Evaluation of Oscillatory Activation Functions on MNIST dataset using GANs

Prith Sharma[1], Aditya Raj[2], Sushant Sinha[3], and Shubham Bharadwaj[4]

[1]VIT University
[1]*prith.sharma2019@vitstudent.ac.in*
[2]*adityaraj.sahoo2019@vitstudent.ac.in*
[3]*sushant.sinha2019@vitstudent.ac.in*
[4]*shubham.bharadwaj2013@vitalum.ac.in*

Abstract

MNIST is essentially a database of handwritten digits and thus, having a dataset such as this with over 70,000 images makes it a great source to perform experimentation on. Thus, in this particular paper, we are working on exploring the use and output of the oscillatory activation functions on the MNIST dataset by generating a GAN to generate these kind of handwritten digits.The code has been written essentially in Pytorch and we have used a number of oscillatory activation functions and viewed the results. A GAN is a network in which there are 2 neural networks, i.e. the generator and discriminator which are pitted against each other. This is essentially done to enable the generator to generate a fake image and the discriminator to classify that image as real or fake. The training is done accordingly. Thus, viewing the results of oscialltory activation functions in GANs might open us up to new possibilites and help us understand whether the oscillatory activation functions would yield better results or not in the case of GANs.

1 Related Work

A. What are GANs:

GANs stands for Generative Adversarial Networks. This type of modelling categorized as unsupervised learning. GANs comprises two models namely: Generator (G(x)) and Discriminator (D(x)). These models are capable of automatically discovering and learning the patterns in the input data. The Generator is responsible for creating fake data which is to be to be trained on the discriminator. It learns how to generate plausible data. The instances generated by the Generator are considered to be the negative training sample for the Discriminator. Here, the Generator attempts to make the Discriminator believe that the generated instances are real. The Discriminator is responsible for differentiating between fake instances generated by the Generator and real instances. Training data for discriminator includes positive samples and negative samples. The Discriminator is penalized for misclassifying the data, using this loss, the weights are updated through back propagation. There are different types of GANs: Vanilla GANs, Deep Convolutional GANs, Conditional GANs and Super Resolution GANs.

Preprint. Under review.

B. MNIST:

MNIST stands for Modified National Institute of Standards and Technology. This database consists of handwritten digits from 0 to 9 and is present in the form of 28x28 pixel grayscale image. The database comprises of 10,000 test examples and 60,000 examples for training. MNIST database is considered as a benchmark because of its simplicity, ease of use and application on real-world data.

C. Oscillatory functions:

Oscillatory activation functions comprises of multiple hyperplanes in their decision boundary. This enables the neurons to make more complex decisions than other popular sigmoidal, ReLu like Swish, and Mish activation functions. The higher representative power of networks with oscillating activation functions allows classification and regression tasks to be solved with fewer neurons. Also, the oscillations in the activation function appear to improve gradient flow and speed up back propagation learning[6]. The results presented in [6] suggest that deep networks with oscillating activation functions might potentially partially bridge the performance gap between biological and artificial neural networks.Oscillatory functions have been proved to outperform many other activation functions. The better performance can be noticed with benchmarking as well.

2 Results and Discussion

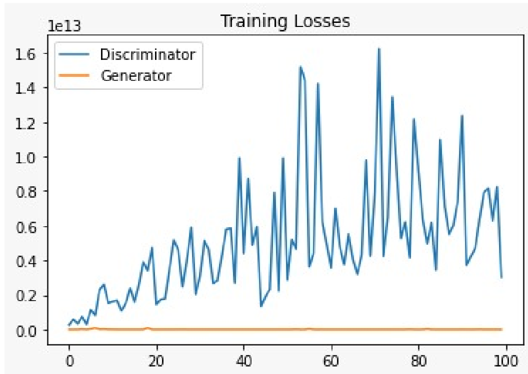

Figure 1: z2ccosz

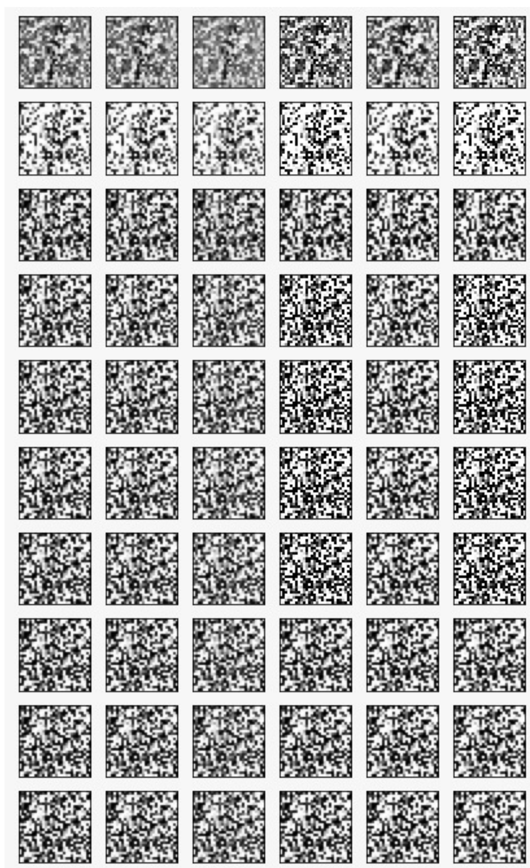

Figure 2: final generated images for z2cosz over time

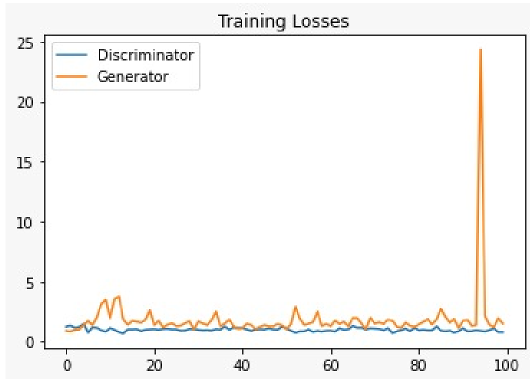

Figure 3: leaky Relu

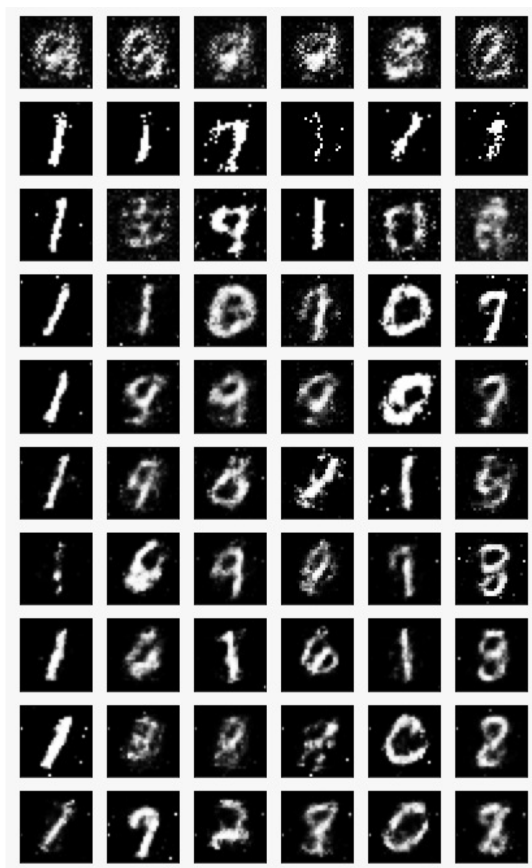

Figure 4: final generated images for leaky Relu over time

Figure 5: sine

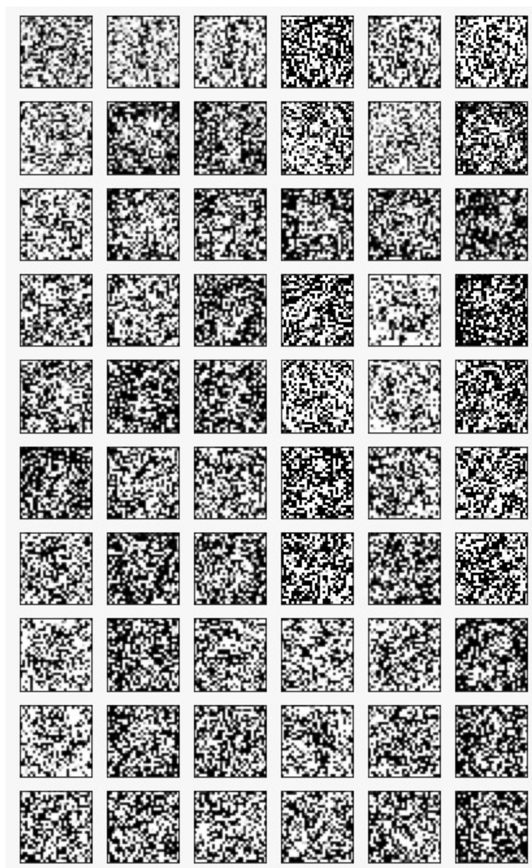

Figure 6: final generated images for sine activation over time

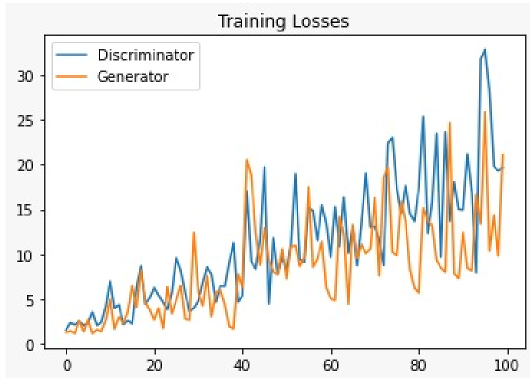

Figure 7: for GCU

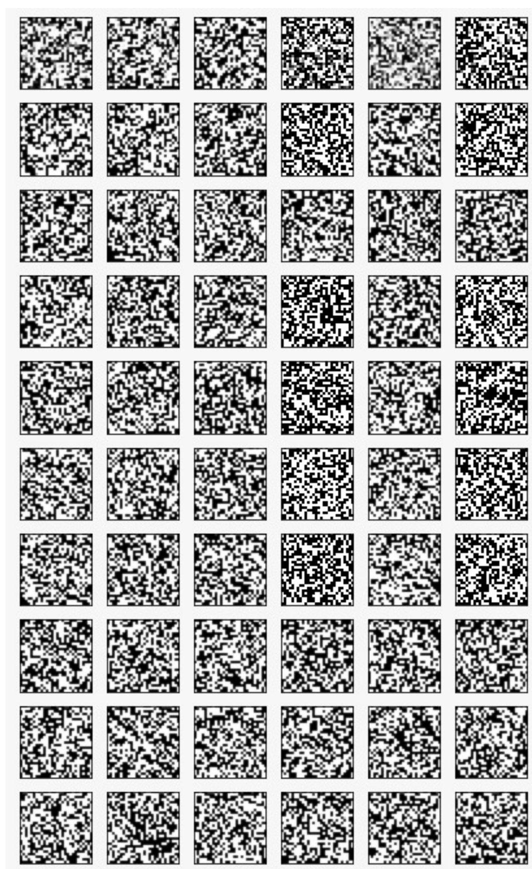

Figure 8: final generated images for GCU overtime

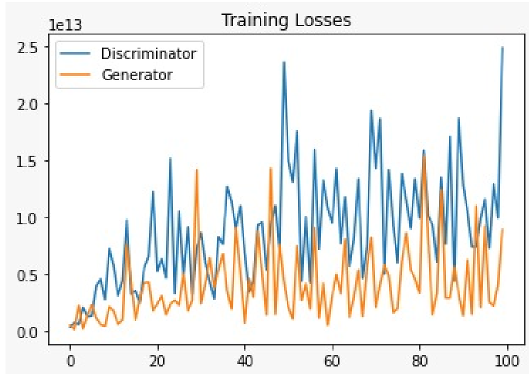

Figure 9: for cos2

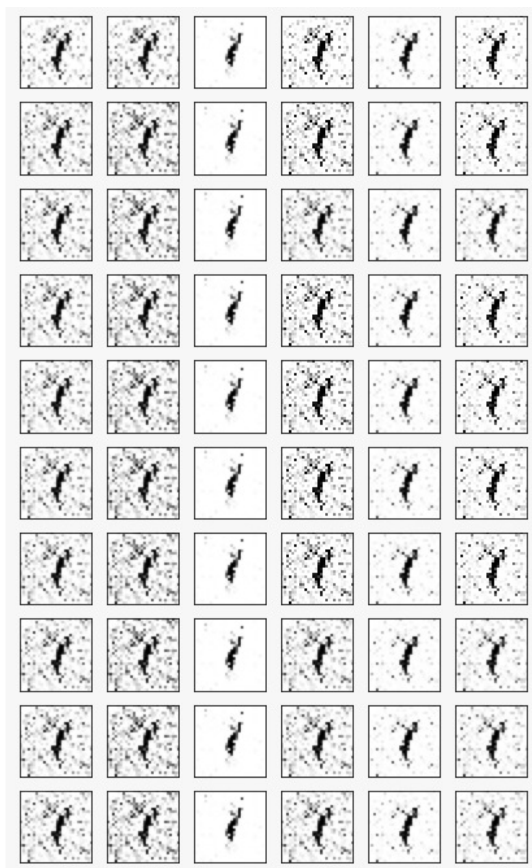

Figure 10: final generated images for cos2

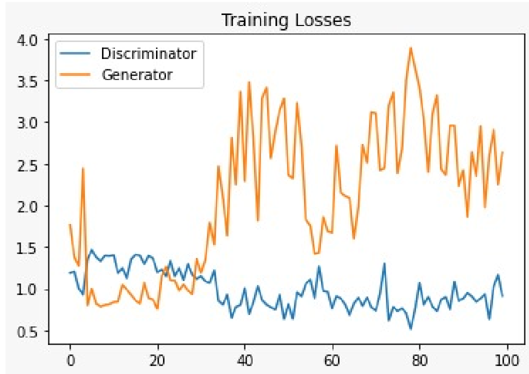

Figure 11: Sign sin activation loss curves

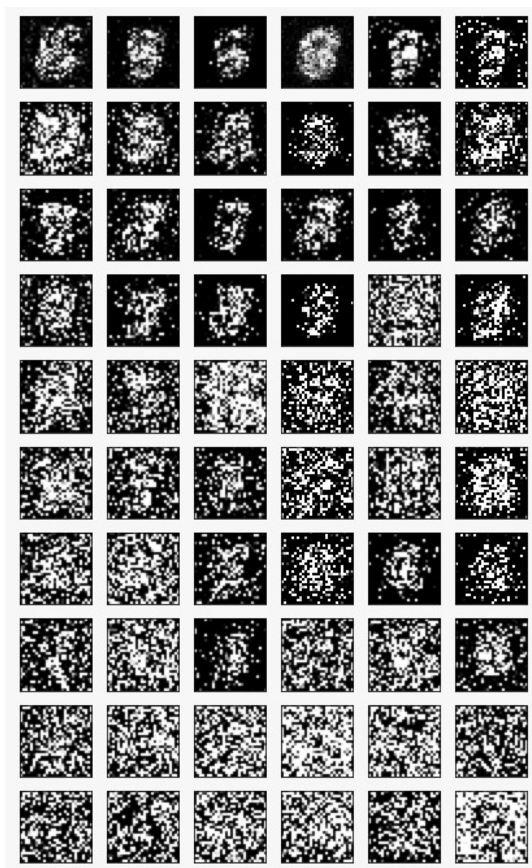

Figure 12: final generated images for sign sin

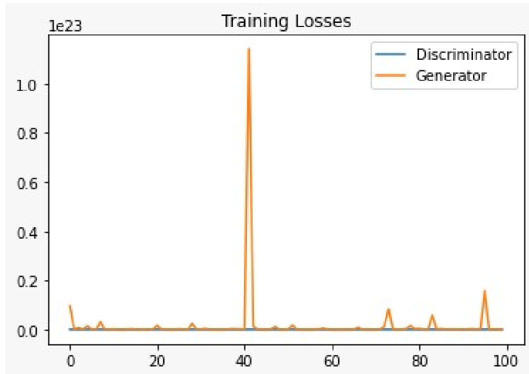

Figure 13: new act 1

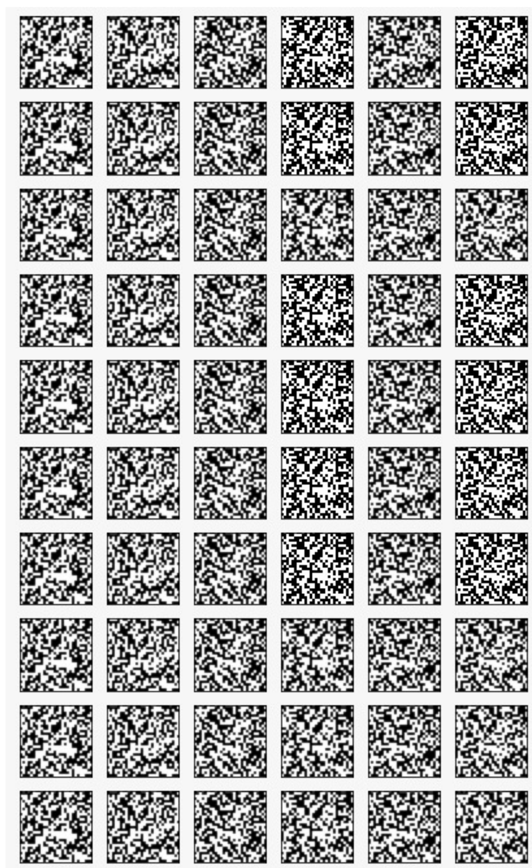

Figure 14: final generated images for new act 1

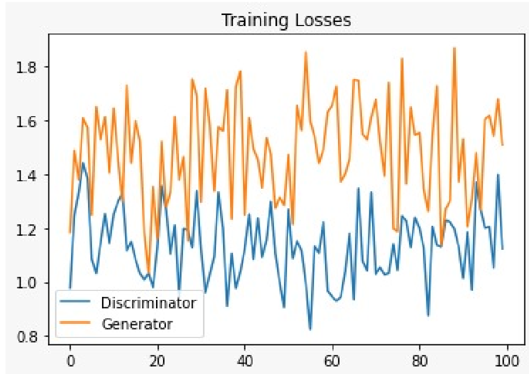

Figure 15: signum

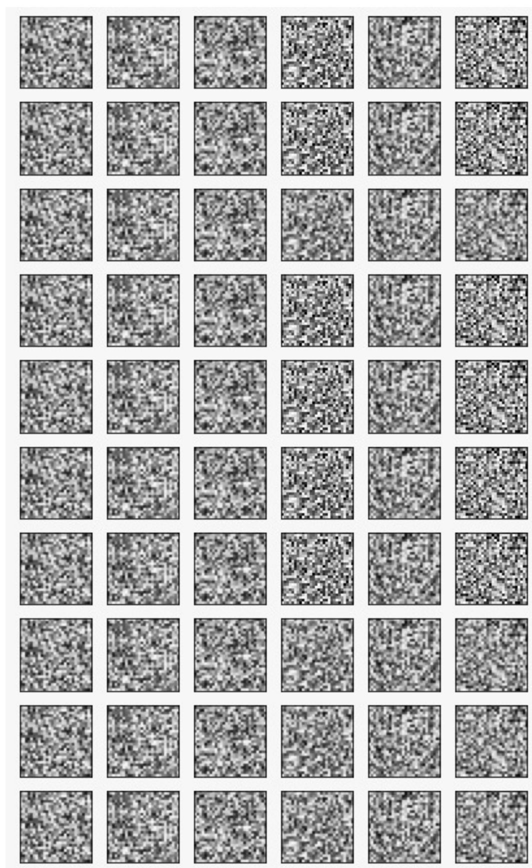

Figure 16: final generated images for signum

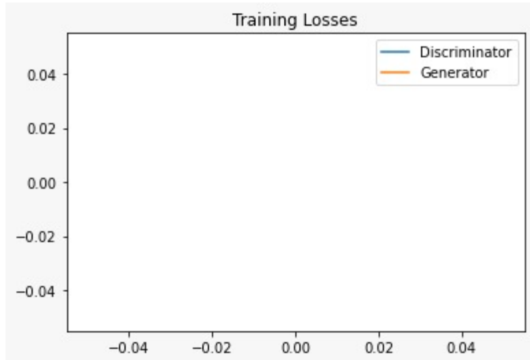

Figure 17: silu

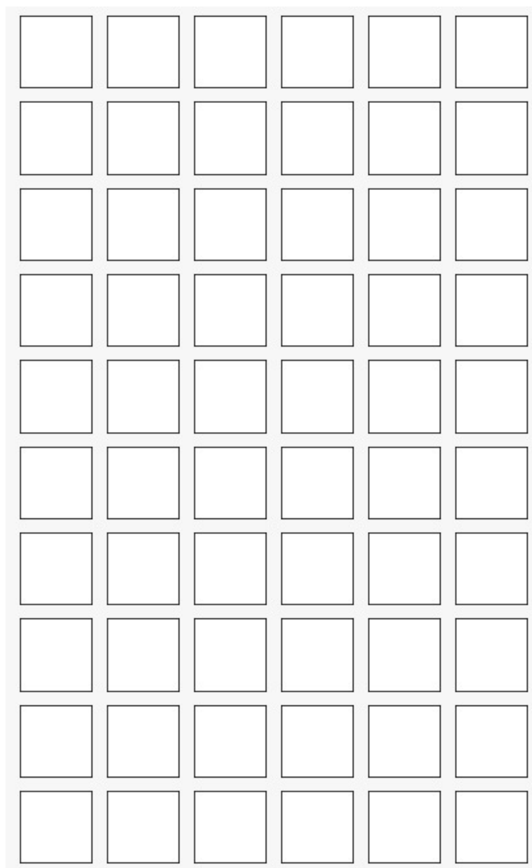

Figure 18: the final images generated(none)

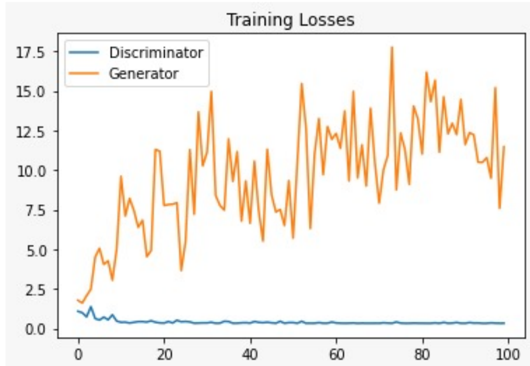

Figure 19: LiSHT activation

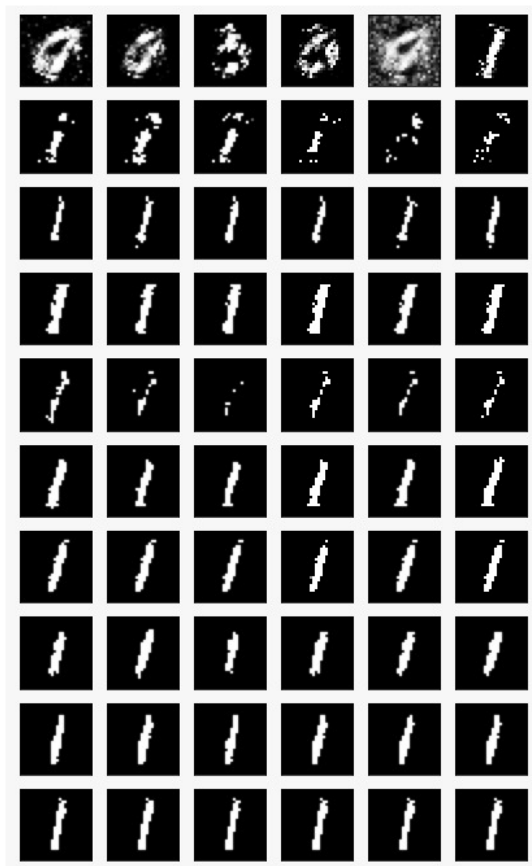

Figure 20: Final generated images for LiSHT

The following 2 are for: act ReSech

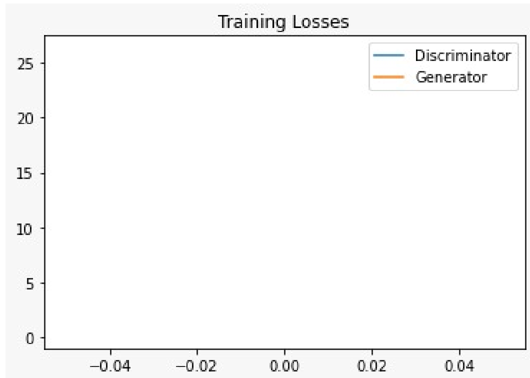

Figure 21: no loss (no output) for ReSech

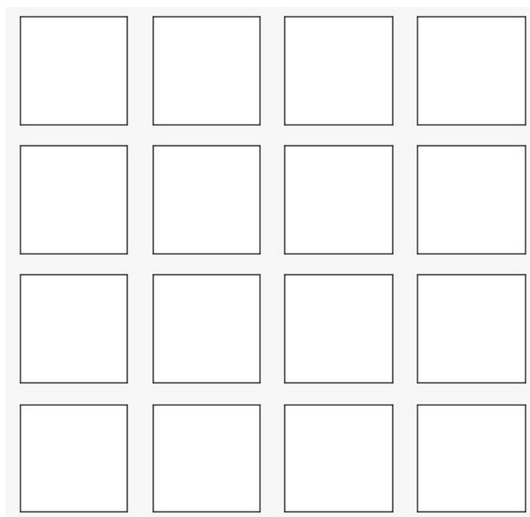

Figure 22: no final generated images for ReSech

Figure 23: bipolar

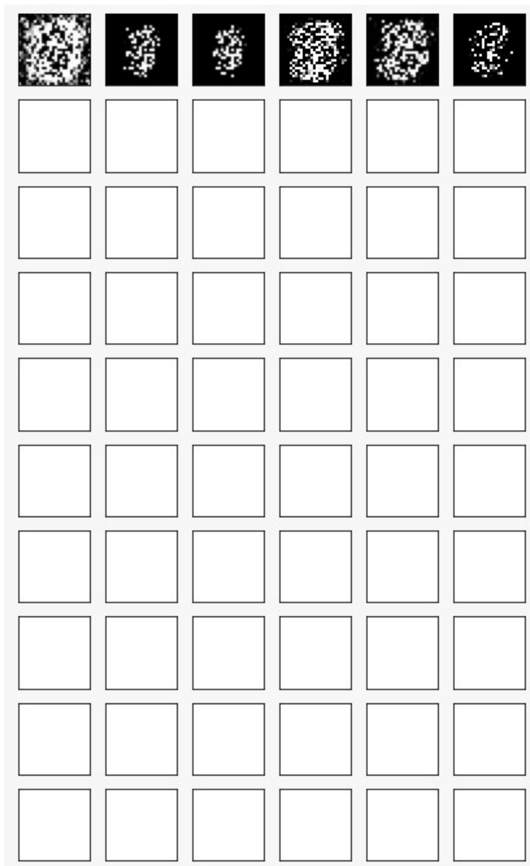

Figure 24: final generated images for bipolar activation over time

....

Figure 25: absolute

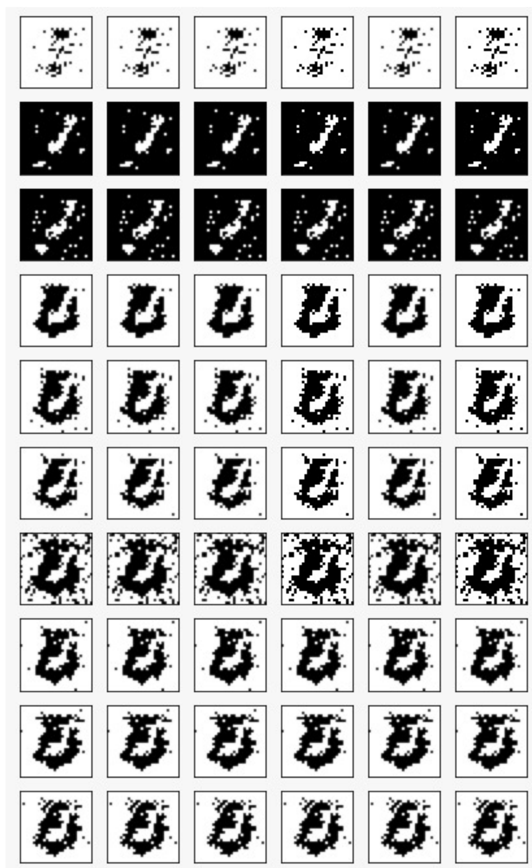

Figure 26: final generated images for absolute activation over time

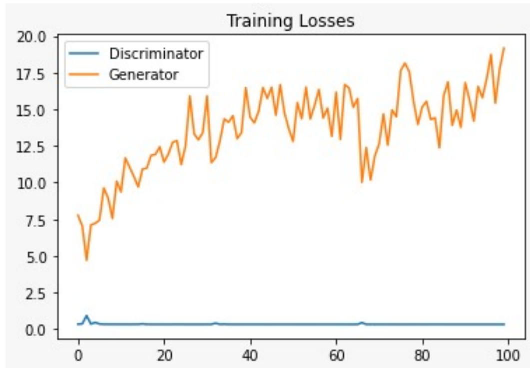

Figure 27: elliott

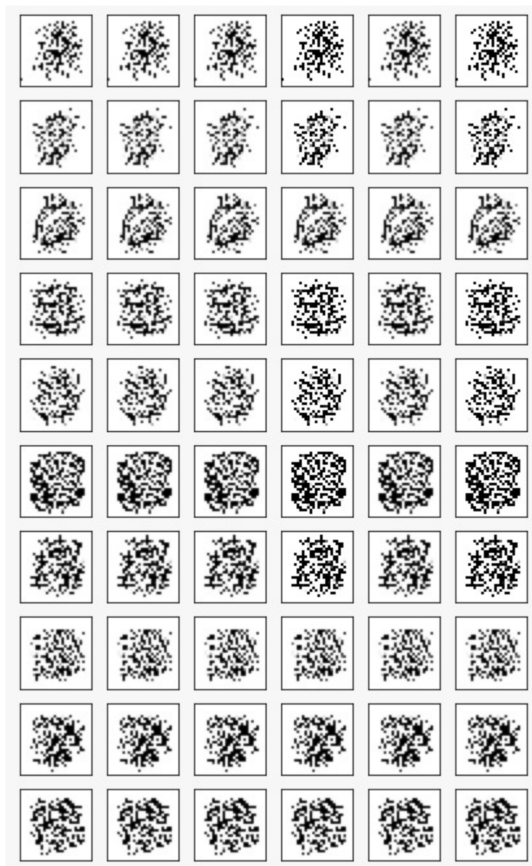
Figure 28: final generated images for elliott activation over time

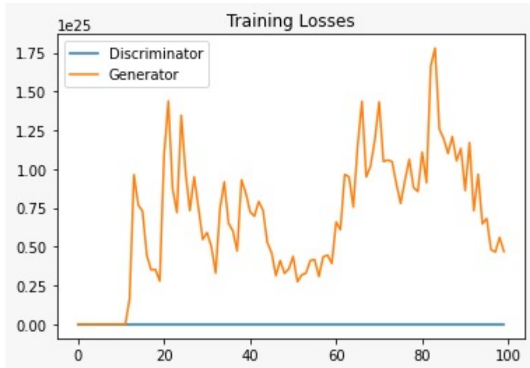

Figure 29: quadratic

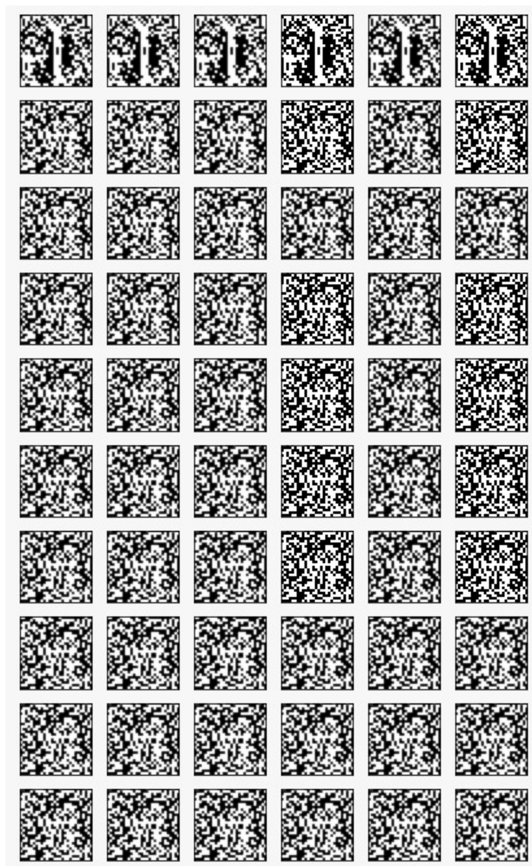

Figure 30: final generated images for quadratic activation over time

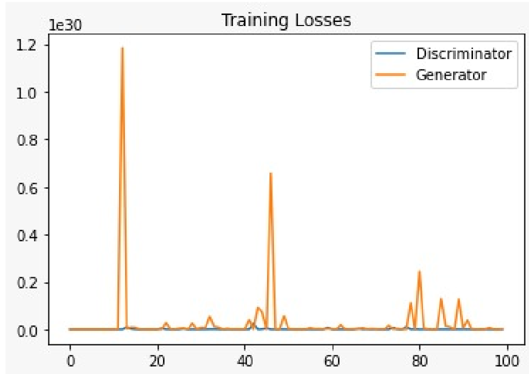

Figure 31: Mcubic

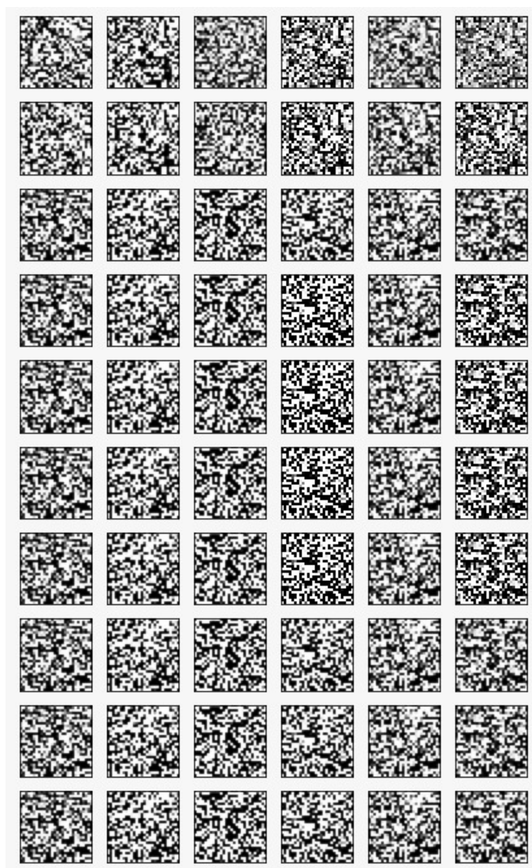

Figure 32: final generated images for Mcubic activation over time

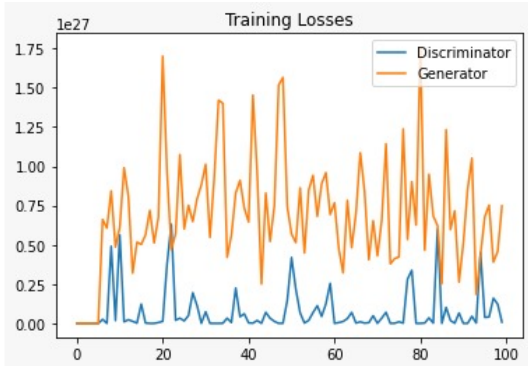

Figure 33: NMcubic

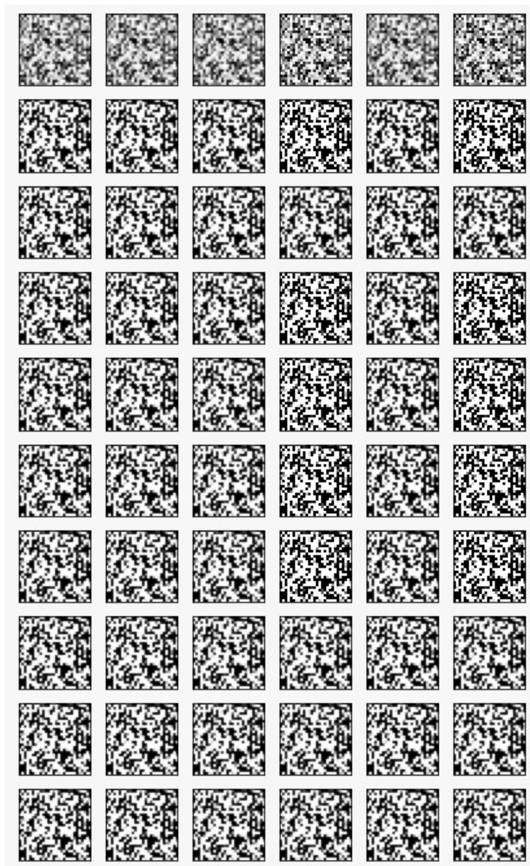

Figure 34: final generated images for NMcubic activation over time

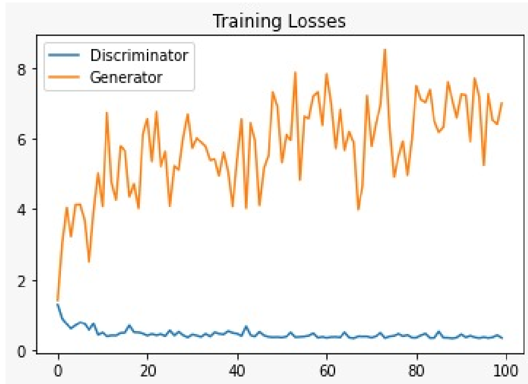

Figure 35: shiftedSinc

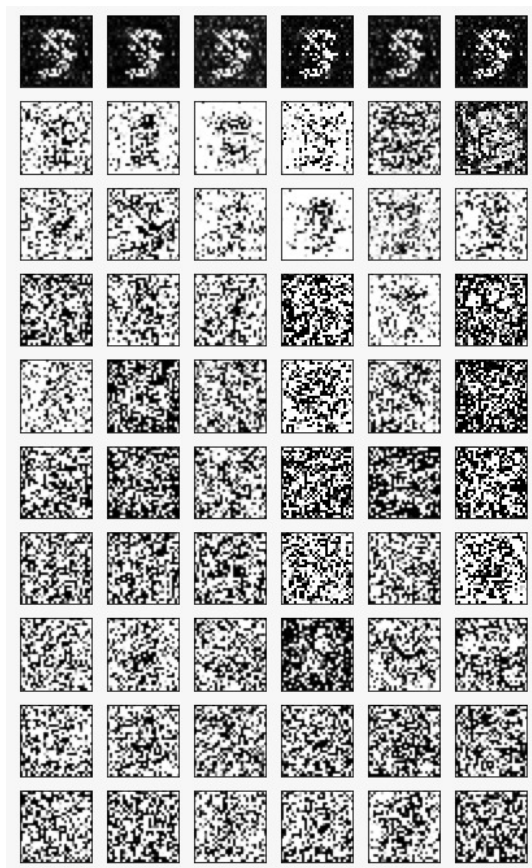

Figure 36: final generated images for shiftedSinc activation over time

Figure 37: DSU

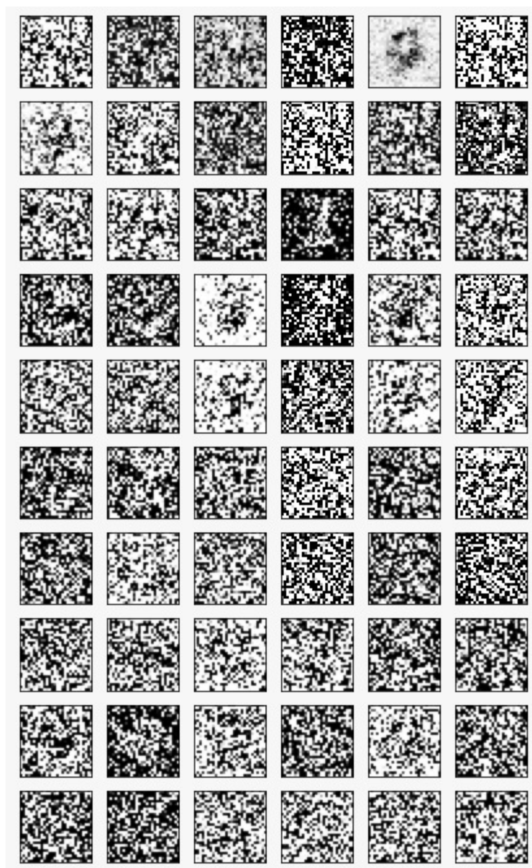

Figure 38: final generated images for DSU activation over time

3 Conclusion and Future Work

We used several oscillatory activation functions on MNIST dataset using GANs. Most of the activation functions were giving good results. Leaky Relu was giving the most reasonable results.

References

[1] How to Identify and Diagnose GAN Failure Modes. https://machinelearningmastery.com/practical-guide-to-gan-failure-modes/.

[2] THE MNIST DATABASE of handwritten digits. http://yann.lecun.com/exdb/mnist/.

[3] Top 6 Metrics To Monitor The Performance Of GANs. https://analyticsindiamag.com/top-6-metrics-to-monitor-the-performance-of-gans/.

[4] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[5] Mathew Mithra Noel, Advait Trivedi, Praneet Dutta, et al. Growing cosine unit: A novel oscillatory activation function that can speedup training and reduce parameters in convolutional neural networks. *arXiv preprint arXiv:2108.12943*, 2021.

[6] Matthew Mithra Noel, Shubham Bharadwaj, Venkataraman Muthiah-Nakarajan, Praneet Dutta, and Geraldine Bessie Amali. Biologically inspired oscillating activation functions can bridge the performance gap between biological and artificial neurons. *arXiv preprint arXiv:2111.04020*, 2021.