L2L-GAN: Layer to Layer Generative Adaversarial Network for Few-shot Generation

Haoxuan Wang
Shanghai Jiao Tong University
#800 Dongchuan Rd.
hatchet25@sjtu.edu.cn

Xinyuan Lu
Shanghai Jiao Tong University
#800 Dongchuan Rd.
lxy9807@sjtu.edu.cn

Abstract

Abundant training samples are often required to generate images for a certain category. However, some particular classes of data are expensive to obtain and rarely exist. Few-shot generation aims to augment these classes of data to aid down-stream tasks such as image classification and segmentation. Though the task is important and profitable, this area of research has not been fully explored. Thus, we propose a L2L-GAN, a fast adaptation framework from the optimization-based view that simulates the target class generation process using only source data. We propose the Generalization Enhancement Module (GEM) that can learn the fast adaptation process with only the source training samples, and can be directly applied to the target domain image generation process. GEM does a layer to layer mapping between the network learned with more samples and the network learned with less samples.

1. Introduction

Image generation has become an increasingly important topic in machine learning and computer vision communities these years due to the limitation of human access to some real world scenes. For example, real-world cameras on vehicles cannot collect enough crucial but rare video images under conditions such as before car crashing and rapid stopping. And the unbalanced dataset may affect downstream tasks such as object recognition and segmentation in autonomous driving. Thus, the ability of automatically generating images based on few source images has become an increasing demand both in research and industry.

Few-shot image generation is a challenging yet not fully explored task. The problem can be formulated as follows: Given the source training sets $\mathbf{X_s}$ and $\mathbf{X_n}$, where images of each class in $\mathbf{X_s}$ are abundant to train a good generator but $\mathbf{X_n}$ only contains very limited amount of data. Few-shot generation attempts to utilize $\mathbf{X_n}$ to produce more diverse

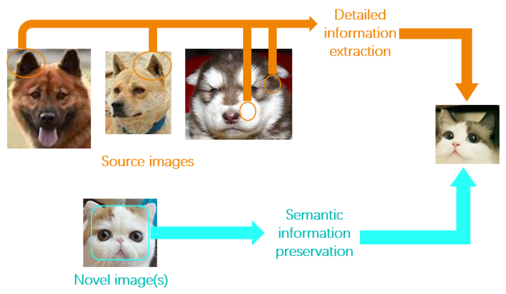
Detailed information extraction
Source images
Novel image(s)
Semantic information preservation

Figure 1. General intuition of how to do few-shot generation.

and realistic images of the given categories. The main difficulties for this task mainly lies in two parts: (1) How to learn/adapt quickly using only few novel images and (2) How to restrict the generated image class to be consistent with the novel class used for learning. Previous works such as [2] and [9] uses meta-learning to direct the model to learn from small amount of samples. [1] focuses on using generative models to augment the dataset for better classification generalizability. [4] [6] utilize conditional images to replace Gaussian noise input and combines information from different layers of the GAN structure. Different from all these works, we propose a learning framework that can learn to make better use of features from different source categories and apply it to the target novel category.

Motivated by the intuition that image generation models generate images by doing smart combinations of features that are seen during the training procedure, we intend to learn the invariant feature for novel classes and combine it with different feature variants learned during the training process. This procedure is shown in 1. However, when trained by limited number of samples, the model could easily overfit and features are learn badly. Thus Generalization Enhancement Module (GEM) is proposed to force the lightly-trained model to yield similar performance as a model that is well-trained on large amount of samples.

Our contributions can be summarized as follows:

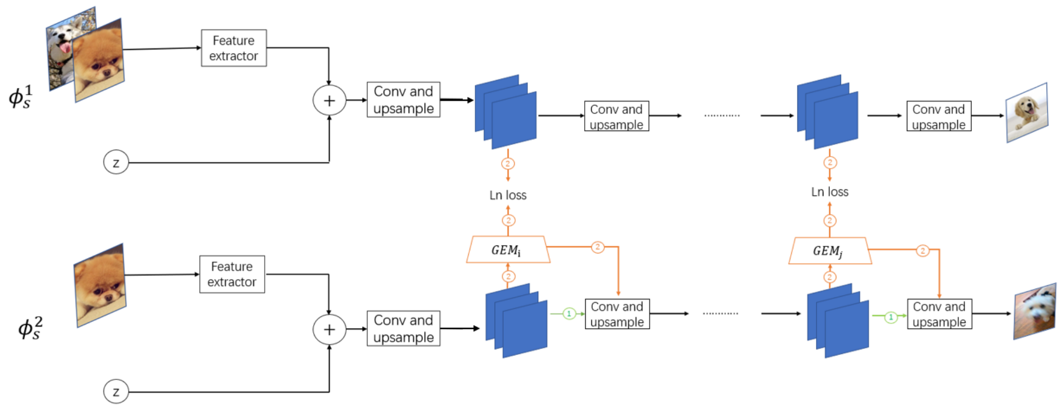

Figure 2. Illustration of our method in terms of generator. In the first step of the two-step training phase, $\mathbf{f_1}$ is trained with abundant base images and $\mathbf{f_2}$ is trained with base images that has similar numbers as novel classes. In the second step, GEM modules are added into $\mathbf{f_2}$ for training. The green lines indicate the first step of training and orange lines for the second step.

- Propose a GEM module for learning transferable features in the source images;
- Propose a new learning framework from the optimization and transformation-based view to do few-shot generation;
- Supplement and provide the few-shot generation community with code written in PyTorch since most works are done using Tensorflow.

2. Related work

There is a growing interest on few-shot image generation recently. Compared with few-shot image classification, generation tasks are often more difficult. We summarize the most significant works in this area below.

2.1. Optimization-based methods

Meta-learning methods has attracted much interest in the area of machine learning due to its superiority in improving model generalizability. As an algorithm that learns how to learn, meta-learning trains the generation model end-to-end via ”adjusting” the optimization direction so that SGD is close to performing search on the novel class loss hyperplane. [9] is a representative algorithm using the GAN structure that optimizes the model using an inner and outer loop. In the inner loop, the gradients for updating the generator and discriminator are computed without updating the actual parameters used for evaluation. Also, it is slightly different from the normal GAN update process: the discriminator used for calculating the generator’s gradient copies the parameters from the last iteration. In the outer process, the gradients are not directly used for backpropagation but reweighted first so as to adjust the optimization procedure. Optimization-based methods are rare due to the limited number of samples we can access for the novel categories and the update directions are ambiguous most of the time.

2.2. Fusion-based methods

Another track of generator models originates from matching networks[12] and relation networks[7], which makes abundant use of different layer feature maps by fusing them. The intuition is that different depth of neural networks exhibits different level of image information, where the shallow layers extract detailed and higher resolution features and deep layers extract high-level and semantic-related information. MatchingGAN[5] and F2GAN[6] are recent works that use this idea for few-shot generation. While their main points of demonstration are different, the general structures are quite similar. MatchingGAN’s main point is to match random vectors sampled from Gaussian distribution with conditional images as the input for GAN. F2GAN illustrates its idea as fusing high-level features and filling in low-level feature details. However, these methods depend heavily on novel images and their performance is restricted by the novel image qualities. The source images are not fully made use of and the model trained from them are only seen as a feature extractor. Also, one-shot learning seems not applicable for these methods. Our method intend to make more use of the features learned from the source classes so that the novel image generation procedure can make use of information from source class images.

2.3. Transformation-based methods

There’s also a range of methods that are transformation-based, which focuses on learning the feature mappings from base category images to novel category images. DAGAN[1] mainly focuses on augmenting the novel dataset for image classification. Instead of using one conditional image for

Algorithm 1 Training procedure of our framework

```
1:  Input: Source data $\mathbf{X_s}$, novel data $\mathbf{X_n}$ with sample number $p$. Source generator $\phi_s^1$, source generator $\phi_s^2$, target generator
    $\phi_t$, GEM module set $\{\tau_i\}_{i=1}^K$. Total number of epochs $T$, GEM modules number $K$.
2:  Initialization: Train $\phi_s^1$ using $\mathbf{X_s}$, random initialize $\phi_s^2$, $\phi_t$ and $\{\tau_i\}_{i=1}^K$. Set current epoch $t$.
3:  For $t = 1$ to $T$:
4:      Randomly sample $p$ samples from $\mathbf{X_s}$ to form a sub-dataset $\mathbf{X_s^t}$
5:      Train $\phi_s^2$ using $\mathbf{X_s^t}$
6:      For $i = 1$ to $K$:
7:          Select a certain layer in $\phi_s^2$ to add $\tau_i$ after it
8:          Calculate the n-norm loss between the feature maps of the selected corresponding layers in $\phi_s^1$ and $\phi_s^2$
9:          Update $\tau_i$ only using the n-norm loss
10: Train $\phi_t^{plain}$ using $\mathbf{X_n}$
11: $\phi_t \leftarrow$ combination of $\phi_t^{plain}$ with $\{\tau_i\}_{i=1}^K$
12: Fine tune $\phi_t$ using $X_n$
13: Return $\phi_t$
```

generation, it uses two images from the same base category to form a class distribution, and interrupts one image with an Gaussian noise to generate the new image. The new image is expected to have the same distribution as the other two base images. However, simply adding Gaussian noise limits the diversity of generated images lead to mode collapse. DeltaGAN[4] adopts the idea of Delta-encoder[11], which learns the gap between different classes and uses the gap (delta) to transfer class distributions. DeltaGAN assumes that the difference between images are consistent for base and novel categories, and attempts to learn the difference using base category images. This makes the generation process suitable for one-shot learning. Our method is different from DeltaGAN as we learn the difference between model layers, making the generation process more flexible and diverse, and is not limited to specific input images.

3. Methodology

3.1. Learning Algorithm

Given the source training dataset $\mathbf{X_s} = \{x_{ij}|i = 1..K_s, j = 1..P_{s_i}\}$ which has K_s class and P_{s_i} samples in i class and novel class $\mathbf{X_n} = \{x_j|j = 1..P_n\}$ which has only P_n samples, we separate source training dataset randomly to be two datasets $\mathbf{X_{ss}} = \{x_{ij}|i = 1..K_s, j = 1..P_{s_i} - P_n\}$ and $\mathbf{X_{sn}} = \{x_{ij}|i = 1..K_s, j = 1..P_n\}$ ($\mathbf{X_{sn}}$ can also be constructed by sampling from $\mathbf{X_{ss}}$). Then we train $\phi_s^1(z)$ on $\mathbf{X_{ss}}$, $\phi_s^2(z)$ on $\mathbf{X_{sn}}$ and $\phi_t(z)$ on $\mathbf{X_n}$ separately. Since ϕ_s^1 and ϕ_s^2 are obtained by using less samples, their generation ability is weak and limited. Thus the Generalization Enhancement Module (GEM) τ is added to enhance models' ability. The goal of τ is to satisfy $\tau(\phi_s^2(z)) = \phi_s^1(z)$. More specifically, by adding GEM, the feature map of $\phi_s^2(z)$ will resemble the corresponding one as in $\phi_s^1(z)$, and we train τ while fixing $\phi_s^1(z)$ and $\phi_s^2(z)$. Our final generative model on novel classes can be written as $\tau(\phi_t(z))$. In the optimal case, GEM has both the ability to improve model generation performance as well as adapting base categories features to novel categories. The specific generator structure of the framework is shown in Figure 2. We currently do not change much of discriminator network. The algorithm is shown in Algorithm 1.

Despite from the conventional discriminator loss and generator loss, penalty on the gradient is also used for optimization to aid the discriminator's learning process. In our two-step learning procedure, the GEM training is guided only by n-norm loss (L1 or L2 loss). Thus, the GEM loss and the total loss function can be written as:

$$\begin{aligned} \mathcal{L}_{\tau_i} &= \mathcal{L}_{norm}(h_{si}, h_{ni}) \\ \mathcal{L}_{total} &= \mathcal{L}_G + \mathcal{L}_D + \sum_{i=1}^{K} \mathcal{L}_{\tau_i} \end{aligned} \tag{1}$$

where h_{si} is the output feature map of ϕ_s^1 at the position of the ith GEM module and h_{ni} is the output feature map of ϕ_s^2 at the position of the ith GEM module. L_{τ_i} intends to make the two feature maps as similar as possible. Since the assumption is that the feature map learned by ϕ_s^2 is much weaker than the one learned by ϕ_s^1, it is expected that GEM can generalize and strengthen the feature representation by introducing more variant elements.

3.2. Generalization Bound

The method's performance bounds can be easily inferenced and found. On the one hand, GEM learns from the difference $\sigma(\phi_s^1, \phi_s^2)$ between ϕ_s^1 and ϕ_s^2. Thus it only contains the variant feature information, which is also within ϕ_s^1's generation range. On the other hand, ϕ_t determines the basic semantic information of the generated images. Our final output model is a combination of ϕ_t and GEM modules, thus the generated images are expected to be a smart composition of semantic parts and invariant features. The generated novel images should be different from provided ones, but the difference between input and output cannot exceed $\sigma(\phi_s^1, \phi_s^2)$. Thus, the performance/generalization bound of the final ϕ_t can be approximated as lower-bounded by ϕ_s^2 and upper-bounded by ϕ_s^1. This gives us insight that L2LGAN's performance greatly depends on the performance of ϕ_s^1, which is a generation model that's trained by abundant samples. This result is also proved, as shown in the discussion part.

Layer	Resample	Norm	Output Shape
image	-	-	96*96*3
encode 0	stride=2	BN	64*48*48
encode block 1	stride=2	BN	64*24*24
encode block 2	stride=2	BN	128*12*12
encode block 3	stride=2	BN	128*6*6
decode block 0*	upsample	BN	128*12*12
decode block 1*	upsample	BN	128*24*24
decode block 2*	upsample	BN	64*48*48
decode block 3	upsample	BN	64*96*96
decode block 4	-	BN	64*96*96
final conv 1	-	BN	64*96*96
final conv 2	-	BN	64*96*96
final conv 3	-	BN	3*96*96

Table 1. Model Architecture

GEM type 1	GEM type 2	GEM type 3
-	conv(stride=1)	conv(stride=1)
-	ReLu	ReLu
conv(stride = 2)	conv(stride = 2)	conv(stride = 2)
upsample	upsample	upsample
-	-	deconv (stride = 1)

Table 2. GEM types

3.3. Model Architecture

The model's implementation is a combination of UNet and ResNet, with upsampling and downsampling layers acting as encoder and decoder, as well as skip layer connections to transfer more information between blocks. This structure is a well established baseline and is very suitable as a startup. Detailed information about the model can be found in Table 1. For our GEM modules, we provide three variations, as shown in Table 2. The variations differ in complexity, with increasing number of convolutional layers and activation functions.

3.4. Learning Process

Despite from various model architectures, we also provide different learning strategies for GEM modules since algorithm 1 only provides a general framework for learning. We list them here:

- Strategy A: Train GEM modules sequentially and independently from shallow to deep;
- Strategy B: Train GEM modules sequentially, but when training deeper modules shallow modules are also updated;
- Strategy C: Train all GEM modules at once.

The detailed learning procedures also differ in many aspects, such as using L1/L2 loss and the number of selected GEM modules. Different settings often lead to different generation performance, and we often choose the better ones.

4. Experiments

4.1. Dataset and baselines

We plan to evaluate the L2L-GAN on the Omniglot[8] dataset and VGGFace[10] dataset. Omniglot consists of hand written characters from 50 different categories. VGGFace is a harder dataset that consists of realworld human faces and was intended for object detection. These two datasets are chosen due to their representativeness and they are used in our baseline DAGAN. The Omniglot contains 1622 classes of different characters, each character has 20 samples. We take 1200 classes of images as source images, 200 as validation images and the remaining class images as novel classes. For novel classes, we only randomly select 2 out of 20 images for each class. VGGFace dataset contains 2354 classes of different human faces from the internet where each human has 100 face samples. We take 1800 classes as source images and 200 classes as validation images. The remaining classes are taken as novel classes, where we randomly select 2 out of 100 images for each class.

4.2. Evaluation metrics

Various metrics for evaluating few-shot generation were proposed, and can classified as subjective metrics and objective metrics. The subjective metrics mainly depend on human eye impression, including directly examining the generated image quality and producing the Interpolated spherical subspace[13]. Objective metrics are numerical numbers that can measure generated image quality. Popular metrics include Fréchet Inception Distance (FID)[3], Inception Scores (IS)[14] and Learned Perceptual Image Patch Similarity (LPIPS)[15]. Also, the generated images can be further used for augmenting few-shot classification dataset, and classification accuracy can be used. However, classification accuracy is optional for us since time is limited.

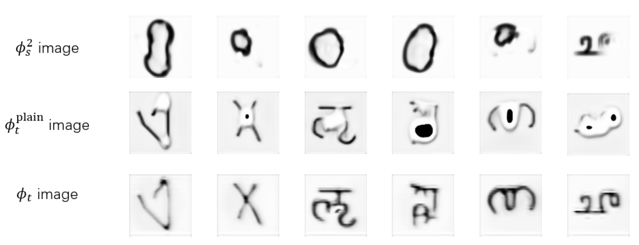

Figure 3. Comparison of L2LGAN generation results on Omniglot dataset with other baselines. L2LGAN has the best performance.

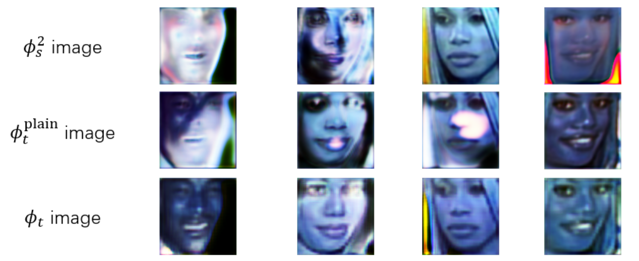

Figure 4. Comparison of L2LGAN generation results on VGGFace dataset with other baselines. L2LGAN has the best performance.

4.2.1 FID

Fréchet Inception Distance(FID) is a improved version of Inception score, which can better measure the similarity between generated images and real images. In detail, it uses the outputs after activation layer to fit a a multidimensional Gaussian, then calculate Fréchet distance between two Gaussian. The lower the FID is, the better our model is.

4.3. Experimental results

Fig 3 shows our generation results on Omniglot dataset and Fig 4 shows the results on VGGFace dataset. The samples are chosen from generated images and we compare our final model ϕ_t with the two baselines that use small amount of samples. ϕ_s^2 is trained on source dataset, approximately 2400 images while ϕ_t^{plain} and ϕ_t are trained and validated on approximately 400 images. The comparison shows that the model only trained on source classes has diffculty in generalizing to novel classes, while L2LGAN is able to preserve the semantic information of the character and greatly improve the quality of generated images at the same time. Table 3 shows the improvement in numbers. Concretely, the GEM module we introduced can close the gap of training sample numbers, which indicates that it can improve the generalizability of original models.

	ϕ_s^2	ϕ_t^{plain}	ϕ_t
GEM type 1	157.9	235.4	**150.1**
GEM type 2	**147.0**	194.0	151.2
GEM type 3	**124.6**	205.2	132.2

Table 3. FID comparison on Omniglot dataset

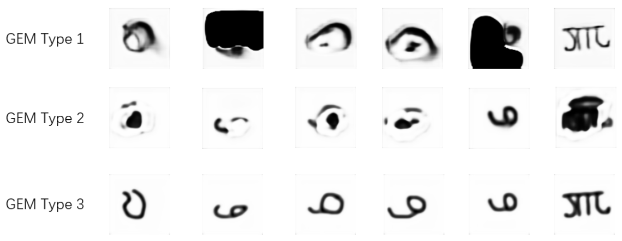

Figure 5. Ablation study on different GEM module structures. GEM type 3 yields the best performance.

4.4. Ablation study

We study how different GEM types (as shown in Table 2) effect the final performance of our L2LGAN. Figure 5 shows the difference between generated images using the three GEM types. We observe that GEM type 3 gives us better quality images. And it indicates that more complicated GEM modules can exhibit more information.

We also conducted ablation studies on different training strategies. Our experimental results shows that the different training strategies does not differ much in terms of generated images seen by bare eyes. Thus we do not explicitly discuss them here. We suspect that different strategies yield similar performance is due to that GEM modules are simple under our settings, and the final ϕ_t's other parts (apart from GEM) are not changed anymore after training for the first time.

5. Conclusion

We propose a new learning framework L2LGAN for doing few-shot generation, and introduce the Generalization Enhancement Module (GEM) for strengthening the generalizability of models trained on few images. Our experiment results on Omniglot dataset and VGGFace shows that the learning framework as well as GEM can justify our proposal. However, drawbacks still exists because our generated images are not variant enough.

To conclude, L2LGAN is a optimization-based few-shot learning framework that has the ability to better generation model performance, but it also has aspects that can be further improved. We believe that this research direction is promising and has a lot of space to explore. Our future directions include adopting more powerful backbone architectures and introducing more techniques for improving the generation ability.

References

[1] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks, 2018.

[2] Louis Clouâtre and Marc Demers. Figr: Few-shot image generation with reptile, 2019.

[3] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018.

[4] Yan Hong, Li Niu, Jianfu Zhang, Jing Liang, and Liqing Zhang. Deltagan: Towards diverse few-shot image generation with sample-specific delta, 2020.

[5] Yan Hong, Li Niu, Jianfu Zhang, and Liqing Zhang. Matchinggan: Matching-based few-shot image generation, 2020.

[6] Yan Hong, Li Niu, Jianfu Zhang, Weijie Zhao, Chen Fu, and Liqing Zhang. F2gan: Fusing-and-filling gan for few-shot image generation, 2020.

[7] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection, 2018.

[8] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[9] Weixin Liang, Zixuan Liu, and Can Liu. Dawson: A domain adaptive few shot generation framework, 2020.

[10] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015.

[11] Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Rogerio Feris, Abhishek Kumar, Raja Giryes, and Alex M. Bronstein. Delta-encoder: an effective sample synthesis method for few-shot object recognition, 2018.

[12] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning, 2017.

[13] Tom White. Sampling generative networks, 2016.

[14] Qiantong Xu, Gao Huang, Yang Yuan, Chuan Guo, Yu Sun, Felix Wu, and Kilian Weinberger. An empirical study on evaluation metrics of generative adversarial networks, 2018.

[15] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018.