Continual Learning with Generative Replay via Discriminative Variational Autoencoder

Woo-Young Kang
Surromind Robotics
Seoul 08826, Republic of Korea
wykang@surromind.ai

Byoung-Tak Zhang
School of Computer Science and Engineering
Seoul National University,
Surromind Robotics
Seoul 08826, Republic of Korea
btzhang@bi.snu.ac.kr

Abstract

Catastrophic forgetting in machine learning is the problem of gradually losing accuracy from previously learned tasks as new tasks are learned sequentially. This problem occurs mainly in modern artificial neural networks using gradient-based training algorithms. In this paper, we suggest a simple but robust generative replay-based model to mitigate the catastrophic forgetting problem. Also, We categorize current methods for incremental learning task as two approaches. Then, we analyze our method on Permuted MNIST task and Split MNIST task. Our experimental results show that our proposed method achieves competitive accuracy compared to other algorithms in Permuted MNIST task and outperforms other algorithms on Split MNIST task.

1 Introduction

Recently, as computing power and the number of available data increase, deep learning models have been showing remarkable performance on many image recognition tasks exceeding human accuracy [1, 2, 3]. However, such deep learning models can suffer from the catastrophic forgetting problem when we want to learn several tasks sequentially without any techniques to mitigate the forgetting problem [4, 5, 6]. Theoretically, the catastrophic forgetting problem can be completely avoided if we store all of the past data. However, it is sometimes difficult to store all of the previous data due to the limitation on memory capacity.

In this paper, we propose a simple method, a classifier-integrated generative model, to mitigate the catastrophic forgetting problem in the incremental learning setting instead of storing all of the past data. Also, we categorize current studies for the incremental learning into two types and analyze some issues of each approach. Our proposed model uses the variational auto-encoder(VAE) [7], especially the conditional VAE(CVAE), as a base architecture for stable training and integrating the CVAE with a classifier.

Our classifier-integrated VAE has several advantages. First, we can reduce the total memory usage for modeling the incremental learning framework by incorporating an encoder network of the VAE with a classifier. Second, distributions of latent features are clustered class-wise discriminatively on the latent space as the VAE is trained jointly with a classification loss. By introducing a prior network, we can also learn the class-wise prior distributions, which is more flexible and natural way than assigning randomly or intuitively. We present the intuitive explanation for the discriminative latent space at Appendix C. In this paper, we call our proposed model as AC-VAE, which is illustrated in Figure 1.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

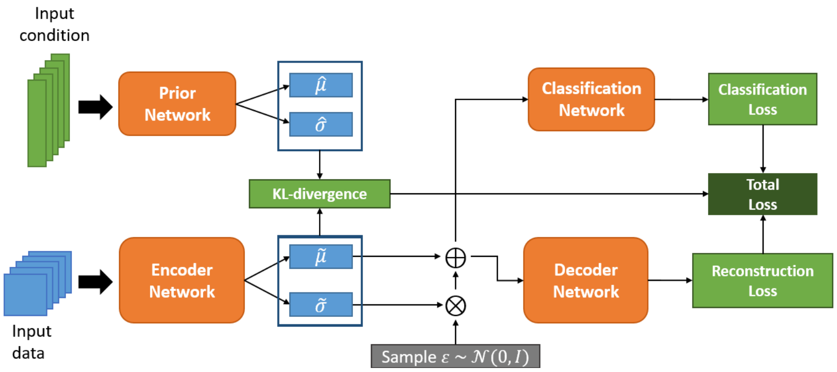


Figure 1: Overview of the AC-VAE. The prior network and classification network can be a simple neural network which has one or two fully connected layers. The $\oplus$ and $\otimes$ mean element-wise addition and multiplication, respectively.

2 Related work

In some studies such as Elastic Weight Consolidation(EWC) [8], Synaptic Intelligence(SI) [9], and Variational Continual Learning(VCL) [10], weight regularization techniques are used to mitigate the catastrophic forgetting problem. In this approach, each of them has a regularization term to constrain the update of network parameters to mitigate the catastrophic forgetting. For example, the EWC uses the Fisher information matrix to calculate the contributions of each weight. Then, highly contributing weights are less updated. In the SI, they decide the contribution of each parameter using the gradient of the loss. The VCL formulates continual learning task as the online variational inference with a Bayesian Neural Network(BNN) [11]. They minimize the Kullback-Leibler(KL) divergence between parameter distributions of the current and the previous network to maintain the performance of the previous task. These approaches can be called as the prior-focused approaches [12].

Other studies which use generative replay have been proposed recently [13, 12]. This approach, called as likelihood-focused approach [12], generates previous task data when a new task data is coming, then optimizes them concurrently. In the Deep Generative Replay(DGR) [13], they use the WGAN-GP [14] to unconditionally generate previous task data and they alternatively optimize the WGAN-GP and a classifier. In the Variational Generative Experience Replay(VGER) [12], they also use a GAN [15] to generate previous data. In contrast to the DGR, they use multiple generative models for each task and use a BNN as their classifier. Then, they optimize a classification error and KL divergence with respect to parameter distributions of the current and the previous classifier. Both methods have two separate stages, learning a generator and learning a classifier, while we propose a simple model by merging a classifier and a generative model, which makes the entire process for incremental learning task with the generative replay simple.

Lastly, dynamic architecture approaches also have been proposed. In these approaches, the structure of a network can be changed as they learn new tasks [16, 17, 18]. It can overcome limited capacity problem due to the fixed network structure of the above two approaches. However, in this paper, we focus only on prior and likelihood-focused approaches.

3 Proposed methods

3.1 Variational auto-encoder

Our proposed model is based on the VAE [7]. Thus, in this subsection, we briefly summarize some preliminaries. Considering we have observations $X = \{x_1, x_2, ..., x_N\}$ where $x_i \in \mathbb{R}^{\mathbb{D}}$, the goal of the VAE is to maximize marginal log likelihoods of each observation $\log p(x_1, x_2, ..., x_N) = \sum_{i=1}^{N} \log p(x_i)$ with respect to both variational parameters and generative parameters. However, this includes an intractable term when we use a complex model. Because the intractable term is always positive, we can optimize the variational lower bound instead of the marginal log likelihood

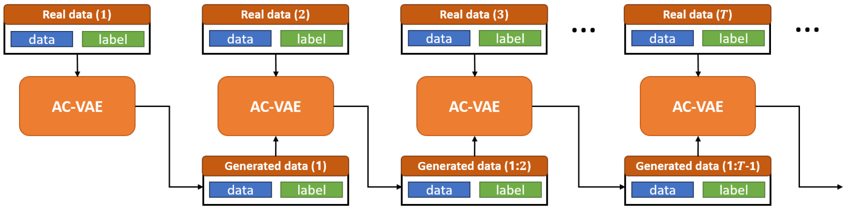


Figure 2: The process for incremental learning with our AC-VAE. At first, the model learns a real data. Then, it generates all previous task data to learn a new task jointly. The orange-colored box can be replaced with any generative models which have a classifier to label generated samples.

as follows:

$$\log p(x_i) \geq \mathbb{E}_{q(z|x_i)}[\log p(x_i|z)] - D_{KL}[q(z|x_i)||p(z)] \tag{1}$$

where $q(z|x_i)$ is the variational posterior (the encoder) and $p(x_i|z)$ is the decoder. In general, both the encoder and the decoder can be modeled with neural networks.

3.2 Auxiliary classifier VAE (AC-VAE)

The encoder of the VAE is generally modeled with a deep neural network. Therefore, we can use the encoder network as a feature learning network for a classifier by jointly optimizing a classification loss and the variational lower bound. To do the joint optimization, we add a simple auxiliary classifier which takes the output of an encoder, see Figure 1. With the joint optimization, we can make discriminative posterior distributions for each class in latent space, which is illustrated in the Appendix C. Figure 6(c). Then, we introduce an additional simple network which approximates the posterior distributions. This additional network is called the prior network in this paper. The prior network replaces the prior distribution $p(z)$, generally fixed as unit Gaussian, in Equation 1. Thus, both the posterior and the prior distribution can be learned, which alleviates the constraint on distributions of z. Using the prior network, we can sample class conditional images from the class-wise discriminative distributions. The loss function of our model can be written as:

$$\begin{gathered}\mathcal{L}_{AC-VAE} = \mathcal{L}_{CVAE} + \lambda \mathcal{L}_{Classification}, \\ where,\ \ \mathcal{L}_{CVAE} = D_{KL}[q_\theta(z|x,c)||p_\psi(z|c)] - \mathbb{E}_{q_\theta(z|\mathrm{x},c)}[\log p_\phi(\mathrm{x}|z,c)],\end{gathered} \tag{2}$$

where, the λ is a weighting value of the classification loss. θ, ψ, and ϕ are network parameters of the encoder, the prior network, and the decoder respectively. Also, the $p_\psi(z|c) = \mathcal{N}(z|\mu_c, \sigma_c^2)$, where c is a class condition. The classification loss, $\mathcal{L}_{Classification}$, is calculated by a classifier modeled with another parameter ω.

3.3 Incremental learning process with a generative model

Firstly, the AC-VAE is trained only with real images and labels of the first task because there is no previous task. After training the AC-VAE, the model can access the previous data only through the previous AC-VAE model which can generate the previous data. Because our proposed model has an auxiliary classifier, we can label the generated data with the classifier. When we train the AC-VAE with a new task, the model learns both the generated previous task data and the real current task data jointly. This is illustrated in Figure 2.

4 Experiments

In this section, we analyze two experimental settings. The results are described at Table 1 and detailed experimental settings are described at Appendix A. In Table 1, numbers written next to the 'VCL - ' mean the size of random coreset data [10] per each task. Firstly, we show experimental results on Permuted MNIST task [5, 8]. In this task, input pixels of each dataset are shuffled with task-wise fixed

Table 1: The averaged accuracy of the Permuted MNIST task and Split MNIST tasks. Each setting is conducted with 5 tasks and accuracy is averaged over 5 runs for each setting. We refer to reported accuracy for the DGR and the VGER from [13, 12].

Approach	Model	Permuted MNIST	Split MNIST
Prior-focused	EWC [8]	94.14	18.77
	VCL-0 [10]	95.53	19.47
	VCL-200 [10]	**96.33**	82.71
Likelihood-focused	DGR [13]	∼92.3	∼94.8
	VGER [12]	∼96	∼95
	AC-VAE(ours)	96.01	**97.92**

permutations. The prior-focused approaches such as the EWC and the VCL achieve high accuracy on Permuted MNIST task even without reference images. However, we need to pay close attention to the results of Split MNIST task. Farquhar & Gal [12] insightfully pointed out drawbacks of the Permuted MNIST task in that unrealistic large differences in each permuted digits lead to artificially lessened forgetting. Also, the prior-focused methods will fail when a new task has a similar pattern with one of the old tasks. This is because they don't have any explicit terms to discriminate the old and similar new tasks in their loss function. So, the old task data will be predicted as a similar one in the new task. That's why prior-focused approaches fail in the Split MNIST setting without reference samples of the previous task. From the Table 1, the VCL-200 can be seen as a prior-likelihood hybrid approach because it uses previous data stored in coreset. So the VCL-200 can maintain performance for previous tasks to some degree

The second setting is Split MNIST task. This setting is more challenging than Permuted MNIST task in that added new tasks could have similar patterns to old tasks (more realistic differences). In this experiment, we consider the five subsets of the MNIST data: the first subset consists of 0, 1, the second subset is composed of 2, 3, and in the same manner for remaining subsets. We can see that prior-focused approaches show low performance on learning new classes of MNIST data incrementally. While likelihood-focused approaches can effectively mitigate the catastrophic forgetting problem in this setting. Furthermore, our proposed model achieves higher accuracy than the DGR and the VGER. This is because our model conditionally generates previous samples, which makes higher quality samples than the DGR and the VGER that uses an unconditional generative model. The generation result of each task for the DGR and our method is illustrated in Appendix B. Figure 5.

5 Discussion

In this paper, we addressed how to mitigate the catastrophic forgetting problem. To solve the forgetting problem, we proposed a conditional generative model based on the VAE architecture to generate previous task data and used them as reference data of previous tasks. Also, we incorporated an auxiliary classifier into the conditional generative model, which can reduce the system complexity of the incremental learning task with generative replay.

Through several experiments, we could see that prior-focused approaches such as the EWC and the VCL perform well in Permuted MNIST task and are not enough to solve Split MNIST task. However, likelihood-focused approaches such as the DGR, the VGER, and the AC-VAE can effectively mitigate the catastrophic problem on both Permuted and Split MNIST tasks. In spite of that, there are two critical issues that likelihood-focused approaches have to consider. First of all, it is hard to extend to more complex data such as ImageNet [19] because it is known as a challenging problem to generate natural images with a generative model. Since we can't store all of the previous task data, we rely on generated samples from a generative model. So, the generated samples should be as highly realistic as possible so that the current model can reference them. This is the reason why likelihood-focused approaches are not easy to extend to a more complex dataset. Secondly, the error of generated samples for previous tasks will gradually accumulate. Since, practically, we can't precisely approximate true data distribution with finite samples, the quality of generated samples may degrade progressively. Thus, in the future, we should overcome these limitations to solve the catastrophic problem more effectively. In our opinion, it will be a good option to carefully integrate the prior- and likelihood-focused approaches.

Acknowledgment

This work was supported by the ICT R&D program of MSIT/IITP. [2017-0-00162, Development of Human-care Robot Technology for Aging Society].

References

[1] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, et al. Going deeper with convolutions. Cvpr, 2015.

[2] 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.

[3] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5987–5995. IEEE, 2017.

[4] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

[5] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

[6] Roger Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285, 1990.

[7] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[8] 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*, 114(13):3521–3526, 2017.

[9] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. *arXiv preprint arXiv:1703.04200*, 2017.

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

[11] Alex Graves. Practical variational inference for neural networks. In *Advances in neural information processing systems*, pages 2348–2356, 2011.

[12] Sebastian Farquhar and Yarin Gal. Towards robust evaluations of continual learning. *arXiv preprint arXiv:1805.09733*, 2018.

[13] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, pages 2994–3003, 2017.

[14] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5769–5779, 2017.

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[16] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[17] 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.

[18] Jeongtae Lee, Jaehong Yun, Sungju Hwang, and Eunho Yang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017.

[19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[20] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

[21] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, pages 3483–3491, 2015.

[22] Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, pages 4790–4798, 2016.

[23] Liwei Wang, Alexander Schwing, and Svetlana Lazebnik. Diverse and accurate image description using a variational auto-encoder with an additive gaussian encoding space. In *Advances in Neural Information Processing Systems*, pages 5758–5768, 2017.

[24] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Appendix

A. Implementation details

Here, we describe the detailed network structure used in our experiments. For the EWC, we stacked two hidden layers, where each layer consists of 800 hidden units with ReLU activation. Also, we applied the dropout to the input layer and hidden layers with 0.2 and 0.5, respectively. We trained this EWC with batch size 256 and 50 epochs for each task. For fine-tuning with respect to λ, we conducted multiple runs with $\lambda = 10^3, 10^4, 10^5, 10^6, 10^7$ and selected the best value $\lambda = 10^6$. For the VCL, we chose the same structure with Nguyen et al. [10]. For the DGR, we had a hard time to reproduce the performance reported in [13], so we carefully referred to the reported accuracy. In our method, we found out that adding Gaussian noise to input images is useful to improve our algorithm for Split MNIST task. Since our loss function includes reconstruction error, there are small differences in pixel values between real images and generated images. So, this error can pass to next tasks and will be accumulated. Thus, the added Gaussian noise plays a positive role in

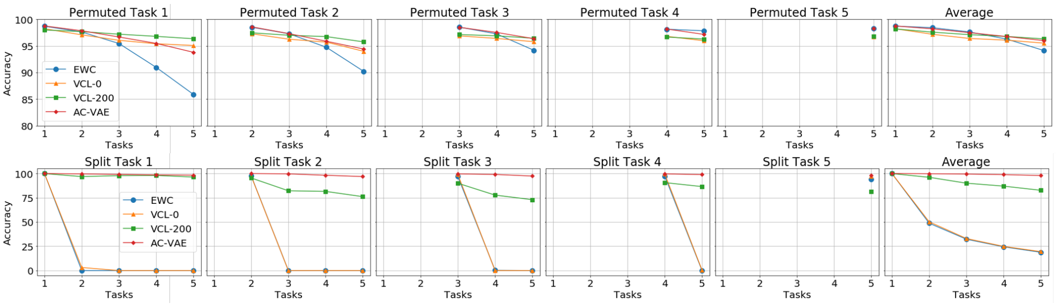
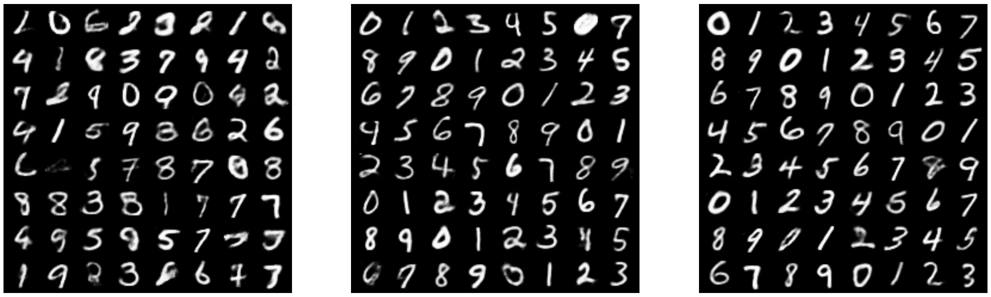


Figure 3: Learning curve for both Permuted and Split MNIST. The top row is Permuted MNIST task and the bottom row is Split MNIST task.

Table 2: Implementation details of the AC-VAE for Permuted MINST and Split MNIST settings

Network	Operation	Kernel or Input dims	Stride/Padding or Output dims	BatchNorm	Dropout	Activation
Encoder	Convolution	3 × 3 × 1 × 32	2/1	√	-	ReLU
	Convolution	3 × 3 × 32 × 64	2/1	√	0.5	ReLU
	Convolution	3 × 3 × 64 × 128	2/1	√	0.5	ReLU
	Convolution	3 × 3 × 128× 256	2/1	√	-	ReLU
	Fully-connected	2 × 2 ×256	256	√	-	ReLU
	Fully-connected	256	128	-	-	ReLU
	Fully-connected	256	128	-	-	Softplus
Decoder	Fully-connected	128	256	√	-	ReLU
	Fully-connected	256	2 × 2 ×256	√	-	ReLU
	Convolution	3 × 3 × 256 × 128	2/1	√	-	ReLU
	Convolution	3 × 3 × 128 × 64	2/1	√	-	ReLU
	Convolution	3 × 3 × 64 × 32	2/1	√	-	ReLU
	Convolution	3 × 3 × 32 × 1	2/1	-	-	Sigmoid
Prior Net	Fully-connected	10	32	√	-	ReLU
	Fully-connected	32	128	-	-	ReLU
	Fully-connected	32	128	-	-	Softplus
Auxiliary classifier	Fully-connected	128	128	√	0.5	ReLU
	Fully-connected	128	10	-	-	Softmax
Hyper-parameters	Optimizer	Adam(lr=0.001, betas=(0.9, 0.999), eps=1e-08)				
	Batch size	64				
	Softplus	beta=1, threshold=20				
	λ for AC-VAE	Permuted: 5e+3, Split: 2e+2				
	Gaussian noise	mu=0, std=0.3				

	Epochs per task	Permuted: {100, 100, 200, 200, 200}
		Split: {100, 100, 200, 200, 200}				



(a) Samples from the VAE[7] (b) Samples from the CVAE[20] (c) Samples from our AC-VAE

Figure 4: Plotting of generated samples from each VAE model.

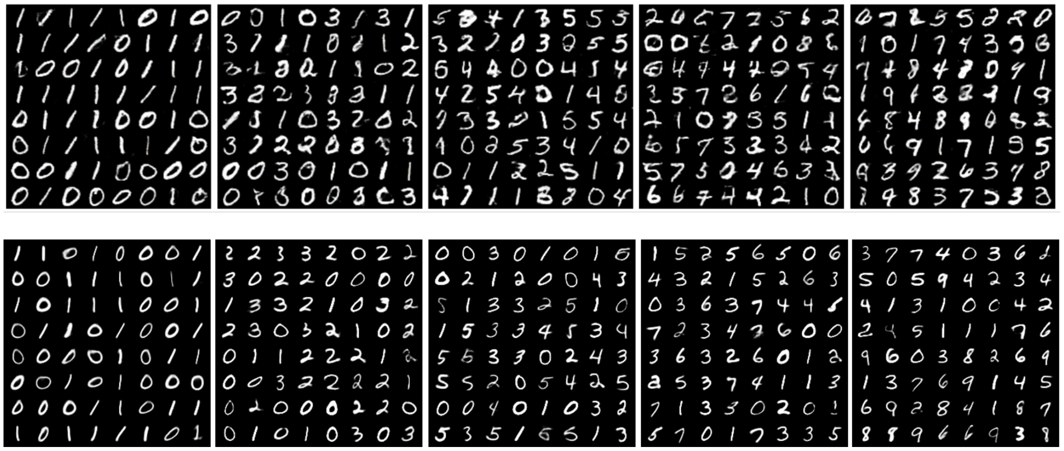
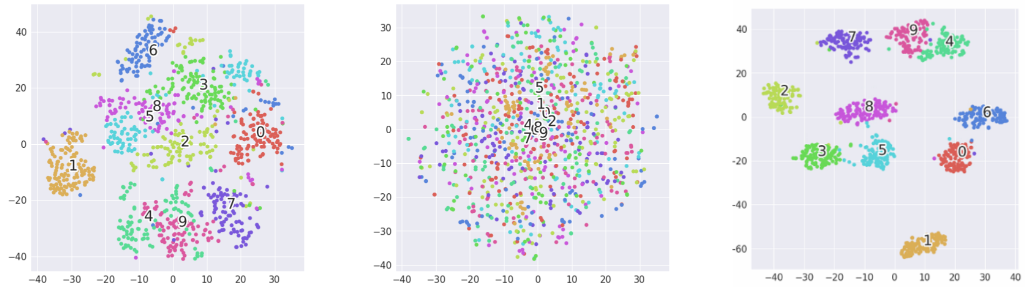


Figure 5: The generated image samples. The top row is generated from the DGR[13], and the bottom row is sampled from our AC-VAE.

training our model to become robust to the error. Adding noise technique is not necessary to reduce catastrophic forgetting for the prior-focused approaches because they don't include such generation process. GAN-based models such as the DGR and the VGER don't need to add noise too. This is because their loss function to generate an image is just classification error of a discriminator, not the pixel-wise mean-squared error or binary cross entropy. Also, when we conducted some experiments with this adding noise technique for the EWC and the VCL, there was no noticeable increase in final accuracy for 5 task incremental learning. The detailed structure of our AC-VAE is described at Table 2 and learning curves for Permuted and Split MNIST tasks are illustrated in Figure 3.

B. Conditional vs Unconditional generation

The quality of the generated images of the previous tasks is crucial to maintain the classification accuracy for incremental learning task. This is because when we learn a new task, previous data we can utilize is the generated samples from a generative model. Thus, if a generative model generates poor quality samples, the accuracy for the previous task may decrease significantly. Furthermore, because the new generative model may also approximate the poorly generated previous data, the error will be accumulated whenever we learn new tasks. Generally, in the VAE, we can get more sharp and realistic samples if we optimize a conditional likelihood of given data[21, 22]. That is why we should use a conditional generative model instead of an unconditional generative model. Sampling qualities of the unconditional VAE, standard conditional VAE, and our proposed model are described in Figure 4. In Figure 4, we can see that some sampled images from the standard VAE which is an unconditional generative model are hard to recognize with the naked eye. The abnormal samples could be a noise factor causing a decrease in accuracy for the previous task. However, the two conditional models generate high-quality samples. This affects the success of the incremental



(a) Latent space of standard VAE

(b) Latent space of standard conditional VAE

(c) Latent space of proposed model

Figure 6: Visualization of the latent space. (a) is the unconditional standard VAE[7], (b) is the standard conditional VAE[20] and (c) is our proposed method. In this visualization, we randomly sampled 1000 images from MNIST test set. Also, We used the t-SNE[24] to reduce the dimension of latent vectors for plotting

learning task. As a result, our AC-VAE achieves higher accuracy than the DGR which unconditionally generates previous data, see Table 1. The generation results of both the DGR and the AC-VAE are illustrated in Figure 5.

C. Selection of prior distribution

Instead of our method, there are several ways to determine the conditional prior distribution $p(z|c)$. For example, we can determine the $p(z|c)$ randomly or intuitively. Then, we may not need to add additional prior network. Actually, there is a study to learn the VAE with class-discriminative distributions. They randomly initialize each class-specific moment of Gaussian distributions as prior distributions[23]. However, this method could assign similar patterns, such as digit 4 and 9, to the distant latent space, so the manifold of data may not be learned properly. While our proposed method explicitly considers the manifold for similar patterns through updating encoder parameters with respect to classification loss. Thus, similar patterns will be closely located in the latent space. Also, we can manually assign the prior distributions considering the pattern similarity with our intuition. However, if there are hundreds of classes, it is difficult to decide moments for each class one-by-one. In contrast, if we use the AC-VAE, we don't need to determine each parameter of distributions; we just need class conditions for our prior network. The projected latent space is illustrated in Figure 6.