Open Access
Full Length Article

ORIGINAL ARTICLE

Continual Deep Learning through Unforgettable Past Practical Convolution

Muhammad Rehan Naeem[1], Muhammad Munwar Iqbal[2], Rashid Amin[3]

[1,2,3] *Department of Computer Science, University of Engineering and Technology, Taxila Pakistan*

ABSTRACT

When trained sequentially on tasks, several machine-learning models forget how to perform previously learned tasks. This phenomenon, called catastrophic forgetting, is an essential challenge to address so that systems can learn continuously. In the face of unexpected circumstances, humans and animals can develop sophisticated predictive models that allow them to correctly and effectively reason about real-world phenomena, and they can adjust these models extremely rapidly. Because of the static and offline design of traditional deep learning methods, the viability of deep neural networks (DNNs) to solve data stream problems still needs extensive research. It is necessary for intelligent systems to continuously learn new skills, but traditional deep learning approaches suffer from disastrous forgetting of the past. With weight regularization, recent works resolve this. Although computationally costly, functional regularization is supposed to work better, but it does so rarely in practice. In this article, we address this problem by using a modern approach to functional regularization that uses a few memorable historical instances that are important to prevent forgetting. Our methodology allows weight-space preparation by using a Gaussian Method formulation of deep networks when defining both the unforgettable history and a practical prior. Our technique achieves state-of-the-art success on traditional metrics and opens up a new path for continual learning where approaches based on regularization and memory are naturally merged. In 10 pieces of training, 10-200 unforgettable examples are set. For split, we use five MNIST-built binary classification tasks 0/1, 2/3, 4/5, 6/7, and 8/9 with a completely linked multi-head network with two hidden layers, each with 256 hidden units. Throughout all tasks, EUHC performs reliably better (except for the first task where it is similar to the best). It also increases by a wide margin over the lower limit (' distinct tasks'). In particular, the EUHC matches the performance of tasks 4-6 with the network trained jointly on all tasks, which means that forgetting is completely avoided there. The overall output is also the highest for all duties. For instance, with 10 such memorable instances, EUHC's careful selection raises the average accuracy to 70% from 45%.

Keywords: Deep Neural Networks, MNIST-built binary classification, traditional deep learning methods,

Author`s Contribution
[1] Data analysis, interpretation, and manuscript writing, Active participation in data collection, [2] Conception, synthesis, planning of research, [3.] Interpretation and discussion

Address of Correspondence
Muhammad Munwar Iqbal
Email: munwar.iq@uettaxila.edu.pk

Article info.
Received: Aug 26, 2021
Accepted: December 17, 2021
Published: December 30, 2021

Cite this article: *Naeem MR, Iqbal MM, Amin R. Continual Deep Learning through Unforgettable Past Practical Convolution J. inf. commun. technol. robot. appl.2017; 8(2):20-29.*

Funding Source: *Nil*
Conflict of Interest: *Nil*

INTRODUCTION

Continuous learning (also called life-long learning and incremental learning) is a very general type of online learning in which data can arrive continuously online, tasks may alter over time (e.g. new classes may be discovered), and entirely new tasks may arise. The ubiquity of deep learning means designing deep,

continuous learning methods. However, finding a balance between adjusting to recent data and maintaining old data information is difficult. Too much plasticity contributes to the disgraceful disaster-forgetting crisis.

Intelligent systems quickly adapt to the change in the environment which is an excellent ability. When intelligent systems use this ability, it is also required to recognize, remember, and recall past practices with new adaption. Unluckily, the latest and standardized deep-learning techniques are unable to support these abilities to remember the past while adapting the new skills. For applications such as robotics, such disastrous forgetting raises a major problem, where new tasks can emerge during training, and data from previous tasks may be inaccessible for retraining. Continuous sources of knowledge are exposed to computer systems running in the real world and are thereby expected to understand and recall various tasks from complex data distributions. For example, to learn from their interactions, an autonomous agent communicating with the environment is necessary and must be able to increase gain, fine-tune, and transfer information over long periods. Continuous or lifetime learning is called the capacity to continuously improve over time by accommodating additional information and maintaining previously acquired experiences. For machine learning and neural networks and thus for the development of artificial intelligence (AI) technologies, such a continuous learning task has become a long-standing challenge.

State-of-the-art theoretical studies show that neural networks increase their depth of representation and generalization capacity (NNs) [1, 2]. Nevertheless, the data stream problem remains an uncharted field of traditional deep neural networks (DNNs). Unlike traditional data stream approaches based on a shallow network structure [3-5], DNNs theoretically provide significant accuracy and the ability to manage unstructured data streams. Due to their substantial computational and memory demand, the direct application of traditional DNNs for data stream analytics is often impossible to implement under limited computational resources [6]. Ideally, data streams should be treated sample-wise without any retraining process to avoid the issue of disastrous forgetting, besides, to scale up with the design of continuous environments [6,7]. Another difficulty comes from conventional DNNs' set, static structure [8]. In other words, before the process runs, network capacity must be estimated. This trait does not mirror data streams' diverse and changing characteristics.

In deep neural networks (DNNs), many techniques have been proposed in recent years to address disastrous forgetting. A common technique is keeping obtained values of previous task/data closer to the network weights but predictions based on the previous task may not ensure a guarantee of the quality. Since the network outputs are complexly dependent on the weights, such weight-convolution cannot be efficient. A simpler solution is to use practical convolution, where we establish the network outputs directly, but this is expensive since it needs output representations at multiple input positions. Existing techniques minimize these costs by deliberately choosing positions, such as using a working memory or Gaussian-Process (GP) influencing points, but they do not reliably surpass existing methods of weight-convolution.

Problem Statement

Adapting the change is the characteristic of Artificial Intelligent (AI) based systems. When working on a particular task there is a requirement to distinguish, memorize, and recall the previous knowledge while adopting new changes. It is not defined what kind of unforgettable examples to be chosen and is there any theory behind this. There is no standard that how many unforgettable examples does anyone uses in experiments and by increasing the number of examples, efficiency is also significantly improved. If there are any standards for this then if we loosen any standards, does it bring changes, and what sort of improvements are required. When we have large data sets, we have to select a more appropriate method to bring more significant results.

LITERATURE REVIEW

State-of-the-art computational studies indicate that increasing the depth of neural networks increases the capacity of neural networks (NNs) for representation and generalization. The data stream problem, however, remains an uncharted field of traditional deep neural networks (DNNs). DNNs can deliver substantial improvements in accuracy and the ability to manage unstructured data streams, unlike traditional data stream techniques based on a shallow network structure. Due to

their substantial computing and memory requirement, direct deployment of traditional DNNs for data stream analytics is often impossible, rendering them difficult to implement under minimal computational resources. Ideally, in addition to scale-up with the design of continuous settings, the data streams should be managed in a sample-wise manner without any retraining stage to avoid the disastrous forgetting issue. The set and rigid form of conventional DNNs brings another challenge [9]. In other words, before the operation runs, the network bandwidth needs to be measured. The diverse and changing properties of data streams do not mirror this trait. Because of the static and offline design of traditional deep learning methods, the viability of deep neural networks (DNNs) to solve data stream problems still needs extensive research. In this research, a deep continuous learning algorithm, namely autonomous deep learning (ADL), is suggested. ADL has a modular framework, unlike conventional deep learning approaches, where the network structure can be designed from scratch with the absence of the original network structure through the self-constructing network structure. By having a different deep structure that can accomplish a trade-off between plasticity and resilience, ADL explicitly tackles disastrous forgetting [10]. The Network Significance (NS) formula is proposed to drive the increasing and pruning process of hidden nodes. The Drift Detection Scenario (DDS) is designed to signal distributional changes to data streams that cause a new hidden layer to be generated. As a complexity reduction module that reduces unnecessary layers, the Maximum Information Compression Index (MICI) approach plays an important role.

Not only is human and animal learning distinguished by the capacity to develop specific skills, but also by the ability to adapt easily as certain skills need to be worked out in new or evolving circumstances. Animals, for example, may rapidly adjust to walking and moving in various environments, and in the case of unpredictable disruptions, people can readily modulate force during reaching gestures. Besides, these events are remembered, and when related disruptions arise in the future, they may be recalled to respond more easily. Since it is impossible to learn completely new models on such short time scales, the author designs algorithms that specifically train models to adapt quickly to small amounts of data. For intelligent systems running in the real world, where evolving conditions and unpredictable disruptions are the rules, such online adaptation is critical. In this article the author proposes an algorithm for fast and continuous online learning that uses deep neural network models to construct and sustain the distribution of tasks, enabling both generalization and task specialization to be naturally created [9, 11]. In the face of unforeseen changes, humans and animals can develop sophisticated predictive models that allow them to correctly and effectively reason about real-world phenomena, and they can adjust these models extremely rapidly. Models of deep neural networks allow us to describe very complex functions but lack this ability to change rapidly online. The purpose of this paper is to develop, using deep neural network models, a framework for continuous online learning from an incoming stream of data [12]. Until designing and retaining a mixture of models to manage non-stationary task distributions, the author formulates an online learning framework that uses stochastic gradient descent to update model parameters and an expectation-maximization algorithm with a Chinese restaurant system. This enables all models to be adapted as appropriate, with new models instantiated for task adjustments and existing models remembered when tasks that were previously seen are again experienced. In addition, meta-learning can be used to meta-train a model in such a way that this direct online SGD adaptation is efficient, which is otherwise not the case for large-scale feature approximations.

Online Learning: Unlike conventional offline learning, where the entire training data must be made accessible before learning the assignment, online learning examines learning algorithms that learn to refine predictive models sequentially over a stream of data instances. For surveys and overviews on the topic, see Bartlett et al. 2007; and Shalev-Shwartz et al. 2011 [13,14]. The first set of online learning algorithms includes various techniques for learning a linear model. This line of research is extended to non-linear models with online learning with kernels [15], but the models are shallow and their output lags behind that of modern deep neural networks. Unfortunately, attempts to learn neural networks online are plagued by issues such as convergence, disastrous interference, and

more. Ramasamy et al. 2018 [15] start with a small network and then adapt the capacity by adding more neurons as new samples arrive, while [16] proposed a method for online deep metric learning based on stacking multiple metric layers. Pernici et al [17, 18] work is close to our first application scenario in terms of applications. They learn face identities by temporal continuity in a self-supervised manner. They use a memory of detected faces and start with the VGG face detector and descriptor. We, on the other hand, begin with a much weaker pre-trained model and update the model parameters over time while they do not. Catastrophic interference, or the extreme forgetting of previous samples while learning new ones, is a common problem in both continuous and online learning. This phenomenon occurs at various scales: in online learning, it occurs when learning samples of different patterns than previous ones; in conventional continuous learning, it occurs over a series of tasks. Hsu et al. group the studied continual learning scenarios into incremental task learning, incremental domain learning, and incremental class learning in [19]. They argue that the last two - namely, approaches that do not involve knowledge of the mission identity - should receive more consideration since this is the case in most realistic scenarios. However, as previously mentioned, the task-based sequential learning setup has been used in the majority of approaches to date. Elastic Weight Consolidation [20], Synaptic Intelligence [21], and Memory Aware Synapses [22] are examples of regularization-based methods. These methods calculate significance weights for each model parameter and penalize adjustments to parameters that were previously considered significant. We'll talk about how to make one of them work in an online environment later. Although Synaptic Intelligence computes the value weights in real-time, it does not update the losses until the task is completed, so it, like the other methods, relies on understanding the task boundaries. Incremental Moment Matching [23] is based on similar principles, but it saves different templates for different tasks and only merges them at the end. As a result, it's unclear how this could be applied to a task-free online environment. The work on Dynamically Expandable Networks [7] is also related. They use the similarities between the current task and previously learned tasks to figure out which neurons can be reused and which neurons need to be added to account for the new information.

A typical and realistic situation that exists during the process of human learning is constantly learning various activities. Usually, the acquisition of new skills from new tasks has a little detrimental effect on previously learned tasks. Besides, it also helps to advance all relevant skills by learning several tasks that are strongly connected. This is not generally the case in modern machine learning for deep neural networks (DNNs), however. DNNs suffer the so-called "catastrophic forgetting" phenomenon when a list of learning tasks is introduced, where they typically "forget" previously mastered tasks after being prepared for a new task [24]. This is a fascinating concept that has lately drawn numerous research attempts. One of the main problems of continuous learning, where machine learning systems are trained for sequential or streaming assignments, is to resolve disastrous forgetting. Despite recent impressive advances in state-of-the-art deep learning, the devastating forgetting dilemma still plagues deep neural networks (DNNs). In continuous learning with DNNs, this paper provides a conceptually clear but general and efficient system for managing disastrous forgetting. The approach proposed consists of two components: an optimization component of the neural system and a learning and/or fine-tuning component of the parameter [25]. Not only is the suggested approach capable of evolving neural systems in an intuitively realistic manner by distinguishing the explicit learning of the neural system and the parameter estimate, but it also reveals good abilities in research to mitigate disastrous forgetting.

RESEARCH METHODOLOGY

There are different datasets are available on the internet for research purposes. Dataset will be selected in this step according to research requirements. After selecting the dataset, a cleansing of data will be conducted to select suitable attributes. In this step, a suitable dataset will be selected to conduct research and get results based on deep learning algorithms. After cleansing the dataset, now deep learning algorithm will be selected which is used to get the results. There are different algorithms available for deep learning, so we have to select the most appropriate algorithm to get

accurate results. After selecting the best algorithm for the current dataset and situation, now we have to select a deep learning tool for the analysis. There are different tools available so we have to select a deep learning tool to analyze the dataset. After tool selection, now we will run different algorithms on the dataset to check the accuracy of the algorithms. Different algorithms will provide different results according to their working style. We will select algorithms with more accuracy and results. After checking the accuracy and working of algorithms, we will select algorithms with more accuracy and results. Now we will train the algorithm on the dataset by selecting the different ratios of the dataset. After training of algorithm on the dataset, now we will test and validate the dataset for the correctness of the results. This step will provide correctness and novelty of results. After testing and validating the results, we will discuss the findings of the research. This step provides the results of the research, limitations, and recommendations for further improvements for future work. The methodology of the proposed research is as follows:

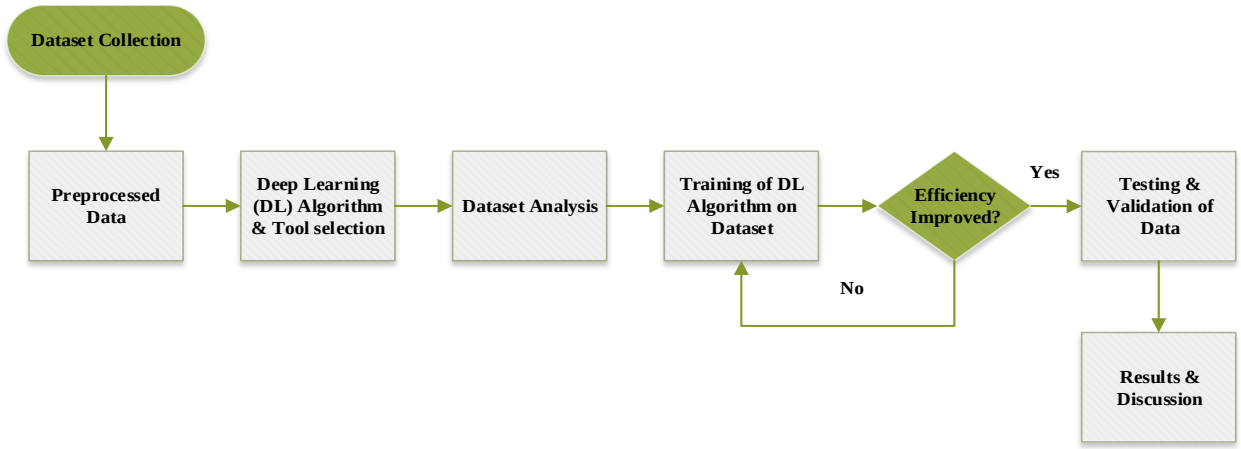


Figure-Error! No text of specified style in document.**1 Methodology for Continual Deep Learning**

Preliminaries of learning enhancement

In an unfamiliar area, reinforcement learning deals with learning a policy for an agent to communicate. Various challenges, such as sports, natural language processing, neural architecture/optimizer search, and so on, have been successfully extended to it. An agent monitors the current state at each step s_t the climate agrees on the action a_t in compliance with a strategy $\pi(a_t \mid s_t)$ and observes a signal of compensation $r_t + 1$. The agents aim to find a policy that maximizes the amount of discounted incentives anticipated $R_t, R_t = \sum_{t'=t+1}^{\infty} \gamma^{t'-t-1} r_{t'}, where\, \gamma \in (0,1)$. The value of potential incentives is a discount factor that decides the value. The meaning function of a policy π, the expected return is described $V_\pi(s) = E_\pi[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \mid s_0 = s]$, and its action-value function $Q_\pi(s,a) = E_\pi[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \mid s_0 = s, a_0 = a]$.

Policy gradient techniques solve the question of choosing a successful policy by executing stochastic gradient descent to maximize an output target over a given family of stochastic parameterized policies $\pi_\theta(a \mid s)$ parameterized by θ. The policy gradient theorem includes expressions concerning the gradient of the average reward and discounted reward targets θ. The goal is specified in the discounted setting concerning a designated start state or dispersed state $s_0 : \rho(\theta, s_0) = E_{\pi\theta}\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \mid s_0\right]$.

Effective Unforgettable Historical Classifier (EUHC)

One of the main problems of continuous learning, where machine learning systems are trained for sequential or streaming assignments, is to resolve disastrous forgetting. Despite recent impressive advances in state-of-the-art deep learning, the devastating forgetting dilemma still plagues deep neural networks (DNNs).

Computational Complexity of the Algorithm

The resulting algorithm for binary classification is seen in the figure. We presume, for binary classification, a sigmoid $\sigma(f_w(x))$ function and lack of cross-entropy. The Jacobian and noise accuracy is as follows:

$J_w(x) = \nabla_w f_w(x)$ *and* $\Lambda_w(x) = \sigma(f_w(x))\left[1-\sigma(f_w(x))\right]$

. We need the diagonal of the covariance, which we denote by v, to determine the mean and kernel. This can be obtained

using $\sum_{*}^{-1} = \sum_{i=1}^{N} J_{w_*}(x_i)^T \Lambda_{w_*}(x_i, y_i) J_{w_*}(x_i) + \delta I_P$ b

ut over $D_{1:t}$, with the number. The following update computes this

$$\frac{1}{V^t} = \left[\frac{1}{V_{t-1}} + \sum_{i \in D_t} diag\left(J_w(x_i)^T \Lambda_w(x_i) J_w(x_i)\right)\right] \quad \text{...Eq.(1)}$$

Where'/' denotes wise division of elements and diag. (A) is A's diagonal. Using this in a phrase equal to

$$m_{w_*}(x) := f_{w_*}(x),\ k_{w_*}(x, x') := J_{w_*}(x) \sum_{*} J_{w_*}(x')^T$$

We can measure a matrix of the mean and kernel:

$$m_{t,s}(w)[i] = \sigma(f_w(x_i)),\ K_{t,s}(w)[i,j] = \Lambda_w(x_i)[J_w(x_i) Diag(v_t) J_w(x_j)^T] \Lambda_w(x_j), \quad \text{...Eq.(2)}$$

Memorable overall examples are Xi, Xj, where Diag(a) denotes a diagonal matrix with an as the diagonal. Using these, the gradient can be written where the gradient of the functional regularization is added to the gradient of the loss as an additional term. The regularization is determined in the figure in the subroutine.

Algorithm:
EUHC for binary classification on task t given $q_{t-1}(w): N(\mu_{t-1}, diag(v_{t-1}))$, and memorable pasts $M_{1:t-1}$.
Function $EUHC(D_t, \mu_{t-1}, v_{t-1}, M_{1:t-1})$:
Get $m_{t-1,s}, K^{-1}_{t-1,s}, \forall$ tasks $s < t$ $(Eq.2)$
Initialize $w \leftarrow \mu_{t-1}$
while not congregated *do*
Random $\{x_i, y_i\} \in D_t$
$g \leftarrow N\nabla_{wi}(y_i, f_w, y_i(x_i))$
$g_f \leftarrow g\_FR(w, m_{t-1}, K^{-1}_{t-1}, M_{1:t-1})$
Adam update with gradient $g + \tau g_f$
end
$\mu_t \leftarrow w$ and compute v_t $(Eq.1)$
$M_t \leftarrow memorable\_past(D_t, w)$
return μ_t, v_t, M_t

Function $g\_FR(w, m_t, K^{-1}_{t-1}, M_{1:t-1})$:
Initialize $g_f \leftarrow 0$
for task $s = 1, 2, 3, 4 \ldots, t-1$ *do*
compute $m_{t,s}$ $(Eq.2)$
$h_i \leftarrow \Lambda_w(x_i) J_w(x_i)^T, \forall x_i \in M_s$
form matrix H with h_i as columns
$g_f \leftarrow g_f + HK^{-1}_{t-1}(m_{t,s} - m_{t-1,s})$
end
return g_f
Function $memorable\_past(D_t, w)$:
Calculate $\Lambda_w(x_i), \forall x_i \in D_t$
return M examples with highest $\Lambda_w(x_i)$

Figure 1: Binary classification algorithm for the specified task and unforgettable sections.

MNIST Permuted and Divided

The Permuted MNIST consists of a sequence of tasks, each of which adds a fixed pixel permutation to the whole MNIST dataset. We use a completely linked single-head network with two hidden layers, each consisting of 100 hidden units, similarly to previous work. We're training for 10 duties. In the 10–200 range, the number of unforgettable examples is set. We are also checking the Split MNIST benchmark, consisting of five MNIST-built binary classification tasks: 0/1, 2/3, 4/5, 6/7, and 8/9. We use a completely linked multi-head network with two hidden layers, each with 256 hidden units, following the settings from previous work. Per mission, we pick 40 unforgettable points.

Table 1. MNIST comparisons: we use 200 references as unforgettable points for Permuted. We use 40 for Break, Split.

Method	Permuted	Split
DLP	82%	61.2%
EWC	84%	63.1%
SI	86%	98.9%
Improved VCL	93%	98%
Random Coreset	94.6%	98.2%
FRCL-RND	94.2%	97.1%
FRCL-TR	94.3%	97.8%
FRORP-L2	87.9%	98.5%
FROMP-L2	94.6%	98.7%
FRORP	94.6%	99%
EUHC	**94.9%**	**99%**

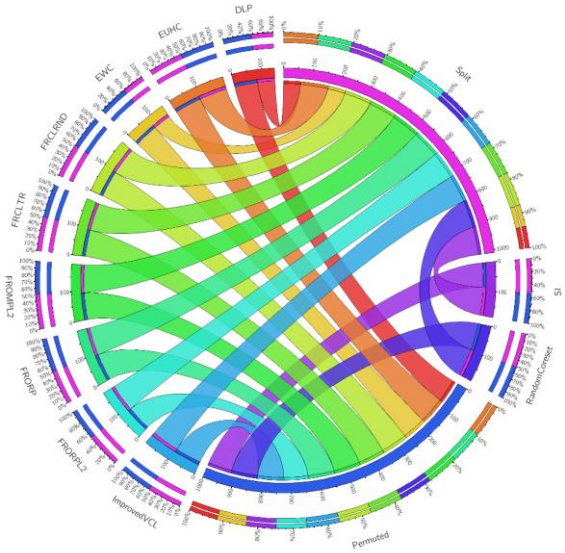


Figure 2a. MNIST comparisons for Permuted and split

Fig. 2a in which EUHC produces greater results than the methods of weight regularization as well as the method of continuous learning function-regularization. EUHC also strengthens and shows the efficacy of the kernel over FRORP-L2 and EUHC-L. The change is not important when opposed to FRORP. We assume that this is because a random unforgettable past already reaches a performance similar to the maximum possible performance, and by deliberately picking the cases, we see little more progress.

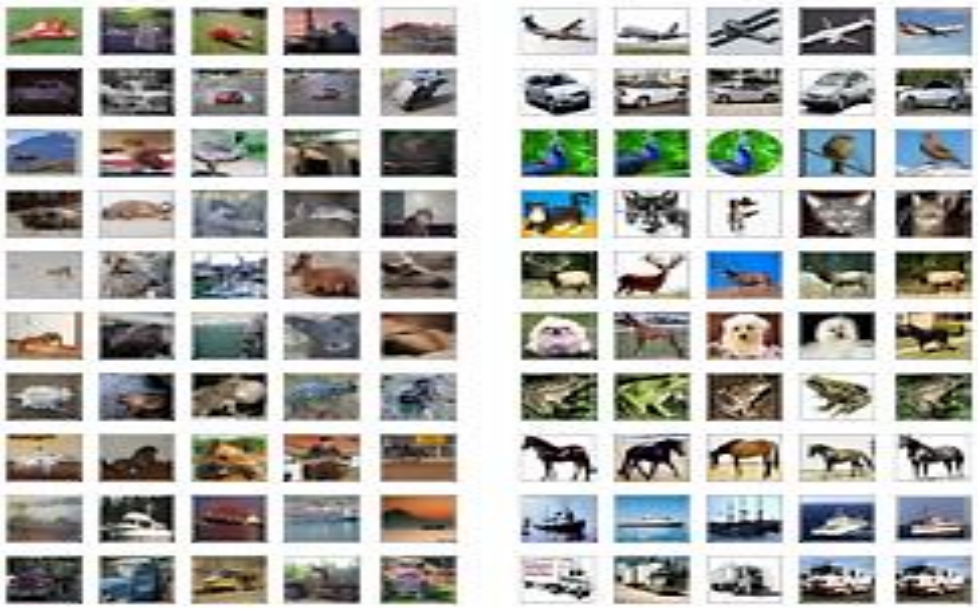
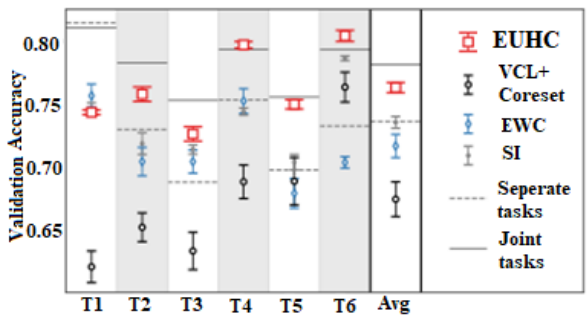


Figure 2b: Most unforgettable (left) vs least memorable (right)



FigureError! No text of specified style in document.**-3a: Divided CIFAR: Specificity of individual tasks and average**

The final precision of the average is seen, however, in Fig. 3c, where the number of unforgettable examples is minimal, we see an increase. It appears to be more difficult to identify the most memorable examples than the least memorable examples, which means that they might lie closer to the judgment frontier.

Divided CIFAR

Divided CIFAR is a benchmark that is more complicated than MNIST, and it consists of 6 tasks. A full CIFAR-10 dataset is the first task, followed by 5 tasks, each consisting of 10 consecutive CIFAR-100 groups. We use a multi-head CNN with 4 convolutional layers for the model architecture, then 2 dense layers with dropouts. In the range 10-200, the number of unforgettable examples is set, and we run each process 5 times. Two additional baselines are compared. The first baseline consists of networks trained independently for each mission. Such teaching does not take advantage of the forward/backward transition from other activities and sets a lower limit that we have to reach. The second foundation is where we collectively train all duties, which will maybe produce the greatest outcomes and which we would prefer to balance. The observations are summarized in Fig. 3a, where we see that EUHC, though outperforming all other strategies, is similar to the upper limit. Although VCL forgets the previous tasks, the weight-regularization methods EWC and SI do not perform well on the latter tasks. Bad VCL output is most likely due to the difficulties of using CNNs3 prop Bayes by Back. Throughout all tasks, EUHC performs reliably better (except for the first task where it is similar to the best). It also increases by a wide margin over the lower limit (' distinct tasks'). In particular, the EUHC matches the performance of tasks 4-6 with the network trained jointly on all tasks, which means that forgetting is completely avoided there. The overall output is also the highest for all duties (see the column 'Avg'). Fig. 3b. Regarding the amount of unforgettable past instances, 3b indicates the results. In a similar manner to Fig. 3c, carefully choosing a memorable example improves the output, especially when there are limited numbers of memorable examples. For instance, with 10 such memorable instances, EUHC's careful selection raises the average accuracy to 70% from 45% obtained by EUHC. Unfortunately, the inclusion of the kernel in EUHC here does not change dramatically over EUHC -L, unlike the MNIST experiment. Fig. 2b presents a few pictures of the most and least memorable examples from the past, where we again see that it may be more difficult to identify the most memorable.

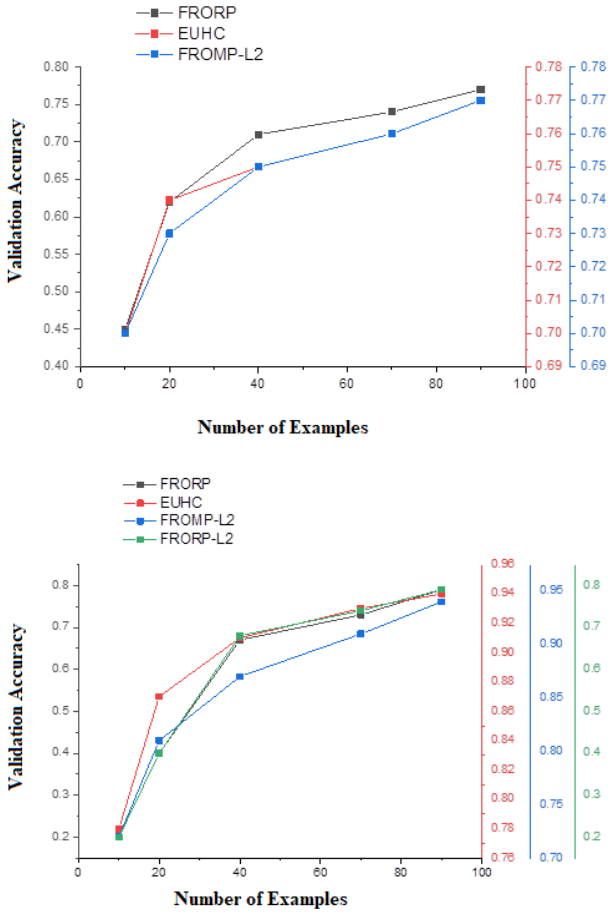


Figure 3c: MNIST Permuted

EUHC has a ranking of 2:6-0:9 percent, which is equivalent to the score of 2:3-1:4 percent for EWC, but

higher than 9:2-1:8 percent for VCL+ core sets. EUHC output is summarized in the table and graphically presented in figure 4.

Table 2. Forward and Backward Transfer Metrics

Split MNIST		Permuted MNIST		Split CIFAR	
Forward	Backward	Forward	Backward	Forward	Backward
0.0±0.1%	-0.2±0.2%	-1.9±0.1%	-1.0±0.1%	6.1%±0.7%	-2.6±0.9%

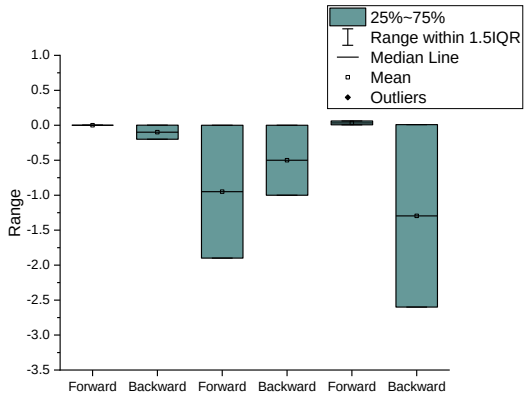


Figure 4. Transfer Metrics Analysis

The figure shows the analysis of the transfer matrices based on backward and forward.

Table 4. Overcome Catastrophic Forgetting, Learn to Develop

Model	Individual	Classifier	Adapter	EUHC (ours)
ImNet	69.84	69.84	69.84	69.84
C100	73.96	77.07	79.82	79.59
SVHN	95.22	93.12	94.21	95.28
UCF	69.94	62.37	70.72	72.03
OGlt	86.05	79.93	85.10	86.60
GTSR	99.97	99.68	99.89	99.72
DPed	99.86	99.92	99.58	99.52
Flwr	41.86	65.88	60.29	71.27
Airc	50.41	36.41	50.11	53.01
DTD	29.88	48.20	50.60	49.89
Avg	71.72	73.14	76.02	77.68
Total size	58.96	6.68	12.50	14.46

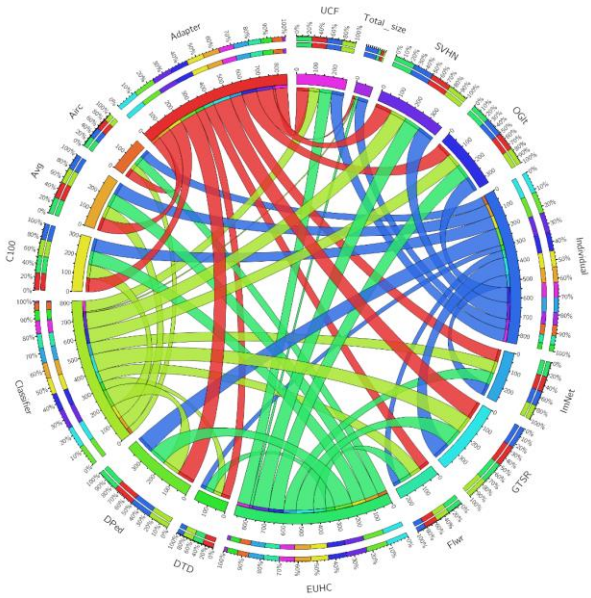


Figure 4. Comparison of different techniques to Overcome Catastrophic Forgetting

Validation classification accuracy (percentage) results in the Visual Domain Decathlon dataset. After practicing the 10 tasks, the final model size is the total parameter size (in Million). Individual models are those that have been conditioned for specific tasks. A task-specific classifier is tuned for each task, as shown by the term "classifier". Figure 5 presents the graphical representation and correlations which are shown in table 4. The experimental findings presented in this section demonstrate the importance of directly continuous structure learning. The applicable parameters from previous tasks can be exploited until the correct frameworks have been learned. Furthermore, since the method for leveraging these parameters is mastered by structure optimization, much less tuning is needed for improved success on new tasks, and therefore forgetting is avoided.

CONCLUSION

We suggest EUHC, a continual learning functional-regularization strategy thus preventing disastrous forgetting. To transform weight-space distributions into function-space, EUHC uses the Gaussian Method formulation of neural networks. We suggested ways to classify specific memorable past examples with this formulation and regularize the training of neural network weights functionally. The EUHC achieves state-of-the-art

benchmark results. The first move in this paper is to incorporate the concepts of neural networks and GP groups while preserving the simplicity of the methods of teaching. There are a lot of potential avenues to discover. In 10 pieces of training, 10-200 unforgettable examples are set. For split, we use five MNIST-built binary classification tasks 0/1, 2/3, 4/5, 6/7, and 8/9. EUHC performs reliably better (except for the first task where it is similar to the best) It increases by a wide margin over the lower limit (' distinct tasks') EUHC matches the performance of tasks 4-6 with the network trained jointly on all tasks, which means that forgetting is completely avoided there. The overall output is also the highest for all duties. The results of the experiments outlined in this section show how important it is to learn structures directly and continuously. Until the correct frameworks have been taught, the relevant parameters from previous tasks may be used. Since structure optimization has learned the method for exploiting these parameters, much less tuning is required for better success on new tasks.

These are a few of the issues that will be discussed in the future. Functional regularization, we argue, is the best way to train deep learning algorithms in the long run. We hope that the techniques presented in this paper will lead to new approaches to deep learning knowledge transfer.

REFERENCES

1. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, vol. 25, pp. 1097-1105, 2012.
2. R. B. Palm, "Prediction as a candidate for learning deep hierarchical models of data," Technical University of Denmark, vol. 5, 2012.
3. A. Ashfahani, M. Pratama, E. Lughofer, Q. Cai, and H. Sheng, "An online RFID localization in the manufacturing shop floor," in Predictive Maintenance in Dynamic Systems, ed: Springer, 2019, pp. 287-309.
4. M. Pratama, E. Dimla, T. Tjahjowidodo, W. Pedrycz, and E. Lughofer, "Online tool condition monitoring based on parsimonious ensemble+," IEEE transactions on cybernetics, vol. 50, pp. 664-677, 2018.
5. M. Pratama, W. Pedrycz, and E. Lughofer, "Evolving ensemble fuzzy classifier," IEEE Transactions on Fuzzy Systems, vol. 26, pp. 2552-2567, 2018.
6. J. Gama and G. Zhang, "Learning under Concept Drift: A Review," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, vol. 31, 2019.
7. J. Yoon, E. Yang, J. Lee, and S. J. Hwang, "Lifelong learning with dynamically expandable networks," arXiv preprint arXiv:1708.01547, 2017.
8. J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," ACM computing surveys (CSUR), vol. 46, pp. 1-37, 2014.
9. A. Ashfahani and M. Pratama, "Autonomous deep learning: Continual learning approach for dynamic environments," in Proceedings of the 2019 SIAM International Conference on Data Mining, 2019, pp. 666-674.
10. H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in Advances in Neural Information Processing Systems, 2017, pp. 2990-2999.
11. C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner, "Variational continual learning," arXiv preprint arXiv:1710.10628, 2017.
12. A. Nagabandi, C. Finn, and S. Levine, "Deep online learning via meta-learning: Continual adaptation for model-based rl," arXiv preprint arXiv:1812.07671, 2018.
13. P. Bartlett, E. Hazan, and A. Rakhlin, "Adaptive online gradient descent," 2007.
14. S. Shalev-Shwartz, "Online learning and online convex optimization," Foundations and Trends in Machine Learning, vol. 4, pp. 107-194, 2011.
15. S. Ramasamy, K. Rajaraman, P. Krishnaswamy, and V. Chandrasekhar, "Online deep learning: growing rbm on the fly," arXiv preprint arXiv:1803.02043, 2018.
16. W. Li, J. Huo, Y. Shi, Y. Gao, L. Wang, and J. Luo, "Online deep metric learning," arXiv preprint arXiv:1805.05510, 2018.
17. F. Pernici, F. Bartoli, M. Bruni, and A. Del Bimbo, "Memory-based online learning of deep representations from video streams," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2324-2334.
18. F. Pernici and A. Del Bimbo, "Unsupervised incremental learning of deep descriptors from video streams," in 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), 2017, pp. 477-482.
19. Y.-C. Hsu, Y.-C. Liu, A. Ramasamy, and Z. Kira, "Re-evaluating continual learning scenarios: A categorization and case for strong baselines," arXiv preprint arXiv:1810.12488, 2018.
20. J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, et al., "Overcoming catastrophic forgetting in neural networks," Proceedings of the national academy of sciences, vol. 114, pp. 3521-3526, 2017.
21. F. Zenke, B. Poole, and S. Ganguli, "Improved multitask learning through synaptic intelligence," ed: CoRR.
22. R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 139-154.
23. S.-W. Lee, J.-H. Kim, J. Jun, J.-W. Ha, and B.-T. Zhang, "Overcoming catastrophic forgetting by incremental moment matching," arXiv preprint arXiv:1703.08475, 2017.
24. N. Kamra, U. Gupta, and Y. Liu, "Deep generative dual memory network for continual learning," arXiv preprint arXiv:1710.10368, 2017.
25. X. Li, Y. Zhou, T. Wu, R. Socher, and C. Xiong, "Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting," arXiv preprint arXiv:1904.00310, 2019.