Methodology.) (2004)
6. Doan, T., Bennani, M.A., Mazoure, B., Rabusseau, G., Alquier, P.: A theoretical analysis of catastrophic forgetting through the ntk overlap matrix. In: International Conference on Artificial Intelligence and Statistics. pp. 1072–1080. PMLR (2021)
7. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International conference on machine learning. pp. 1126–1135. PMLR (2017)
8. Golkar, S., Kagan, M., Cho, K.: Continual learning via neural pruning. arXiv preprint arXiv:1903.04476 (2019)
9. Goodfellow, I.J., Mirza, M., Xiao, D., Courville, A., Bengio, Y.: An empirical investigation of catastrophic forgetting in gradient-based neural networks. arXiv preprint arXiv:1312.6211 (2013)
10. Jung, S., Ahn, H., Cha, S., Moon, T.: Continual learning with node-importance based adaptive group sparse regularization. Advances in Neural Information Processing Systems **33**, 3647–3658 (2020)
11. Ke, Z., Liu, B., Huang, X.: Continual learning of a mixed sequence of similar and dissimilar tasks. Advances in Neural Information Processing Systems **33**, 18493–18504 (2020)
12. Keskar, N.S., Mudigere, D., Nocedal, J., Smelyanskiy, M., Tang, P.T.P.: On large-batch training for deep learning: Generalization gap and sharp minima. arXiv preprint arXiv:1609.04836 (2016)
13. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences **114**(13), 3521–3526 (2017)
14. Lee, J., Park, S., Mo, S., Ahn, S., Shin, J.: Layer-adaptive sparsity for the magnitude-based pruning. In: International Conference on Learning Representations (2020)
15. Li, Z., Hoiem, D.: Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence **40**(12), 2935–2947 (2017)
16. Mallya, A., Lazebnik, S.: Packnet: Adding multiple tasks to a single network by iterative pruning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 7765–7773 (2018)
17. McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: The sequential learning problem. In: Psychology of learning and motivation, vol. 24, pp. 109–165. Elsevier (1989)

18. Mirzadeh, S.I., Farajtabar, M., Pascanu, R., Ghasemzadeh, H.: Understanding the role of training regimes in continual learning. Advances in Neural Information Processing Systems **33**, 7308–7320 (2020)
19. Nguyen, C.V., Achille, A., Lam, M., Hassner, T., Mahadevan, V., Soatto, S.: Toward understanding catastrophic forgetting in continual learning. arXiv preprint arXiv:1908.01091 (2019)
20. Parisi, G.I., Kemker, R., Part, J.L., Kanan, C., Wermter, S.: Continual lifelong learning with neural networks: A review. Neural Networks **113**, 54–71 (2019)
21. Raghavan, K., Balaprakash, P.: Formalizing the generalization-forgetting trade-off in continual learning. Advances in Neural Information Processing Systems **34** (2021)
22. Ramasesh, V.V., Dyer, E., Raghu, M.: Anatomy of catastrophic forgetting: Hidden representations and task semantics. In: International Conference on Learning Representations (2020)
23. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. arXiv preprint arXiv:1606.04671 (2016)
24. Saha, G., Garg, I., Ankit, A., Roy, K.: Space: Structured compression and sharing of representational space for continual learning. IEEE Access **9**, 150480–150494 (2021)
25. Shin, H., Lee, J.K., Kim, J., Kim, J.: Continual learning with deep generative replay. Advances in neural information processing systems **30** (2017)
26. Sokar, G., Mocanu, D.C., Pechenizkiy, M.: Spacenet: Make free space for continual learning. Neurocomputing **439**, 1–11 (2021)
27. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. Advances in neural information processing systems **29** (2016)
28. Wang, Z., Jian, T., Chowdhury, K., Wang, Y., Dy, J., Ioannidis, S.: Learn-prune-share for lifelong learning. In: 2020 IEEE International Conference on Data Mining (ICDM). pp. 641–650. IEEE (2020)
29. Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., Zhang, Z., Fu, Y.: Incremental classifier learning with generative adversarial networks. arXiv preprint arXiv:1802.00853 (2018)
30. Yin, D., Farajtabar, M., Li, A., Levine, N., Mott, A.: Optimization and generalization of regularization-based continual learning: a loss approximation viewpoint. arXiv preprint arXiv:2006.10974 (2020)
31. Yoon, J., Yang, E., Lee, J., Hwang, S.J.: Lifelong learning with dynamically expandable networks. In: International Conference on Learning Representations (2018)
32. Zenke, F., Poole, B., Ganguli, S.: Continual learning through synaptic intelligence. In: International Conference on Machine Learning. pp. 3987–3995. PMLR (2017)