Improved Training of Wasserstein GANs: A Summary

Dhruv Bhardwaj (SR19280) and Bhartendu Kumar (SR19649)
MTech (AI), Indian Institute of Science, Bangalore
dhruvb@iisc.ac.in, bhartenduk@iisc.ac.in

***Abstract*—This is a summary of "Improved Training of Wasserstein GANs" [1], and is a submission as a term paper for the course Advanced Deep Representation Learning (E9 333). In general, Estimating the Lipschitz constant of a neural network is NP-Hard and constraining lipschitz bound is intractable. Thus, the problem that this paper attempts to solve is to have a tractable and easily implementable methodology to bound lipschitz while not degrading the performance of network. The paper [1] introduces a gradient penalty term in loss of WGAN, which helps stabilize the discriminator of a Generative Adversarial Network (GAN) during training.**

***Index Terms*—Generative Adversarial Networks, Normalization, Unsupervised Learning, Generative Models**

I. Introduction

THE main problem with GANs is their instability in training. Introduction of WGANs promises to solve this problem but then for training a WGAN it needs the discriminator (or the critic) to be 1-Lipschitz. This again is a problem as it is very difficult to limit the space of functions that a neural network learns to be 1-Lipschitz functions.

Formally, the game between the generator G and the discriminator D is the minimax objective:

$$\min_G \max_D \mathbb{E}_{\mathbf{x}\sim\mathscr{P}_r}[logD(\mathbf{x})] - \mathbb{E}_{\tilde{\mathbf{x}}\sim\mathscr{P}_g}[log(1-D(\tilde{\mathbf{x}}))] \quad (1)$$

where $\mathscr{P}_r$ is the data distribution and $\mathscr{P}_g$ is the model distribution implicitly defined by $\hat{x} = G(z)$, $z \sim p(z)$(z is sampled from a simple distribution). The main issue is a powerful discriminator, if this happens then though the above loss is equivalent to minimizing the Jensen-Shannon divergence between $\mathscr{P}_r$ and $\mathscr{P}_g$ [2], but doing so often leads to vanishing gradients as the discriminator saturates.
Divergences which GANs typically minimize are potentially not continuous with respect to the generator's parameters, leading to training difficulty. WGAN tries to solve this problem as the **Wasserstein Divergence** $W(q,p)$ is under mild assumptions, continuous everywhere and differentiable almost everywhere [3]. WGAN objective is:

$$\min_G \max_{D\in\mathscr{D}} \mathbb{E}_{\mathbf{x}\sim\mathscr{P}_r}[D(\mathbf{x})] - \mathbb{E}_{\tilde{\mathbf{x}}\sim\mathscr{P}_g}[G(\tilde{\mathbf{x}})] \quad (2)$$

where $\mathscr{D}$ is the set of 1-Lipschitz functions.

The contributions are as follows:

1) On toy datasets, they demonstrate how critic weight clipping can lead to undesired behavior.
2) Propose gradient penalty (WGAN-GP), which does not suffer from the same problems.
3) Demonstrate stable training of varied GAN architectures, performance improvements over weight clipping, high-quality image generation, and a character-level GAN language model without any discrete sampling.

Need for this method:

1) Implementing a k-Lipshitz constraint via weight clipping biases the critic towards much simpler functions.
2) WGAN optimization process is difficult because of interactions between the weight constraint and the cost function, which result in either vanishing or exploding gradients

Advantages of this method:

1) Two-sided penalty: Method encourage the norm of the gradient to go towards 1 (two-sided penalty) instead of just staying below 1 (one-sided penalty). Empirically this seems not to constrain the critic too much
2) model's ability to train a large number of architectures which we think are useful to be able to train

Section II briefly describes method of GAN training using Gradient penalty strategy. Experimental details and analysis are presented in section III. The summary concludes in section IV.

II. Method

A differentiable function is 1-Lipschtiz if and only if it has gradients with norm at most 1 everywhere. In this method the gradient norm of the critic's output with respect to its input is constrained. for tractability , enforce a soft version of the constraint with a penalty on the gradient norm is enforced for random samples $\hat{\mathbf{x}} \sim \mathscr{P}_{\hat{\mathbf{x}}}$

A. New Objective

$$\min_G \max_{D\in\mathscr{D}} \mathbb{E}_{\mathbf{x}\sim\mathscr{P}_r}[D(\mathbf{x})] - \mathbb{E}_{\tilde{\mathbf{x}}\sim\mathscr{P}_g}[G(\tilde{\mathbf{x}})] + \lambda \mathbb{E}_{\hat{\mathbf{x}}\sim\mathscr{P}_{\hat{\mathbf{x}}}}[(||\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})||_2 - 1)^2] \quad (3)$$

Sampling Distribution $\mathscr{P}_{\hat{\mathbf{x}}}$ sampling uniformly along straight lines between pairs of points sampled from the data distribution $\mathscr{P}_r$ and the generator distribution $\mathscr{P}_g$.

$$\tilde{\mathbf{x}} \leftarrow G_\theta(\mathbf{z})$$
$$\hat{\mathbf{x}} \leftarrow \varepsilon x + (1-\varepsilon)\tilde{\mathbf{x}}$$

We get $\tilde{\mathbf{x}}$ from generator $\leftarrow G_\theta(\mathbf{z})$ having some noise vector z. They take an interporation on a line between real image and generated image. They take a random number ε between 0 and 1.

B. Normalization

The normalization which don't introduce correlations between examples can only be used like layer normalization [3] as a drop-in replacement for batch normalization.

III. Experiments

A. Modeling discrete data with a continuous generator

The KL divergences between two discrete distributions (not coincident) are infinite,and so the JS divergence is saturated. But WGAN behaves nicely in this scenario too and the Lipschitz constraint forces critic to provide a linear gradient to come inside the simplex constituted by discrete points.

B. Meaningful loss curves and detecting overfitting

Experiments showed that loss correlates with sample quality and converges toward a minimum. Loss converges as the generator minimizes $W(P_r, P_g)$.
In WGAN-GP, the training loss gradually increases even while the validation loss drops,when given enough capacity and too little training data.

IV. Conclusion

Gradient penalty seems to constrain the critic at the same time having a large space of function to learn which contribute to stability in training and better quality of images.

V. Future Work

Implementation is easy but whether it scales beyond a toy language model is unclear.

References

[1] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron Courville. Improved Training of Wasserstein GANs.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. NIPS, pp. 2672–2680, 2014.

[3] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. arXiv preprint arXiv:1701.07875, 2017

Proofs we came up with

1. Finding the Lipschitz bound of a CNN

Getting Lipschitz bound of a CNN is **NP-Hard** problem and to the best of our knowledge there is no efficient method for it. So, we started by having a lipschitz bound for just 1 convolutional layer. We got a closed form for Lipschitz of 1 layer convolution, though this **result was present in literature but the method is novel**.

Notations:

- Assume $\mathbf{T}$ as the operator equivalent to one layer convolutional neural network.
- C to be the number of channels of image at input. D to be the number of channels at output of this layer.
- **Filters** are $h_{i,j}$, $i \in 0,1,..,C-1$ and $j \in 0,1,2....,D-1$. Thus there are CxD filters representing the weights of the layer.
- We do not consider bias in this exploration.
- f be the input image with C channels. $\mathbf{T}f$ be the output.
- l: be length of image and b: be width of image.
- $\hat{h_{i,j}}$ be the fourier transform (DFT of filter)
- $\hat{f}$ be fourier transform of image
- ω be the lattic of frequencies in fourier domain of image
- Ω be total number of frequencies in fourier domain of image

$$||\mathbf{T}f||^2 = \sum_{d=1}^{\mathbf{D}} \sum_{c=1}^{\mathbf{C}} ||h_{cd} * f_c||^2 \qquad [definition]$$

$$= \sum_{d=1}^{\mathbf{D}} \sum_{c=1}^{\mathbf{C}} (\frac{1}{lb})^2 \sum_{\omega}^{\Omega} (\hat{h_{cd}}(\omega).\hat{f(\omega)})^2 \qquad [parseval]$$

$$\implies \frac{||\mathbf{T}f||^2}{||f||^2} = \frac{(\frac{1}{lb})^2 \sum_{d=1}^{\mathbf{D}} \sum_{c=1}^{\mathbf{C}} \sum_{\omega}^{\Omega} (\hat{h_{cd}}(\omega))^2 (\hat{f(\omega)^2}}{\sum_{c=1}^{\mathbf{C}} ||f_c||^2}$$

$$= \frac{(\frac{1}{lb})^2 \sum_{d=1}^{\mathbf{D}} \sum_{c=1}^{\mathbf{C}} \sum_{\omega}^{\Omega} (\hat{h_{cd}}(\omega))^2 (\hat{f(\omega)^2}}{(\frac{1}{lb})^2 \sum_{c=1}^{\mathbf{C}} ||\hat{f_c}||^2} \qquad [parsevals]$$

$$= \frac{\sum_{\omega}^{\Omega} \sum_{d=1}^{\mathbf{D}} \sum_{c=1}^{\mathbf{C}} (\hat{h_{cd}}(\omega))^2 (\hat{f(\omega)^2}}{\sum_{\omega}^{\Omega} \sum_{c=1}^{\mathbf{C}} ||\hat{f_c}||^2} \qquad [change of order]$$

$$= \frac{\sum_{\omega=1}^{\Omega} \hat{f_\omega}^T \mathscr{M}_\omega^T \mathscr{M}_\omega \hat{f_\omega}}{\sum_{\omega}^{\Omega} \hat{f_\omega}^T \hat{f_\omega}}$$

here $\mathscr{M}_\omega := \mathscr{M}_\omega(d,c) = \hat{h_{dc}}(w) \in \mathbb{R}^{\mathbf{D}\times\mathbf{C}}$
$\hat{f_\omega}$:= [$\hat{f_1}(\omega)$ .. $\hat{f_2}(\omega)$ ... $\hat{f_{\mathbf{C}}}(\omega)$] $\in \mathbb{R}^{\mathbf{C}}$

$$\frac{||\mathbf{T}f||^2}{||f||^2} \leq \frac{\sum_{\omega=1}^{\Omega} \lambda_{max}(\omega) \hat{f_\omega}^T \hat{f_\omega}}{\sum_{\omega}^{\Omega} \hat{f_\omega}^T \hat{f_\omega}}$$

where $\lambda_{max}(\omega) :=$ largest eigen value of the matrix $\mathscr{M}_\omega^T \mathscr{M}_\omega$
let $\sum_{\omega}^{\Omega} \hat{f_\omega}^T \hat{f_\omega} = 1$ as its fraction.

$$\frac{||\mathbf{T}f||^2}{||f||^2} \leq \sum_{\omega}^{\Omega} \lambda_{max}(\omega)\alpha_\omega \quad (\sum_{\omega}^{\Omega} \alpha_\omega = 1)$$

$$\leq max_\omega\{\lambda_{max}(\omega)\} \qquad [max of convex combination is largest]$$

Thus, :

$$\implies \frac{||\mathbf{T}f||^2}{||f||^2} \leq max_\omega\{\lambda_{max}(\omega)\} \qquad [\lambda_{max}(\omega) := \lambda_{max}(\mathscr{M}_\omega^T \mathscr{M}_\omega)]$$

$$\frac{||\mathbf{T}f||}{||f||} \leq max_\omega\{\sigma_{max}(\omega)\} \qquad (\sigma_{max}(\omega) = (\lambda_{max}(\omega))^{1/2})$$

This gives, Lipschitz constant, $\mathbb{L} = max_{\omega}\{\sigma_{max}(\mathscr{M}_{\omega})\}$ where $\sigma_{max}(.)$ is largest singular value
Hence Proved.

2. Gradient of Network w.r.t input is upper bound on Lipschitz

Setting:
Assuming a Fully Connected feed - forward DNN having relu activations.

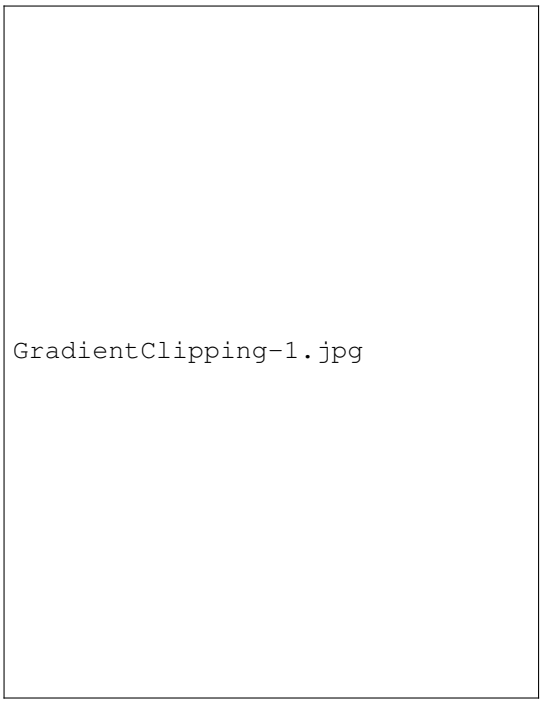

Fig. 1. Part 1 of Proof of Claim 2

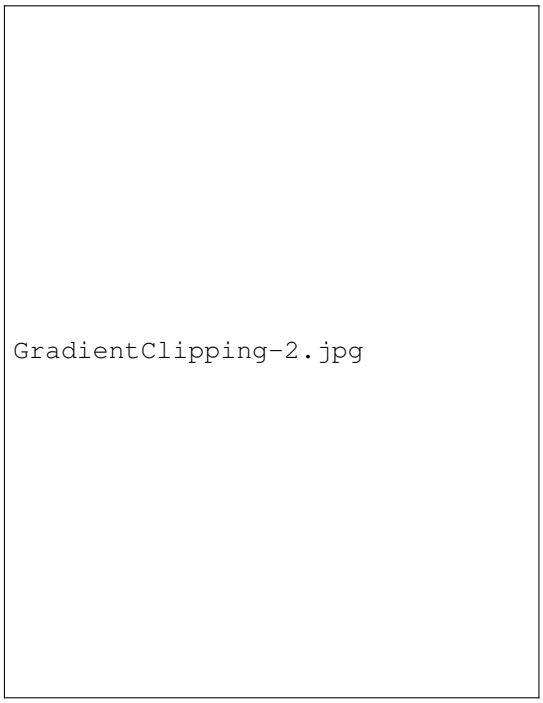

Fig. 2. Part 2 of Proof of Claim 2

3. Gradient Clipping can give Lipschitz 1

Setting:
Assuming a Fully Connected feed - forward DNN having relu activations.

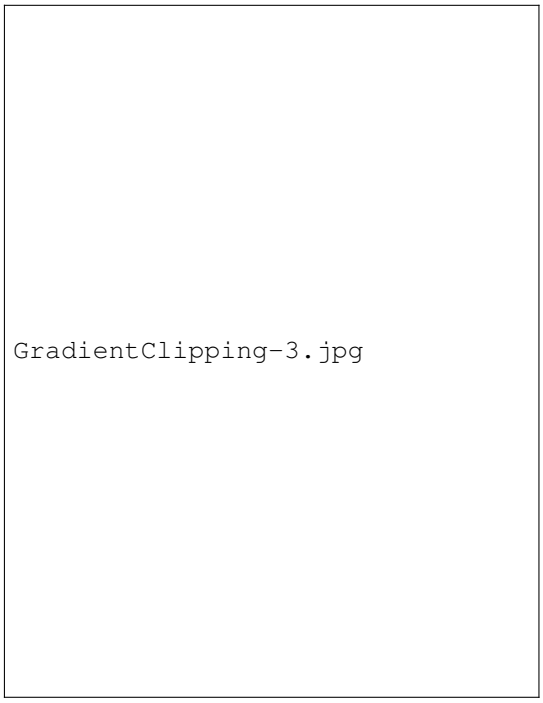

Fig. 3. Proof of Claim 3