Low Rank Approximation of Weight Matrices in GANs

Arnav Garg
Department of Computer Science
University of California - Los Angeles
Los Angeles, CA 90025
arnavgarg@cs.ucla.edu

Prateek Malhotra
Department of Computer Science
University of California - Los Angeles
Los Angeles, CA 90024
prateekmalhotra@cs.ucla.edu

Tanmay Sardesai
Department of Computer Science
University of California - Los Angeles
Los Angeles, CA 90025
tanmays@cs.ucla.edu

Abstract

Generative Adversarial Networks are hard to train and several recent works [16, 10, 17] have focused on improved regularization by controlling the spectra of weight matrices. Most recently, [10] proposed a new reparameterization technique which learns the Singular Value Decomposition of each weight matrix in the network - thus, allowing us to directly manipulate the spectra of the matrices. Our work builds on this existing body of literature by introducing a generalized method for training neural networks using this reparameterization and reducing the number of parameters by restricting the rank of each weight matrix. For a GAN, we find a theoretical upper bound on the distance between the original discriminator and its k-rank approximation along with good results on the CIFAR-10 dataset by using matrices with restricted rank. Furthermore, we demonstrate high accuracy on the MNIST dataset by using low rank weight matrices and show a significant decrease in the number of parameters required as compared to a network composed of traditional convolutional layers.

1 Introduction

Generative Adversarial Networks were first proposed by Goodfellow et al. [6] for learning complex distributions and it has since been used for various tasks such as Image to Image translation [21], image generation [18], inpainting [15], denoising [12] , and dialogue generation [19]. In a more general manner, GANs introduce a min-max problem where there is a competition between the discriminator and the generator network where the generator approximates the empirical data distribution and the discriminator tries to tell generated samples from true data samples. Recent research in GANs has focussed on finding the distance between distributions in a way suitable for the architecture [2], improving the generalization bounds and dealing with mode collapse [13], and understanding convergence along with finding equilibrium [14]. We now write the following min-max optimization problem as mentioned above:

$$\min_{\theta} \max_{\mathcal{W}} f(\theta, \mathcal{W}) = \frac{1}{n} \sum_{i=1}^{n} \phi(\mathcal{A}(\mathcal{D}_{\mathcal{W}}(x_i))) + \mathbb{E}_{x \sim \mathcal{D}_{G_\theta}} [\phi(1 - \mathcal{A}(\mathcal{D}_{\mathcal{W}}(x)))] \tag{1}$$

$\phi(x) : \log x$ and $\mathcal{A}(x) = \text{sigmoid(x)}$ is used in the Original GAN paper and $\phi(x) : x$ and $\mathcal{A}(x) = x$ denotes the Wasserstein GAN [1]. In the above equation $\{x_i\}_{i=1}^n$ denote n real data points and G_θ denotes the generative network parameterized by θ, $\mathcal{D}_\mathcal{W}$ is the discriminator parameterized by $\mathcal{W}$. ϕ maps the output of the discriminator to a value between $[0, 1]$. In the above context, the generator tries to maximize the objective function in equation (1) and the discriminator tries to minimize the objective with the important point being that the gradients obtained by using backpropagation are interdependent. Both of the networks can be trained by using gradient descent schemes such as SGD [5].

Despite the improvement in theoretical properties of GANs including generalization and finding of local nash equilibrium [2], these methods require strong assumptions which are not satisfied in practice very often. However, recent work has focussed on empirically improving the performance of GANs using a multitude of regularization techniques [16, 1, 17] which result in better performance on standard datasets including CIFAR-10 and the STL-10 datasets. Our focus in this paper is related to normalization techniques which focus on regularizing the spectra of weight matrices which includes spectral norm as introduced by [16] and the D-optimal regularization technique introduced by [10] which was an improvement on previous effort. More specifically, our paper uses the reparameterization trick [10] to separate each weight matrix into three distinct components similar to singular value decomposition. Not only does this enable us the control the spectra of weight matrices but it also helps us to control the rank of the weight matrix by changing the dimension of the diagonal singular value matrix.

The reparameterization is done as follows: Each weight matrix W_i is reparameterized as $U_i E_i V_i^T$ before the training process is started. So the discriminator becomes:

$$\mathcal{D}(x; \mathcal{U}, \mathcal{E}, \mathcal{V}) = U_L E_L V_L^T \sigma_{L-1}(U_{L-1} E_{L-1} V_{L-1}^T ... \sigma_1(U_1 E_1 V_1^T x)) \tag{2}$$

And this reparameterization can be used for regularization by adding relevant constraints to the loss term. One thing to note here is that different constraints can be added to each matrix which gives us control over the properties of each separate component. In (2), σ denotes the activation function, U and V denote orthonormal matrices and E denotes the diagonal matrix of singular values. The subscript included for each matrix denotes the layer to which that matrix belongs; $\mathcal{U}, \mathcal{E},$ and $\mathcal{V}$ are the sets of $\{U_1, ..., U_L\}$, $\{E_1, ..., E_L\}$ and $\{V_1, ..., V_L\}$ respectively. Here the dimension of E_i, where i denotes the layer number, is equal to $\min(d_i, d_{i+1})$ where d_i and d_{i+1} are the dimensions of the weight matrix of the i^{th} and $(i+1)^{th}$ layer respectively.

Our work extends this concept of regularization by further restricting the rank of each weight matrix W. We show that by empirically choosing a scaling factor S (which regulates the rank of W), we can obtain a significant decrease in the number of parameters and the size required to store the network in memory. We further demonstrate that, empirically, the number of network parameters is linearly dependent on S. Furthermore, this work gives the theoretical bound on the difference between the output of the discriminator network and the low rank discriminator network; the theorem introduced in our work bounds the value $\left\|D(x) - D^k(x)\right\|_\infty$ based on the largest excluded singular value e_{k+1}. We provide justifications for this theorem and later on speak more about its applicability.

2 Related Work

Previous work on GAN research has focused on finding theoretical upper bounds for the generalization gap in GANs which is measured using the $\mathcal{F}$ divergence distance metric introduced in [2]. [16] introduced the spectral normalization technique as an improvement on orthogonal regularization for GANs because they believed that orthogonal regularization was too restrictive and hampered the learning process. In their approach, they normalized the singular value of the weight matrices on every iteration such that the largest singular value was always equal to 1. To achieve the above task, they used the one step power method to approximate the singular value decomposition of each weight matrix.

However, [10] showed that there is another way to do this by using a method they called SVD reparameterization which is faster in practice and often more accurate because the one step power method underestimates the SVD almost consistently. They found a tighter upper bound than [4] for the generalization gap and showed good results by empirically slowing down the rate of singular value decay in weight matrices. Our work is related to [10] and we build on their reparameterization technique to introduce low rank matrices which result in low space complexity. To this end, our paper is consistent with their use of the D-optimal regularizer which gives good empirical results and achieves a slow singular value decay as claimed in their paper.

Furthermore, there have been a number of studies on reducing the size of convolutional neural networks to make them more efficient while training and reduce their space complexity [8, 11, 9]. [8] uses depth-wise separable convolutions to produce lightweight CNNs which can be used for mobile devices, [9] use low precision weight matrices (1 bit) at runtime and at training time quantized weights and gradients are used for speeding up computation. Other methods focus on compression using trained quantization and huffman encoding [7] which can significantly reduce the size of neural networks without affecting accuracy.

Our work is related to the first set of literature in the sense that we introduce an upper bound on the difference between the true discriminator and it's k-rank approximation while building on the framework introduced by [10] to reduce the rank of each weight matrix by a hyperparameter S which we call the scaling factor. The second body of literature relates to our work as all of the mentioned methods work to reduce the space complexity of neural networks while minimizing the loss in accuracy. Some of these methods introduce sophisticated techniques such as trained quantization and encoding methods to encourage the algorithm's compression power. To compare against these methods and to combine their methodology with ours is left as a direction of future research in this area.

3 Methodology

3.1 Rank K Approximation

Singular value decomposition is a general method that can be applied to any matrix of any shape to decompose it into three separate components:

$$W = UEV^T$$

where U and V denote orthonormal matrices and E denotes a diagonal matrix $\{e_1, e_2, e_3, ..., e_m\}$ where e_i denotes the i^{th} singular value and $e_1 \geq e_2 \geq ... \geq e_m$. Here, if $W \in \mathbb{R}^{p \times q}$. $m = \min(p, q)$. Thus the number of entries in the diagonal matrix is the minimum dimension of the matrix. In practice, if the matrix is very large, we can approximate it using only the k largest singular values $e_1, ..., e_k$ where $k \leq m$. The figure below illustrates the concept neatly:

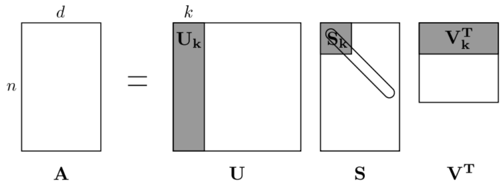

Figure 1: Visualization of K-Rank approximation of a matrix of any given size

This linear approximation is, in practice, very useful in compression and can retain most pertinent information regarding the matrix which can be used to create a reconstruction which is very similar [20]. For our work, we consider this restriction on the rank of a matrix to be another form of regularization which restricts the expressive power of the network. As noted in [16], orthogonal regularization restricts the network too much by forcing each singular value to be 1. Our low rank method, however, removes such restrictions while still limiting the network's expressive power.

3.2 SVD-Rank-k Reparameterization and D-Rank-k-optimal Regularization

In this section, we describe the modifications to the SVD reparameterization technique as introduced by [10]. We call this method SVD-Rank-k reparameterization. In SVD-rank-k reparameterization each weight matrix W_i is reparameterized as $U_i E_i V_i^T$ before the training process is started. So the discriminator becomes:

$$\mathcal{D}(x; \mathcal{U}, \mathcal{E}, \mathcal{V}) = U_L E_L V_L^T \sigma_{L-1}(U_{L-1} E_{L-1} V_{L-1}^T ... \sigma_1(U_1 E_1 V_1^T x)) \tag{3}$$

where E_i is a diagonal matrix with k_i rows where k_i is chosen according to a scaling factor S.

$$k_i = S * \min(d_i, d_{i+1}) \tag{4}$$

We also modify U_i and V_i by reducing the number of rows and columns so that the matrix multiplication works correctly.

Given that we need to make U and V orthonormal (singular value decomposition), we can directly add a regularization term in the objective function - penalizing deviance form the orthonormal behaviour. Thus the terms: $\left\|U_iU_i^T - I_i\right\|_F^2$ and $\left\|V_iV_i^T - I_i\right\|_F^2$ ensure that these two matrices are orthonormal. Specifically, we use this reparameterization to optimize the following objective function:

$$\min_{\theta} \max_{\mathcal{Q}(\mathcal{E}),\mathcal{U},\mathcal{V}} f(\theta, \mathcal{Q}(E), \mathcal{U}, \mathcal{V}) - \lambda \underbrace{\sum_{i=1}^{L}(\left\|U_iU_i^T - I_i\right\|_F^2 + \left\|V_iV_i^T - I_i\right\|_F^2)}_{\text{Orthogonal Regularization}} - \gamma\mathcal{R}(\mathcal{E})$$

where $\mathcal{Q}(\mathcal{E}) = \left\{\frac{E_1}{e_1^1}, ..., \frac{E_L}{e_1^L}\right\}$ and $\mathcal{R}(\mathcal{E})$ is the some other regularization term and f is defined using (1).

We use the D-optimal Regularization used in Jiang et al which ensures slow singular value decay [10]. The formula looks the same but note in our implementation only top k singular values are used as we have reduced the size of E_i

$$\mathcal{R}(\mathcal{E}) = \frac{1}{2}\sum_{i=1}^{L-1} \log(|(E_i^T E_i)^{-1}|) = -\sum_{i=1}^{L-1} \log(\prod_{k=1}^{r_i} e_k^i) \tag{5}$$

Thus, the reparameterization technique helps us in manipulating separate components of the matrix to ensure desirable properties in each one of them. We now mention a few properties and intuition behind this regularization method. D-optimal design [3] is a method in experimental design to estimate statistical parameters by using the minimum number of experiments. It focuses on maximizing the Fischer information matrix and allows correlation between individual features. The D-optimal regularizer also, in the same manner, focuses on increasing the log gram of the weight matrix to ensure a slower singular value decay - resulting in high correlation between features. We later show empirically how the performance of the network changes with the hyperparameter S on the MNIST dataset and the CIFAR-10 dataset.

4 Theory

In our paper, we make an assumption that $W_i^{k_i}$ has approximately the same top k singular values as W_i after each iteration, for all $i \in [L]$ where L is the number of layers in the network. Let $\mathcal{F}$ be the collection of composite functions $\mathcal{A}(D(\cdot))$, where $D(\cdot)$ is the L-layer discriminator network defined by (3). Then under the assumption that the input data $x_i \in \mathbb{R}^{d_i}$ is bounded, i.e., $\|x_i\|_2 \le B_x$ for $i \in [n]$, the activation operator σ_i is 1-Lipschitz with $\sigma_i(0) = 0$ for any $i \in [L-1]$, ϕ is ρ_ϕ-Lipschitz and the spectral norms of the weight matrices are bounded respectively, i.e., $\|W_i\|_2 \le B_{W_i}$ for any $i \in [L]$. the bound proven by Jiang et. al. [10] with probability $1-\delta$ over a joint distribution of $x_1, \cdots, x_n$,

$$d_{\mathcal{F},\phi}(\mu, v_n) \le \inf_{v \in D_G} d_{\mathcal{F},\phi}(\mu, v) + \mathcal{O}\left(\frac{\rho_\phi \beta\sqrt{d^2 L \log(\sqrt{dn}L\beta}}{\sqrt{n}} +_\phi \beta\sqrt{\frac{\log\frac{1}{\delta}}{n}}\right) + \epsilon$$

would hold in our case as well, as the $\|W_i^{k_i}\|_2 = \|W_i\|_2 \le B_{W_i}$. However, as each $W_i^{k_i}$ is a low k rank approximation of the weight matrix W_i, the difference between our discriminator, $D^k(x)$ with low k rank matrices and the discriminator, $D(x)$ used in Jiang et. al.'s work, can be shown in theorem 1.

Theorem 1: Let $D^k(x)$ be the k low rank approximation of the discriminator D(x), where $D(x) = W_L\sigma_{L-1}(\cdots\sigma_1(W_1x)\cdots)$ and $D^k(x) = W_L^{k_L}\sigma_{L-1}(\cdots\sigma_1(W_1^{k_1}x)\cdots)$ and $\|W_i\|_2 \le B_{W_i}$ for all $i \in [L]$ and $\|x_i\|_2 \le B_x$ for all $i \in [n]$, then we prove that:

$$\|D(x) - D^k(x)\|_\infty \le \sum_{i=1}^{L} \frac{B_x \prod_{j=1}^{L} B_{W_i}}{B_{W_i}} e_i^{k_i+1}$$

where $e_i^{k_i+1}$ is the largest singular value that was not included in our weight approximation for all for $i \in [L]$. View the Appendix for proof.

5 Evaluations

In this section we provide empirical results on the techniques we have described above. To demonstrate our proposed new methods we run experiments on CIFAR-10 and MNIST dataset. We train DC-GAN on CIFAR-10 and show comparable results on between model using scale 1 and scale 0.5. We also train a convolutional neural network for classification on MNIST dataset and provide results on accuracy and loss in the network over 4 different scales of 0.25, 0.5, 0.75 and 1. For both of these cases we also provide size of the model. All our implementaions were done in PyTorch.

5.1 DC-GAN

Most of the results in this section are limited due to our lack of resources to train the model. A future paper will include more experiments and empirical data on GAN training. We use a 4-Layer CNN for both our Discriminator and Generator. We set our training parameters $\lambda \in [1, 100]$ and $\gamma \in [0.25, 5]$. We noticed that performance wasn't sensitive to tuning these parameters. We were not able to recreate the results in Jiang et al. by setting scale to 1 as we were limited by computational constraints. We trained the GAN 2 times, first with scale of 1 and second with scale of 0.5. Results for the 100th epoch of the training are available in the appendix. We noticed that the Discriminator model size is almost linearly proportional to the scale, 11 MB for scale 1 vs 6MB for scale 0.5. Our future work will talk about more results on training of GANs

5.2 MNIST Classification

For MNIST classification we used a 3-layer CNN. Even here we set our training parameters $\lambda \in [1, 100]$ and $\gamma \in [0.25, 5]$. Here we have results for training for 5 epochs with a batch size of 64. We trained the same exact model for 4 different scales - 0.25, 0.5, 0.75 and 1.0.

Figure 2 shows a plot of the accuracy of the model. Note that for scale 0.5, 0.75 and 1.0 the accuracy is comparable to each other whereas the accuracy drops significantly when using 0.25. The Y axis notes the number of correctly classified samples where the test dataset size is 10,000.

Figure 3 shows a plot of the loss of the model at different scales. One thing to note here is that the loss at scale 0.25 is more volatile compared to the rest.

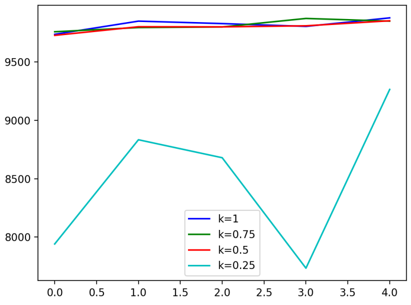

Figure 2: Accuracy

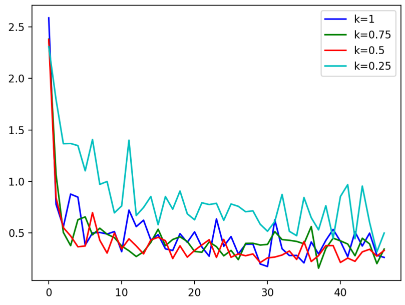

Figure 3: Loss

Scale	Number of Parameters	Size (in KB)
1.0	36663	145
0.75	26837	107
0.5	18373	74
0.25	8547	35

Table 1: Table comparing number of parameters and size at different scales

We also compare the size of the model and show significant improvement. Using the convolutional layer available in PyTorch we saw that the size of the model was 132 KB and number of parameters was 33580 layers. Using our method at scale 1 which is similar to the methods in Jiang et al. the size of the model increases to 145 KB and number of parameters increase to 36663. This is expected as instead of storing one weight matrix we are storing 2 orthonormal matrices and 1 diagonal matrix. As you can see in Table 1 varying the scale linearly reduces the size of the model. Intuitively we can also think that this will affect the computational complexity of the model but we are not able to test this for a small model that we have used for this task.

6 Future Work

As part of our future work, we would like to relax the assumption that $W_i^{k_i}$ has approximately the same top k singular values as W_i and find a tighter bound for the our generalization error over the function class $\mathcal{F}_k$, which consists of our low k-rank approximation matrices. Empirical future steps include more testing on GANs using SVD-rank-k reparameterization. This includes training at with different scales and comparing inception and FID scores. We also need more evaluation on the current experiments like checking if U_i and V_i are near orthonomal. We also need evaluation to compare how far apart the singular values are at different scale.

7 Conclusion

In this paper, we propose a new method to leverage the power of SVD reparmeterization by applying rank-k approximation to reduce the number of parameters in a given network while achieving competitive performance. We find a theoretical upper bound for the difference between the output of the discriminator network and its rank-k approximation which can be used to bound the generalization gap in GANs. Our experiments on the CIFAR-10 dataset and the MNIST dataset show that this method, both as a way of regularization and a way of reducing the space complexity, achieves comparable performance on the following two tasks: image classification and image generation. Furthermore, we empirically show a linear relation between the number of parameters in the network and the scaling factor proposed in our technique.

References

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

[2] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans). In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 224–232. JMLR. org, 2017.

[3] Bhavesh S Barot, Punit B Parejiya, Hetal K Patel, Mukesh C Gohel, and Pragna K Shelat. Microemulsion-based gel of terbinafine for the treatment of onychomycosis: optimization of formulation using d-optimal design. *AAPS PharmSciTech*, 13(1):184–192, 2012.

[4] Peter L. Bartlett, Dylan J. Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. *CoRR*, abs/1706.08498, 2017.

[5] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[6] 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.

[7] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[8] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[9] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *The Journal of Machine Learning Research*, 18(1):6869–6898, 2017.

[10] Haoming Jiang, Zhehui Chen, Minshuo Chen, Feng Liu, Dingding Wang, and Tuo Zhao. On computation and generalization of generative adversarial networks under spectrum control. 2018.

[11] Jonghoon Jin, Aysegul Dundar, and Eugenio Culurciello. Flattened convolutional neural networks for feedforward acceleration. *arXiv preprint arXiv:1412.5474*, 2014.

[12] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. *CoRR*, abs/1803.04189, 2018.

[13] Eric V Mazumdar, Michael I Jordan, and S Shankar Sastry. On finding local nash equilibria (and only local nash equilibria) in zero-sum games. *arXiv preprint arXiv:1901.00838*, 2019.

[14] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? *arXiv preprint arXiv:1801.04406*, 2018.

[15] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

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

[17] Henning Petzka, Asja Fischer, and Denis Lukovnicov. On the regularization of wasserstein gans. *arXiv preprint arXiv:1709.08894*, 2017.

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

[19] Sai Rajeswar, Sandeep Subramanian, Francis Dutil, Christopher Pal, and Aaron Courville. Adversarial generation of natural language. *arXiv preprint arXiv:1705.10929*, 2017.

[20] Nathan Srebro and Tommi Jaakkola. Weighted low-rank approximations. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 720–727, 2003.

[21] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017.

APPENDIX

Proof (Theorem 1)

We bound the term $\|D(x) - D^k(x)\|_\infty$

$$\begin{aligned}
\|D(x) - D^k(x)\|_\infty &\le \|D(x) - D^k(x)\|_2 \\
&\le \|W_L\sigma_{L-1}(\cdots\sigma_1(W_1x)\cdots) - W_L^{k_L}\sigma_{L-1}(\cdots\sigma_1(W_1^{k_1}x)\cdots)\|_2 \\
&\le \|W_L\sigma_{L-1}(\cdots\sigma_1(W_1x)\cdots) - W_L^{k_L}\sigma_{L-1}(\cdots\sigma_1(W_1x)\cdots)\|_2 + \\
&\qquad W_L^{k_L}\sigma_{L-1}(\cdots\sigma_1(W_1x)\cdots) - W_L^{k_L}\sigma_{L-1}(\cdots\sigma_1(W_1^{k_1}x)\cdots)\|_2 \\
&\le \|W_L - W_L^{k_L}\|_2\|\sigma_{L-1}(\cdots\sigma_1(W_1^{k_1}x)\cdots)\|_2 + \\
&\qquad \|W_L^{k_L}\|_2\|\sigma_{L-1}(\cdots\sigma_1(W_1x)\cdots) - \sigma_{L-1}(\cdots\sigma_1(W_1^{k_1}x)\cdots)\|_2
\end{aligned}$$

We know that σ_i is 1-Lipchitz with $\sigma_i(0) = 0$, therefore:

$$\begin{aligned}
\|D(x) - D^k(x)\|_\infty &\le \|W_L - W_L^{k_L}\|_2\|W_{L-1}^{k_L}(\cdots\sigma_1(W_1^{k_1}x)\cdots)\|_2 + \\
&\qquad \|W_L^{k_L}\|_2\|W_{L-1}(\cdots\sigma_1(W_1x)\cdots) - W_{L-1}^{k_{L-1}}(\cdots\sigma_1(W_1^{k_1}x)\cdots)\|_2
\end{aligned}$$

Using the Eckart-Young theorem, we know $\|W_L - W_L^k\|_2 = e_L^{k+1}$

$$\begin{aligned}
\|D(x) - D^k(x)\|_\infty &\le e_L^{k_L+1}B_x\prod_{j=1}^{L-1}B_{W_i} + \\
&\qquad \|W_L^{k_L}\|_2\|W_{L-1}(\cdots\sigma_1(W_1x)\cdots) - W_{L-1}^{k_{L-1}}(\cdots\sigma_1(W_1^{k_1}x)\cdots)\|_2 \\
&\le e_L^{k_L+1}B_x\prod_{j=1}^{L-1}B_{W_i} + B_{W_L}\Big(\sigma_i^{k_{L-1}}B_x\prod_{j=1}^{L-2}B_{W_i} + B_{W_{L-1}}(\cdots)) \\
&\le \sum_{i=1}^{L}\frac{B_x\prod_{j=1}^{L}B_{W_i}}{B_{W_i}}e_i^{k_i+1}
\end{aligned}$$

Empirical Results

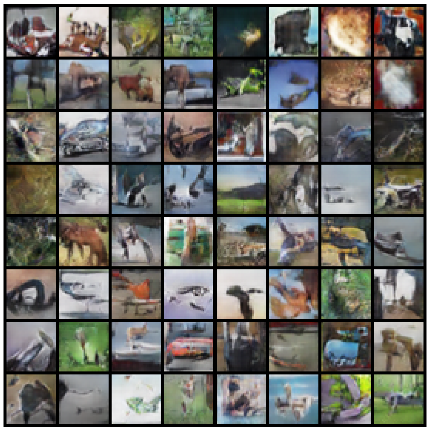

Figure 4: Scale = 1.0 (11MB)

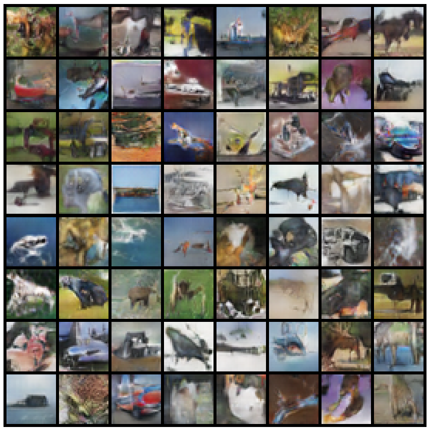

Figure 5: Scale = 0.5 (6MB)