UNSUPERVISED GENERATIVE VARIATIONAL CONTINUAL LEARNING

Liu Guimeng[2], *Guo Yang*[1], *Cheryl Wong Sze Yin*[1]
Ponnuthurai Nagartnam Suganathan[2,3], *Ramasamy Savitha*[1*]

[1]Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore
[2] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
[3]KINDI Center for Computing Research, College of Engineering, Qatar University, Qatar

ABSTRACT

Continual learning aims at learning a sequence of tasks without forgetting any task. While most of the existing literature in continual learning is aimed at class incremental learning in a supervised setting, there is an enormous potential for unsupervised continual learning using generative models. This paper proposes a combination of architectural pruning and neuron addition in generative variational models toward unsupervised generative continual learning (UGCL). Evaluations on standard benchmark data sets demonstrate the superior generative ability of the proposed method.

Index Terms— Continual Learning, Unsupervised, Variational Inference

1. INTRODUCTION

Continual learning is aimed at learning a sequence of tasks, without catastrophically forgetting any task [1, 2]. This is accomplished through regularization, network architectural adaptation or replay of samples from past tasks. Some methods for regularization include the Laplace Propagation (LP) present Laplace approximation to evaluate the importance of parameters [3], Elastic weight consolidation (EWC) that use Fisher information matrix to identify the importance of each parameter [4], Synaptic Intelligence (SI) that identifies the important parameters those contribute more to the loss [5]. Recently, Bayesian inference based approaches for regularization based variational continual learning (VCL) is gaining traction. In VCL, the prior of new task is restricted to the posterior of the network trained with past tasks [6].

In addition to regularization, architectural adaptations that include iterative pruning and binary masks to prevent any updates on important weights [7] are also a topic of interest. Such models require precise estimation of important parameters. PackNet [8] and UCB [9] preserve the most important representations of the past tasks through weight pruning. While PackNet is based on preserving the important weights with large magnitude, UCB leverages the uncertainty estimates in Bayesian neural networks for estimating importance. While deep discriminative networks were very common in the early research on continual learning, methods developed for discriminative models are not directly extendable to the generative setting due to the difference in the architecture and objective function [10]. However, deep generative models have become popular due to their practicable modeling of high-dimensional data distributions. Therefore, there is a need to develop deep continual learning approaches based on generative approaches. Deep Bayesian Unsupervised Lifelong Learning (DBULL) proposes an approximate Bayesian inference algorithm that perform knowledge preservation by efficiently representing latent representations [11]. This method relies on an evolving clustering approach to adapt the representations of an unsupervised model, with the streaming data. However, it does not adapt its architecture during the learning process. Furthermore, the Variational Continual Learning [6] uses multiple decoders corresponding to multiple tasks, resulting in a huge network and hence, heavy computation.

In this paper, we propose a task agnostic, unsupervised generative continual learning (UGCL) approach based on variational inference. The algorithm uses weight pruning, neuron addition and regularization to learn a sequence of tasks. Regularization is aimed at freezing the important parameters to preserve the learned knowledge and free up redundant parameters for learning new tasks. Freeing up redundant parameters is enabled through pruning, where the most important parameters are estimated through the uncertainty of the weight parameters. When we have depleted of all the network resources by continual freezing and learning of subsequent tasks, we also add neurons to the network to allow the model to always have sufficient free parameters for learning more tasks. In addition, the network weights are also fine-tuned to compensate for the decline in performance caused by pruning redundant parameters. In all these, we use a variational auto-encoder with a single encoder-decoder. This results in a compact network. We evaluate the ability of continual unsupervised representation of the proposed method, in comparison to other continual learning approaches, on two data sets. Performance studies demonstrate the potential of the proposed method for efficient unsupervised continual learning.

2. BACKGROUND: GENERATIVE VARIATIONAL CONTINUAL LEARNING MODEL

In this section, we will briefly introduce variational auto-encoder (VAE) and Bayesian inference.

2.1. Preliminaries: Variational Continual Learning

A variational auto-encoder (VAE) is an unsupervised Bayesian model. For observed data x and latent variable z, a VAE decoder can be modeled by prior $p(z)$ and desired posterior distribution $p(x|z,\theta)$. In the batch form of given data $D_t = \{x_t^{(n)}\}_{n=1}^N$ for certain task t, the optimization of parameters in VAE is proceeded by maximizing the variational lower bound with respect to the variational parameters of the decoder θ and encoder ϕ:

$$\mathcal{L}_{VAE}^t(\theta,\phi) = \sum_{n=1}^{N_t} \mathbb{E}_{q_\phi(z_t^{(n)}|x_t^{(n)})}\left[\log \frac{p(x_t^{(n)}|z_t^{(n)},\theta)p(z_t^{(n)})}{q_\phi(z_t^{(n)}|x_t^{(n)})}\right] \quad (1)$$

However, VAEs does not offer parameter uncertainty estimates, which are necessary in continual learning setting for weighting information learnt from previous data, Bayesian inference is introduced in variational continual learning (VCL). VCL is based on Bayesian neural networks (BNN), whose parameters are represented by distribution $P(y|\theta,x)$ (usually, Gaussian) over possible value, parameterized as $\theta = (\mu,\sigma)$. Therefore, training a BNN is to compute the posterior $p(\theta|x)$, while the prior is drawn from $p(\theta)$. As the posterior is intractable and approximate inference is often required, the true posterior is approximated as $q_t(\theta) \approx p(\theta|D_{1:t})$ with observing t dataset. The approximate posterior q_t is obtained by maximizing the full variational lower bound [6]:

$$\mathcal{L}_{VCL}^t(q_t(\theta),\phi) = \mathbb{E}_{q_t(\theta)}\left[\mathcal{L}_{VAE}^t(\theta,\phi)\right] - \mathrm{KL}(q_t(\theta)||q_{t-1}(\theta)) \quad (2)$$

The VCL is a "multi-head" generative model with shared network parameters. The "head networks" including task-specific encoders to compress input data to latent variable z and the lower-level of decoders to generate the intermediate level representations h from z. The global-level shared representations of decoder parameters across multiple tasks and generates observations x from h. However, a "multi-head" network requires more capacity than a "single-head" network, and the capacity will increase further as new tasks introduced to the network. In this paper, we proposed a compact continual learning model whose decoder do not contain task-specific part. Noted that encoders are not required during inference, the proposed approach greatly reduce the network capacity.

3. UNSUPERVISED GENERATIVE CONTINUAL LEARNING

Our proposed Unsupervised Generative Continual Learning (UGCL) uses regularization and architectural adaptation strategies for efficient unsupervised continual learning as shown in algorithm 1. Assume that the model has been trained to represent tasks $1,\dots t-1$ and is currently being presented to learn task t. We prune insignificant neurons after training task $t-1$ to ensure sufficient network resources for the subsequent task. As weights in the BNN are approximated as a Gaussian distribution, we estimate the signal-to-noise ratio (SNR) of individual weights. SNR is a classic measurement of mean and standard deviation [9] ($\Omega = \mathrm{SNR}(\mu,\sigma) = |\mu|/\sigma$), is used to measure the importance of the weight parameter. The higher the SNR, the more important the parameter is for the given task. We introduce a gradual pruning of the network where 67% of the weights are pruned after task 1 and 90% are pruned for each subsequent task. This is based on the premise that as the number of tasks increases, the percentage of significant weight parameters decreases. The important weight parameters are frozen, and shared for representation of subsequent tasks. As pruning can result in catastrophic forgetting of previous tasks, we fine-tune the pruned network. During inference, a binary mask that denotes the important parameters of certain tasks is applied to the weight, so that the network state matches the one learned during training. Thus, UGCL learns continually through weight pruning and fine-tuning. If, after training $t-1$ tasks, all the weight parameters are deemed significant, we add an additional 10% neurons to the existing network to train the subsequent task. Suppose VCL approach is used in UGCL framework, we present the optimization of unsupervised generative variational continual learning (UGVCL):

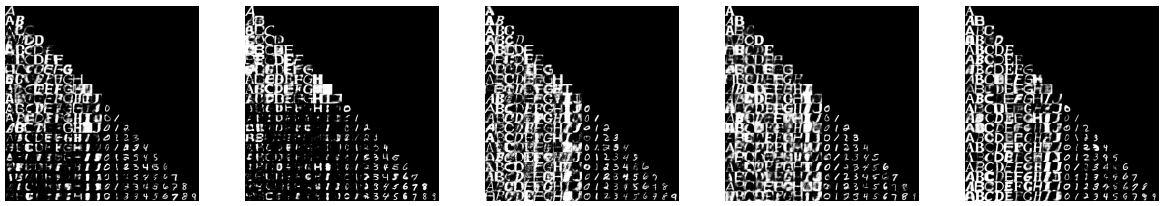
$$\mathcal{L}_{UGVCL}(q_t(\theta_f^{t-1},\theta_p^t),\phi) = \mathbb{E}_{q_t(\theta_f^{t-1},\theta_p^t)}\left[\mathcal{L}_{VAE}^t(\theta_f^{t-1},\theta_p^t,\phi)\right] - \mathrm{KL}(q_t(\theta_f^{t-1},\theta_p^t)||q_{t-1}(\theta_f^{t-1})) \quad (3)$$

where θ_f and θ_p denotes the frozen and pruned parameters, and are are complementary to each other.

Algorithm 1 Unsupervised Generative Continual Learning

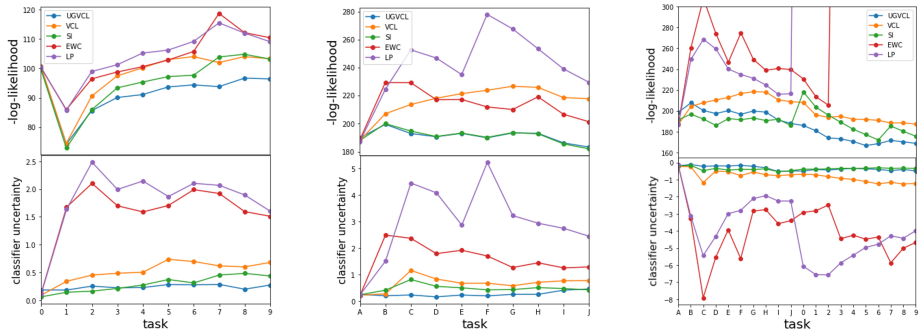
Input: Training data $D = \{x^{(n)}\}_{n=1}^N$, trainable parameters (μ, σ), learning rate α
Output: output result

Initialize μ and σ
$\theta = \mu + \sigma \circ \epsilon$ ◁ θ represented weight w and bias b
for every task t **do**
  if insufficient free parameters **then**
    add neurons with predetermined quantity
  while loss not plateaus **do**
    $\epsilon \in \mathcal{N}(0,I)$
    $g_1 = \nabla\mathcal{L}_{UGCL}(q_t(\theta_f^{t-1},\theta_p^{t-1}),\phi)$
    $\theta_p^{t-1} = \theta_p^{t-1} - \alpha g_1$ ◁ training
    for each parameter j in each layer l **do**
      $\Omega[j] = \mathrm{SNR}(\mu[j],\sigma[j])$
      if $\Omega[j] \notin$ top $k\%$ of Ω in l **then**
        $\theta_p^t = \theta_p^{t-1}[j] = 0$ ◁ weight pruning
      else
        $g_2 = \nabla\mathcal{L}_{UGCL}(q_t(\theta_f^{t-1},\theta_p^{t-1}[j]),\phi)$
        $\theta_f^t = (\theta_f^{t-1},\theta_p^{t-1}[j] - \alpha g_2)$ ◁ fine-tuning
  re-initialize θ_p^t
return result



(a) EWC (b) LP (c) SI (d) VCL (e) UGVCL

Fig. 1. Generated images from each of the generators after training notMNIST + MNIST. Each of the columns shows the images generated from a specific task's generator, and each row represent the generations from trained task.



(a) MNIST (b) notMNIST (c) notMNIST + MNIST

Fig. 2. Overall testing result on MNIST, notMNIST and MNIST+notMNIST (the lower the better)

4. RESULTS

We evaluate the performance of the UGCL, in comparison to other state of the art continual learning approaches, for unsupervised representation. Our comparisons are based on the reconstruction error of the network for individual tasks.

4.1. Experiment setting and data description

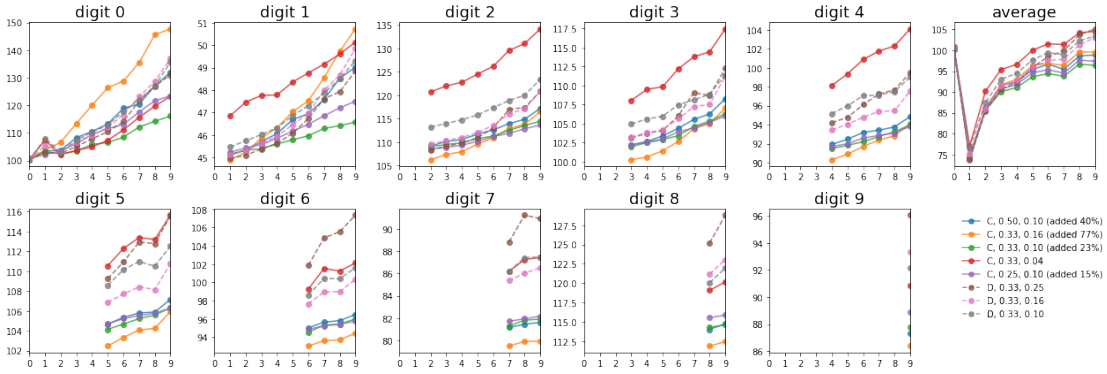
In this paper, we perform three image generation experiments to demonstrate the effectiveness of the proposed method. We first experiment on MNIST and notMNIST (small) datasets which contain 10 digits/letters images, corresponding to 10 tasks respectively. All the tasks are trained one by one, only model with superior 'memory' can reproduce old tasks while learning new ones. Then we also consider a long-term continual learning with more tasks, which combine MNIST and notMNIST together with a sequence of 20 tasks. This can evaluate the generalization ability of through learning the two data sets in sequence. The MNIST and notMNIST data sets are similar in the pixel settings, but are different in the content of the images (digits and letters images). We also evaluated the performance of the model with neuron addition, supplementing existing networks with additional parameters for new tasks (this also allow us to have more choices on pruning strategies, for example, the cumulative proportion of frozen weights can be larger than 1). The models are evaluated by two metrics: log-likelihood and classifier uncertainty, where classifier uncertainty is to assess the quality of the generated images by training a discriminative classifier and see whether the generated images are correctly classified in high confidence.

4.2. Performance Study

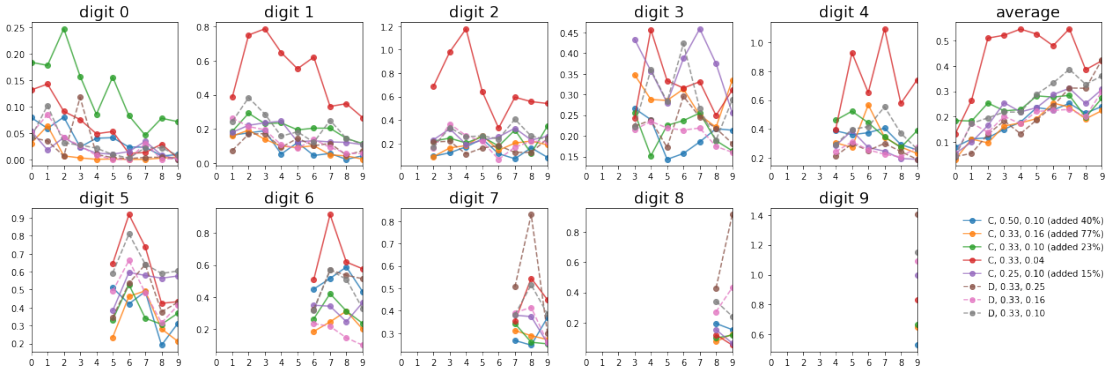
Figure 2 shows the proposed method compared with 4 VAE-based benchmarks (EWC, LP, SI, VCL) on all 3 experiments. It is evident that the proposed UGVCL method outperforms VCL, EWC and LP in overall performance of all 3 tasks on both metrics, and it is slightly better than SI, these implies that UGVCL has an outstanding long-term memory of previous tasks while is still able to learn incremental data of subsequent tasks. It is noteworthy that the proposed UGVCL method has a single encoder-decoder unlike the 'multi-head' decoders for individual tasks in VCL, thus making it task agnostic. Figure 1 shows samples from the generative models, it is apparent that UGVCL helps to prevent catastrophic forgetting, while VCL and SI fail at accurate representation of tasks in notMNIST.

4.3. Ablation Study: Pruning Strategy

In our proposed method, the only hyper parameter is the proportion of eligible weight to be removed from each layer. As the representations learnt from task I will be shared across all the subsequent tasks, we prefer to freeze a larger fraction



(a) Test minus log-likelihood result on MNIST (the lower the better)



(b) Classifier uncertainty result on MNIST (the lower the better)

Fig. 3. Results of different pruning strategies. The two alternative pruning strategies are indicated by the first letter in the legend. 'D': Freeze a predetermined percentage of the eligible weights, with the number of frozen parameters decreasing to enable shared representation with the subsequent task; 'C': Freeze a constant fixed fraction of weights parameters in all tasks, ignoring the effect of shared representation. The numbers indicate the ratios of pruning after each task. For example, 'C, 0.33, 0.10' denotes 33% of weights are frozen in the initial network, and for all the subsequent tasks, 10% of the total amount of weights are frozen (note that only significant weights are frozen and this subset is disjoint to those pruned).

of weight after training the model with the data from I. Figure 3 illustrates the result of UGVCL with different pruning strategies on MNIST. It can be seen that UGVCL with pruning strategy 'D' failed to learn the incremental tasks, as the minus log-likelihood is clearly higher than the model with pruning strategy 'C' from the task "digit 5". This indicates that global-level sharing of features across multiple tasks does not perform very well . Of all the model with pruning strategy 'C', 'C, 0.33, 0.10' performs excellent and robust in all tasks, and adding a large volume of neurons don't necessarily result in a good performance. It can be observed from this study that the percentage of weights to be pruned should be well selected lest the model either is incapable to retain knowledge from old tasks or unable to learn incremental tasks. It is noteworthy that this trend also applies to notMNIST, and pruning strategy 'C, 0.33, 0.10' still has an outstanding performance.

5. CONCLUSION

In this work, we present a generative continual learning framework with weight pruning that compresses the model size of a 'multi-head' generative model without compromising accuracy. We implement this framework on VCL in which the redundant parameter can be identified according to the uncertainty distribution in BNN. During inference, appropriate masks are applied to recover the sparsity during training. Experimental results indicate an outstanding performance even when the benchmarks have larger network capacity. We also develop an add neuron option in our work which enable the model to adapt to more emerging tasks.

Acknowledgement

This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-RP-2021-027)

6. REFERENCES

[1] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[2] Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[3] Alexander J Smola, Vishy Vishwanathan, and Eleazar Eskin, "Laplace propagation.," in *NIPS*. Citeseer, 2003, pp. 441–448.

[4] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al., "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[5] Friedemann Zenke, Ben Poole, and Surya Ganguli, "Continual learning through synaptic intelligence," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3987–3995.

[6] Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner, "Variational continual learning," *arXiv preprint arXiv:1710.10628*, 2017.

[7] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang, "Learning efficient convolutional networks through network slimming," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2736–2744.

[8] Arun Mallya and Svetlana Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7765–7773.

[9] Sayna Ebrahimi, Mohamed Elhoseiny, Trevor Darrell, and Marcus Rohrbach, "Uncertainty-guided continual learning with bayesian neural networks," *arXiv preprint arXiv:1906.02425*, 2019.

[10] Timothée Lesort, Hugo Caselles-Dupré, Michael Garcia-Ortiz, Andrei Stoian, and David Filliat, "Generative models from the perspective of continual learning," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[11] Tingting Zhao, Zifeng Wang, Aria Masoomi, and Jennifer Dy, "Deep bayesian unsupervised lifelong learning," *Neural Networks*, 2022.