Generalizability and Training Stability of GANs: Literature Survey

Aamir Raihan
56357908

Sarah Elhammadi
97667166

Ke Ma
27694165

Abstract

Goodfellow et al. [2014] opened an entire new frontier in unsupervised learning by proposing a generative model called Generative Adversarial Network (GAN). Although GANs have been successfully applied in image generation, they suffer from several problems that primarily arise from training instability and generalization properties. In this project, we review several variants of GANs and their problems before summarizing the related work and techniques that have been proposed to address these problems. We also conduct two experiments. In the first experiment, we apply a GAN based semi-supervised architecture on the dataset provided by the ongoing Kaggle competition: Google Landmark Recognition Challenge and report the classification accuracy. In the second experiment, we verify the effectiveness of GAN variants and compare their learning curves on different datasets.

1 Introduction

In Generative Adversarial Networks (GANs), generator G and discriminator D networks are trained simultaneously. G's task is to generate realistic data (e.g. image) from noise z, and D's task is to classify a data point as either real or fake. These two adversaries compete with each other in the form of a min-max game: G tries to fool the D while D tries to improve its classification accuracy. Formally, the game between the generator G and the discriminator D is the min-max objective:

$$\min_G \max_D \quad E_{x \sim P_{real}}[f(D(x))] + E_h[f(1 - D(G(h)))]. \qquad (1)$$

GAN training usually continues until the generator wins, meaning the discriminator is unable to distinguish between the samples from P_{data} and $P_{generated}$. However, the training in practice is oscillatory where the objective goes up and down. This differs from the usual deep net training, where training objective steadily goes down. Hence, it raises some open discussions about the generalization properties of GANs such as why there will be an equilibrium at which the generator wins, and even if the generator wins, why it means that $P_{generated}$ is close to P_{data}. This oscillatory behavior also makes the training process highly unstable and introduces problems such as mode collapse, low diversity and low resolution of generated image. In addition, GANs perform poorly in the case where large amount of classes are present, and thus generate non-sensible objects.

This paper discusses two prevailing problems of GANs: 1) generalization properties, and 2) training instability, before reviewing techniques that have been proposed in literature to address these problems. We conduct experiments on the dataset provided by the Google Landmark Recognition Challenge, an ongoing Kaggle competition, and report the classification accuracy. Then, we conduct experiments using different variants of GANS, namely, Deep Convolutional GAN, Wasserstein GAN, and Wasserstein GAN with gradient penalty on three different datasets (MNIST, CIFAR10 and Google Landmark Recognition Dataset) and analyze their learning curves.

2 Generalization Properties of GANs

2.1 Problems with finite discriminator

The fundamental issues about the generalization in GANs have been debated since the emergence of GANs. For example, in what sense is the learned distribution close to the target distribution, and also, what kind of equilibrium exists between generator and discriminator. Goodfellow et al. [2014] showed that given sufficiently large number of samples and sufficiently large discriminator nets, the learned distribution is close to the target distribution. However, the recent analysis by [Arora et al., 2017] challenged these claims for a discriminator of finite size. They formally defined the notion of generalization where high generalization means that the population distance between the true and generated distribution is close to the empirical distance between the empirical distributions. It was further shown that discriminator of finite size n is unable to distinguish between a true distribution and a distribution with support of $O(\frac{n}{\epsilon^2})$, and thus finite size discriminator is unable to enforce learned distribution to have large diversity, let alone to enforce $P_{generated} \approx P_{real}$. Thus, the training of GANs may not have good generalization properties, meaning that even when training appears to be successful, the trained distribution may be far from the actual distribution. Furthermore, even JS divergence and Wasserstein don't generalize with any polynomial number of examples.

2.2 Estimating support size of the generated distribution

Arora and Zhang [2017] showed that GANs learn distributions of low support and proposed a test based upon birthday paradox for estimating the support size of the generated distribution. The birthday paradox suggests that for a distribution of support N, the sample size of $\sqrt{N}$ is quite likely to have duplicate. If duplicate is found on a sample size of s then using the birthday paradox suggests that distribution have support size of s^2. The only failure mode of birthday paradox test would be if the distribution would assign a large probability to a single image and be uniform on a huge number of other images. Then the sample size s would have a huge probability to have the duplicate of that image even when the support size is huge. But such non-uniformity is the only failure mode of the birthday paradox test calculation, and such non-uniformity would itself be considered a failure mode of GANs training. It was also shown that the diversity of learned distribution grows near linearly with the discriminator size. A good way of quantifying the performance of GANs is in the context of diversity. The result suggests that the Deep Convolutional GAN (DCGAN, a variant of GAN for convolutional architecture) and MIX+GAN (a training framework that uses the mixture of generator nets) have low diversity compared to ALI or BiGANs which have support size of around a million whereas DCGAN and MIX+GAN have only a support size of around 160000.

2.3 Mixture of GANs

A weaker notion of generalization may exist in a neural net distance within which the two distributions will be close. Arora et al. [2017] proved that ϵ approximate equilibrium exists where the generator wins, i.e if the discriminator is a deep net with p parameters, then a mixture of $O(\frac{plog\frac{p}{\epsilon}}{\epsilon^2})$ generator nets can produce a distribution D that the discriminator is unable to distinguish from D_{real} with probability more than ϵ. The basic premises is that every density can be approximated using the infinite mixture of Gaussians and since generator can at least approximate the Gaussian, so the mixture of generators should win the game. In fact, they proved even a finite mixture of reasonable size can approximate the performance of infinite mixture that win the game. Thus, the theory guarantees existence of approximate equilibrium and suggests that GANs training may be better and more stable by replacing the simple generator and discriminator with mixtures of generators. The experiment result of using a mixture of generators shows significant improvement in inception score (a metric used to evaluate the quality of generated samples) and training stability over the Wasserstein GAN or DCGAN. However, some limitations lie in the fact that the effect of backpropagation algorithm is not considered in the training process. Also, the mixture of GANs does not have much practical significance as they quadratically increased the computational power required during the training of network.

3 Variants of GAN

Training GAN requires finding Nash equilibrium between highly non-convex game between the generator and discriminator with lots of parameters involved and the objective is minimized using the gradient descent method. However, the training processes is extremely susceptible to network structure and parameter tuning and the most common training instability is mainly due to gradient vanishing and mode collapse. Vanishing gradient becomes a serious problem when P_{data} and $P_{generated}$ become disjointed and so discriminator perfectly separates real and fake data, so training stops even though $P_{generated}$ is far from P_{data}. Mode collapse is another common problem of GANs where the generator repeatedly produces the same or similar output since that point easily fools the discriminator. Thus, lots of GAN variants have been proposed in the literature to mitigate these training instability.

3.1 Deep Convolutional Generative Adversarial Networks (DCGAN)

Radford et al. [2015] proposed DCGAN, a variant of GAN for convolutional architecture. It is one of the foundational work on improving the training stability of GAN. DCGAN enforces architectural constraints for convolutional architecture for generative adversarial network to stabilize the training and produce better high-resolution samples over a wide range of database. The set of architecture constraints introduced by the paper include: 1. Replace any pooling layers with strided convolutions(discriminator) and fractional-strided convolutions (generator). 2. Use batch normalization (BN) in both the generator and the discriminator. 3. Remove fully connected hidden layers for deeper architectures. 4. Use ReLU activation in generator for all layers except for the output, which uses Tanh. 5. Use LeakyReLU activation in the discriminator for all layers

Batch normalization (BN) tackles poor initialization and helps gradient flow in deeper models. It also helps to prevent the generator from mode collapse, one of the most common problems which arises in generator where generator collapse all samples to a single point. However, batch normalization should not be applied to all layers since they produce sample oscillation and mode instability. Similarly, bounded activation allows the model to learn more quickly to saturate and cover color space of the training distribution.

3.2 Wasserstein GAN (WGAN)

Arjovsky et al. [2017] used an example of learning parallel lines to show that there are scenario where only Earth Mover(EM) distance converge, whereas other distance such as the total variation distance, KL divergence and JS divergence don't. This indicates that there are scenario where probability distribution over a low dimensional manifold can be learned using EM distance, but not by other distances or divergence since the loss function is not continuous. To tackle this issue, two theorems are showed providing a compelling reason for using Wasserstein distance as the loss function of generative adversarial networks.:

- If the parametric function g_θ, which transforms a fixed distribution $p(z)$ to P_θ, is continuous then the $W(P_r, P_\theta)$ (Wasserstein distance) is continuous. Furthermore, if g is locally Lipschitz and continuous then $W(P_r, P_\theta)$ is continuous and differentiable almost everywhere. The theorem gives a strong argument supporting earth mover distance because to optimize the model P_θ based on some $d(P_r, P_\theta)$, it is desirable to have a loss function that is continuous and differentiable everywhere. The paper points out the feedforward neural network satisfying both of the property.
- The $W(P_r, P_\theta)$ is the weakest of the group. In other words, every distribution that converges under the KL, reverse-KL, TV and JS divergences also converges under the Wasserstein divergence.

Note that an approximate way of computing the Wasserstein distance is to use Kantorovich-Rubinstein duality.

In Wasserstein GAN, the discriminator learns the K-Lipschitz continuous function to help compute the Wasserstein distance. As the loss function decreases in the training, the Wasserstein distance gets smaller and the generator model's output grows closer to the actual distribution. Weight clipping is used to maintain the K-Lipschitz continuity of f.

Empirical experiments show that the Wasserstein GAN critic provides clean gradients as compared to GAN discriminator when the discriminator and critic are trained to optimality, which removes the need of balancing the discriminator update with the generator update. In fact, the better trained the critic is, the better the estimate of Wasserstein distance will be and consequently the estimate of the gradient is better. Also, there is one to one correlation between the lower loss function and better samples generated by the generator and thus they provide a meaningful interpretation to the loss.

3.3 Bidirectional Generative Adversarial Networks (BiGANs)

Donahue et al. [2016] proposed a Bidirectional Generative Adversarial Network (BiGAN), a new unsupervised feature learning framework. The proposed model not only include generator and discriminator but also an encoder E which maps the data x to its latent representation.At the global optimum, G and E are each other reverse. The bidirectional discriminator not only uses the data space but also the information of the latent space provided by the encoder.

3.4 Boundary equilibrium generative adversarial networks (BEGAN)

The problem with GAN is that since it tries to minimizes the Kullback-Leibler divergence between the real distribution and generated distribution. Thus, the discriminator might provide meaningless gradient if the discriminator becomes good too quickly. Instead of matching the data distribution, Berthelot et al. [2017] matched the distribution of reconstruction loss derived from the Wasserstein distance between the reconstruction losses of real and generated data and the network is trained with this loss with an additional equilibrium term to balance D and G. The loss objective is similar to WGAN, the difference being that the loss objective is obtained by matching the loss distribution rather than data distribution, and there is no need for the discriminative function to be K-Lipschitz. The equilibrium term, also called diversity ratio, is introduced to maintain the equilibrium between the generator and discriminator and provide a way to control the image diversity and sample quality. The BEGAN framework is the only other framework besides WGAN that provides a way of measuring the convergence.

4 Training of GANs

GANs support a wide variety of functions in that the generator and discriminator need not be multilayer perceptron where the entire network can only be trained by back-propagation [Rumelhart et al., 1985]. In addition, training of GAN is robust to overfitting since the generator is not directly updated with data examples, but only with the gradients flowing through the discriminator. To address the training instability problem, a set of architectural features and training procedures are recommended by Salimans et al. [2016]:

- Feature matching: Instead of maximizing the output of the discriminator, the new objective requires the generator to generate data that matches the expected value of the internal feature of an intermediate layer of the discriminator.
- Minibatch discrimination: one of the common failure modes for GAN is the mode collapse where the generator always emits the same points. The solution is to look at multiple samples in combination rather than isolation.
- Historical averaging: adding a term to each player's cost function that captures the difference between the current parameter values and those over the last t time steps.
- One sided label smoothing: in discriminator's cost function, replacing the target of a classifier with smooth values prevents the discriminator from giving a very large gradient signal to generator, while it also prevents extrapolating to avoid extreme samples.

- Virtual batch normalization: In virtual batch normalization, the samples are normalized using a reference batch.

Using DCGAN architecture, the training result on ImageNet dataset with these techniques outperforms the traditional training method. Also, the inception score correlates very well with human judgments.

Alternatively, Xiang and Li [2017] showed that although BN accelerates training in the beginning, the use of BN can be unstable and negatively impact the quality of the trained model. Instead, weight normalization (WN) is introduced and it was claimed that stability is similar to the one in WGAN, provided the learning rate is set to 0.0002 or lower.

One noticeable weight normalization technique is spectral normalization (SN) [Miyato et al., 2018], borrowed from the classic idea of spectral regularization and applied in deep learning by Yoshida and Miyato [2017]. The key idea is that the product of spectral norm of gradients at each layer serves as a good approximation of the overall Lipschitz continuity of the entire discriminating network. It was shown that weight clipping suffered from the rank degeneracy. In contrast, SN-GAN allows for more non-zero singular value, which gives rise to a more powerful discriminator and generator. In addition, SN-GAN is the first GAN to ever fit all 1000 ImageNet classes in one GAN. However, while spectral normalization is less costly, it is not intended to claim that spectral normalization better controls the Lipschitz constant than the gradient penalty method. Another method to improve the stability of training is to replace the usual GAN loss with a softmax cross-entropy loss. Using the DCGAN architecture, Lin [2017] showed that the soft-max GAN is still robust to training. They use the DCGAN architecture and simply change the loss and remove the batch normalization and other empirical techniques used to stabilize training. They show that the soft-max GAN is still robust to training.

Lipschitz constraint removes mode-collapse and improves training stability of WGAN. As a result, no empirical evidence of mode collapse is found in WGAN. However, the main limitation of WGAN is that enforcing K-Lipschitz by weight clipping. Large clipping value can increase the convergence time of critic till optimality, while small clipping can lead to vanishing gradient if the number of layers is large or batch normalization is not used. Also, Wasserstein GAN training becomes unstable using the momentum based optimizer such as Adam (an optimization method that is a variant of stochastic average gradient descent). Gulrajani et al. [2017] proposed that the training process of WGAN can be further refined by enforcing K-Lipschitz continuity by penalizing the norm of the gradient of the critic(discriminator) with respect to its input. The author showed that there are many problems associated with enforcing K-Lipschitz continuity using weight clipping like optimization difficulties and even pathological value function for critic. Critic trained with weight clipping fail to capture the higher order moments of data distribution, so the critic learns much simpler function. In fact, even deeper WGAN critics with batch normalization sometimes fail to converge. The other problem associated with weight clipping is that the optimization difficulty causes either vanishing or exploding gradients without careful tuning of the clipping threshold c. Thus, Gulrajani et al. [2017] proposed an alternate way called gradient penalty (GP) using the fact that every differential function is 1-Lipschitz if and only if it has gradients with norm at most 1 everywhere, which entails that directly constraining the gradient norm of the critic's output with respect to its input can enforce 1-Lipschitz. To circumvent the tractability issue, a soft version of the constraint with a penalty on the gradient norm for random samples P_x is proposed. P_x sampled uniformly along the straight lines between pairs of points sampled from the data distribution P_r nd P_g. Empirically it was found that enforcing along the straight lines seems sufficient and gives good performance.

While the use of gradient penalty in training of WGAN can increase the stability, it only estimates continuity between generated and real points, and thus the Lipschitz-ness is not enforced in the early stage of training because at the beginning stage the sampled points could be far away from the manifold. Wei et al. [2018] proposed a new regularizer for WGAN in addition to the gradient penalty, such that Lipschitz continuity is imposed over the manifold and its surrounding regions supporting the real data distribution. The key conceptual idea is to perturb each data point twice where a Lipschitz constant is used to bound the difference in the discriminator's response on the perturbed points. Since, as the results showed, perturbing the data with Gaussian noise directly yields no plausible results, it is therefore proposed to perturb the hidden layers using dropout. With this consistency term, the results are impressive on MNIST and CIFAR-10 for supervised learning

Daskalakis et al. [2017] argues that one source of training instability comes from the limit cycle problem (i.e. loss oscillates but not converge). To address this problem, they proposed a simple but effective modifications of training of WGAN where the conventional gradient descent is replaced with Optimistic Mirror Decent (OMD), which was claimed to accelerate the convergence of zero-sum convex-concave games. Despite of the concern on the performance of the average iterate, the theoretical result of the extended OMD shows that the last iterate provides a good estimate of the value of bilinear games, which avoids the cycling behavior. Experiment on generating DNA sequence demonstrates a smaller KL divergence and training on CIFAR10 achieves better inception score than that of Adam.

Karras et al. [2017] proposed an entire new methodology of training, where training starts with both generator and discriminator at low spatial resolution of 4x4 pixels and as the training advances, new layers are incrementally added in the Generator and Discriminator, which increases the spatial resolution. All existing layers remain trainable throughout the training process. This allows stable synthesis at high resolution and considerably speeds up the training. Due to the progressive growth of generator and discriminator, most of the iterations are spent in training the low spatial resolution and so comparable quality is often obtained 2-6 times faster depending upon the final output resolution.This was the first GAN architecture which successfully generated image at a resolution of 1024x1024 and the generated image quality is really impressive, capturing all the minor details of the human face.

5 Experiments

We conducted some experiments on the dataset provided for the Kaggle competition: Google Landmark Recognition Challenge. This competition is part of the Landmark Recognition Workshop at CVPR 2018 where teams are required to build models that recognize landmarks. We adopted the GAN based semi-supervised learning to improve the accuracy of our classifier by using the unlabelled examples generated by the generator. It was a difficult challenge since the total number of classes present in the original training datasets is around 15000 and the number of training examples per each class is much less than those generally available in MNIST, CIFAR10 and ImageNet dataset. Therefore, we selected the GAN based semi-supervised architecture for our classifier since it improves the accuracy by learning from the unlabelled GAN generated images. The architecture was inspired from Salimans et al. [2016], where the discriminator is turned into a multi-class classifier and learns the probabilities of each classes in addition to being normal critic to the generator.

The architecture for our classifier have 15 separate GAN based classifiers trained on different 1000 classes, which generate the probability of the samples belonging to these classes and then use a fully connected neural network at the output layer whose input is the output probabilities of these 15 classifier. But due to the limitation of the computational resources, we were not able to train the final neural network layer. The classification accuracy obtained by the 15 GAN based classifiers are shown in the Table 1.

We also trained some of the commonly used GANs such as DCGAN, WGAN, and WGAN-GP on three different datasets (MNIST, CIFAR10 and Kaggle Landmark Recognition Dataset) and plot the training curves in figure 1. The training curve for DCGAN and WGAN/WGAN-GP is different since DCGAN GAN uses sigmoid cross entropy loss whereas the cost function of WGAN and WGAN-GP is the average number of incorrect prediction given by the discriminator. From the curve, its quite evident that the training is much more stable for WGAN-GP. Also, the fluctuation in the training curve is higher for Kaggle Landmark Recognition Rataset since it doesn't have lots of examples available for a particular class and so it isn't able to generalize well to all the classes. Some generated images from WGAN and WGAN-GP can be found under Appendix A.

6 Conclusion

This paper provides an overview of generative adversarial networks (GANs), and the prevalent problems of training instability suffered by GANs. Drawing on the latest research on GANs including learning theory, variants of original GANs and modified training techniques, we investigate the possible causes of training instability and corresponding enhancements, and point out the remaining challenges in their theory and applications. To verify the effectiveness of different methods for constructing and training of GANs on less-known dataset, we apply DCGAN, WGAN, WGAN-

classifier	Number of Classes	Number of Training Examples	Accuracy
classifier 1	1000	57806	64
classifier 2	1000	61015	69
classifier 3	1000	79612	73
classifier 4	1000	59436	69
classifier 5	1000	62895	67
classifier 6	1000	69222	73
classifier 7	1000	126376	82
classifier 8	1000	56976	64
classifier 9	1000	61952	68
classifier 10	1000	114429	80
classifier 11	1000	68570	69
classifier 12	1000	58054	64
classifier 13	1000	64640	67
classifier 14	1000	60331	65
classifier 15	950	41267	64

Table 1: Classification accuracy of the 15 GAN based classifiers on Google landmark recognition challenge dataset

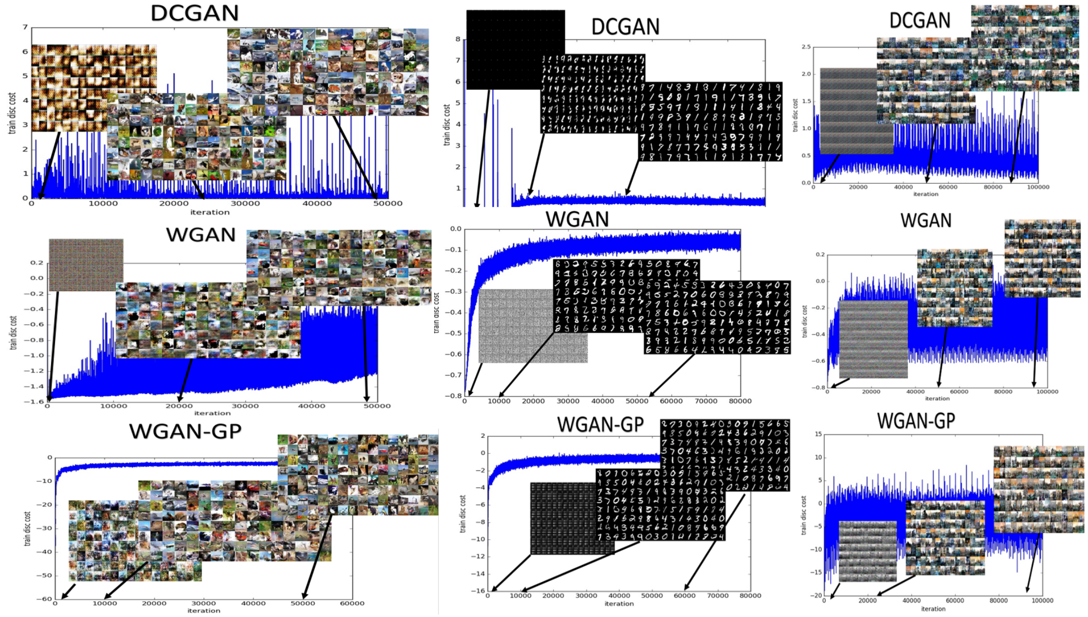

Figure 1: Learning curves and sample images for DCGAN, WGAN, and WGAN-GP on MNIST, CIFAR10 and Kaggle Landmark Recognition Dataset.

GP to Kaggle Landmark Recognition Dataset and achieve plausible results, demonstrating the generalizability of the latest GANs.

The code we used to train and evaluate our models is available at `https://github.com/AamirRaihan/cpsc540project`.

References

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Sanjeev Arora and Yi Zhang. Do gans actually learn the distribution? an empirical study. *arXiv preprint arXiv:1706.08224*, 2017.

Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans). *arXiv preprint arXiv:1703.00573*, 2017.

David Berthelot, Tom Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training gans with optimism. *arXiv preprint arXiv:1711.00141*, 2017.

Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

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.

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.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Min Lin. Softmax gan. *arXiv preprint arXiv:1704.06191*, 2017.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

Xiang Wei, Boqing Gong, Zixia Liu, Wei Lu, and Liqiang Wang. Improving the improved training of wasserstein gans: A consistency term and its dual effect. *arXiv preprint arXiv:1803.01541*, 2018.

Sitao Xiang and Hao Li. On the effects of batch and weight normalization in generative adversarial networks. *stat*, 1050:22, 2017.

Yuichi Yoshida and Takeru Miyato. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*, 2017.

Appendix A

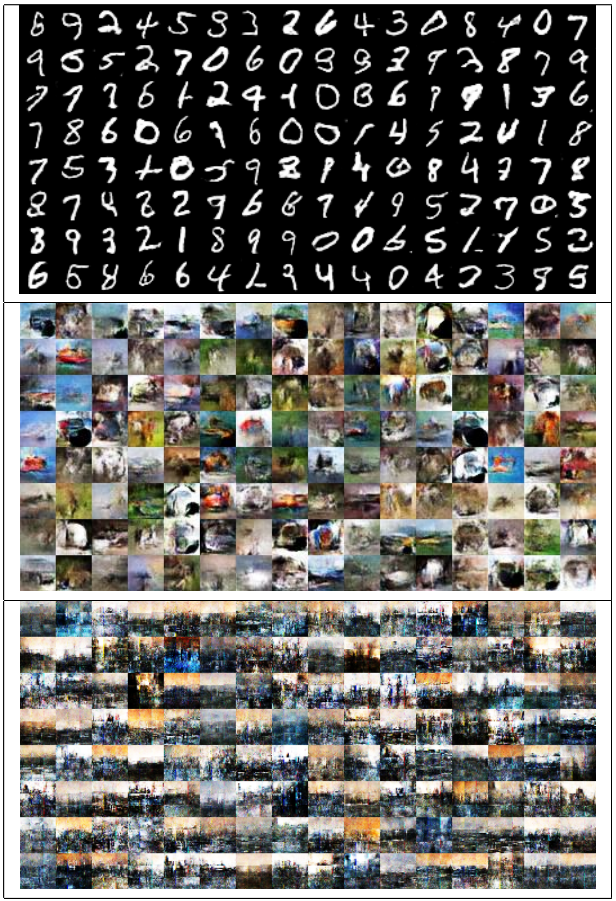

Table 2: Sample generated by WGAN

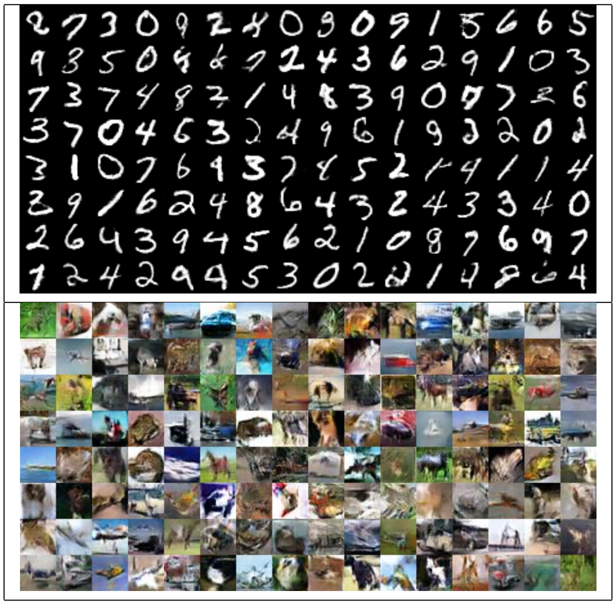

Table 3: Sample generated by WGAN-GP