Algorithms and Optimization for Big Data

Project Report

GAN Training using Regret Minimization

Group 5

Contributors :
Chintan Gandhi (201501019)
Jay Mohta (201501036)
Ativ Joshi (201501040)
Pratik Padalia (201501084)

1. Abstract

In this experiment we try to find an efficient way to find a single set of weights, which when tested with a number of inputs provide minimum loss. The Generative Adversarial Network (GAN) based model is experimented in different fashions using different window sizes. The different approaches are discussed with respect to the updation of loss functions in the given scenario.

2. Introduction

Generative Adversarial Network (GAN) is a framework for estimating generative models via an adversarial process. We simultaneously or alternatively train two feedforward neural networks: generator (*G*) and discriminator (*D*) and this setting corresponds to a minimix two-player game. The input to generator includes input from latent space mixed with noise. The network outputs some distribution we would like to achieve. The discriminator takes as input a set of data, either real (*x*) or generated (*G(z)),* and outputs a probability of that data being real (*D(x)*).

In these set of experiments we have tried to study the following topics :

a. Convergence of the generator and discriminator loss with respect to the window size
b. Accuracy comparison for different window sizes using SVM classifier

3. Related Works

The original GAN approach as an adversarial game was described in GoodFellow et al. (2014). Challenges in the traditional GAN training approaches still prevail as they battle against the issues of mode collapse and cycling. Kodali et al. (2017) propose studying GAN training dynamics as regret minimization and provide a proof for its convergence in artificial convex-concave case. Grnarova et al. (2017) formulate the problem of training GAN as finding a mixed strategy in a zero-sum game. They propose a novel training algorithm called Chekhov GAN. This alternate approach considers finding mixed strategies and leverages online learning algorithms. In our work, we have tried to introduce the concept of local regret in GAN training dynamics.

4. Loss Function

$$\min_G \max_D V(D, G) = \underline{\mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]} + \underline{\mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]}.$$

At each iteration, the weights of both the networks D and G changes. Thus we get a 'new' loss function which we need to store for calculating regret. Thus effectively, we need the old weights of D and G before the backpropagation.

5. Regret Minimization

Regret minimization is the framework by which we try to minimize the loss incurred by taking into account loss in previous networks for the current input. Basically, the weights of the neural network are not updated only by the current loss but also includes the previous losses. The loss function defined for a generative adversarial network is, in fact, regret function but with a window size 1. As window size increases, training time increases, fluctuations are minimized and there is faster convergence to the equilibrium condition.

6. Convergence

We have seen the concept of regret minimization above. In this scenario we have tried to apply regret minimization function on different sets of losses to analyze what gives us the best results. Using regret minimization of both the losses with a window size, w = 2, the generator loss and the discriminator loss, a random data generation was observed with no convergence of the loss.

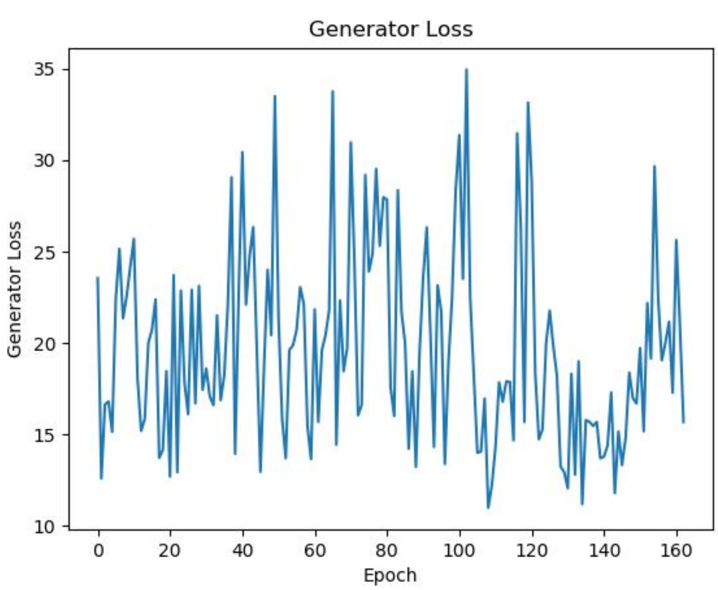

Fig 1. Window Size (w) = 2

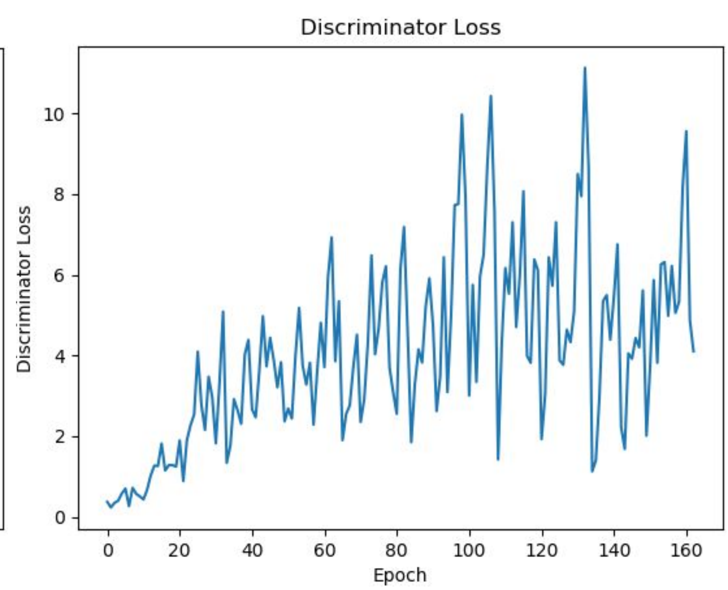

Fig 2. Window Size (w) = 2

On applying regret minimization only on the generator loss with window size, w = 2, the results generated had a better convergence rate for the generator loss when compared with the convergence rate for the generator loss with window size, w = 1.

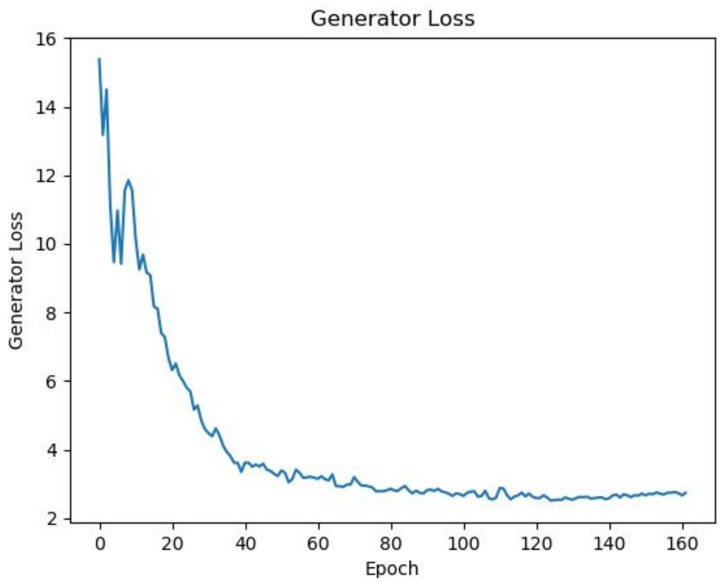

Fig 3. Window Size (w) = 2

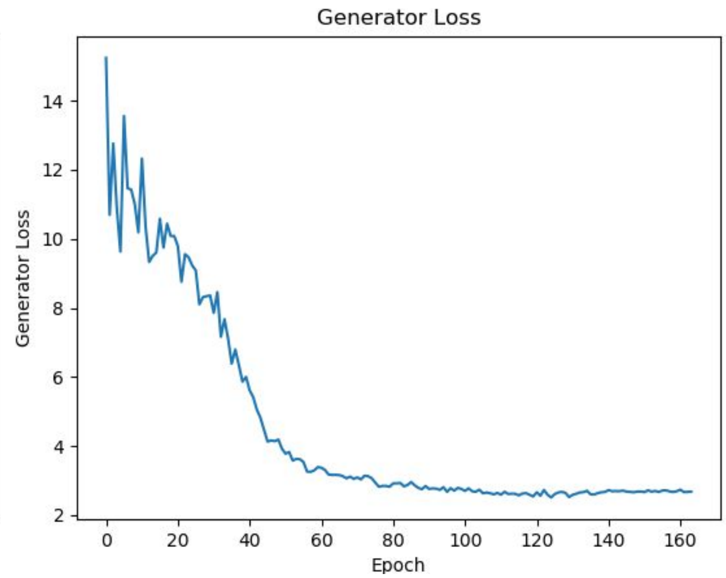

Fig 4. Window Size (w) = 1

The discriminator loss in both cases have similar results. The discriminator loss for window size, w=2 converges faster than that for window size, w=1.

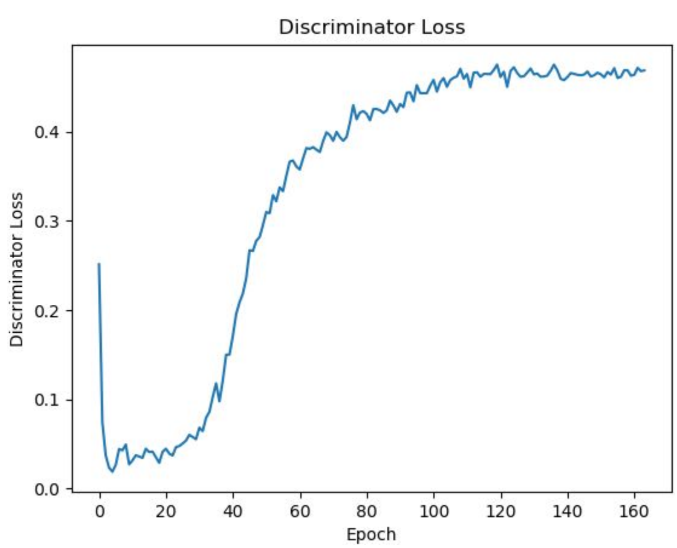

Fig 5. Window Size (w) = 2

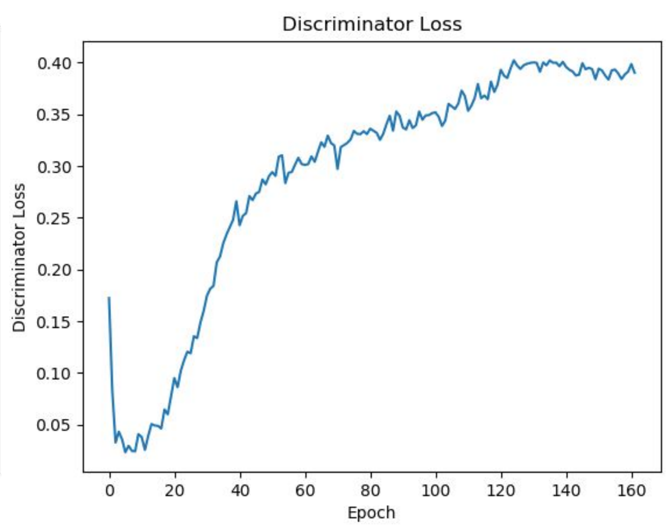

Fig 6. Window Size (w) = 1

Window size 3 also gives a similar convergence both in the case of generator and discriminator.

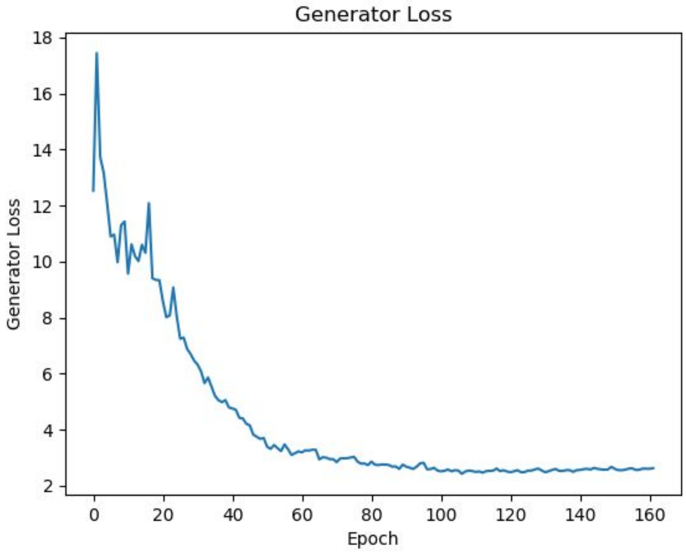

Fig 7. Generator Loss, Window Size (w) = 3

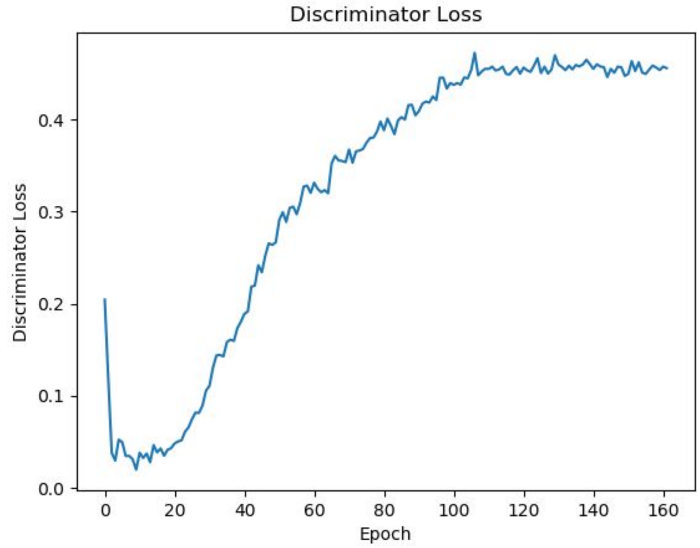

Fig 8. Discriminator Loss, Window Size (w) = 3

We observe a faster convergence for a bigger window size because a larger window size will result into more stability. The errors can be in either directions and thus for a smaller window, say, w = 1 the losses in different directions are going to neutralise each other wasting two steps and leads to more fluctuations and instability and hence slower convergence rate. On the other hand when we consider a larger size window, say, w = 3, it takes the average of losses and thus does not vary much, leading to a smoother curve and faster convergence rate. However, the step size decreases for a larger window size.

7. Classifier and Accuracy

After training the GAN for 161 epochs, we analyse the accuracy of the discriminator on the MNIST dataset. We used SVM to find the accuracy with which it can classify digits which were generated by GANs. The SVM hyperparameters used were C=5 and Gamma=0.05 for RBF Kernel.
For window size, w = 1, we have an accuracy of 95% and for window size, w = 2, we have an accuracy of 96%.
Both the cases have a good accuracy and not much of difference in between them, this says that the GAN is trained equally well after 161 epochs in both the cases.

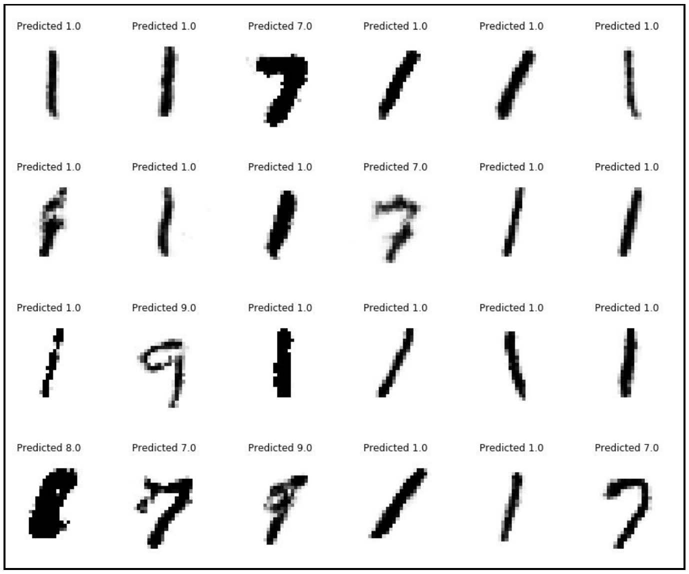

Fig 9. Results : Classifier Predictions Window Size (w) = 1

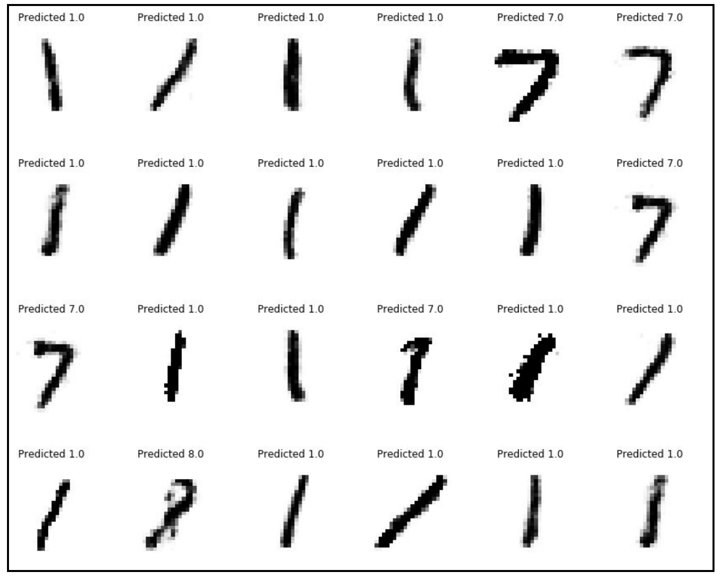

Fig 10. Results : Classifier Predictions Window Size (w) = 2

8. Conclusion

From these experiments, we figured out that we can have the concept of windowing for training GANS with a higher convergence rate with a greater window size on the generator window while it does not work out when applied to the discriminator. Although, when we incorporate greater window size, the computation for each step increases and this leads to greater training time.

9. Future Works

There are several questions unanswered like, what window size would lead to highest convergence rate. Will the accuracy of the images generated vary with window size? There can be some correlation between regret minimization technique and Adam Optimiser and can work together that could result into a more stable environment.

10. References

1. Goodfellow, Ian, 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*, pp. 2672-2680. 2014.
2. Grnarova, Paulina, Kfir Y. Levy, Aurelien Lucchi, Thomas Hofmann, and Andreas Krause. "An online learning approach to generative adversarial networks." *arXiv preprint arXiv:1706.03269* (2017).
3. Kodali, Naveen, James Hays, Jacob Abernethy, and Zsolt Kira. "On convergence and stability of gans." (2018).
4. SVM classfier - https://github.com/ksopyla/svm_mnist_digit_classification
5. https://www.cs.toronto.edu/~duvenaud/courses/csc2541/slides/gan-foundations.pdf