Generalised Controller Design using Continual Learning

Diana Benavides-Prado[1], Chathura Wanigasekara[2], and Akshya Swain[2]

[1] Auckland University of Technology, Auckland, New Zealand
diana.benavides.prado@aut.ac.nz
[2] The University of Auckland, Auckland, New Zealand
{c.wanigasekara,a.swain}@auckland.ac.nz

Abstract. In control applications, controllers for different plants are usually designed with different methods. Although these plants share common characteristics, these are generally designed in isolation. Recently, several researchers have studied the problem of continuously learning a sequence of related learning tasks. A challenge in continual learning is the phenomenon of catastrophic forgetting of knowledge of previous tasks which have been integrated into a neural network model. In this paper we evaluate the feasibility of modelling different controllers using continual learning. We explore regression versions of state-of-the-art methods and demonstrate that even the simplest continual learning approach decreases the overall Mean Average Error (MAE) by 39% of the MAE achieved by a non-continual strategy. Furthermore, a method based on dynamically expanding the network can achieve an overall MAE which is only 18% of the non-continual MAE. Given these results, we also propose a set of new metrics that allow us to characterise the nature of catastrophic forgetting that occurs for these continual learning methods.

Keywords: Continual learning · catastrophic forgetting.

1 Introduction

Many control methods are available for controlling a wide variety of systems (plants) [8, 12, 17, 23, 28, 32]. Typical control schemes include PID control, feedback control, sliding mode control among others [30, 31]. Most of these systems usually have similar characteristics and can be reduced to standard forms such as state-space models. Most of the practical systems are subjected to variations due to various factors such as heat, dust, wear and tear [29]. However, control methods are commonly designed in isolation for each system. This results in redesign/tuning/re-calibration of existing controllers, which is a time-consuming and tedious task which might need to be carried out a few times each year.

An alternative strategy to avoid redesign/re-calibration is by treating these plant control schemes as continual learning tasks. Each of these schemes can be considered as a single task which is learned sequentially. This can help to avoid the need for learning from scratch by potentially using the knowledge acquired in other plant control schemes. Therefore, a single neural network can be used to control different plants which can be used in a wide range of similar problems. An example of this approach is depicted in Figure 1.

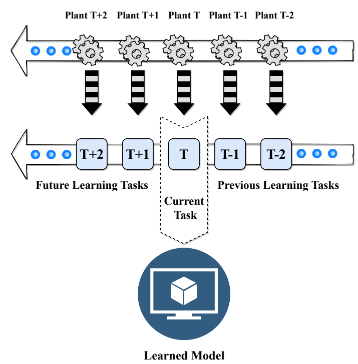

Fig. 1: Generalised controller modelled as a continual learning problem.

Continual learning has been an increasingly active area of research in deep neural networks [16]. In continual learning, a machine learning system observes a sequence of tasks from a particular domain. Training examples of these tasks are observed sequentially. A long-standing challenge in continual learning systems is the problem of catastrophic forgetting of knowledge of previous tasks. This is also known as the *stability-plasticity dilemma* which has been studied for decades [5]. A single network that is used to learn a sequence of tasks should be *plastic* or *adaptive* enough to accommodate knowledge of new tasks, while at the same time be stable enough to not forget knowledge of previous tasks.

A range of methods have been proposed to tackle the problem of catastrophic forgetting in supervised continual learning systems. These methods can be classified in three groups [9]: 1) memory replay methods, which rely on storing or generating some training examples of previous tasks which are re-used in future tasks, 2) regularisation-based methods, which rely on regularising the objective function to be optimised for each incoming task therefore controlling how parameters or weights learned for previous tasks change, and 3) parameter-isolation methods, which rely on allocating sub-networks to specific tasks, by possibly changing the network size as more tasks are sequentially observed. More recent methods combine two or more of these strategies.

In this paper, we study the problem of designing generalised controllers as a continual learning problem. Experiments are carried out using three existing state-of-the-art continual learning methods. Since these existing continual learning approaches were originally proposed to solve classification problems, regression versions of these algorithms are proposed for the generalised controller domain. We accompany this exploration by two new metrics to characterise the amount and the type of catastrophic forgetting. Our main contributions are:

1. We propose an approximation to the problem of optimising multiple controllers using continual learning. Each of these problems is treated as a learning task with training examples observed sequentially.
2. We perform a systematic evaluation of continual learning methods for learning a generalised controller, which includes state-of-the-art metrics in continual learning such as overall accuracy, accuracy per task and time complexity. For this we cast existing continual learning methods which are originally designed for classification problems as regression methods.
3. We propose two new metrics for better characterising the levels and types of catastrophic forgetting occurring in a system.

2 Existing Research

The challenge of learning systems that learn a sequence of tasks was first studied more than two decades ago [26]. Several approaches from transfer, multitask, and lifelong learning have been categorised as alternatives for learning a sequence of tasks [27]. These approaches explored the ability of a learning system to improve the performance while more training examples were observed and tasks were learned. Silver [25] described lifelong learning systems that retain the knowledge and use that to learn new tasks more efficiently and effectively. Silver and Mercer [24] studied lifelong learning in the context of neural networks. More recently, three core properties of lifelong learning systems were identified [6,7]: 1) learning new tasks by leveraging knowledge from previous tasks, 2) learning continuously and incrementally, 3) retaining knowledge acquired during previous tasks.

Continual learning tackles the problem of lifelong learning of a sequence of tasks using deep neural networks. Recently, continual learning has gained increasing interest in the context of deep neural networks. Research has focused on the problem of catastrophic forgetting of knowledge of previous tasks while learning new tasks and integrating knowledge into an existing deep neural network. Parisi et al. [16] and De Lange et al. [9] describe methods to tackle the problem of catastrophic forgetting, which are typically categorised into: 1) regularisation-based methods to impose constraints on how the network changes as new tasks are observed [13], 2) memory management and dual-memories for memory replay, e.g. long-term and short-term memories [15,18,19] and, 3) dynamic network architectures that can change as more tasks are observed, e.g. by expanding or shrinking sub-networks [22,33]. The problem of catastrophic forgetting was also studied in the context of knowledge consolidation in neural networks [20, 21] which recently has been extended to deep neural networks [1]. More recently, research has been conducted that goes beyond catastrophic forgetting and work towards knowledge improvement as tasks are learned sequentially [3, 4].

An additional remarkable challenge in continual learning systems is measuring their performance. Diaz-Rodriguez et al. [10] surveyed and proposed a set of metrics to measure a variety of characteristics of continual learning systems. These metrics include: accuracy, backward transfer of knowledge, and forward transfer of knowledge [15], among others. Other studies have proposed specific metrics to determine the gain in performance while tasks are learned sequentially [2], and to determine the ratio of catastrophic forgetting at the end of learning [14].

3 Methodology

Continual learning systems are composed of a set of $\mathcal{T} = \{T_1, T_2, \ldots, T_l\}$ tasks observed in sequence. In supervised learning, each of these tasks is about learning a mapping from an input feature space $\mathcal{X}$ to an output feature space $\mathcal{Y}$. This mapping is represented by a function $f = \mathcal{X} \to \mathcal{Y}$. A training set $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, for training vectors $\boldsymbol{x}$ sampled from the input feature space $\mathcal{X}$ and their corresponding outputs y sampled from the output feature space $\mathcal{Y}$, is usually available for learning. A challenge in supervised continual learning is that the distributions $\mathcal{D}$ of multiple tasks usually differ, *i.e.* $\mathcal{D}_{T_1} \neq \mathcal{D}_{T_2} \neq \ldots \mathcal{D}_{T_l}$, making it hard for a model to fit well all the tasks. A deep neural network model learns a set of parameters or weights $\boldsymbol{\theta}$ as a representation of the learned function $f = \mathcal{X} - > \mathcal{Y}$.

The phenomenon of catastrophic forgetting is experienced when a model loses its ability to retain knowledge of previous tasks as more tasks are learned. Therefore, the accuracy of these tasks is affected while more tasks are learned. A number of approaches have been proposed to deal with this problem, including methods to protect existing knowledge, methods to retain part of the data from previous tasks, and methods to dynamically expand a network [9, 16]. Section 2 provides a summary of existing research in this area.

We explore three existing methods that tackle catastrophic forgetting from three different angles: 1) EWC [13], which regularises learning of new tasks with respect to existing knowledge, 2) OWM, which combines regularisation and retention of data from previous tasks for replay and, 3) DEN [33], a method that allows to dynamically expand a network as new tasks are learned. We explore variants of these methods for our sequence of regression tasks for multiple controllers. We also propose two new metrics to characterise the level and type of forgetting experienced by each of these approaches in the context of a generalised controller.

3.1 Methods

EWC [13] is a regularisation-based method for continual learning. The problem of EWC at a given task T is to find a set of parameters $\boldsymbol{\theta}_T$ that are optimal for that task while avoiding too much deviations from the set of parameters $\boldsymbol{\theta}_{T-1}$ learned for tasks observed before task T. The function $\mathcal{L}$ to be optimised at task T is given by:

$$\mathcal{L}(\boldsymbol{\theta}_T) = \mathcal{L}(\boldsymbol{\theta}_T) + \sum_i \frac{\lambda}{2} F_i(\theta_{T,i}, \theta^*_{T-1,i}) \tag{1}$$

where F_i is the Fisher information matrix applied to each parameter i for the current task, $\theta_{T,i}$, and each parameter i for a previous task, $\theta^*_{T-1,i}$. The parameter λ controls the influence of previous tasks. In our regression version of EWC, named EWCReg, the function $\mathcal{L}$ is a loss function for regression such as Mean Absolute Error (MAE).

OWM regularises learning of new network parameters by forcing parameters or weights learned on new tasks to be orthogonal to the subspace spanned by inputs from previous tasks. To find the direction orthogonal to these inputs, the method first finds a projector $\mathbf{P} = \mathbf{I} - \mathbf{A}(\mathbf{A}^T\mathbf{A} + \alpha\mathbf{I})^{-1}\mathbf{A}$, where $\mathbf{A} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ i.e. the columns of $\mathbf{A}$ consists of past inputs, $\mathbf{I}$ is the unit matrix and α is a small constant. Note that $\mathbf{A}^T$ denotes the transpose of $\mathbf{A}$. During gradient descent at a learning task T, the parameter vector $\boldsymbol{\theta}$ is modified according to:

$$\boldsymbol{\theta}_T = \eta \mathbf{P}_{T-1} \boldsymbol{\theta}_T \tag{2}$$

where $\mathbf{P}_{T-1}$ is the projector of previously learned inputs for the task $T-1$. The parameter η is the learning rate. Similar to OWM for classification problems, the loss function of the proposed regression variant, named OWMReg, can be any loss function used for the regression.

DEN is a dynamic network expansion method that tackles the problem of catastrophic forgetting during learning of a new task in three steps: 1) selective retraining of parameters affected by the new task, 2) dynamic expansion of selected layers and units of the network, 3) split and duplication of selected units of the network. A new task is first trained on the current version of the network while enforcing its parameters or weights to be sparse. Then, in the first step, a sub-network S is identified. This sub-network contains parameters that are connected to the outputs of the current task. Re-training of this sub-network is performed by minimising:

$$\min_{\boldsymbol{\theta}_T^S} \mathcal{L}(\boldsymbol{\theta_T}^S; \boldsymbol{\theta}_{T-1}^S, \mathcal{D}_T) + \mu \left\| \boldsymbol{\theta}_T^S \right\|_2 \tag{3}$$

where $\boldsymbol{\theta}_T^S$ corresponds to the parameters for the sub-network S on the current task T, θ_{T-1}^S is the set of parameters for this sub-network on the previous task and $\mathcal{D}_T$ is the training data for the current task. μ is a regularisation parameter.

The second step uses group sparse regularisation to dynamically decide the number of neurons to be added to a particular layer L, by minimising:

$$\min_{\boldsymbol{\theta}_\mathcal{N}^L} \mathcal{L}(\boldsymbol{\theta}_\mathcal{N}^L; \boldsymbol{\theta}_{T-1}^L, \mathcal{D}_T) + \mu \left\| \boldsymbol{\theta}_\mathcal{N}^L \right\|_1 + \gamma \sum_g \left\| \boldsymbol{\theta}_\mathcal{N}^{L,g} \right\|_2 \tag{4}$$

where $g \in \mathcal{G}$ is a group defined on the parameters for each neuron. The network is expanded using (4), when the loss is above a user-specified threshold. In that case, the network is expanded by k units, with k is a user-defined parameter.

In the final step, the network is split/duplicated by solving:

$$\min_{\boldsymbol{\theta}_T} \mathcal{L}(\boldsymbol{\theta}_T; \mathcal{D}_T) + \lambda \left\| \boldsymbol{\theta}_T - \boldsymbol{\theta}_{T-1} \right\|_2^2 \tag{5}$$

where λ is the L_2 regularisation parameter. In our regression version of DEN, named DENReg, the loss functions $\mathcal{L}$ used in (3), (4) and (5) can be any typical loss function for regression problems such as MAE.

3.2 Metrics to Characterise catastrophic forgetting

In this section, we propose two new metrics whose objective is to provide more insights into the behaviour of a continual learning system. The first metric determines the level of forgetting of a task once the full sequence of tasks has been learned. This metric is similar in nature to the catastrophic forgetting ratio proposed by Lee et al. [14], which measures the final performance on a task with respect to the best performance that can be achieved for that particular task. In the present study, the final performance of a task is compared with respect to its performance for the first time. This identifies the level of forgetting which is caused by including that task as part of a continual learning system rather than learning it in a non-continual manner. The forgetting level for a particular task T is formally defined as $FL_T = P_T^{t_l} - P_T^{t_T}$ where the overall level of forgetting of task T, FL_T, is the difference between the performance P at the final time step of the sequence t_l and the performance P on that task at the initial time step on which that task was learned, t_T. Note that the level of forgetting has different behaviours depending on the type of performance metrics used. For example, for performance metrics measuring accuracy, a task which experiences low levels of forgetting has an FL_T close to zero. Small negative values denote low levels of forgetting, while positive values would denote a gain in performance. Similarly, for performance metrics measuring error, such as MAE, a task experiencing low levels of forgetting should have an FL_T close to zero. However, in this case small positive values denote low levels of forgetting, while negative values for this metric will denote gain in performance.

This first metric can effectively help quantify the level of forgetting. However, it is interesting to look at various types of forgetting which may often occur in a continual learning system. Gama et al. [11] provide some ideas into a useful categorisation about changes or *drift* in dynamic learning systems. Similarly to online learning, in the context of continual learning it is important to understand the nature of changes in performance, which could occur: 1) abruptly (i.e when tasks experience high levels of forgetting suddenly at a single time step in the sequence), 2) incrementally (i.e. when tasks experience and accumulate forgetting across several consecutive time steps of the sequence) or 3) gradually (i.e. when forgetting levels are experienced across several time steps with a *seasonal* pattern of performance increasing and decreasing over consecutive time steps). This categorisation may help to profile forgetting, and therefore to react to this more appropriately for different tasks.

To determine if abrupt forgetting is occurring for a task, we first need to determine the maximum level of forgetting for that task at any pair of consecutive time steps using: $MF_T = \max(P_T^t - P_T^{t-1})$, $\forall\, t \in \{0, 1, \ldots, l\}$.

Next, given a constant threshold τ_a, a task is said to be experiencing abrupt forgetting if:

$$\frac{MF_T}{FL_T} \leq \tau_a \quad \text{and} \quad MF_T \times FL_T > 0 \tag{6}$$

To determine if a task is experiencing incremental forgetting up to some level τ_i for l_i consecutive time steps, the following metric can be used:

$$\left(\sum \frac{P_T^t - P_T^{t-1}}{FL_T} \leq \tau_i \right) \geq l_i \tag{7}$$

for all $t \in \{0, 1, \ldots, l\}$.

Finally, to determine if a task is experiencing gradual forgetting up to some level τ_g for up to l_g consecutive time steps, the following metric can be used:

$$\left(\sum abs\left(\frac{P_T^t - P_T^{t-1}}{FL_T} \right) \leq \tau_g \right) \geq l_g \tag{8}$$

Note that this last metric could effectively identify gradual forgetting irrespective of the direction of forgetting.

4 Experiments and Results

We evaluate the feasibility of various methods described in earlier sections for an example of controlling a DC motor. We investigate learning a sequence of plant control schemes using the three continual learning methods explained in Section 3: EWCReg, OWMReg and DENReg. The performance of these learning methods are compared with Vanilla CL and Vanilla NonCL. Note that Vanilla CL learn tasks sequentially without considering the effects of catastrophic forgetting while Vanilla NonCL learns all the tasks at once. We also measure the level and nature of forgetting using the two metrics proposed in Section 3.

Experiments are carried out by generating datasets for 20 tasks. For each task, the parameters of the DC motor were changed. A dataset for a specific task is generated by making the DC motor to follow a fixed trajectory. Each task is composed of $7,500$ training examples, $1,500$ validation examples and $1,500$ test examples, and 21 input features. One of the input features corresponds to previously observed speeds of a DC motor, while the other 20 features correspond to the values of y in the previous 20 times. The output feature y corresponds to speed of the motor. Note that for Vanilla NonCL, all training examples available for each task are used. For the other methods, the first $1,000$ training examples are used, to simulate real-world continual learning scenarios where training data is scarce. We arranged 30 randomly selected task orders, ensuring that each task is the first task of the sequence for at least one of these orders. Results are averaged across task orders, unless stated otherwise. To make these tasks more varied, random noise is added to 50% of the training examples. Furthermore, the order of the input features are shuffled randomly for each task, except for Vanilla NonCL which is not subject to any of the above types of noise.

Table 1: Mean MAE and total training time after training all 20 tasks sequentially, averaged across task orders.

Method	Mean MAE	Training Time (sec.)
DENReg	0.252 ± 0.001	501.0 ± 18.3
EWCReg	0.401 ± 0.048	$6,330.0 \pm 190.34$
OWMReg	0.364 ± 0.038	276.0 ± 12.4
Vanilla CL	0.538 ± 0.106	179.00 ± 19.1
Vanilla NonCL	1.36 ± 0.068	7.18 ± 0.34

4.1 Hyper-Parameter Settings

For all methods under evaluation, a two-layer neural network with 200 units in each layer is trained. We use $1,000$ epochs of batches containing 128 training examples per batch. The learning rate is set to 0.001 in all cases. In all cases except OWMReg, we use gradient descent to optimise MAE. For OWMReg, a momentum optimiser of value 0.99 is used, which also optimises MAE. EWC λ parameter is set to the number of tasks, 20. We use 200 validation examples from each previous tasks to construct the Fisher information matrix in EWCReg. For OWMReg, α parameter is set to 10. For DENReg, we set the lambda sparsity parameter L_1 to 0.001 and L_2 to 0.0001. The group LASSO lambda is set to 0.001, the number of units to be increased in the expansion process is set to 5, the threshold for dynamic expansion is 0.1 and the threshold for split and duplication is set to 0.1.

4.2 Overall Performance

The MAE from sequential learning of all the tasks and the total training time at the end of the sequence of tasks, averaged across task orders, are presented in Table 1. A naive method such as Vanilla CL, with no control for catastrophic forgetting, outperforms the approach of Vanilla NonCL. EWCReg and OWMReg achieve lower MAE than Vanilla CL, demonstrating the ability of these methods to avoid catastrophic forgetting. However, the training time of EWCReg is approximately 35 times more compared to Vanilla CL. DENReg clearly outperforms counterparts with a final MAE of 0.252, which is only 18% of Vanilla NonCL and 47% of Vanilla CL. In terms of training time, DENReg requires only 3 times more training time than Vanilla CL. The mean MAE averaged across task orders at each timestep of the sequence is presented in Figure 2 (left), for DENReg, EWCReg, OWMReg and Vanilla CL. Although the MAE of Vanilla CL decreases with addition of more tasks over time, its performance is poorer compared to other parameter-isolation methods such as DENReg, OWMReg and EWCReg.

4.3 Performance per Task

Figure 2 (right) shows MAE of each task as each consecutive task is learned. This result is shown as an example for one of the task orders used in the experiments.

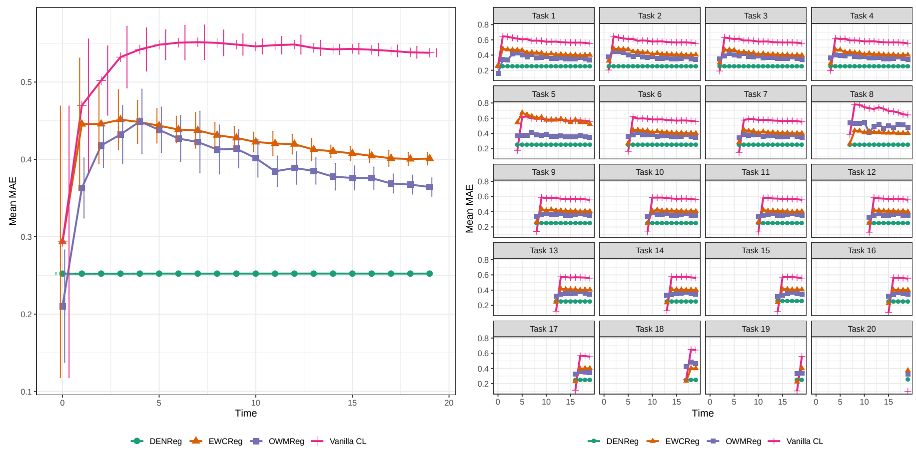

Fig. 2: Mean MAE of tasks learned sequentially, at each timestep, across all task orders, overall (left) and per task (right).

Vanilla CL achieves a low MAE for each of these tasks, when these are learned for the first time. However, high levels of catastrophic forgetting are experienced as new tasks are learned. EWCReg also experiences forgetting after a task is learned for the first time, although at a lower rate than Vanilla CL. OWMReg experiences forgetting during the initial task (Task 1). However, it is capable of retaining knowledge of previous tasks with small forgetting later in the sequence. DENReg is stable for the full sequence of 20 tasks. The result for DENReg is consistent with previous findings for this method, where it has been shown that DEN performs well when the number of tasks is relatively small [4].

Similarly, Figure 3 (left) explores MAE of each task when these are learned for the first time, averaged across task orders. Vanilla CL learns tasks for the first time with low MAE. However, contrasting to Figure 3 (right), tasks learned using this method are always affected in their performance in the next timestep. On the other hand, OWMReg tends to perform worse than other methods while learning tasks for the first time. However, as depicted in Figure 3 (right), this allows the method to control catastrophic forgetting later in the sequence. EWCReg and DENReg achieve values of MAE which are more similar to Vanilla CL when tasks are learned for the first time.

4.4 Characterisation of catastrophic forgetting

Figure 3 (right) shows levels of forgetting for tasks presented in Figure 3 (left), for all the methods under evaluation. Overall forgetting levels, for all tasks in this sequence, are: Vanilla CL, 0.376, EWCReg, 0.129, OWMReg, 0.0137, and DENReg, 0.0. Consistent with previous results, DENReg does not experience forgetting once the sequence of tasks is finished. Table 2 shows the types of forgetting experienced by each method and task, for a specific task order. The

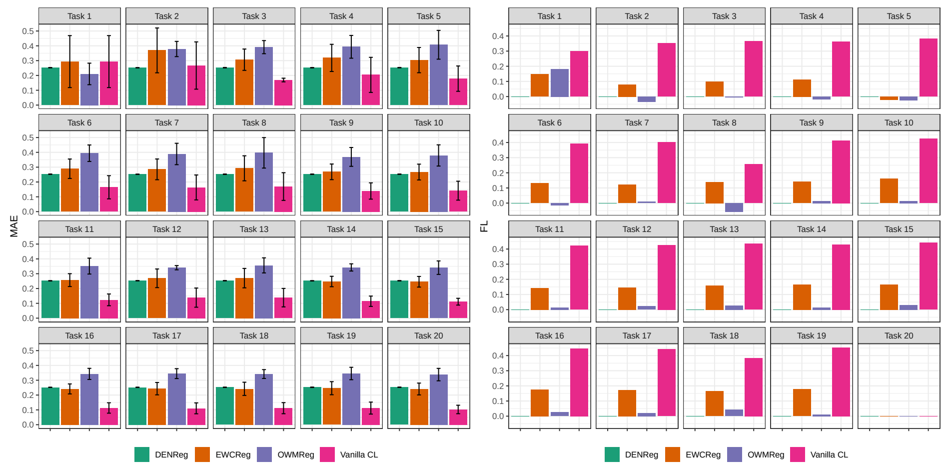

Fig. 3: Left: Mean MAE of tasks when tasks are learned for the first time, for all task orders. Right: Forgetting level for each task, measured as the difference of MAE at the last timestep of the sequence and the MAE on the first time a task was learned, for a specific task order.

abrupt forgetting threshold τ_a was set to 0.95. Incremental forgetting would occur if the level of forgetting is at least $\tau_i = 0.05$ of the level of forgetting for that task for at least 3 consecutive timesteps. Similarly, gradual forgetting would occur if the level of forgetting is at least $\tau_g = 0.01$ of the level of forgetting for that task for at least 2 consecutive timesteps, regardless of the direction of forgetting. Methods such as EWCReg and Vanilla CL experience abrupt forgetting for all the tasks. OWMReg, on the contrary, experiences different kinds of forgetting. DENReg experiences no forgetting at all according to the thresholds set, as shown previously in Figure 2 (right).

Table 2: Types of forgetting experienced, for a specific task order.

Type of Drift	**DENReg**	**EWCReg**	**OWMReg**	**Vanilla CL**
Abrupt	0	19	11	19
Incremental	0	0	1	0
Gradual	0	0	1	0
No/Unclassified	20	1	7	1

5 Conclusions

We investigated the applicability of a continual learning approach to the problem of learning a sequence of controllers. We explored a variety of state-of-the-art continual learning methods. Experiments demonstrated that the problem of learning multiple controllers can be formulated as a continual learning problem. This approach achieves much lower levels of error than learning all these controllers in a non-sequential manner. Furthermore, state-of-the-art methods that allow dynamic network expansion showed a potential to retain the knowledge of previous controller tasks and help to avoid the problem of catastrophic forgetting in this context, for a small number of tasks. These promising results leave as future work the problem of exploring more complex scenarios of controllers that are designed to systems such as generic classes of linear systems, T-S fuzzy nonlinear systems, or control systems composed of a larger number of tasks.

References

1. Atkinson, C., McCane, B., Szymanski, L., Robins, A.: Pseudo-Recursal: Solving the Catastrophic Forgetting Problem in Deep Neural Networks. arXiv preprint arXiv:1802.03875 (2018)
2. Benavides-Prado, D., Koh, Y.S., Riddle, P.: Measuring cumulative gain of knowledgeable lifelong learners. In: NeurIPS Continual Learning Workshop. pp. 1–8 (2018)
3. Benavides-Prado, D., Koh, Y.S., Riddle, P.: Selective hypothesis transfer for lifelong learning. In: 2019 International Joint Conference on Neural Networks (IJCNN). pp. 1–10. IEEE (2019)
4. Benavides-Prado, D., Koh, Y.S., Riddle, P.: Towards knowledgeable supervised lifelong learning systems. Journal of Artificial Intelligence Research **68**, 159–224 (2020)
5. Carpenter, G.A., Grossberg, S.: The art of adaptive pattern recognition by a self-organizing neural network. Computer **21**(3), 77–88 (1988)
6. Chen, Z., Hruschka, E., Liu, B.: Lifelong Machine Learning and Computer Reading the Web. ACM SIGKDD Tutorial (2016), https://www.cs.uic.edu/ liub/Lifelong-Machine-Learning-Tutorial-KDD-2016.pdf
7. Chen, Z., Ma, N., Liu, B.: Lifelong Learning for Sentiment Classification. arXiv:1801.02808 (2018)
8. Cloosterman, M.B.G., Hetel, L., Van De Wouw, N., Heemels, W.P.M.H., Daafouz, J., Nijmeijer, H.: Controller synthesis for networked control systems. Automatica **46**(10), 1584–1594 (2010)
9. De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., Tuytelaars, T.: A continual learning survey: Defying forgetting in classification tasks. arXiv preprint arXiv:1909.08383 (2019)
10. Díaz-Rodríguez, N., Lomonaco, V., Filliat, D., Maltoni, D.: Don't forget, there is more than forgetting: new metrics for continual learning. arXiv preprint arXiv:1810.13166 (2018)
11. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A survey on concept drift adaptation. ACM computing surveys (CSUR) **46**(4), 1–37 (2014)
12. Heij, C., Ran, A.C., Schagen, F.V.: Introduction to mathematical systems theory: Linear systems, identification and control. Basel: Birkhäuser (2007)
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, S., Stokes, J., Eaton, E.: Learning shared knowledge for deep lifelong learning using deconvolutional networks. In: IJCAI. pp. 2837–2844 (2019)
15. Lopez-Paz, D., Ranzato, M.: Gradient episodic memory for continual learning. In: Advances in neural information processing systems. pp. 6467–6476 (2017)
16. 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)
17. Patel, N., Nguang, S.K., Coghill, G., Swain, A.: Online implementation of servo-controllers using bit-streams. In: Proceedings of IEEE Region 10 Conference(TENCON). pp. 1–6. IEEE (2005)
18. Ramapuram, J., Gregorova, M., Kalousis, A.: Lifelong generative modeling. arXiv preprint arXiv:1705.09847 (2017)
19. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 2001–2010 (2017)
20. Robins, A.: Catastrophic Forgetting, Rehearsal and Pseudorehearsal. Connection Science **7**(2), 123–146 (1995)
21. Robins, A.: Consolidation in Neural Networks and in the Sleeping Brain. Connection Science **8**(2), 259–276 (1996)
22. 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)
23. Shang, Y., Chen, B., Lin, C., Liu, Z.: Adaptive tracking control for a class of uncertain nonlinear multi-agent systems with input quantization based on neural approach. IEEE Access **7**(1), 167300–167309 (2019)
24. Silver, D.L., Mercer, R.E.: Selective Transfer of Neural Network Task Knowledge. The University of Western Ontario (Canada) (2000)
25. Silver, D.L., Yang, Q., Li, L.: Lifelong Machine Learning Systems: Beyond Learning Algorithms. In: AAAI Spring Symposium: Lifelong Machine Learning. vol. 13, pp. 49–53 (2013)
26. Thrun, S.: Is learning the n-th thing any easier than learning the first? In: Advances in neural information processing systems. pp. 640–646 (1996)
27. Thrun, S., Pratt, L.: Learning to learn: Introduction and overview. In: Learning to learn, pp. 3–17. Springer (1998)
28. Wanigasekara, C., Almakhles, D., Swain, A., Nguang, S.K., Subramaniam, U., Padmanaban, S.: Performance of neural network based controllers and $\Delta\Sigma$-based PID controllers for networked control systems: A comparative investigation. In: Proceedings of 19th International Conference on Environment and Electrical Engineering (EEEIC) (2019)
29. Wanigasekara, C., Swain, A., Nguang, S.K., Prusty, B.G.: Improved learning from small data sets through effective combination of machine learning tools with VSG techniques. In: Proceedings of the International Joint Conference on Neural Networks. pp. 1–6 (2018)
30. Wanigasekara, C., Almakhles, D., Swain, A., Nguang, S.K.: Delta-modulator-based quantised output feedback controller for linear networked control systems. IEEE Access **8**(1), 175169–175179 (2020)
31. Wanigasekara, C., Almakhles, D., Swain, A., Zhou, L.: Design of dynamic fuzzy Q-learning controller for networked wind energy conversion systems. In: Proceedings of 20th International Conference on Environment and Electrical Engineering (EEEIC) (2020)

32. Wanigasekara, C., Almakhles, D., Zhou, L., Swain, A., Subramaniam, U., Padmanaban, S.: Design of $\Delta\Sigma$ based PID controller for wind energy systems. In: Proceedings of 19th International Conference on Environment and Electrical Engineering (EEEIC) (2019)
33. Yoon, J., Yang, E., Lee, J., Hwang, S.J.: Lifelong Learning with Dynamically Expandable Networks. arXiv e-prints (2017)