Tackling More Lifelong Learning Properties with Dynamic Weights Consolidation and Sparse Sub-networks

Anonymous Authors[1]

Abstract

Humans are the best example of agents that can learn a variety of skills incrementally over the course of their lives, and imbuing machines with this skill is the goal of lifelong machine learning. Ideally, lifelong learning should avoid the stumbling blocks of catastrophic forgetting, inability to learn new tasks (known as intransigence), and confusion, while achieving the key properties of backward transfer, forward transfer, and the capacity for few-shot learning. In previous approaches, the focus has been given to subsets of these properties, often by fitting together with an array of separate mechanisms. In this paper, we propose a simple and powerful approach that can demonstrate each of the desired properties. Our novel approach defines a small number of weight consolidation parameters that work in tandem with efficiently located sparse neural sub-networks based on the "Lottery Tickets Hypothesis".

1. Introduction

Humans have a sustained ability to acquire knowledge and skills, refine them on the basis of novel experiences, and transfer them across multiple domains with few new samples over a lifespan (Calvert, 2004; Bremner, 2012; Barnett & Ceci, 2002). It is no surprise that the learning abilities of humans have been inspiring machine learning approaches for decades. Lifelong learning (Thrun., 1995) has been proposed to explore the capacity for machine intelligence to learn sequential tasks by accommodating new tasks while retaining previously learned experiences. Although neural networks and deep-learning algorithms have been very successful at learning in the single-task scenario (Yann LeCun & Burges., 1998; Alex Krizhevsky & Hinton., 2012), a lifelong learning approach has remained a long-standing challenge (Hassabis & Botvinick, 2017; Thrun & Mitchell, 1995), with primary difficulties abstractly visualized in Figure 1.

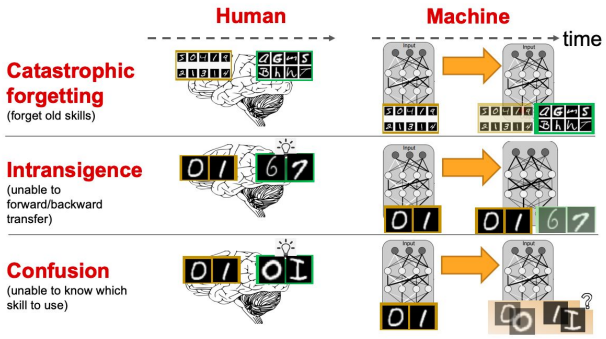

Figure 1. Illustrations of how key lifelong learning issues manifest in machine learning approaches.

1.1. Avoid Catastrophic Forgetting

A key property of successful lifelong learning is learning without succumbing to catastrophic forgetting (McCloskey & Cohen, 1989), the forgetting previous skills upon learning new ones. As pictured in Figure 1, humans generally avoid catastrophic forgetting to learning two separate skills of recognizing digits and letters. In contrast, if a neural network first learns to classify digits, subsequently learning to recognize letters will weaken skills associated with digits. By trying to fit a new task data distribution, drifting of the network weights away from those tuned for the old task occurs. Overcoming catastrophic forgetting indicates non-negative *backward transfer*. Positive backward transfer indicates the power to improve the previous task's predictive accuracy after learning the related new tasks. In this paper, we group both catastrophic forgetting and positive backward transfer under the term backward transfer.

1.2. Avoid Intransigence

Intransigence reflects the inability of a model to learn new tasks (Chaudhry et al., 2018a), and overcoming this indi-

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

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

cates non-negative *forward transfer*, reflecting the ability to more efficiently adapt to new complex tasks following simpler tasks. In this paper, we group both intransigence and positive forward transfer under the term forward transfer. For example, a child may require many examples of the digits *0* and *1* before properly learning to distinguish between them, but after doing so, there is decreased difficulty in learning to recognize more visually complex digits like *6* or *7* (illustrated in figure 1's intransigence part). In contrast, after learning to recognize one digit, the weights of a neural network may be ill-suited for learning to recognize new digits and fewer weights are available for learning new digits without interfering with previous digits. Ideally, a lifelong learning approach should be able to leverage previous skills learned to accelerate the learning of new skills.

1.3. Avoiding Confusion

In a lifelong learning circumstance, another difficulty that may occur is inter-task confusion. When a human learns to distinguish between the digits *0* and *1*, and then separately learns to distinguish between the letters *o* and *I*, the difference between the characters is usually sufficient to allow for distinguishing between all four characters at once. In contrast, after engaging in a similar training setup, a neural network may consistently mis-classify *o* and *I* for *0* and *1* simply because they were learned first (Hou et al., 2019), or it may confuse the visually similar classes because the features necessary to distinguish them were not necessary in the original tasks. Ideally, a lifelong learning approach should be able to distinguish between classes learned in different tasks. However, so far, little work has been done to solve the inter-task confusion in a lifelong learning approach.

While a lifelong learning approach should address all of the aforementioned phenomena, previous approaches deal with subsets of these properties or require combining several mechanisms. For example, regularization-based approaches (Kirkpatrick et al., 2016; Zenke et al., 2017; Maltoni & Lomonaco, 2018) alleviate catastrophic forgetting by restraining the network weights drifting from old tasks to a new one. Moreover, (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2018b) propose the episodic memory-based approaches to prevents catastrophic forgetting and achieve positive backward-transfer. Dynamic architecture approaches (Yoon et al., 2018; Rusu et al., 2016; Rebuffi et al., 2016) are capable of addressing intransigence and forgetting, but at the expense of increasing computational requirements.

In this paper, we propose a unified approach to bring task-incremental lifelong learning closer to a more realistic human learning paradigm by addressing these known issues. Abstractly, our approach works by identifying the sub-networks critical for performance on each learned task (inspired by the lottery ticket hypothesis (Frankle & Carbin, 2018)) and intelligently using this information to inform weight consolidation policies. By placing high consolidation strength on identified sub-networks and considering larger sub-networks, we can avoid catastrophic forgetting. Applying weaker consolidation to smaller sub-networks and ensuring that certain weights tangential to the sub-networks are free to be modified allows for avoiding intransigence. Additionally, we can control the consolidation of sub-network weights during rehearsal in order to avoid confusion and increase backward transfer.

Our primary contributions of this work are thus:

- We describe a lifelong learning approach which works by identifying sub-networks critical to individual tasks in order to intelligently controlling weight consolidation.
- We demonstrate that by changing a small number of hyperparameters, our approach can achieve forward and backward transfer as well as assist in decreasing confusion during rehearsal.

2. Our Unified Approach

Our lifelong learning approach is designed to be able to exhibit several important abilities by leveraging the same underlying mechanism. In this section, we first describe the primary component of the model and then discuss how it functions with respect to backward transfer, forward transfer, and confusion.

2.1. Identifying Critical Sub-networks (Lottery Tickets)

Central to our approach is the ability to identify a critical sub-network associated with each skill learned by a neural network, where each skill is associated with a single output dimension (although the technique can generalize to multiple output dimensions per skill). We refer to these sub-networks as lottery tickets (LTs), taking inspiration from the lottery ticket hypothesis (Frankle & Carbin, 2018). The visual intuition behind our approach to identifying LTs is provided in Figure 2 – where the final LT weights for class 0 are shown with solid edges in (e). Our particular approach is designed so that even when very sparse sub-networks are chosen, our method guarantees the presence of a full path (from input to output nodes) and it is not easy for non-LT weights to influence the output of the corresponding LT class.

To identify the weights making up a lottery ticket, we first obtain the rescaled magnitudes of all weights such that for each node, n, the maximum weight magnitude pointing to it is 1:

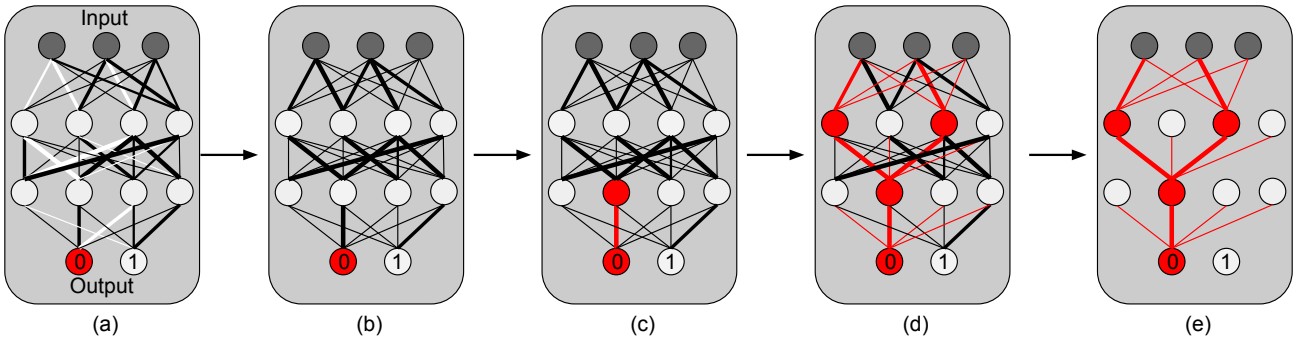

Figure 2. Lottery ticket sub-network identification. From (a) to (b): take absolute value of weight, normalize input weights per node, and filter weights by threshold. Weights below threshold are not shown in (b). From (b) to (c): $N(depth-1)$ consists of the red node in the penultimate layer. From (c) to (d): the remaining important LT nodes are identified by following the weights over the threshold. From (d) to (e), all LT weights are displayed in solid red edges. LT backward-incidental weights are shown with thin solid edges and LT forward-incidental weights are shown with dashed edges. Central LT weights are shown with thick red edges. LT nodes are red.

$$in(n) := \frac{|in(n)|}{max(|in(n)|)}, \tag{1}$$

where $in(n)$ is the set of weights pointing into a node. Next, we recursively identify the LT *nodes* in a network for class index c:

$$\begin{aligned} N_c(depth) &= \{c\} \\ N_c(l) &= \{n \in layer(l) | \\ &\quad (\exists w | \, (w \in out(n) \cap in(N(l+1)) \wedge w > \tau))\} \end{aligned} \tag{2}$$

where $N(l)$ is the set of important nodes for layer l, $out(\cdot)$ and $in(\cdot)$ are correspondingly the set of weights pointing out of and into of any nodes in the argument, and $\tau \in [0, 1]$ is the pruning threshold that defines how sparse the selected sub-networks are. The base case of $N_c(depth) = \{c\}$ is seen in Figure 2 as the highlighted output node for which the LT is identified. Given N, the set of LT weights for class c is finally defined by:

$$LT(c) = \{w \in layers | w \in in(N_c)\} \tag{3}$$

For the biases of a layer, the same condition is used. After the LT for a class has been identified, the set of weight locations responsible are stored in memory.

Given an LT, we define three specific types of weights: central LT weights, backward-incidental weights, and forward-incidental weights. Central LT weights are weights feeding into LT nodes which have normalized magnitudes $> \tau$, backward-incidental weights are those feeding into LT nodes which have normalized magnitudes $< \tau$, and forward-incidental weights are those leading out of LT nodes into non-LT nodes.

2.2. Backward Transfer

As LTs are critical for a neural networks performance on a task, it follows that by preventing the weights of a LT from changing, the performance on the corresponding task can been maintained. We manifest this insight with a quadratic loss during training to prevent LT weights from drifting too far from the target values and use the inclusion of a weight in a LT to determine the consolidation strength during training. Our training loss while training on task t is thus:

$$L(w) = L_t(w) + \sum_i consolidation(w_i)(w_i - w_{ti})^2, \tag{4}$$

where $consolidation(w_i)$ specifies the consolidation strength for an individual weight, and is defined by:

$$consolidation(w) = \begin{cases} b_{nf} & \text{if } w \in LT(C) \\ b_{fw} & \text{if } w \in out(N(C)) \\ 0 & otherwise \end{cases} \tag{5}$$

where C is the set of classes that have already undergone training. In other words, b_{nf} is applied to central and backward-incidental LT weights, and $b_f w$ is applied to forward-incidental weights.

Given a fixed pruning value, τ, modulating b_{nf} allows us to control how strongly past skills are maintained. Modulating b_{fw} however, has a larger effect on intransigence and forward transfer, which we discuss next.

2.3. Forward Transfer

The inability for a network to learn new skills, known as intransigence, will inevitably occur in a network of fixed

capacity which is required to learn a large amount of tasks. In the context of weight consolidation, intransigence may occur if too many weights are used in maintaining old skills, leaving an insufficient number of weights to modify to learn new skills. For this reason, there is a balance between avoiding forgetting and avoiding intransigence. To control this balance in favor of forward transfer with out LT consolidation technique, we have the following mechanisms.

Intelligent consolidation. By ensuring the availability of weights that can be tuned for a new task, intransigence can be reduced. In our approach, this suggests that intransigence can be minimized by decreasing the consolidation strength of existing LTs (with b_{nf}) as well as increasing the LT freezing selectivity (τ) so that fewer weights are frozen per skill. Additionally, we hypothesize that allowing forward-incidental LT weights to be modified for the new task is critical for forward transfer, as these are the weights are used to transfer existing skills in the form of useful features captured by individual network nodes.

Selective forgetting. Due to the ability to track LTs for each learned skill (class), our approach is capable of performing selective forgetting of skills in order to free weights for new memories. With our approach, forgetting a skill simply corresponds to setting $b_{nf} = 0$ for those weights which are only used by the skill being forgotten.

While both intransigence and forgetting can be observed in the *multi-head* evaluation scenario (at test time, the tasks indices are provided. Only accordant task's outputs will be checked and evaluated), realistically deploying a lifelong learning model in the real world may be better modelled with *single-head* evaluation (at test time, the task index is not known. Other task's output may also has a higher prediction given a task), allowing classes from different tasks to become confused. Next, we discuss how this problem is addressed by our model.

2.4. Confusion

It is possible for a neural network to have both very good positive backward and forward transfer in the multi-head scenario, but still perform poorly in the more realistic single-head scenario. This is a result of the introduction of inter-task class confusion. During training, if a model is only presented with the data from a subset of classes at a time, the model learns to differentiate between that set of classes, but does not learn to differentiate between those new classes and previously seen ones. For example, in Figure 3, we can see that although a model learns to differentiate between class 0a and 1a and classes 0b and 1b very well, due to the similarities between classes, confusion occurs between 0a/0b and 1a/1b. To produce this figure, we use a *2-fractured Split-MNIST task* described in the next section. For now, it is sufficient to known that when two classes are labelled by the same number but different letter, they are visually similar. For further experimental setup information behind this figure, refer to confusion (A) in Table 1.

	0a	1a	2a	3a	4a	5a	6a	7a	8a	9a	0b	1b	2b	3b	4b	5b	6b	7b	8b	9b
0a	0.77	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1a	0.00	0.91	0.00	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00
2a	0.14	0.01	0.15	0.00	0.03	0.02	0.17	0.00	0.01	0.00	0.21	0.01	0.12	0.00	0.02	0.01	0.02	0.01	0.08	0.00
3a	0.04	0.11	0.00	0.24	0.01	0.08	0.04	0.04	0.08	0.00	0.07	0.01	0.05	0.01	0.00	0.13	0.01	0.00	0.08	0.00
4a	0.04	0.20	0.05	0.02	0.19	0.00	0.06	0.05	0.03	0.02	0.07	0.02	0.09	0.00	0.01	0.05	0.01	0.02	0.04	0.02
5a	0.25	0.01	0.01	0.09	0.00	0.22	0.03	0.00	0.01	0.01	0.20	0.00	0.01	0.02	0.00	0.02	0.02	0.00	0.09	0.00
6a	0.24	0.00	0.09	0.00	0.00	0.02	0.26	0.00	0.00	0.01	0.23	0.00	0.04	0.00	0.02	0.00	0.03	0.00	0.04	0.00
7a	0.17	0.06	0.01	0.03	0.01	0.00	0.00	0.26	0.00	0.08	0.14	0.00	0.05	0.01	0.00	0.01	0.00	0.09	0.06	0.01
8a	0.02	0.21	0.01	0.06	0.04	0.01	0.06	0.05	0.14	0.00	0.05	0.02	0.08	0.00	0.00	0.16	0.01	0.01	0.08	0.01
9a	0.23	0.02	0.04	0.04	0.01	0.01	0.04	0.03	0.00	0.20	0.16	0.00	0.03	0.01	0.07	0.00	0.01	0.02	0.07	0.01
0b	0.56	0.00	0.02	0.01	0.00	0.02	0.03	0.00	0.00	0.00	0.29	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.02	0.00
1b	0.00	0.56	0.00	0.00	0.06	0.00	0.02	0.03	0.08	0.00	0.00	0.12	0.03	0.00	0.00	0.08	0.00	0.00	0.00	0.00
2b	0.03	0.17	0.03	0.04	0.03	0.01	0.12	0.01	0.05	0.01	0.08	0.04	0.15	0.00	0.00	0.09	0.03	0.01	0.09	0.01
3b	0.04	0.11	0.00	0.29	0.00	0.14	0.01	0.04	0.02	0.01	0.09	0.00	0.02	0.05	0.00	0.05	0.01	0.00	0.10	0.00
4b	0.36	0.01	0.09	0.01	0.01	0.00	0.03	0.02	0.00	0.11	0.15	0.00	0.03	0.00	0.10	0.00	0.00	0.02	0.04	0.01
5b	0.04	0.13	0.01	0.08	0.04	0.02	0.04	0.06	0.09	0.00	0.07	0.00	0.08	0.01	0.00	0.24	0.01	0.02	0.05	0.01
6b	0.04	0.21	0.02	0.02	0.03	0.01	0.15	0.01	0.04	0.00	0.14	0.01	0.10	0.00	0.01	0.08	0.10	0.00	0.05	0.00
7b	0.02	0.27	0.01	0.03	0.05	0.00	0.02	0.15	0.03	0.01	0.05	0.02	0.10	0.00	0.00	0.10	0.00	0.10	0.02	0.03
8b	0.10	0.13	0.01	0.13	0.01	0.03	0.05	0.05	0.03	0.01	0.12	0.01	0.06	0.01	0.00	0.07	0.01	0.02	0.17	0.00
9b	0.02	0.29	0.01	0.06	0.10	0.00	0.03	0.11	0.03	0.02	0.05	0.03	0.07	0.00	0.00	0.07	0.00	0.03	0.04	0.04

Figure 3. The confusion matrix from single-head evaluation on the 2-fractured Split-MNIST task without rehearsal (which consists of 10 pairs of classes which are visually similar). The labels for each row correspond to the true class labels, and the labels for each column are for the predicted labels. We can see that there is a strong bias for classes to be mis-classified as belonging to the first task (0a and 1a). Classes are also commonly confused for their visually similar classes (e.g. 3b confused for 3a or 2a confused for 2b).

One way to combat confusion is with *rehearsal* (also known as experience replay). At a high level, rehearsal works with our approach in the following way: when new task data is presented to the model, it is all used for training, applying the loss function to only the current outputs. Next, samples from each new class are chosen randomly and stored in memory. Finally, rehearsal is performed by training the network on stored samples from all seen classes. We use a fixed size total memory capacity, M, so the number of samples stored per class after seeing the set of classes C is up to $M/|C|$. When the set of classes, $C_r \subset C$ is being rehearsed, up to $R/|C_r|$ samples per class are used for training, where R is the *rehearsal budget*. For weight consolidation during rehearsal, we use the policy described in Equation 6, which ensures that 1) the backward-incidental weights which can have a forward transfer effect on LT nodes have consolidation controlled by b_{fw}, 2) the weights which are not part of any LT are free to change, and 3) weights which are part of an LT for a non-rehearsed class can be penalized for

changing with b_{nf}.

$$consolidation(w) = \begin{cases} b_{fw} & \text{if } w \in LT(C_r) \wedge w < \tau \\ 0 & \text{if } w \notin LT \\ b_{nf} & otherwise \end{cases} \tag{6}$$

Naturally, we may expect that as more samples are able to be chosen for rehearsal and when forward transfer is able to happen during rehearsal, confusion can be reduced.

In the following section, we discuss metrics to determine how well our approach is able to express each of the properties discussed in this section and experimentally evaluate our approach.

3. Experimental Setups and Results

For each of backward transfer, forward transfer, and confusion, we discuss the metrics we will use for measuring them independently and then examine experimental results of our approach.

3.1. Backward Transfer

Metrics Catastrophic forgetting refers to the tendency for a neural network to forget old skills as new ones are presented. Given a sequence of tasks to solve, the extent of catastrophic forgetting can be measured with *backward transfer interference* (BTI) (Riemer et al., 2018), which is the average difference between the *retained accuracy* (RA) of each task and the *learned accuracy* (LA). The LA for a task is the test accuracy as measured immediately after the task is learned, while RA is the test accuracy as measured after all tasks have been learned. Intuitively, catastrophic forgetting will cause the retained accuracy for old tasks to be much smaller than the learned accuracy. When $BTI < 0$, we consider forgetting (or negative backward transfer) to have occurred. When $BTI > 0$ (old task skills improve as new ones are presented) we consider positive backward transfer to have occurred. In this work, we also introduce the *normalized backward transfer interference* score, $nBTI$:

$$nBTI = \frac{RA - LA}{1 - LA} \tag{7}$$

The $nBTI$ allows us to put models on a more equal footing when comparing backward transfer. Consider the following two models whose error rates suffer proportionally from forgetting: (1) where $LA = 0.99$ and $RA = 0.98$ ($BTI = -0.01$), and (2) where $LA = 0.9$ and $RA = 0.8$ ($BTI = -0.1$). For both models, $nBTI = -100\%$, indicating that the error rate doubles. In our corresponding backward transfer-focused experiments in the following section, BTI is calculated using multi-head evaluation, so that at test time, the task index is known to the model.

Tasks Both experiments for backward transfer make use of the Split-MNIST task sequence (Zenke et al., 2017), which consists of five tasks: 0 vs 1, 2 vs 3, etc. Task orders are shuffled across trials for these experiments to remove any confounding backward transfer effects potentially caused by particular task orders.

Experiment 1: pruning and LT consolidation Figure **??** demonstrates how τ and b_{nf} can work together to achieve positive backward transfer. As we expect, there is a clear trend for backward transfer to improve as consolidation of LT weights increases in strength. However, once τ is below 0.95, $nBTI$ does not appear to be sensitive to changes in the hyperparameter. For experimental setup, refer to backward (A) in Table 1.

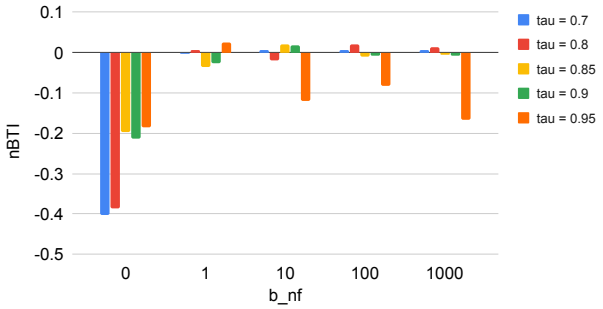

Figure 4. Backward transfer experiment 1. The effect of τ and b_{nf} on $nBTI$.

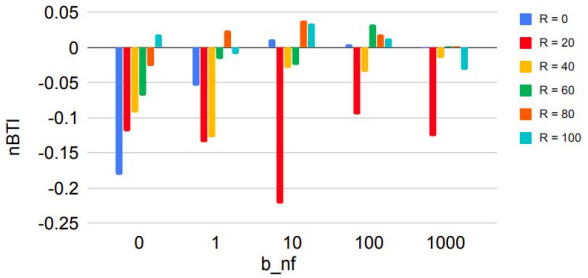

Figure 5. Backward transfer experiment 2. The effect of rehearsal budget and b_{nf} on $nBTI$ for the Split-MNIST task. Since there are 10 classes and we evaluate with 10-shot learning, a rehearsal budget of 100 corresponds rehearsing on all seen training data.

Experiment 2: rehearsal and LT consolidation To answer the question of how rehersal works with weight consolidation to influence $nBTI$, consider Figure 5. We can observe that both increased consolidation strength and increased rehearsal budget generally lead to improved $nBTI$. Interestingly, when $b_{nf} > 0$ and the rehearsal budget is very small but non-zero, the performance is worse than when

experiment	task	task shuffling	trials	tau	b_nf	b_fw	memory capacity	rehearsal budget	architecture
backward (A)	Split-MNIST	TRUE	10	[0.95, 0.9, 0.85, 0.8, 0.7]	[0, 1, 10, 100, 1000]	0	0	0	[256, 256]
backward (B)	Split-MNIST	TRUE	10	0.8	[0, 1, 10, 100, 1000]	0	100	[0, 20, 40, 60, 80, 100]	[256, 256]
forward (A)	Split-MNIST	TRUE	10	[0.95, 0.9, 0.85, 0.8, 0.7]	[0, 1, 10, 100, 1000]	[0, 0.1, 1, 10]	0	0	[256, 256]
forward (B)	Split-EMNIST	FALSE	30	0.9	1000	0	0	0	[50, 50]
forward (C)	binary EMNIST	NA	5	0.8	1000	0	0	0	[256, 256]
confusion (A)	Split-MNIST (f=2)	NA	20	0.9	1000	0	0	0	[256, 256]
confusion (B)	Split-MNIST (f=2)	TRUE	5	0.9	1000	[0, 0.1, 1]	200	[0, 1*20, 2*20, 4*20, 6*20, 8*20, 10*20]	[256, 256]

Table 1. Experimental setups for experiments in this section. For all experiments, the following hyperparameters are shared: batch size 64, 10 training epochs (also applied to rehearsal), 10-shot learning is used, and training is done with the Adam optimizer (Kingma & Ba, 2014) with default Keras (Chollet et al., 2015) settings.

there is no rehearsal. This may be due to the very noisy gradient present during rehearsal. For experimental setup, refer to backward (B) in Table 1.

3.2. Forward Transfer

Metrics Intransigence refers to the inability for a model to learn new skills. Positive forward transfer reflects the ability to learn new tasks faster (and possibly with fewer weights needed) as more tasks are learned (able to transfer knowledge to new tasks). To measure intransigence and forward transfer, we build off of FTI (forward transfer and interference) (Lopez-Paz & Ranzato, 2017), which is the difference between the LA for a task in the lifelong setting, and the LA for a task when learned independently ($LA_{\text{non LLL}}$). When $FTI < 0$, we consider intransigence (negative forward transfer) to occur, and when $FTI > 0$ we consider positive forward transfer to have occurred. However, to make the metric independent of the overall performance of the model, we use a *normalized forward transfer* metric:

$$nFTI = \frac{LA - LA_{\text{non LLL}}}{1 - LA_{\text{non LLL}}} \tag{8}$$

To provide some intuition behind this augmented forward transfer metric: when $nFTI = -1$: using the lifelong learning approach causes the learned error for each task to increase by 100% (2X worse), when $nFTI = 0$, using non-LLL is the same as LLL with regard to learned accuracy, and when $nFTI = 0.5$: learned errors rates are decreased by half.

Tasks For the first experiment, we again use Split-MNIST with task shuffling. For the second experiment, we use Split-EMNIST, which is set up the same as Split-MNIST, except uses the 62 classes from the EMNIST dataset (Cohen et al., 2017). Task shuffling is not performed for the second experiment.

Experiment 1: pruning and consolidation In Figure 6, we can observe the effect of τ and consolidation on $nFTI$. Most notably, when $b_{nf} = 0$, or τ is near 1, $nFTI$ is maximized. In contrast, this is when $nBTI$ is minimized. Additionally, $nFTI$ does prove to be sensitive to b_{fw}, especially when b_{nf} is large, suggesting that when forward-incidental weights are not free to adapt to the new task, modifying LT weights suffices.

		b_nf				
b_fw	**tau**	**0**	**1**	**10**	**100**	**1000**
0	**0.7**	-0.04	-0.28	-1.12	-2.63	-2.21
	0.8	-0.09	-0.26	-0.67	-1.84	-2.55
	0.85	-0.10	-0.22	-0.70	-1.46	-1.55
	0.9	-0.13	-0.23	-0.46	-0.83	-0.95
	0.95	-0.16	-0.08	-0.22	-0.20	-0.12
1	**0.7**	0.06	-0.41	-1.20	-2.15	-3.09
	0.8	0.00	-0.36	-1.25	-1.99	-2.10
	0.85	-0.09	-0.35	-0.87	-1.39	-2.15
	0.9	-0.15	-0.32	-0.74	-1.29	-1.41
	0.95	-0.09	-0.14	-0.50	-0.21	-0.27
10	**0.7**	-0.06	-0.32	-2.21	-3.30	-3.00
	0.8	-0.07	-0.34	-2.18	-2.92	-3.37
	0.85	-0.07	-0.33	-2.02	-2.13	-2.19
	0.9	-0.07	-0.46	-1.73	-2.50	-2.49
	0.95	-0.14	-0.42	-0.63	-1.04	-0.79

Figure 6. Forward transfer experiment 1. The effects of τ, b_{nf}, and b_{fw} on $nFTI$.

Experiment 2: selective forgetting In Figure 7, we report the per-task increase in learned accuracy as a result of selective forgetting on a challenging setup. We use a model of very limited size (two hidden layers of width 50) and a total of 62 classes (presented in 31 tasks) For more setup details, refer to forward (B) in Table 1. In this setting, the number of free weights in the model will decrease very quickly. When performing selective forgetting in this problem, we forget (i.e. unfreeze weights used only for that skill) a skill when it is 10 tasks old. Indeed we can observe that the ability to forget has an overall positive effect on the learned accuracies of tasks in this long task sequence.

3.3. Confusion

Metrics In this work, we define confusion to occur when at test time, a class is mis-classified as another that is not a member of the same task (and thus only applies to single-head evaluation, where the task index is not known to the model). To measure confusion, we introduce the $autonomy$

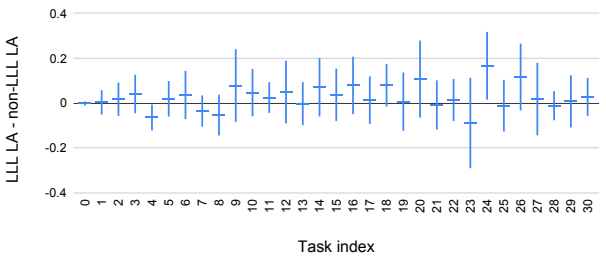

Figure 7. Forward transfer experiment 2. The increase in learned accuracy induced by selective forgetting during a long task sequence. Results are everaged across 30 trials and error bars cover one standard deviation.

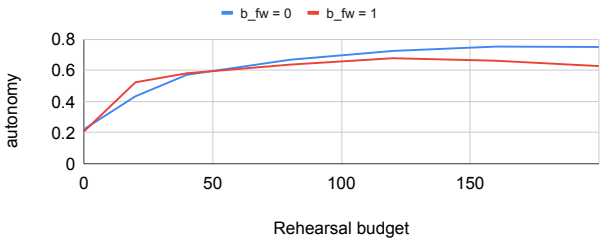

Figure 8. Confusion experiment. Effect of rehearsal and b_{fw} on $autonomy$. The effect of b_{nf} was also explored, but found not to have a significant effect.

metric to measure the presence of confusion (so-named because it reflect the ability for a model to classify samples without being told what task it is for):

$$autonomy = \frac{\text{single-head RA}}{\text{multi-head RA}} \tag{9}$$

When $autonomy = 0$, the model is fully dependent on labels, and when $autonomy = 1$, single-head accuracy is same as multi-head accuracy. **Tasks** For the following confusion experiment, we use the novel *k-fractured Split-MNIST* task sequence, which is derived from Split-MNIST in the following way: for each class, we split it into two classes by first using PCA to reduce the dimensionality of each sample to $k/2$ and then clustering these points with k-Means++ (Arthur & Vassilvitskii, 2007) into k clusters. Test samples are clustered with the principal components and cluster centers identified with the training data. The motivation behind this task sequence is that it represents an extreme case where inter-task confusion can very easily happen while the classes are designed to posess visual differences. Figure 3 shows how the tasks are ordered and how confusion occurs between classes during single-head evaluation when no rehearsal is used.

Experiment: rehearsal and consolidation With this experiment, we address the question of how rehearsal and b_{fw} work together to reduce confusion. In Figure 8, we can see that a small initial increase in rehearsal budget confers a substantial increase in $autonomy$. However, the effect of b_{fw} is less clear: at large rehearsal budgets, small values are better, but with small rehearsal budgets, b_{fw} has no obvious effect. For experimental setup, refer to confusion (B) in Table 1.

4. Related Work

In this section, we will review several representative achievements related to our approach.

Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2016) is one influential approach to addressing catastrophic forgetting and serves as inspiration for our work. This approach consists of a quadratic penalty during training on the difference between the network parameters for the old and the new tasks. Weight consolidation importance is computed offline with the diagonal of the Fisher information metric. A similar method, SI (Zenke et al., 2017) proposes an algorithm that penalizes changes to the most relevant synapses, where per-parameter regulation strengths can be efficiently calculated during the training period. While these approaches mainly focus on avoiding catastrophic forgetting, our approach is also able to consider forward transfer and confusion with some success.

Appearing prominently in our work, the notions of forward transfer and backward transfer were explored by the authors of (Lopez-Paz & Ranzato, 2017) with a Gradient Episodic Memory (GEM) approach, which stores a subset of the observed examples to improve backward transfer. An optimized approach, A-GEM (Chaudhry et al., 2018b), was proposed to achieve a better trade-off between forgetting and transferring with few-shots data while maintaining a smaller memory footprint. While A-GEM also considers both backward and forward transfer (although with a learning-curve based forward transfer metric), we do so with an approach not dependent on rehearsal.

Another set of approaches focus on architecture expansion in order to solve continually introduced tasks (Rusu et al., 2016; Yoon et al., 2018). This family of approaches aim to work around intransigence caused by reduction of free network capacity as more skills are learned. Inevitably, this results in increasing computational load and requires retraining to properly utilize the modified architecture. Comparatively, by using smaller LTs or through selective forgetting, our approach can control the rate at which the representational capacity of the neural network is used up, so that longer task sequences may be accommodated.

5. Conclusions

In this work, we presented an approach for few-shot task-incremental lifelong learning. This approach tackles a difficult problem that captures many important aspects of human learning. Progress in this area is critical for the development of computationally efficient and flexible machine learning algorithms. The success at this problem reduces the demand for training data while a single model learns to solve more and more tasks.

Our new approach works in a unified way: a single mechanism is employed to address the several considered lifelong learning properties. In particular, the approach performs weight consolidation guided by identified critical sub-networks – which we refer to as LTs. To achieve positive backward transfer, setting large consolidation values for LT and LT-forward-incidental weights is beneficial, as well as selecting larger LTs. To achieve positive forward transfer, selective small LTs and using low consolidation values is beneficial. Finally, to avoid inter-task confusion, we can decrease the consolidation associated with all skills during single-head rehearsal.

References

Alex Krizhevsky, I. S. and Hinton., G. E. Imagenet classification with deep convolutional neural networks. *NIPS*, pp. 1097–1105, 2012.

Arthur, D. and Vassilvitskii, S. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035. Society for Industrial and Applied Mathematics, 2007.

Barnett, S. and Ceci, S. When and where do we apply what we learn? a taxonomy for fartransfer. *Psychological Bulletin128*, pp. 612–637, 2002.

Bremner, A., L. D. . S. C. Multisensory development. *Oxford University Press*, 2012.

Calvert, G., S. C. . S. B. The handbook of multisensory processes. *Cambridge,MA: The MIT Press*, 2004.

Chaudhry, A., Dokania, P. K., Ajanthan, T., and Torr, P. H. S. Riemannian walk for incremental learning: Understanding forgetting and intransigence. *CoRR*, abs/1801.10112, 2018a. URL http://dblp.uni-trier.de/db/journals/corr/corr1801.html#abs-1801-10112.

Chaudhry, A., Ranzato, M., Rohrbach, M., and Elhoseiny, M. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018b.

Chollet, F. et al. Keras, 2015.

Cohen, G., Afshar, S., Tapson, J., and van Schaik, A. Emnist: an extension of mnist to handwritten letters. *CoRR*, abs/1702.05373, 2017. URL http://dblp.uni-trier.de/db/journals/corr/corr1702.html#CohenATS17.

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

Hassabis, D., K. D. S. C. and Botvinick, M. Neuroscience-inspired artifi-cial intelligence. *Neuron Review95(2)*, pp. 245–258, 2017.

Hou, S., Pan, X., Loy, C. C., Wang, Z., and Lin, D. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 831–839, 2019.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. Overcoming catastrophic forgetting in neural networks, 2016. URL http://arxiv.org/abs/1612.00796. cite arxiv:1612.00796.

Lopez-Paz, D. and Ranzato, M. Gradient episodic memory for continual learning. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *NIPS*, pp. 6467–6476, 2017. URL http://dblp.uni-trier.de/db/conf/nips/nips2017.html#Lopez-PazR17.

Maltoni, D. and Lomonaco, V. Continuous learning in single-incremental-task scenarios. *CoRR*, abs/1806.08568, 2018. URL http://dblp.uni-trier.de/db/journals/corr/corr1806.html#abs-1806-08568.

McCloskey, M. and Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. *The psychology of learning and motivation*, pp. 109–165, 1989.

Rebuffi, S.-A., Kolesnikov, A., and Lampert, C. H. icarl: Incremental classifier and representation learning. *CoRR*, abs/1611.07725, 2016. URL http://dblp.uni-trier.de/db/journals/corr/corr1611.html#RebuffiKL16.

Riemer, M., Cases, I., Ajemian, R., Liu, M., Rish, I., Tu, Y., and Tesauro, G. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. Progressive neural networks. *CoRR*, abs/1606.04671, 2016. URL http://dblp.uni-trier.de/db/journals/corr/corr1606.html#RusuRDSKKPH16.

Thrun., S. A lifelong learning perspective for mobile robot control. *Intelligent Robots and Systems*, 1995.

Thrun, S. and Mitchell, T. M. Lifelong robot learning. *Robotics and Autonomous Systems*, 15(1-2):25–46, 1995. URL http://dblp.uni-trier.de/db/journals/ras/ras15.html#ThrunM95.

Yann LeCun, C. C. and Burges., C. J. The mnist database of handwritten digits. 1998.

Yoon, J., Yang, E., Lee, J., and Hwang, S. J. Lifelong learning with dynamically expandable networks. In *ICLR (Poster)*. OpenReview.net, 2018. URL http://dblp.uni-trier.de/db/conf/iclr/iclr2018.html#YoonYLH18.

Zenke, F., Poole, B., and Ganguli, S. Continual learning through synaptic intelligence. In Precup, D. and Teh, Y. W. (eds.), *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3987–3995. PMLR, 2017. URL http://dblp.uni-trier.de/db/conf/icml/icml2017.html#ZenkePG17.