Assignment: Deep Generative Perspective on Continual Learning 2024

Matteo Gioia
Sapienza University of Rome
matteo.gioia@uniroma1.it

Daniele Trappolini
Sapienza University of Rome
daniele.trappolini@uniroma1.it

June 15, 2024

Abstract

This study investigates the challenge of continual learning by utilizing a Variational Autoencoder (VAE) as an alternative to the traditional replay buffer. Continual learning enables models to acquire new skills incrementally without losing previously learned knowledge, a phenomenon often termed catastrophic forgetting. We incrementally train a Conditional VAE on the MNIST dataset and compare its performance with the traditional replay buffer method. Our findings indicate that while the VAE-based approach does not yet surpass the performance of a simple greedy buffer, it offers potential scalability benefits for larger datasets and higher intra-class variance. The study highlights the need for improved VAE architectures to better retain past knowledge and avoid overfitting on new tasks.

Contents

1 Introduction **2**

2 Methodology **2**

2.1 Model Architecture 2

2.2 Training Procedure 2

2.3 Evaluation Metrics 3

3 Results **3**

3.1 General Performance and Hyperparameter Tuning 4

3.2 Memory Usage and Scaling Capabilities 4

4 Conclusion **5**

1 Introduction

Catastrophic forgetting is a significant obstacle in developing truly adaptable and effective continuous learning systems for real-world applications. Traditional learning models typically train on a static dataset. However, many real-world scenarios require models to adapt continuously to sequential data without forgetting previous knowledge. This issue is prevalent in AI systems used for predictive maintenance, medical diagnosis, and speech recognition, where data and context can evolve over time. Catastrophic forgetting occurs because neural network models update their weights based on new data, often at the expense of previous knowledge. Weight updates are a global process that does not differentiate between new and previously learned skills. To combat catastrophic forgetting, several approaches have been proposed, including generative models such as Variational Autoencoders (VAE) [5], Generative Adversarial Networks (GAN) [3], and Diffusion Models [4]. These models can generate synthetic data that represent previous knowledge, enabling the model to learn new information without forgetting the old. In our project, we test the performance improvement in a continual learning context using VAE as a generative model.

2 Methodology

2.1 Model Architecture

Our model is composed of two components:

- Conditional Variational Autoencoder (CVAE): A generative model that learns the underlying data distribution and generates synthetic samples based on a conditioning label.
- Multi-Layer Perceptron (MLP): A feedforward neural network for classification tasks.

The Conditional VAE includes an encoder, which maps input data to a latent space, and a decoder, which reconstructs data from this latent space. Both components support conditioning with a label concatenated to the hidden features, enabling the generation of new samples mimicking the original data distribution, with the CVAE serving as a substitute for a static replay buffer.

2.2 Training Procedure

The training process is organized into a sequence of tasks. Each task presents the model with new classes from the MNIST dataset [2] in a class incremental fashion. For each new task, the following steps are executed:

- Training the CVAE: The CVAE is trained on the current task's data, learning latent representations and optimizing the reconstruction loss while preserving previous knowledge. Once trained, the CVAE generates synthetic samples resembling the current task's training data.
- Generating Synthetic Samples: Synthetic samples generated by the trained CVAE represent knowledge of both current and past tasks.
- Training the MLP: The MLP is trained using synthetic samples from the CVAE, helping the model learn both new and past tasks.

2.3 Evaluation Metrics

To evaluate the continual learning approach, we compute the following metrics:

- Backward Transfer (BWT) [6]: Measures the impact of new task learning on the performance of previously learned tasks, defined as the difference in final performance on previous tasks before and after learning new tasks.
- Forward Transfer (FWT) [1]: Assesses the impact of prior task learning on new task performance, computed as the difference in performance on new tasks with and without prior task learning.
- Average Accuracy: The mean accuracy across all tasks at the end of the learning sequence, providing a general measure of the model's overall performance in a continual learning setting.

Additionally, we briefly examine memory usage, scaling capabilities, and limitations of this approach in subsequent sections.

3 Results

In this study, we performed hyperparameter tuning using Optuna to optimize the performance of our model. The following hyperparameters were tuned during the optimization process:

- hidden_size: This hyperparameter determines the size of the hidden layers in the neural network. We tested values from the categorical set {32, 64, 128, 256}.
- latent_dim: This hyperparameter controls the dimensionality of the latent space. We explored values from the categorical set {32, 64, 128, 256}.
- num_epochs: This hyperparameter defines the number of epochs for training the model. We allowed this parameter to vary between 5 and 30.

- num_epochs_VAE: This hyperparameter specifies the number of epochs for training the Variational Autoencoder (VAE). We adjusted this parameter to vary between 20 and 40.

3.1 General Performance and Hyperparameter Tuning

Trial	Hid. Size	Lat. Dim	N.Eps	N.Eps VAE	Acc.	BWT	FWT
0	256	64	27	37	0.243	-0.180	-0.127
1	256	128	22	39	0.105	-0.227	**-0.032**
2	32	256	8	32	0.308	-0.279	-0.118
3	256	128	13	23	0.177	-0.175	-0.119
4	64	256	28	30	0.139	-0.269	-0.120
5	32	32	10	21	0.122	-0.267	-0.096
6	64	32	10	33	0.164	-0.384	-0.091
7	64	32	7	33	**0.396**	-0.066	-0.125
8	32	64	13	21	0.123	-0.200	-0.120
9	256	256	28	34	0.220	**-0.027**	-0.154

Table 1: Performance metrics table with varying parameters

This table shows how the model performance varies with different parameters used in various experiments. The parameters vary in terms of hidden size, latent dimension, number of epochs, and number of VAE epochs. The performance metrics include average accuracy, BWT (Backward Transfer), and FWT (Forward Transfer).

Observations:

- Increasing the hidden size or latent dimension does not always lead to better average accuracy. For example, trial 7 with a hidden size of 64 and latent dimension of 32 achieves the highest average accuracy of 0.396.
- The number of epochs and VAE epochs also show significant impact. For instance, trial 2, with relatively few epochs (8) and VAE epochs (32), achieves a high average accuracy of 0.308.
- Negative BWT values indicate interference of the new tasks with the old ones, and this varies across trials. For example, trial 6 has the lowest BWT (-0.384), suggesting the highest interference.

3.2 Memory Usage and Scaling Capabilities

The model occupies approximately 928 MB of memory when loaded onto a GPU in Google Colab. It is worth noting that the approach using Variational Autoencoders (VAE) in these models does not result in significant memory savings because the models themselves are relatively small. However, as we scale to larger models, the memory savings become

substantial. This indicates that while the immediate benefits of using VAE might not be apparent with smaller models, their impact on memory efficiency becomes pronounced with the increase in model size, highlighting the scalability advantages of VAEs in more complex scenarios.

4 Conclusion

Currently, the biggest limitation of the model is the architecture of the VAE itself, which seems to still forget the representations of previous classes and overfit on the current task. Applying standard techniques such as regularization or EWC did not seem to help either, suggesting that using a different architecture, as discussed in the lectures, might help combat catastrophic forgetting. On the other hand, a simple greedy buffer vastly outperforms our results with similar memory usage.

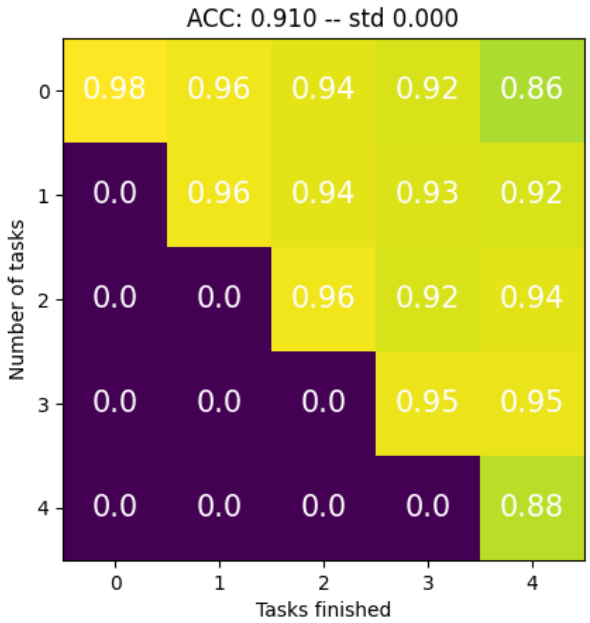

Figure 1: Performance of the MLP trained with a greedy buffer

However, this strategy might not be worthwhile if the dataset size grows together with the intra-class variance, and further testing in a similar situation where the CVAE might actually perform better is needed.

References

[1] Jiefeng Chen, Timothy Nguyen, Dilan Gorur, and Arslan Chaudhry. Is forgetting less a good inductive bias for forward transfer? *arXiv preprint arXiv:2303.08207*, 2023.

[2] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[5] Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

[6] Sen Lin, Li Yang, Deliang Fan, and Junshan Zhang. Beyond not-forgetting: Continual learning with backward knowledge transfer. *Advances in Neural Information Processing Systems*, 35:16165–16177, 2022.