Plastic and Stable Gated Classifiers for Continual Learning

Nicholas I-Hsien Kuo[*,1], Mehrtash Harandi[2], Nicolas Fourrier[4],
Christian Walder[1,3], Gabriela Ferraro[1,3], and Hanna Suominen[1,3,5]
[*]CBDRH, The University of New South Wales, Sydney, Australia
[1]RSCS, The Australian National University and [2]ECSE, Monash University, Australia
[3]Data61, CSIRO, Australia, [4]Pôle Universitaire Léonard de Vinci, Paris La Défense, France
[5]Department of Future Technologies, The University of Turku, Finland
n.kuo@unsw.edu.au, mehrtash.harandi@monash.edu, nfourrier@gmail.com
{christian.walder, gabriela.ferraro}@data61.csiro.au, hanna.suominen@anu.edu.au

Abstract

Conventional neural networks are mostly high in plasticity but low in stability. Hence, catastrophic forgetting tends to occur over the sequential training of multiple tasks and a backbone learner loses its ability in solving a previously learnt task. Several studies have shown that catastrophic forgetting can be partially mitigated through freezing the feature extractor weights while only sequentially training the classifier network. Though these are effective methods in retaining knowledge, forgetting could still become severe if the classifier network is over-parameterised over many tasks. As a remedy, this paper presents a novel classifier design with high stability. ***Highway****-Connection* ***Classifier Networks*** *(**HCNs**) leverage gated units to alleviate forgetting. When employed alone, they exhibit strong robustness against forgetting. In addition, they synergise well with many existing and popular continual learning archetypes.*
We release our codes at ***https://github.com/Nic5472K/CLVISION2021_CVPR_HCN***

1. Introduction

Continual learning studies the sequential acquisition of functions for a single neural network. This remains a difficult problem because *catastrophic forgetting* [17] may occur when a trained network learns a new skill. Catastrophic forgetting refers to the abrupt loss of knowledge for solving an old task when information for solving a new task is incorporated; and this is due to parametric modifications for satisfying the learning objectives of a new task.

Forgetting can be effectively mitigated by freezing feature extractor weights while only sequential learning the classifiers[13, 14, 16][1]. The effectiveness of this network compartmentalisation is similar to how offline meta-learning achieves rapid knowledge acquisition [18]. To elaborate, this specific setup first reuses acquired feature representations and then provides a less significant amount of parametric modification.

Nonetheless, catastrophic forgetting will inevitably occur due to over-parameterisation in the classifier weights. To strengthen this simple yet practical experimental setup, we propose a novel architectural design for classifier networks to increase their robustness against forgetting. Our work is based on an analysis in the neural network backward pass, and we present **Highway**-Connection **Classifier Networks** (**HCNs**) with gated units to alleviate forgetting.

2. Related Work

There are three main archetypes of continual learning methods – *replay*, *regularisation*, and *dynamic architecture*. Regularisation methods append the conventional learning objective[2] with a secondary loss to preserve learnt network mappings [11, 30, 7]; and that dynamic architectures either instantiate new modules to host more knowledge [22, 29], or seek to fully utilise the backbone network capacity by reducing representation overlap [4, 23]. However, [28] found that only replay methods could successfully prevent forgetting in all of their identified continual learning scenarios[1].

Replay methods [21] interleave external data while training a backbone learner on new tasks. Some replay exact copies of past data [2] and some pseudorehearse with random inputs to achieve function approximation [1]. Representation learning can also be leveraged to select an exemplar set of observed data [19]; or alternatively, one can incorporate generative models to the backbone learner to cre-

[1]This includes Mai *et al*. [16]'s approach which won the CVPR CLVision 2020 challenge. Their best practice applied replay techniques only to the deep layers of their backbone network.

[2]Such as cross entropy loss for image recognition.

Table 1: The Forward and Backward Pass of Residual Connections and Highway Connections

Refer to the notations in Section 4.

Residual Connection		
Forward pass [5]	$\mathbf{n}_{L_\gamma} = \mathbf{n}_{L_{\gamma-1}} + \mathscr{H}(\mathbf{n}_{L_{(\gamma-1)}}, \mathbf{W}_{L_\gamma})$	(1)
Backward pass [12]	$J = \dfrac{\partial \mathbf{n}_{L_\gamma}}{\partial \mathbf{n}_{L_{(\gamma-1)}}} = (\mathbb{I} + \mathbf{D}^\gamma \mathbf{W}_{L_\gamma})$ with diagonal $\mathbf{D}^\gamma$ such that $\mathbf{D}^\gamma_{(r,r)} = \mathscr{H}'(\mathbf{n}_{L_{(\gamma-1)}}, \mathbf{W}_{L_\gamma})_{(r)}$	(2)
Highway Connection		
Forward pass [25]	$\mathbf{k}_{L_\gamma} = (1 - \mathbf{T}_{L_\gamma}) \odot \mathbf{k}_{L_{\gamma-1}} + \mathbf{T}_{L_\gamma} \odot \mathscr{H}(\mathbf{k}_{L_{(\gamma-1)}}, \mathbf{W}_{L_\gamma})$ with gated unit: $\mathbf{T}_{L_\gamma} = \mathbf{T}(\mathbf{k}_{L_{(\gamma-1)}}, \mathbf{Q}_{\mathbf{L}_\gamma})$	(3)
Backward pass [25]	$J = \dfrac{\partial \mathbf{k}_{L_\gamma}}{\partial \mathbf{k}_{L_{(\gamma-1)}}} = \begin{cases} \mathbb{I}, & \text{if } \mathbf{T}(\mathbf{k}_{L_{(\gamma-1)}}, \mathbf{Q}_{\mathbf{L}_\gamma}) = \mathbf{0} \\ \mathscr{H}'(\mathbf{k}_{L_{(\gamma-1)}}, \mathbf{W}_{L_\gamma}), & \text{if } \mathbf{T}(\mathbf{k}_{L_{(\gamma-1)}}, \mathbf{Q}_{\mathbf{L}_\gamma}) = \mathbf{1} \end{cases}$	(4)

ate data which encapsulate features of past tasks [24].

However, replay techniques incur a high computational cost because they require the concurrent training of stored data with those of the new task. While a practitioner can lower the incurred computational intensity by replaying with less externally stored memory, [9] found that this could lead the backbone learner to severely overfit on past tasks while underfit on subsequent tasks. By adopting the network compartmentalisation of [13, 14, 16] and limit sequential learning only to the classifiers, computation can be made less costly in the backward pass while updating the weights of the backbone learner.

3. The Continual Learning Framework

As described in [15], continual learning employs a base learner $F = F(x_i, i)$ to solve a continuum of data $\mathcal{C}$ with n locally independent and identically distributed (iid) tasks

$$(x_1^1, \omega_1, y_1^1), \ldots, (x_\beta^i, \omega_i, y_\beta^i), \ldots, (x_B^n, \omega_n, y_B^n). \quad (5)$$

Each task maps inputs x_β^i to labels y_β^i from datasets $(x_\beta^i, y_\beta^i) \in \mathcal{D}_{\omega_i}$ for all B data where $\beta = 1, \ldots, B$. Once learner F have progressed to observe data from the j-th task, it is prohibited to backtrack and learn from an earlier dataset $\mathcal{D}_{\omega_i}$ for $i < j$. We evaluate and store the progress of the learner accuracy in matrix $\mathscr{A} \in \mathbb{R}^{(n+1)\times n}$. The accuracy of all n tasks of an initialised F is stored in $\mathscr{A}_{1,(1:n)}$; and after observing all data from $\mathcal{D}_{\omega_i}$, all updated accuracy are stored in $\mathscr{A}_{(i+1),(1:n)}$.

4. Highway-Connection Classifier Networks

Depth is important to the success of neural networks [27]. However, deep networks are hard to optimise and *highway connections* [25] took inspirations from the *Long Short-Term Memory* Recurrent Neural Network (LSTM RNN) [6] to ease the training of very deep models.

A conventional layer in feedforward neural network is

$$\mathbf{a}_{L_\gamma} = \mathscr{H}(\mathbf{a}_{L_{(\gamma-1)}}, \mathbf{W}_{L_\gamma}) \quad (6)$$

where layer γ has input $\mathbf{a}_{L_{\gamma-1}}$, output $\mathbf{a}_{L_\gamma}$, and a non-linear transformation $\mathscr{H}$ parameterised with $\mathbf{W}_{L_\gamma}$. A highway connection modifies Equation (6) as Equation (3) shown in Table 1 where $\odot$ is the element-wise product and that gated unit $\mathbf{T}_{L_\gamma}$ has non-linear transformation $\mathbf{T}$ parameterised by $\mathbf{Q}_{L_\gamma}$. The gated unit uses the sigmoid transformation σ, hence each element of $\mathbf{T}(\mathbf{k}_{L_{(\gamma-1)}}, \mathbf{Q}_{\mathbf{L}_\gamma})$ lies within [0, 1] and Equation (3) provides a more flexible transformation where the output extrapolates between $\mathbf{k}_{L_{\gamma-1}}$ and $\mathscr{H}(\mathbf{k}_{L_{(\gamma-1)}}, \mathbf{W}_{L_\gamma})$ dependent on the input.

4.1. HCNs Reduce Parametric Modification

A Γ-layer network with input $\mathbf{a}_{L_0} = \mathbf{x}$, output $a_{L_\Gamma} = \hat{\mathbf{y}}$, and target $\mathbf{y}$ has the following derivative to cost C:

$$\frac{\partial C}{\partial \mathbf{W}_{L_\gamma}} = \frac{\partial C}{\partial \hat{\mathbf{y}}} \left[\prod_{\zeta=\gamma}^{\Gamma-1} \frac{\partial \mathbf{a}_{L_{(\zeta+1)}}}{\partial \mathbf{a}_{L_\zeta}}\right] \left\{\frac{\partial \mathbf{a}_{L_\gamma}}{\partial \mathbf{W}_{L_\gamma}}\right\}$$
$$= \frac{\partial C}{\partial \hat{\mathbf{y}}} \left[\prod_{\zeta=\gamma}^{\Gamma-1} \mathscr{H}'(\mathbf{a}_{L_\zeta}, \mathbf{W}_{L_{(\zeta+1)}})\right] \left\{\frac{\partial \mathscr{H}(\mathbf{a}_{L_{(\gamma-1)}}, \mathbf{W}_{L_\gamma})}{\partial \mathbf{W}_{L_\gamma}}\right\}.$$

As shown in Table 1, both residual connection and highway connection introduced the identity matrix towards the $[.]$ term to facilitate deep learning. However, let us consider the $\{.\}$ term. A conventional feedforward network has $\frac{\partial \mathbf{a}_{L_\gamma}}{\partial \mathbf{W}_{L_\gamma}} = \frac{\partial \mathscr{H}(\mathbf{a}_{L_{(\gamma-1)}}, \mathbf{W}_{L_\gamma})}{\partial \mathbf{W}_{L_\gamma}}$; and residual networks are identical with $\frac{\partial \mathbf{n}_{L_\gamma}}{\partial \mathbf{W}_{L_\gamma}} = \frac{\partial \mathscr{H}(\mathbf{n}_{L_{(\gamma-1)}}, \mathbf{W}_{L_\gamma})}{\partial \mathbf{W}_{L_\gamma}}$ as $\mathbf{n}_{L_{\gamma-1}}$ is independent of $\mathbf{W}_{L_\gamma}$ for all $\mathbf{n}_{L_\gamma}$. On the contrary, highway networks have $\frac{\partial \mathbf{k}_{L_\gamma}}{\partial \mathbf{W}_{L_\gamma}} = \mathbf{T}_{L_\gamma} \frac{\partial \mathscr{H}(\mathbf{k}_{L_{(\gamma-1)}}, \mathbf{W}_{L_\gamma})}{\partial \mathbf{W}_{L_\gamma}}$. Since all units of gate $\mathbf{T}_{L_\gamma}$ lie between $[0, 1]$, the gated units also limit the extent of update to parameters $\mathbf{W}_{L_\gamma}$. This is desirable and makes catastrophic forgetting harder to occur.

4.2. Demonstration

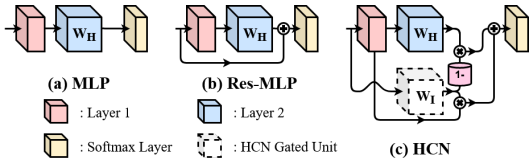
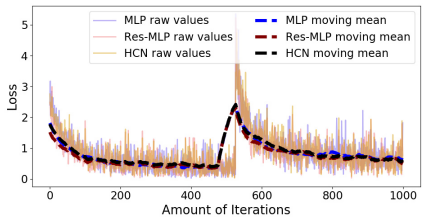
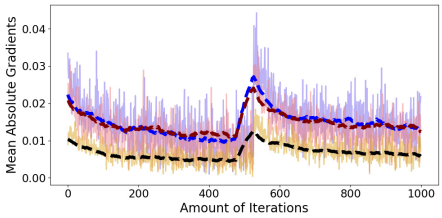


Figure 1: Different Classifier Network Designs



(a) Loss against Iteration



(b) Gradient against Iteration

Figure 2: Differences in the Losses and Gradients

To verify parametric reduction, we tested three classifier designs and compared their gradients and losses. The designs were the *Multiple Layer Perceptrons* (MLPs), MLPs with *Residual* connections (Res-MLPs), and MLPs with *Highway Connections* (HCNs) shown in Figure 1 overleaf. All classifiers were first trained on MNIST [10] for 525 iterations then sequentially on KMNIST [3] for 525 iterations.

The MNIST database has grey-scale handwritten digits of numbers between [0-9]. There are 60K and 10K images for training and testing respectively; and the images are presented on 28×28-pixel boxes. KMNIST has images of the identical format, but of Japanese Hiragana characters. We flattened the images as vector inputs with the length of 784 $(= 28 \times 28)$ and processed them with the classifiers. All of our classifiers had 2 layers with 100 hidden dimensions.

We present our findings in Figure 2, and the gradients shown were the mean absolute values of $\mathbf{W}_{L_\gamma}$s of Equations (6), (1), and (3) for the MLPs, Res-MLPs, and HCNs, respectively. While we found virtually no differences in the losses, the gradients behaved qualitatively differently. The gradients of HCNs were much lower than those of their rivals; while those of Res-MLPs and MLPs behaved similarly. These findings verified that the gates of highway connections scaled down gradient values in the backward pass.

5. Experiments

Table 2: An Overview of the Results

Method	ACC	BWT
Section 5.3 : Naïve Implementation for *Permuted MNIST*		
HCN (Ours)	**71.49** ± 11.57	**-7.71**
MLP	65.54 ± 19.76	-14.39
Res-MLP	65.36 ± 19.85	-15.28
Section 5.3 : Naïve Implementation for *Incremental Cifar100*		
HCN (Ours)	**74.82** ± 15.48	**-6.10**
MLP	73.08 ± 17.49	-8.40
Res-MLP	70.84 ± 18.78	-11.34
Section 5.4 : In-Class Comparison for *Permuted MNIST*		
EWC + HCN (Ours)	**71.74** ± 7.84	**-7.02**
EWC + MLP	66.72 ± 13.78	-12.67
GEM + HCN (Ours)	74.19 ± 8.24	**6.22**
GEM + MLP	**74.52** ± 4.27	3.52
HAT + HCN (Ours)	**70.72** ± 8.32	**-3.39**
HAT + MLP	62.50 ± 20.41	-5.70
Section 5.4 : In-Class Comparison for *Incremental Cifar100*		
EWC + HCN (Ours)	**75.47** ± 13.47	**-5.34**
EWC + MLP	73.12 ± 18.16	-8.27
GEM + HCN (Ours)	80.84 ± 9.18	**0.70**
GEM + MLP	**80.96** ± 9.71	0.61
HAT + HCN (Ours)	**76.04** ± 14.15	**-2.75**
HAT + MLP	73.23 ± 16.54	-3.69

We tested two datasets on three techniques for the three classifier designs with our results summarised in Table 2. All classifiers had 2 layers with 100 hidden dimensions; and were trained with SGD [20] with learning rate 0.01.

5.1. Datasets and Sequential Tasks

Our first experiment tested *image permutation* [7] with MNIST. We flattened MNIST images as a vector of 784 pixels and each task applied a unique permutation over all images of that task. We tested 20 tasks for permuted MNIST on the three classifiers shown in Figure 1. Following the setup of [15], each task observed 1000 data.

Our second experiment tested *incremental classes* [19] with Cifar100 [8]. Cifar100 contains 50K training images and 10K test images of 100 classes of objects presented on 32×32 colour images. Following [16]'s best practice[1], we compartmentalise our backbone network with a feature extractor and a classifier network. Our feature extractor was the reduced ResNet18 used in [19] and we reserved 50 classes to thoroughly pre-train it with 200 epochs. Upon sequential learning, we froze the weights of the ResNet18 and replaced its original classifier with the three classifiers shown in Figure 1. Sequential learning was conducted with the remaining 50 classes. We introduced 5 classes per task

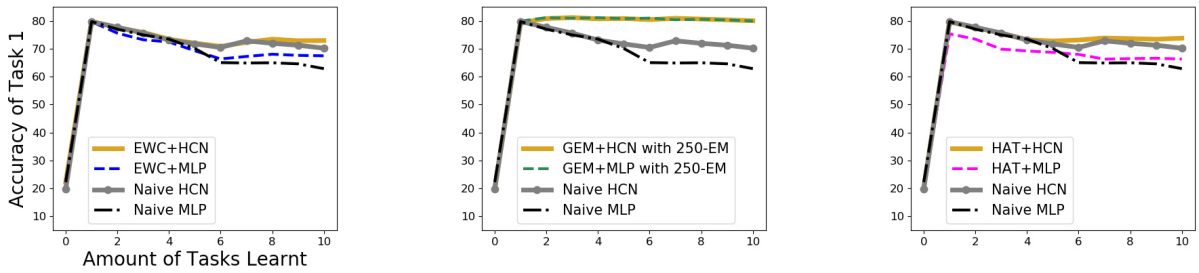


Figure 3: Accuracy of Task 1 with Different Continual Learning Techniques for Incremental Cifar100

(10 tasks in total) and each task observed 1 epoch of data.

5.2. Metrics

We adopted two metrics from [15] with the progressive accuracy matrix $\mathscr{A}$ defined in Section 3:

Average accuracy (ACC)
$\frac{1}{n}\sum_{i=1}^{n}\mathscr{A}_{(n+1),i}$ and

Backwards transfer (BWT)
$\frac{1}{n-1}\sum_{i=1}^{n-1}(\mathscr{A}_{(n+1),i}-\mathscr{A}_{(i+1),i})$,

ACC computed the mean accuracy after observing all tasks $\mathscr{A}_{(n+1),i}$. and BWT quantified the task-wise forget since $\mathscr{A}_{(i+1),i}$ immediately after training on a task's own data. For both metrics, the larger and more positive the better.

5.3. On Naïvely Sequential Learning

We first naïvely trained HCNs, MLPs, and Res-MLPs on permuted MNIST and incremental Cifar100. The ACC and BWT scores in Table 2 showed that HCNs outperformed MLPs and Res-MLPs, and reflected 2 important remarks.

Remark 1: Plasticity and Stability
Conventional neural networks like MLPs are high in plasticity while low in stability. By incurring less BWT, we empirically showed that HCNs functioned as an architectural design that possessed much higher stability. In addition, HCNs' superior ACC scores also meant that the HCN design maintained a high degree of network plasticity.

Remark 2: Connection Specificity
As previously demonstrated in Section 4, residual connections incur the same level of catastrophic forgetting as MLPs. This was verified again by the BWT scores in Table 2, and this showed that the highway connections were specifically required for preventing forgetting.

5.4. HCNs vs MLPs with Continual Learning Techniques Applied

We then tested HCNs against MLPs with continual learning techniques[3]. This included the regularisation-based *Elastic Weight Consolidation* (EWC) [7], replay-based *Gradient Episodic Memory* (GEM) [15], and dynamic architecture-based *Hard Attention to the Task* (HAT) [23].

[3]Mostly following [15]'s repository of https://github.com/facebookresearch/GradientEpisodicMemory.

Of these techniques, EWC regularised the learning objective with the Fisher matrix scaled by the relative importance[4] of the old tasks compared to the new ones. GEM externally stored a few data per task[5] to pre-condition the native gradients by solving a quadratic problem to redirect misaligned gradients. In contrast, HAT learnt a nullification mask[6] per task to generate sparse activations to lower feature overlapping among the tasks for the classifiers.

As shown in Table 2, HCNs outperformed MLPs on both ACCs and BWTs when conjointly trained with EWC and HAT. However, when GEM was applied, both HCNs and MLPs showed no signs of forgetting (see their positive BWT) and they performed comparably in ACC.

In order to further analyse the advantages in employing HCNs, we plotted the accuracy of Task 1 over the entire course of sequential learning for incremental Cifar100 in Figure 3. The results showed another 2 important remarks.

Remark 3: Compatible with All Archetypes of Approaches
HCNs were compatible with all 3 main continual learning archetypes mentioned in Section 2. This was because that all archetypes externally modified the sequential learning experimental setup, while we made modifications internally to the classifier architecture.

Remark 4: High Robustness to Forgetting
From the figure, we see that naïve HCNs performed better than MLPs with EWC and than MLPs with HAT. This showed that though continual learning techniques can be powerful, catastrophic forgetting was perhaps more tightly linked to the backbone architectural design of choice.

6. Discussion, Limitations, and Future Work

This paper introduced **Highway**-Connection **Classifier Networks** (HCNs) to prevent forgetting in sequential learning multiple tasks. The proposed design was based on a mathematical analysis on the shortcut connection in the backward pass. We demonstrated that HCNs exhibited higher stability than MLPs, and that the same level of

[4]See [7], and we defaulted regularisation coefficient $\lambda = 3$.

[5]See [15], and we defaulted the externally stored episodic memory as 250 data per task.

[6]The nullification masks were only applied on the linear layers; we prohibited the masks on the gated units of HCNs.

effectiveness in mitigating forgetting cannot be replaced with residual connections. HCNs were also versatile and could be employed in conjunction with the regularisation-based EWC, the replay-based GEM, and the dynamic architecture-based HAT. More importantly, our remarks found that while continual learning techniques were important, it also required a careful selection on the most appropriate backbone network design to minimise forgetting.

In this work, we adopted the style of network compartmentalisation proposed in [13, 14, 16] and limited updated to the classifier networks. Crucially, the classifier networks were only composed of linear layers but a recent paper has shown that convolutional layers were significantly more susceptible to parametric corruption than linear layers [26]. It would hence be important as a future work to verify the validity of applying highway connections on the entirety of the backbone network including the feature extractors.

Acknowledgement

This work was done while the first author was at the Australian National University. The research was supported by the Australian Government Research Training Program (AGRTP) Scholarship.

References

[1] Craig Atkinson, Brendan McCane, Lech Szymanski, and Anthony Robins. **Pseudo-Rehearsal: Achieving Deep Reinforcement Learning without Catastrophic Forgetting**. *Neurocomputing*, 428:291–307. 1

[2] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. **On Tiny Episodic Memories in Continual Learning**. *International Conference on Machine Learning Workshop on Multi-Task and Lifelong Reinforcement Learning*, 2019. 1

[3] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. **Deep Learning for Classical Japanese Literature**. *Conference on Neural Information Processing Systems Workshop on Machine Learning for Creativity and Design*, 2018. 3

[4] Robert M French. **Using Semi-Distributed Representations to Overcome Catastrophic Forgetting in Connectionist Networks**. In *Proceedings of the 13th Annual Cognitive Science Society Conference*, volume 1, pages 173–178, 1991. 1

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. **Deep Residual Learning for Image Recognition**. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2

[6] Sepp Hochreiter and Jürgen Schmidhuber. **Long Short-Term Memory**. *Neural Computation*, 9(8):1735–1780, 1997. 2

[7] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. **Overcoming catastrophic forgetting in neural networks**. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. 1, 3, 4

[8] Alex Krizhevsky. **Learning Multiple Layers of Features from Tiny Images**. In *Tech Report of the University of Toronto*, 2009. 3

[9] Nicholas I Kuo, Mehrtash Harandi, Nicolas Fourrier, Christian Walder, Gabriela Ferraro, Hanna Suominen, et al. **Learning to Continually Learn Rapidly from Few and Noisy Data**. *Meta-Learning and Co-Hosted Competition of the AAAI Conference on Artificial Intelligence; arXiv preprint arXiv:2103.04066*, 2021. 2

[10] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. **Gradient-based learning applied to document recognition**. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 3

[11] Zhizhong Li and Derek Hoiem. **Learning without Forgetting**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017. 1

[12] Zenan Ling and Robert C Qiu. **Spectrum Concentration in Deep Residual Learning: A Free Probability Approach**. *IEEE Access*, 7:105212–105223, 2019. 2

[13] Vincenzo Lomonaco and Davide Maltoni. **Core50: A New Dataset and Benchmark for Continuous Object Recognition**. In *Conference on Robot Learning*, pages 17–26. PMLR, 2017. 1, 2, 5

[14] Vincenzo Lomonaco, Davide Maltoni, and Lorenzo Pellegrini. **Rehearsal-Free Continual Learning over Small Non-IID Batches**. *arXiv preprint arXiv:1907.03799*, 2019. 1, 2, 5

[15] David Lopez-Paz and Marc'Aurelio Ranzato. **Gradient Episodic Memory for Continual Learning**. In *Advances in Neural Information Processing Systems*, pages 6467–6476, 2017. 2, 3, 4

[16] Zheda Mai, Hyunwoo Kim, Jihwan Jeong, and Scott Sanner. **Batch-level Experience Replay with Review for Continual Learning**. In *Computer Vision and Pattern Recognition Workshop*, 2020. 1, 2, 3, 5

[17] Michael McCloskey and Neal J Cohen. **Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem**. In *Psychology of Learning and Motivation*, volume 24, pages 109–165. Elsevier, 1989. 1

[18] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. **Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML**. In *International Conference on Learning Representations*, 2019. 1

[19] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. **iCaRL: Incremental Classifier and Representation Learning**. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 1, 3

[20] Herbert Robbins and Sutton Monro. **A Stochastic Approximation Method**. *The annals of Mathematical Statistics*, pages 400–407, 1951. 3

[21] Anthony Robins. **Catastrophic Forgetting, Rehearsal and Pseudorehearsal**. *Connection Science*, 7(2):123–146, 1995. 1

[22] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. **Progressive Neural Networks**. *arXiv preprint arXiv:1606.04671*, 2016. 1

[23] Joan Serrà, Didac Suris, Marius Miron, and Alexandros Karatzoglou. **Overcoming Catastrophic Forgetting with Hard Attention to the Task**. In *Proceedings of the International Conference on Machine Learning*, 2018. 1, 4

[24] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. **Continual Learning with Deep Generative Replay**. In *Advances in Neural Information Processing Systems*, pages 2990–2999, 2017. 2

[25] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. **Highway Networks**. *arXiv preprint arXiv:1505.00387*, 2015. 2

[26] Xu Sun, Zhiyuan Zhang, Xuancheng Ren, Ruixuan Luo, and Liangyou Li. **Exploring the Vulnerability of Deep Neural Networks: A Study of Parameter Corruption**. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021. 5

[27] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. **Going Deeper with Convolutions**. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 2

[28] Gido M Van de Ven and Andreas S Tolias. **Three Scenarios for Continual Learning**. *the Continual Learning Workshop of Advances in Neural Information Processing Systems*, 2018. 1

[29] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. **Lifelong Learning with Dynamically Expandable Networks**. In *International Conference on Learning Representations*, 2018. 1

[30] Friedemann Zenke, Ben Poole, and Surya Ganguli. **Continual Learning through Synaptic Intelligence**. In *International Conference on Machine Learning*, pages 3987–3995. PMLR, 2017. 1