Preprint. To appear in: Proceedings of the International Joint Conference on Neural Networks (IJCNN 2019), Budapest, Hungary

Effect of Pruning on Catastrophic Forgetting in Growing Dual Memory Networks

Wei Shiung Liew, Chu Kiong Loo
Faculty of Computer Science and Information Technology
University of Malaya
Kuala Lumpur, Malaysia
liew.wei.shiung@gmail.com,
ckloo.um@um.edu.my

Vadym Gryshchuk, Cornelius Weber, Stefan Wermter
Department of Informatics
University of Hamburg
Hamburg, Germany
2gryshch@informatik.uni-hamburg.de,
weber@informatik.uni-hamburg.de,
wermter@informatik.uni-hamburg.de

***Abstract*—Grow-when-required networks such as the Growing Dual-Memory (GDM) networks possess a dynamic network structure, expanding to accommodate new neurons in response to learning novel concepts. Over time, it may be necessary to prune obsolete neurons and/or neural connections to meet performance or resource limitations. GDM networks utilize an age-based pruning strategy, whereby older neurons and neural connections that have not been activated recently are removed. Catastrophic forgetting occurs when knowledge learned by the networks in previous learning iterations is lost due to being overwritten by newer learning iterations, or to the pruning process. In this work, we investigate catastrophic forgetting in GDM networks in response to different pruning strategies. The age-based pruning method was shown to significantly sparsify the GDM network topology while improving the networks ability to recall newly acquired concepts with a slight decrease in performance with respect to older knowledge. A significance-based pruning method was tested as a replacement for the age-based pruning, but was not as effective at pruning even though it performed better at recalling older knowledge.**

***Index Terms*—catastrophic forgetting, grow-when-required networks, network topology**

I. Introduction

Grow-when-required networks [1] are a type of neural network that dynamically allocates new neurons in response to novel inputs, bypassing the need to pre-determine the size and structure of the neural network before training. The growth of the network has to be carefully balanced. The fast-growing network may be able to perfectly encode all learned information but taking up a lot of computational power and storage space. Neuron proliferation is mitigated by adding new neurons to learn sufficiently novel or unique information not already known by the neural network. When an input is otherwise insufficiently dissimilar, the neuron that is closest or most similar to the input is activated and learning is performed by adapting the neuron weights to closely match the incoming input.

In a continuous learning process, the network structure will continue to expand to add new neurons to accommodate new knowledge. Due to hardware or software constraints however, it may be necessary to remove older neurons that contain knowledge that is no longer relevant. Catastrophic forgetting refers to a side-effect of pruning and the neural network learning process that results in loss of existing knowledge. For example, neuron weights that are updated in response to new inputs may not fully match the older inputs used to train the neural network. Also, network pruning may sometimes result in catastrophic forgetting of learned knowledge.

The Growing Dual-Memory (GDM) network [2] consists of two hierarchically arranged recurrent self-organizing networks. The episodic memory layer (G-EM) is highly dynamic, quickly adapting the neural structure and adding new neurons to learn spatio-temporal representations of novel episodic events. The semantic memory layer (G-SM) on the other hand, operates slower in order to construct more compact representations of the episodic events over larger temporal windows. An intrinsic replay system in the GDM constantly replays neural activity patterns to consolidate learned knowledge and mitigate catastrophic forgetting. No pruning mechanism was implemented, however. In a continuous learning scenario, the GDM networks will continue adding neurons over time in response to novel inputs while still retaining older neurons containing obsolete information. Given hardware or software resource constraints, a GDM with excessive topology size will cause either a reduction in performance speed or generalization. An appropriate pruning strategy is required to fulfill several objectives; ensuring that the growth of the GDM topology in response to new knowledge is regulated by pruning obsolete knowledge; and identifying the appropriate neurons and/or synapses to be pruned so that only the least relevant knowledge is removed.

In this work, we investigate the use of several different conditional pruning strategies on the GDM network. Particularly because the GDM consists of two networks with different neural behavior, the G-EM and G-SM networks may require different pruning strategies to achieve optimal performance. The networks performance was benchmarked using several metrics to measure the topology of the network

as well as the retention and acquisition of knowledge. We briefly explain the operations of the GDM network in section II, and the pruning strategies in section III. The experiment is outlined in section IV and the results are presented in section V and discussed in section VI.

II. Growing Dual-Memory Networks

The GDM network [2] consists of two hierarchically-arranged recurrent self-organizing networks for learning spatio-temporal representations from a sequence of images or video. Neurons in the episodic memory are highly dynamic, quickly adapting neural representations using Hebbian learning and adding new neurons to learn the spatio-temporal representations of novel episodic experiences. Neurons in the semantic memory layer gradually develop condensed representations of statistical regularities from episodic events. Both memory layers are modeled as Grow-When-Required networks [1] that adapt neural map plasticity in response to novel sensory observations. While the learning process in the episodic memory layer is unsupervised, the semantic memory layer utilizes learned class labels to regulate the generation of new neurons and neural update rate. Consolidation of existing knowledge was achieved using internally-generated activity patterns in the episodic memory layer to be replayed to the semantic memory, thus mitigating the effect of catastrophic forgetting during incremental learning tasks. The operation of the GDM network is briefly summarized in the next section. For more details, refer to the authors' work [2].

A. Episodic Memory

The episodic memory (G-EM) layer consists of a Gamma-Grow-When-Required self-organizing network [3] with a dynamic number of neurons and synapses in a competitive map that learns the spatio-temporal structure of a multi-dimensional input, preserving its topological properties. The Gamma-GWR determines the winning neuron in response to an input while taking into account the activity of the network and a temporal context. Each neuron in the map consists of a weight vector w_j and a number K^{em} of context descriptors c_j^k for encoding prototype sequence-selective snapshots of the learning input. Given a network with N recurrent neurons, the best matching unit (BMU) b was computed with respect to the learning input $x(t)$ as follows:

$$b = \underset{j \in \mathcal{N}}{\operatorname{argmin}}(d_j) \tag{1}$$

$$d_j = \alpha_0 \|x(t) - w_j\|^2 + \sum_{k=1}^{K} \alpha_k \|C_k(t) - c_{k,j}(t)\|^2 \tag{2}$$

$$C_k(t) = \beta \cdot w_{I-1} + (1 - \beta) \cdot c_{k-1,I-1} \tag{3}$$

$$a(t) = \exp(d_b) \tag{4}$$

where a_i and $\beta \in (0; 1)$ are constants regulating the influence of the learning input relative to previous neural activity, w_{I-1} is the weight of the BMU in the previous learning iteration $t-1$, and $C_k \in \mathbb{R}^n$ is the global context of the network with $C_k(t_0) = 0$, and $a(t)$ is the activity of the network in response to the current learning input. Each neuron has a habituation counter $h_i \in [0, 1]$ representing how frequently it has fired (i.e. activated as BMU), expressed by a habituation rule as:

$$\Delta h_i = \tau_i \cdot \kappa \cdot (1 - h_i) - \tau_i \tag{5}$$

where κ and τ_i are constants regulating the rate of decrease of a neurons habituation counter [1].

When a learning input is presented to the network, a new neuron is inserted if the activity and habituation of the BMU are smaller than the activity threshold a_T and habituation threshold h_T respectively. Training of the activated neuron is carried out by adapting the weight vectors and context descriptors of the BMU according to:

$$\Delta w_i^{em} = \epsilon_i \cdot h_i \cdot (x(t) - w_i^{em}) \tag{6}$$

$$\Delta c_{k,i}^{em} = \epsilon_i \cdot h_i \cdot (C_k^{em}(t) - c_{k,i}^{em}) \tag{7}$$

where ϵ_i is a constant learning rate. Topological neighbors were updated at a significantly lower learning rate.

B. Semantic Memory

The semantic memory layer (G-SM) consists of a Gamma-GWR network combining bottom-up drive from the G-EM and top-down, task-relevant signals to develop overlapping representations over a larger temporal scale. The neural activity from the G-EM (i.e. the BMU of the G-EM in response to a learning input) is used as input to the G-SM. Neurogenesis is regulated by imposing another condition: a new neuron is added only if the activity of a habituated BMU is below a threshold, and if the label of the learning input is different from the winner label of the BMU. If the BMU correctly predicts the class label, then the update rate of the weight vectors and the context descriptors were decreased by a factor ϵ^c. Thus (6) and (7) become:

$$\Delta w_i^{sm} = \epsilon_i \cdot h_i \cdot \epsilon^c \cdot (w_b^{sm} - w_i^{sm}) \tag{8}$$

$$\Delta c_{k,i}^{sm} = \epsilon_i \cdot h_i \cdot \epsilon^c \cdot (C_k^{sm}(t) - c_{k,i}^{sm}) \tag{9}$$

The additional factors regulating neurogenesis and neural update can be seen as a regularized learning process where task-relevant signals create new neurons only when the network misclassifies the class label of the input, and reducing the learning rate of bottom-up observations when the prediction is correct. As a result, the G-SM network will develop more compact, overlapping representations of the learned concepts that cannot reconstruct episodic events,

but are activated in response to semantically related input (i.e. the same neuron may be activated for the same object seen from different angles).

III. Network Pruning

The following network pruning strategies were proposed for the GDM network. Synapse-aging pruning [4] removes synapses that have not been activated for a length of time. Habituation-based pruning [5] considers how often a neuron had been activated and removing neurons that were rarely activated. Significance-based pruning [6] computes the significance of synapses based on the activation function of the neurons.

A. Pruning by Synapse Age

Synapses are created connecting any two fired neurons in response to learning inputs. Each synapse has an aging counter that increments with each learning iteration, and resets to zero whenever the two connecting neurons are activated simultaneously. Synapses with ages exceeding a threshold are pruned. Similarly, neurons that have all their synapses pruned and are thus isolated will also be removed.

Selecting the appropriate age threshold is non-trivial problem to avoid catastrophic forgetting, especially in incremental learning scenarios where neurons coding for consolidated knowledge might not fire for a large number of iterations.

B. Pruning by Neuron Habituation

An alternate metric to synapse aging was proposed by Gryshchuk [5], bypassing the problem of selecting the age threshold. Neuron habituation is a mechanism to gradually desensitize a neuron after repeated activation [1]. The habituation value of a neuron can thus be associated to the relevance or importance of the information encoded in the neuron. Neurons that have been activated frequently in response to learning inputs will typically have a lower habituation value, given by the habituation equation (5). The proposed method defines the removal of a neuron as a threshold function of its habituation:

$$v = \mu(H) + \sigma(H) \tag{10}$$

where H is a vector representation of the habituation of all the neurons in the network, μ is the mean function, and σ is the standard deviation. Neurons with habituation values above the threshold will be pruned.

C. Pruning by Neuron and Synapse Significance

Scardapane et al. [6] proposed an unsupervised sparsification method for evaluating the significance of neurons and connecting synapses in an echo state network architecture. The pruning strategy considers the relative significance of a synapse in terms of the correlation between its input and output neurons. The significance of a synapse at a particular time instant n is defined as:

$$s_{ij}(n) = \frac{1}{T} \sum_{z=n-T}^{n} \frac{(x_i(z-1) - \hat{\mu}_x)(x_j(z) - \hat{\mu}_x)}{\hat{\sigma}_x^2} \tag{11}$$

where T is a time interval chosen *a priori*, and $\hat{\mu}_x$ and $\hat{\sigma}_x$ are the empirical estimations of the mean and standard deviation of the neuron states. x_i and x_j denote the state of the neurons i and j in the reservoir respectively. Generally the time instant is incremented as each learning input is presented to the network during training. The probability that the synapse between neurons i and j would be removed is represented as:

$$p_{ij}(n) = \exp\left(-\frac{|s_{ij}(n)|}{t(n)}\right) \tag{12}$$

where $t(\cdot)$, also called the temperature parameter of the system, is a positive, monotonically decreasing function of n. This is to ensure that the probability of removing a synapse is higher at the beginning of the network learning process and decreases over time. Temperature $t(n)$ is defined as:

$$t(n) = \alpha^{\frac{n}{Q}-1} t_0 \tag{13}$$

for conducting synapse pruning at every Q time instants. The temperature parameter is scaled by a factor α at every pruning step, given by $(\frac{n}{Q}) - 1$, and t_0 is the initial temperature value chosen *a priori*. The significance of a neuron is defined as the weighted average of the significance of its connecting synapses:

$$s_j(n) = \frac{1}{2|\mathcal{J}_j(n)|} \sum_{z \in \mathcal{J}_j(n)} s_{jz}(n) + \frac{1}{2|\sigma_j(n)|} \sum_{z \in \sigma_j(n)} s_{zj}(n) \tag{14}$$

where $|\cdot|$ denotes the cardinality of the set. Neurons with low significance, i.e. having insignificant synapse connections, will be denoted with a small value in the equation, with the corresponding probability of removal using the same equation as (12) but for neurons instead of synapses.

While the significance-based pruning method was originally introduced for echo state networks, here it is modified for recurrent weight networks. In (11), the state values of neurons in GDM networks are calculated using the GDMs neuron activity in (4). The pruning strategy as outlined in (11) is an online strategy that evaluates neurons based on their activities in a moving time window, and is suitable for GDM networks operating in a continuous learning environment. The settings for T and Q should be considered in the context of the application of the GDM. For example, GDMs that were constantly trained with large quantities of samples in a short time would require a large T parameter so that significant neurons are not prematurely removed. The Q parameter balances pruning between earlier and later training samples. A low Q may cause excessive pruning among

earlier training samples, causing catastrophic forgetting of prior information, while neglecting later training samples, producing low generalization of newly acquired information.

IV. Experiment

An experiment was conducted to evaluate the performance of the GDM network when subjected to different pruning strategies: no pruning, synapse-aging pruning, neuron-habituation pruning, and significance pruning. A dataset for continuous object recognition from video sequences is used for benchmarking using an incremental learning approach.

A. Dataset Pre-processing

The CORe50 dataset [7] consists of 50 objects recorded in 11 different environment conditions, backgrounds, and object poses. For this experiment, sessions 1 and 2 were used for training while 3 was used for testing. Frames were selected for processing from each video sequence at 1 frame-per-second. A deep convolutional network VGG16 [8] was used to extract a 256-dimension feature vector from each image frame. Relief-F was conducted for feature selection.

An incremental learning approach was formulated as follows. Of the ten object classes in the dataset, half were selected for the first training and testing session. Of the remaining five object classes, each object was placed in a separate training session. During the experiment, each session was presented sequentially to the GDM network for training and testing. After each training and testing session, the network was benchmarked to measure catastrophic forgetting using indices explained in the next section. In this manner, the GDM network trained with an initial knowledge base will encounter and learn new and unknown objects and then tested on previous data to measure the loss of information as well as new information gained.

B. Evaluation Metrics

Network performance and catastrophic forgetting were measured using a number of indices as proposed by Kemker et al. [9] and Chaudry et al. [10]. Immediately after the network was trained using the first training session, testing was performed using the same training input. Testing accuracy was assumed as the networks ideal performance: acc_{ideal}.

When trained with a new object class in subsequent training sessions, the network was tested using the testing data for each of the prior training sessions, and labeled as $\text{acc}_{k,j}$: the accuracy evaluated on the held-out test set of the j^{th} session ($j \leq k$) after training the network incrementally from sessions 1 to k. Testing was also conducted for all prior test sets up to the current session simultaneously and labeled as $\text{acc}_{k,all}$.

Intransigence [10] was measured relative to a standard classification model which had access to all the datasets at all times. The reference classification model was tested with the testing data of the k^{th} session: acc_k^*. Intransigence for the k^{th} session was then calculated as:

$$I_k = acc_k^* - acc_{k,k} \tag{15}$$

As intransigence was defined as the difference between the accuracy of an incremental-learned network and a reference model, negative intransigence (i.e. $I_k < 0$) implies that incremental learning up to session k positively impacts the models knowledge about it.

Forgetting for a specific training session was defined as the difference between the maximum knowledge gained about the session throughout the learning process in the past, and the knowledge the network currently has about it [10]. Quantifying forgetting for the j^{th} session after incrementally training the network up to session k:

$$f_j^k = \max_{l \in \{1,\ldots,k-1\}} acc_{l,j} - acc_{k,j}, j < k \tag{16}$$

The average forgetting at the k^{th} training session is then written as:

$$F_k = \frac{1}{k-1} \sum_{j=1}^{k-1} f_j^k \tag{17}$$

The metrics proposed by Kemker et al. [9] instead measure the networks knowledge retention and acquisition. The networks ability to recall the knowledge from the first training session is represented as:

$$\Omega_{base} = \frac{1}{k-1} \sum_{j=2}^{k} \frac{acc_{j,1}}{acc_{ideal}} \tag{18}$$

The networks ability to immediately recall newly acquired knowledge is calculated as:

$$\Omega_{new} = \frac{1}{k-1} \sum_{j=2}^{k} acc_{j,j} \tag{19}$$

The networks ability to retain prior knowledge and acquire new knowledge is represented as:

$$\Omega_{all} = \frac{1}{k-1} \sum_{j=2}^{k} \frac{acc_{j,all}}{acc_{ideal}} \tag{20}$$

Normalization was performed against acc_{ideal} to facilitate a fair comparison between different datasets.

C. Methodology

The experiment was conducted to study the networks ability to acquire new knowledge and measure catastrophic forgetting. The settings for the GDM hyperparameters are listed in Table I and are described here briefly.

Insertion thresholds a_T set the minimum activity of a neuron in response to an input in order to add a new neuron. G-EM networks typically have higher thresholds (i.e. easier to add new neurons) in order to encode more fine-grained and non-overlapping representations as compared to

TABLE I: Hyperparameters for GDM Networks

Hyperparameters	Value
Insertion thresholds	$a_T^{EM} = 0.3$ $a_T^{SM} = 0.001$
Habituation	$h_T = 0.1$ $\tau_b = 0.3$ $\tau_n = 0.1$ $\kappa = 1.05$
Context descriptors	$K^{EM} = 2$ $K^{SM} = 2$
Temporal context	$\alpha = [0.67, 0.25, 0.09]$ $\beta = 0.7$
Learning rates	$\epsilon_b = 0.5$ $\epsilon_n = 0.005$

G-SM networks. The habituation hyperparameters control the rate in which a neuron is habituated after being fired. Setting a strict habituation condition (i.e. high h_T, τ, and κ) will result in fired neurons quickly becoming habituated and the necessity for additional neurons to be added. Context descriptor parameters are used for encoding the spatio-temporal structure of the input. Setting a large value would allow neurons to be encoded with longer temporal sequences, but with a potential for over-generalization. The temporal context parameters α and β modulate the influence of the current input with respect to previous neural activity and the global context of the network. Learning rates ϵ control how much the neurons adapt in response to training inputs.

The efficacy of the selected pruning method should take into consideration the parameter settings of the GDM. Using the pruning methods in this study as an example: neurons in G-EM networks encode specific episodic prototypes. Using a strict pruning method for G-EM may remove dormant neurons that act as temporal links between other neurons, resulting in catastrophic forgetting. However, a strict habituation function may produce similar neurons that may act as redundancies. The interplay between the various hyperparameters drastically affects the behavior of the GDM, and consequently, a pruning methodology that works in one application may not be applicable with different hyperparameter settings.

In this work however, parameter optimization for G-EM and G-SM was not performed in this experiment and was set following the authors work [2]. The intrinsic replay mechanism was not used in this experiment. With replay, the significance-based pruning method will be affected considering (13), whereby the frequency of pruning is a function of the number of training inputs presented to the network. Implementing intrinsic replay will, therefore, bias most of the pruning towards the earlier training sessions. Alternately, this can be addressed by setting Q to a larger value to accommodate the increased number of training iterations from the intrinsic replays.

While selecting an appropriate age threshold for synapse-aging pruning is a non-trivial problem, in this experiment we tested several values as fractions of the number of learning iterations of the dataset. For example, setting the age threshold as 50% of the dataset size for a dataset with 5000 training samples would set the maximum age of a synapse to 2500. We tested for several age thresholds equal to [50%, 60%, 70%, 80%, and 90%] of the number of training data in the dataset.

Habituation-based pruning was conducted at the end of each training and testing session, i.e. after the GDM network was presented with a new object class.

For significance-pruning, two scaling factor parameters regulate the pruning of neurons and synapses respectively. We tested all possible combinations of value pairs for the scaling parameters in the interval range $[0.1, 1.0]$. Other parameters such as n, Q, and t_0 in (13) were set following the authors work [6].

The GDM networks and the experiment were coded in Python. Random elements were minimized by setting the random number generator to a predetermined seed value just prior to running the experiment for each pruning method.

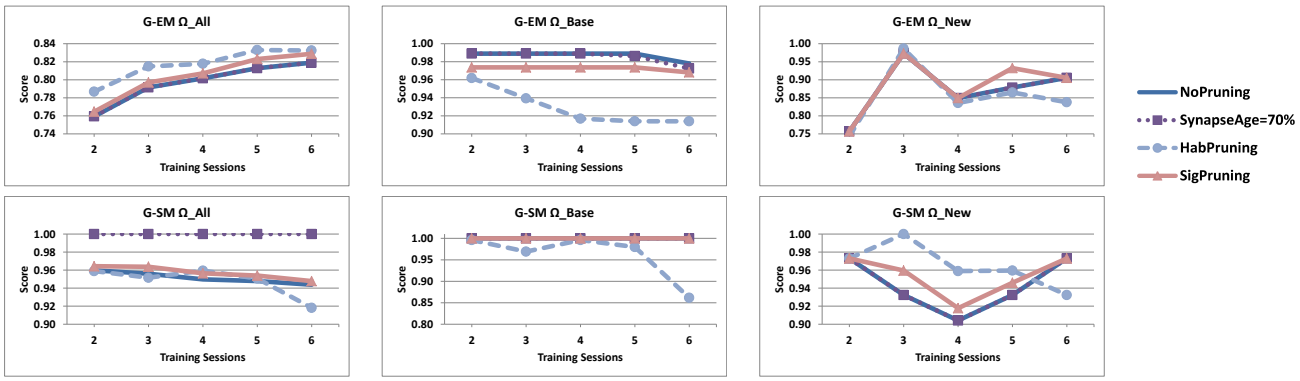

Fig. 1: Metrics of G-EM (top row) and G-SM networks (bottom row) for Ω_{all} (left column), Ω_{base} (middle column), and Ω_{new} (right column)

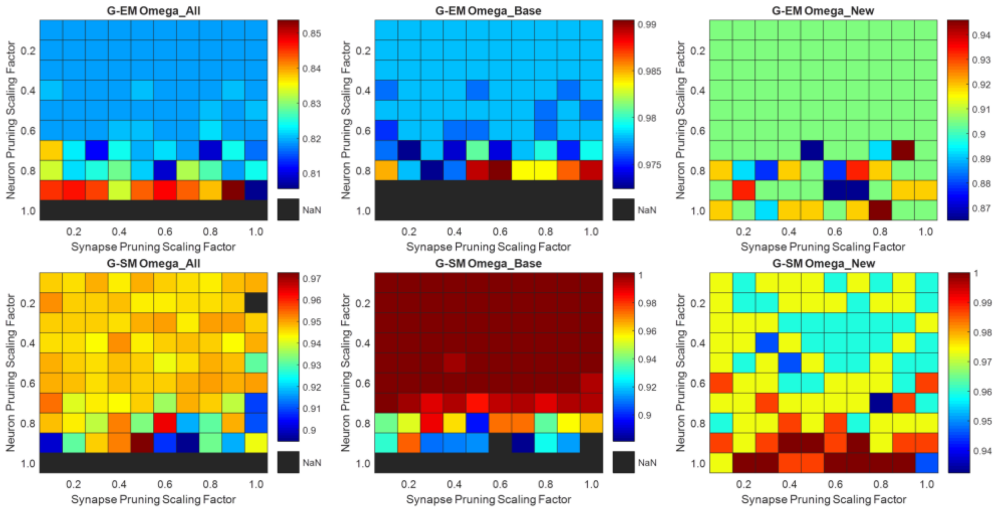

Fig. 2: Performance metrics of G-EM (top row) and G-SM networks (bottom row) using significance-based pruning for Ω_{all} (left column), Ω_{base} (middle column), and Ω_{new} (right column). Extreme outliers were excluded. Values range from poor (blue) to good (red).

V. Experiment Results

A. Generalization Performance

Generalization performance of a network is characterized using three metrics: the networks ability to retain knowledge from the first training session after learning new object classes (18), the ability to immediately recall a newly learned object class (19), and the networks generalization performance on past and present knowledge (20). Fig. 1 compares the metrics for different pruning methods after each training session. Synapse-aging pruning and significance-based pruning were represented using the best overall result.

For synapse-aging pruning, setting the age threshold to 60% or lower significantly affected the networks ability to recall the first training session with successive training. As the GDM in this experiment does not use intrinsic replay, neurons created during the first training session are rarely activated in later sessions, and are subsequently pruned when their ages exceeded the threshold. Setting the age threshold to 70% or higher was able to avoid significant catastrophic forgetting. 80% is the ideal age threshold for maximizing pruning while minimizing reduction in generalization performance: in G-EM networks, Ω_{all} and Ω_{new} were equivalent to that of unpruned networks while Ω_{base} was slightly lower (0.9753 vs 0.9780). In G-SM networks, Ω_{base} and Ω_{new} were equivalent to unpruned networks while Ω_{all} was slightly higher (0.9451 vs 0.9438).

Habituation-based pruning in G-EM networks produced worse results for Ω_{base} (0.9140 vs 0.9780) and Ω_{new} (0.8378 vs 0.9054), while Ω_{all} was slightly better (0.8325 vs 0.8190) as compared to unpruned networks. This suggests that habituation-pruning was able to identify outlier neurons, resulting in slightly worse performance in specific domains but with improved overall generalization. In G-SM networks, habituation-based pruning produced worse generalization performance in all three metrics compared to unpruned networks.

Fig. 2 shows heatmaps of the GDM networks performance for every combination of neuron pruning and synapse pruning scaling factors for the significance-pruning method. For overall testing performance Ω_{all}, G-EM achieved peak results when neuron pruning scaling factor was set to 0.9. G-SM, on the other hand, showed better results on average with less neuron pruning, except in two cases (n=0.8, s=0.6; and n=0.9, s=0.5) where testing performance significantly exceeded the other outcomes with the same neuron pruning scaling factor. For testing performance on the first session Ω_{base}, G-EM showed good results at neuron pruning scaling factor 0.8 and below, while G-SM has better results at 0.7 and below. For testing performance on the most recently learned object class Ω_{new}, G-EM was comparatively less influenced by the scaling factors, with a few exceptions producing slightly above average performance. G-SM, on the other hand, performed better with large neuron pruning.

To summarize for significance-pruning, G-EM networks are able to reach good generalization when neuron pruning scaling factor was set to a high value. While overall performance improved with a larger neuron pruning value, some knowledge from earlier training sessions are lost when the older neurons were considered less significant and were pruned. Setting neuron pruning too high may disrupt the temporal connections in the G-EM, resulting in a drastic reduction in generalization. In addition, more pruning did not guarantee better recall for newly learned object classes. Pruning G-SM networks may be counterproductive as the

Fig. 3: Comparison of Forgetting and Intransigence scores for G-EM and G-SM networks using various pruning methods.

topology is already consolidated. As seen in Fig. 2, setting a high pruning scaling factor resulted in a trade-off between the generalization performance for older consolidated knowledge and for newly acquired knowledge.

B. Network Topology

This section explores the effect of the pruning strategies on the topologies of the G-EM and G-SM networks. The number of neurons and synapses of the resulting networks were compared against G-EM and G-SM networks that were trained without any pruning.

For the synapse-aging pruning strategy, setting the age threshold to a low value resulted in significant synapse and neuron pruning, at the cost of loss of network generalization for earlier knowledge. Setting the threshold to 70

In comparison, the habituation-based pruning method resulted in 63

For significance-based pruning in G-EM networks, the synapse pruning scaling factor had minimal effect on the topology compared to the neuron pruning scaling factor. Even so, the pruned G-EM was nearly equivalent to the unpruned G-EM unless the neuron pruning scaling factor was set to 0.7 or higher, with 0.9 and 1.0 producing significant catastrophic pruning. At 0.8, approximately 2

In G-SM networks however, a significant reduction in network topology was achieved even with minimal scaling factors. The optimum scaling factors to minimize network size while maintaining equal or better generalization than unpruned G-SM networks is 0.6 for neuron pruning and 0.8 for synapse pruning, resulting in a reduction of 20

C. Forgetting and Intransigence

An ideal network would have a score close to -1 for both forgetting and intransigence to denote the positive impact of training sessions on the networks generalization ability. In Fig. 2, the drastic increase of forgetting for the synapse-aging pruning method was the result of pruning older neurons containing knowledge encoded from earlier training sessions.

Habituation-based pruning showed good intransigence scores in G-EM and G-SM for the 3rd and 4th training sessions. However from the 5th session onwards, pruning resulted in higher catastrophic forgetting and intransigence, especially in G-SM networks where over 90

Fig. 4 shows the forgetting and intransigence scores for significance-based pruning. In G-EM networks, setting the neuron pruning scaling factor lower than 0.8 resulted in equivalent forgetting scores to unpruned networks, while 0.8 produced better scores, and higher than 0.8 created significant catastrophic forgetting. Similarly for intransigence, the G-EM was equivalent to unpruned networks at neuron scaling factors lower than 0.7, while peak intransigence occurred with maximum neuron pruning and selected synapse pruning scaling factors. For G-SM networks, less neuron and synapse pruning was desirable to reduce catastrophic forgetting. Intransigence however was better with significant neuron pruning. Precise tuning for synapse scaling factor was required to achieve good intransigence.

VI. Discussion and Conclusion

This work investigated the effects of various pruning methods on the episodic memory and semantic memory in Growing-Dual-Memory networks. The pruning methods used in this study were synapse-aging, neuron habitation, and significance-based pruning. The performance of the pruned networks was benchmarked against that of the unpruned networks using seven performance metrics: number of neurons, number of synapses, testing accuracy of prior

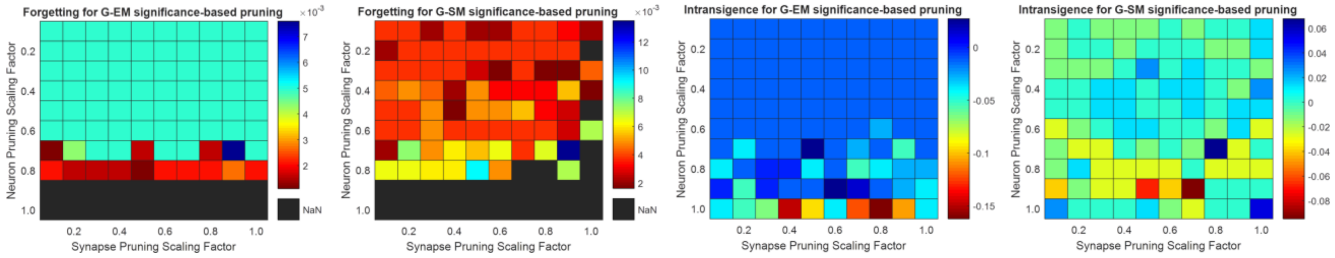

Fig. 4: Forgetting and Intransigence for G-EM and G-SM using significance-based pruning. Extreme outlier values were excluded. Scores range from poor (blue) to good (red).

knowledge, testing accuracy of current knowledge, overall testing accuracy, forgetting, and intransigence. An incremental learning scheme was conducted to simulate the GDM learning new object classes over time in addition to its initial knowledge base. After a training session in which a novel object class was presented, the GDM was benchmarked.

Networks using synapse-aging pruning rarely outperformed unpruned networks. Precise tuning of the age threshold is needed to ensure that older neurons which contain important knowledge from earlier training sessions are not pruned. This may not be possible in lifelong learning scenarios, where there is no fixed size for training data.

Pruning using neuron habituation was proposed as an alternative to address the problems of pruning by age. As neuron habituation is affected by activation frequency, neurons that were rarely activated may be considered redundant and be safely removed. From the experiment, habituation-pruning removed outliers in G-EM resulting in slight losses in testing accuracy. In G-SM however, habituation-pruning resulted in more significant catastrophic forgetting, as neurogenesis was more regulated for a compact topology. In addition, as the topology becomes larger, newer neurons may be pruned before achieving sufficient habituation, resulting in poor testing accuracy of newly acquired knowledge. This may be addressed in future work by a hybrid of age and habituation to give newer neurons time to settle down before being evaluated for pruning.

Significance-based pruning evaluates synapses and neurons by their recent activations in response to learning inputs. Two scaling factors control the rate in which neurons and synapses were pruned. Precise tuning was required to avoid excessive pruning. As observed from the experiment, G-EM and G-SM networks responded differently to neuron and synapse pruning. G-SM networks were more severely affected by neuron and synapse pruning. On the other hand, G-EM networks were generally equivalent to unpruned networks unless the neuron scaling factor was set to a high value, although excessive pruning can also result in poor performance. When benchmarking G-EM and G-SM networks against unpruned networks using the seven performance metrics and tallying up the wins, draws, and losses, optimum scaling factors were found to be 0.8 for neuron and synapse pruning for G-EM networks, and 0.6 and 0.1 for neuron and synapse pruning for G-SM networks.

In conclusion, pruning strategies have to take into consideration the characteristics of the GDM networks. In G-EM where neurongenesis is a common occurrence, pruning strategies should take into account multiple factors such as age, habituation, and significance before deciding which neuron to prune. In G-SM however, each neuron encodes consolidated information that may not be as expendable, and thus a more strict and careful pruning strategy is required.

Acknowledgment

This research was supported by the Georg Forster Research Fellowship for Experienced Researchers from Alexander von Humboldt-Stiftung/Foundation.

References

[1] S. Marsland, J. Shapiro, and U. Nehmzow, A self-organising network that grows when required, Neural Networks, vol. 15, no. 89, pp. 10411058, 2002.

[2] G. I. Parisi, J. Tani, C. Weber, and S. Wermter, Lifelong Learning of Spatiotemporal Representations with Dual-Memory Recurrent Self-Organization, arXiv preprint arXiv:1805.10966, 2018.

[3] G. I. Parisi, J. Tani, C. Weber, and S. Wermter, Lifelong Learning of Humans Actions with Deep Neural Network Self-Organization, Neural Networks, vol. 96, pp. 137149, 2017.

[4] T. M. Martinetz, K. J. Schulten, A neural-gas network learns topologies, in Artificial Neural Networks, T. Kohonen, K. Makisara, O. Simula, and J. Kangas, Eds. North-Holland, Amsterdam, 1991, pp. 397-402.

[5] V. Gryshchuk, Learning to forget in self-organizing memory, unpublished, University of Hamburg, 2018.

[6] S. Scardapane, D. Comminiello, M. Scarpiniti, and A. Uncini, Significance-Based Pruning for Reservoirs Neurons in Echo State Networks, in Advances in Neural Networks: Computational and Theoretical Issues, Springer, Cham. 2015, pp. 31-38.

[7] V. Lomonaco, and D. Maltoni, CORe50: A New Dataset and Benchmark for Continuous Object Recognition, Proceedings of the 1st Annual Conference on Robot Learning, PMLR 78:17-26, 2017.

[8] K. Simonyan, K., and A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556, 2014.

[9] R. Kemker, M. McClure, A. Abitino, T. Hayes, and C. Kanan, Measuring catastrophic forgetting in neural networks, arXiv preprint arXiv:1708.02072, 2017.

[10] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. Torr, Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence, arXiv preprint arXiv:1801.10112, 2018.