A practical Approach to Handwritten Digit Generation with Generative Adversarial Networks on MNIST

S. K. Abhyudhy[1], Dr. Savitha. C[2], Kareem Khan[3]

[1,3]Dept of AI&ML, Sri Siddhartha Instititue of Technology, Tumakuru
[2]Assistant Professor, Dept of AI&ML, Sri Siddhartha Instititue of Technology, Tumakuru

Abstract

Implementation of a Generative Adversarial Network (GAN) utilizing the MNIST (Modified National Institute of Standards and Technology database) dataset of handwritten digits, the GAN comprises a generator creating synthetic images from random noise and a discriminator to classifying these images as real or fake. The Generator utilizes leaky ReLU (rectified linear unit) activations, Batch normalization and reshaping to produce 28x28 grayscale images, while the Discriminator uses dense layers, leaky ReLU, and dropout to enhance classification accuracy. Both networks are trained using the Adam optimizer to improve stability and performance. The GAN is trained in an adversarial setup where the Generator seeks to create convincing images and the Discriminator aims to correctly classify them. Results demonstrate the GAN's ability to generate high-quality handwritten digit images, displaying its effectiveness and providing valuable insights into best practices for GAN implementation and training.

Keywords: Handwritten Digit Generation, Neural Networks, Image Synthesis, batch normalization, Adam Optimizer, Generative Modelling.

1. Introduction

The resolution and quality of images generated by generative techniques, such as generative adversarial networks (GANs), have shown significant advancements in recent years[1]. The properties of the latent space, which refers to the space where data is encoded into a lower-dimensional representation, are not thoroughly understood. This includes the commonly demonstrated technique of latent space interpolations, where intermediate points between two representations are generated, revealing the potential variations within the data.[2,3,4].

In a generative model without conditioning, the data is generated without any control over the specific characteristics or modes of the generated data. However, when additional information is incorporated to condition the model, such as class labels, specific parts of the data for inpainting, or data from a different modality, it becomes possible to guide the data generation process towards desired outcomes.[5].

Understanding the mechanics of GANs and the training of deep convolutional neural network models within a GAN architecture for image generation can be a complex endeavour. For those new to this domain, it is recommended to start by practicing the development and application of GANs on established image datasets commonly used in computer vision, such as the MNIST handwritten digit dataset. By working with manageable and well-understood datasets, it becomes feasible to swiftly develop and train smaller models, thereby enabling a sharper focus on the model architecture and the process of image generation itself. [6].

In the context of Generative Adversarial Networks (GANs), it is important to note that a notable global equilibrium point exists when both the generator and discriminator are able to access their complete strategy sets. However, it's crucial to consider that this equilibrium is not assured when their strategy sets are limited [7] .

Within a Generative Adversarial Network (GAN), the generator model holds a pivotal role in the process of creating new and precise data. This component is designed to take random noise as its input and then transform it into sophisticated data samples, such as text or images. Visually, the generator is often depicted as a deep-rooted neural network. During the training phase, the generator is tasked with capturing the inherent distribution of the training data by leveraging layers of adaptable parameters in its architecture. To achieve this, the generator fine-tunes its parameters through the use of backpropagation. By doing so, it adjusts its outputs to produce samples that closely emulate real data. The generator's triumph lies in its capability to generate top-quality, varied samples that are capable of deceiving the discriminator. This deceptive quality ultimately results in the successful operation of the GAN[8]. :

The original MNIST images were adjusted to fit in a 20x20 pixel box and then placed in a 28x28 image using the centre of mass of the pixels. Some methods work better when images are centred using a bounding box instead of the centre of mass.The MNIST database was created from NIST's Special Database 3 and Special Database 1. The training set has 30,000 patterns from each database, while the test set has 5,000 patterns from each. The training and test sets have different groups of writers.SD-1 contains 58,527 digit images written by 500 different writers. Writer identities were used to unscramble the writers, and the database was split into two sets of nearly 30,000 examples each. Some methods involved deskewing input images, while others involved augmenting the training set with distorted versions of the original samples[9].

Learning reusable feature representations from large unlabelled datasets, such as images and videos, is an active area of research. Generative Adversarial Networks (GANs) can be used to train good image representations and later reuse parts of the networks as feature extractors for supervised tasks like image classification. GANs provide an alternative to maximum likelihood techniques and offer attractive qualities for representation learning, despite being known for being unstable to train. There is limited published research on understanding and visualizing what GANs learn and their intermediate representations. With the incredible progress in technological advancements, particularly in the field of Generative Adversarial Networks (GANs), it has become feasible to produce exceedingly lifelike content that can easily deceive human perception. Deepfake technology represents the forefront of visual and audio manipulation, employing sophisticated deep learning techniques to create remarkably realistic and convincing content. Visual deepfakes encompass a wide range of categories, including lip sync, attribute manipulation, full-image synthesis, body re-enactment, and face manipulation. While deepfake technology has brought significant benefits to the realms of education and entertainment, its detrimental potential is evident through its capacity to tarnish individuals' reputations, sow societal discord, and threaten national security[10]

2. Literature Review

The GAN operates on three foundational principles. First, the generative model learns to create data by using a probabilistic representation, allowing for the generation of new, realistic data. Second, the training of the model can adapt to conflicting scenarios, fostering robustness and versatility. Lastly, the system utilizes deep learning neural networks and sophisticated artificial intelligence algorithms for comprehensive and effective training.[11]

In the examination of the significant impact of 2D to 3D image conversion using GAN, the initial step involves retrieving live data and establishing a benchmark with key features from the corresponding dataset. Subsequently, image merging is undertaken to calculate the threshold and suitability score. Image data pre-processing steps encompass image segmentation and cleansing, which are followed by GAN training. The anticipated outcomes include pattern analysis and the precision of image generation.[10].

Array programming via NumPy provides a powerful, concise, and expressive syntax for working with data in vectors, matrices, and higher-dimensional arrays. It is crucial in various research fields and has been essential in significant scientific discoveries. NumPy serves as the foundation of the scientific Python ecosystem and supports diverse scientific and industrial analysis.[12]

All existing layers in both networks remain trainable throughout the training process. When new layers are added, they are smoothly integrated to avoid disrupting the training of the existing layers. This method has several benefits. Initially, generating smaller images is more stable because there is less information to handle. By increasing the resolution step by step, the process, making it easier to create higher quality images. In practice, this approach stabilizes the training enough to reliably create high-quality megapixel-scale images using WGAN-GP and LSGAN loss functions. [13]

In a recent study, a new technique for transferring the style of one image onto another was introduced. The process involves extracting statistics from a pre-trained convolutional network and using them to generate a stylized image. Despite impressive results, the method is computationally inefficient, prompting recent works to develop faster alternatives. [14]

Generative adversarial networks (GANs) have emerged as a powerful framework for creating generative models. They are used across various applications and datasets. GANs are designed to replicate a specific distribution using a model distribution generated by a generator and distinguished by a discriminator. During the training process, the goal is to minimize the distinction between the model and target distributions using the most effective discriminator available. GANs have garnered widespread interest due to their capability to learn structured probability distributions and their theoretical underpinnings. Training the discriminator is akin to training an accurate estimator for the density ratio between model and target distributions, enabling variational optimization without requiring direct knowledge of the density function.[15]

Stochastic gradient descent (SGD) is the primary optimization algorithm used in deep learning. While SGD is effective at finding minima that generalize well, each parameter update only takes a small step towards the objective. Recently, there has been a growing interest in large batch training in an attempt to increase the step size and reduce the number of parameter updates required to train a model. Large batches offer the advantage of being parallelized across multiple machines, thereby reducing training time. However, a common issue with increasing the batch size is the observed decrease in test set accuracy.To provide insight into this surprising observation, researchers have proposed interpreting SGD as integrating a stochastic differential equation. They have demonstrated that the scale of random fluctuations in the SGD dynamics,. Additionally, the researchers have discovered that there exists an optimal fluctuation scale, g, that maximizes test set accuracy (at a constant learning rate), and this suggests an optimal batch size that is proportional to the learning rate when . These empirical findings have been leveraged to successfully train ResNet-50 to achieve a 76.3% ImageNet validation accuracy in just one hour.[16]

Deep Neural Networks (DNNs) have shown remarkable progress in image recognition, but they often become over-confident when training on specific samples, affecting their ability to generalize to new test samples. To address this, researchers have introduced various methods to prevent overfitting, such as Label Smoothing, Bootstrap, CutOut, MixUp, DropBlock, and ShakeDrop. Label smoothing, a simple yet effective tool, uses soft labels derived from an average between hard labels and a uniform distribution over labels, providing strong regularization. However, it may treat non-target categories equally, limiting its effectiveness. Recent research suggests a new method that considers the relationships among different categories by maintaining a moving label distribution for each category, updating them during the training process. These evolving label distributions effectively build relationships between target and non-target categories, providing more reliable soft labels. [17].

The Convolutional Neural Network (CNN) has been highly successful in various computer vision tasks, such as image classification, object detection, and tracking. Despite its depth, a key characteristic of modern deep learning systems is the use of non-saturated activation functions to replace their saturated counterparts. The advantage of using non-saturated activation functions lies in two aspects: Firstly, it solves the problem of "exploding/vanishing gradient." Secondly, it accelerates the convergence speed. Among all non-saturated activation functions, the most notable one is the Rectified Linear Unit (ReLU). In simple terms, it is a piecewise linear function that prunes the negative part to zero and retains the positive part. One of its desirable properties is that activations are sparse after passing through ReLU.[18]

Stochastic gradient-based optimization is important in various scientific and engineering disciplines. It can efficiently optimize scalar parameterized objective functions, making it valuable for tasks such as maximization or minimization. Gradient descent is a common approach for differentiable functions, and stochastic gradient descent (SGD) has proven effective for optimizing objectives with stochastic characteristics as seen in machine learning and deep learning.[19]

3. METHODOLOGY

The methodology employed in this study revolves around optimizing the training process of a GAN for the MNIST dataset, with the primary goal of achieving faster convergence while maintaining high output quality in fewer epochs. Figure 1 represents the Gans working style. A Generative Adversarial Network (GAN) is a type of neural network that consists of two main components: a generator and a discriminator. The generator network takes random noise as input and transforms it into an image, while the discriminator network attempts to distinguish between real images from the dataset and fake images generated by the generator. [20] These two networks compete against each other in a zero-sum game, with the generator aiming to produce images that the discriminator cannot correctly classify as fake, and the discriminator aiming to correctly identify all fake images. The generator and discriminator are trained simultaneously through backpropagation, with the error signals from the discriminator used to update the weights of both networks. This iterative process leads to the improvement of both the generator and discriminator, resulting in the generation of increasingly realistic images. GANs have a wide range of applications, including image generation and manipulation, data augmentation, style transfer, super-resolution, medical image analysis, anomaly detection, and video generation. [21]

The approach includes several key modifications:

1. **Progressive Growing:** The GAN begins its training process by working with 14x14 images before advancing to 28x28 images midway through the training. This method provides the model with an opportunity to initially understand the broader features and patterns present in the images before transitioning its focus towards capturing finer details. Consequently, this sequential learning strategy results in quicker and more stable convergence of the model.

2. **Instance Normalization and Spectral Normalization:** The Generator network utilizes Instance Normalization instead of Batch Normalization. This approach is beneficial for stabilizing the training process because it normalizes individual samples rather than entire batches, resulting in smoother learning. On the other hand, the Discriminator incorporates Spectral Normalization, which is a technique used to limit the spectral norm of the weight matrices. This further contributes to stabilizing the training process by controlling the magnitude of the weights, leading to improved training dynamics.
3. **Learning Rate Scheduling:** During the training process, a learning rate scheduler is utilized to systematically reduce the learning rate. This approach allows the GAN to finely adjust its parameters as it progresses through training, which helps to prevent overshooting and enhances convergence.
4. **Increased Batch Size and Label Smoothing:** Raising the batch size during training stabilizes the process and enables the model to converge more rapidly by yielding more consistent gradient estimates. Moreover, label smoothing is employed on the real images to prevent the Discriminator from becoming excessively sure of its predictions, thereby fostering a more balanced and constructive adversarial training dynamic.

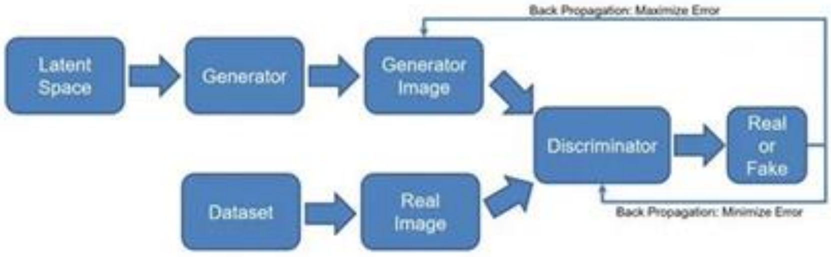

Figure 1: Block diagram

4. Results

In the first 20 epochs shown in Figure (2), the GAN began to learn basic digit structures, focusing on coarse features due to the implementation of Progressive Growing. By Epoch 10, the model started producing images where digit shapes were recognizable, but the images were still blurry and lacked detail. The Instance Normalization in the Generator helped maintain consistent feature scaling, which was crucial for the Generator's early learning stages. By Epoch 20, the images became significantly clearer, with most digits easily identifiable, although some still showed minor artifacts. The combination of Spectral Normalization in the Discriminator and label smoothing helped maintain a balanced training dynamic, preventing the Discriminator from overpowering the Generator. At this point, the GAN was showing promising results, setting the foundation for further refinement in subsequent epochs.

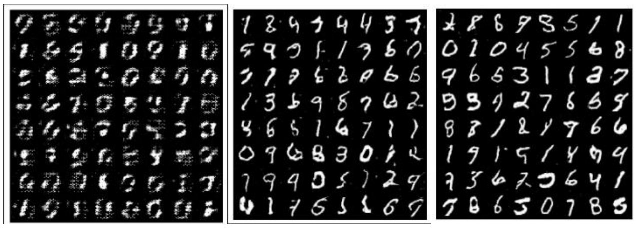

Figure: (2) Early Learning and Initial Quality, (3) Clarity and Stability Improvements, (4) High-Quality and Consistent Results.

Between Epochs 21 and 40, shown in Figure (3) the GAN made substantial progress in improving image clarity and reducing noise. By Epoch 30, the generated images were of high quality, with well-defined digits and minimal artifacts. The Spectral Normalization continued to stabilize the Discriminator, ensuring that the training process remained balanced and preventing mode collapse. Instance Normalization helped the Generator refine details, making the digits more distinct and realistic. The learning rate scheduler began to lower the learning rate gradually, which allowed the model to fine-tune its parameters without destabilizing the training process. By the end of Epoch 40, the GAN consistently produced high-quality images that were nearly indistinguishable from real MNIST digits, marking a significant achievement in the training process.

From Epochs 41 to 60, shown in Fig (4) the GAN reached a phase where the generated images exhibited consistently high quality, with all digits appearing clear and well-formed. By Epoch 50, the model had effectively mastered the generation of realistic handwritten digits, with the training process becoming more focused on fine-tuning and maintaining quality rather than making significant leaps in performance. The increased batch size played a crucial role in this phase, ensuring stable gradient updates and furtherenhancing the efficiency of training. The learning rate scheduler continued to reduce the learning rate, which allowed for delicate adjustments that improved image detail and consistency. By Epoch 60, the model was producing high-quality images with minimal variation, demonstrating that the training had stabilized and the GAN was performing at an optimal level.

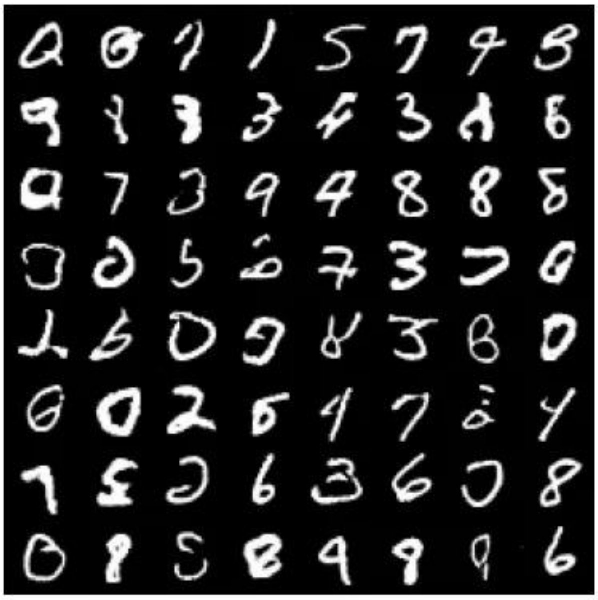

Fig (5): Saturation, Final Refinement, and Output Consistency

During Epochs 61 to 100, shown in figure 5 the Generative Adversarial Network (GAN) reached a point where it had produced the best possible quality of generated images. Further enhancements were negligible as the output remained consistently high-quality, featuring clear and distinct digits. The training process during this phase was dedicated to maintaining this level of excellence. The utilization of Spectral Normalization and Instance Normalization continued to ensure that the training dynamics were stable and balanced, thus preventing any decline in image quality. The learning rate scheduler effectively minimized the learning rate at this stage, allowing only fine adjustments without introducing any instability. The increased batch size and label smoothing also played a crucial role in ensuring the stability of training, preventing any imbalance between the Discriminator and Generator. Throughout this period, the GAN consistently generated high-quality images, and the final epochs were focused on further refining the already excellent output.

5. CONCLUSION

The study looks at improved Generative Adversarial Networks (GANs) for the MNIST dataset. Advanced techniques are used to make training faster and produce better images. The work with low-resolution images which gradually improved their quality using Progressive Growing. This led to faster training and more stable results. And also used Instance Normalization in the Generator instead of Batch Normalization. This made normalization consistent and helped with early learning. Additionally, also applied Spectral Normalization in the Discriminator to ensure stable weight updates, making training more reliable. And also used a learning rate scheduler to gradually reduce the learning rate during training, which allowed for finer adjustments and prevented destabilization. Moreover, increasing the batch size improved gradient stability and speed up training. Learned that using label smoothing prevented the Discriminator from being too confident. By combining these methods, the GAN was able to create high-quality images of handwritten digits in significantly fewer training sessions compared to traditional methods. This underscores the importance of architectural and training strategies in improving GAN performance.

REFERENCES

1. Generative Adversarial Networks Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David WardeFarley, Sherjil Ozair, Aaron Courville, Yoshua Bengio
2. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
3. K. Eves and J. Valasek, “Adaptive control for singularly perturbed systems examples,” Code Ocean, Aug. 2023.
4. A. Dosovitskiy, J. T. Springenberg, and T. Brox. Learning to generate chairs with convolutional neural networks. CoRR, abs/1411.5928, 2014.
5. Alec Radford & Luke Metz indico Research Boston, MA, Soumith Chintala Facebook AI Research New York, NY arXiv:1511.06434v2 [cs.LG] 7 Jan 2016.
6. Z. Zhao, J. Zhao, Y. Sano, O. Levy, H. Takayasu, M. Takayasu, et al., Fake news propagate differently from real news even at early stages of spreading, 2018.
7. Farnia, Farzan; Ozdaglar, Asuman (November 21, 2020). "Do GANs always have Nash equilibria?". International Conference on Machine Learning. PMLR: 3029–3039.Alec Radford & Luke Metz indico Research Boston, MA, Soumith Chintala Facebook AI Research New York, NY arXiv:1511.06434v2 [cs.LG] 7 Jan 2016.
8. B. D. Horne and S. Adali, This just in: fake news packs a lot in title uses simpler repetitive content in text body more similar to satire than real news, 2017.
9. THE MNIST DATABASE of handwritten digits Yann LeCun, Courant Institute, NYU, Corinna Cortes, Google Labs, New York, Christopher J.C. Burges, Microsoft Research, Redmond
10. OkToegang verkry: Nov 03, 2020. Available at http://arxiv.org/abs/1610.07584 .
11. Tex-ViT: A Generalizable, Robust, Texturebased dual-branch cross-attention deepfake detector Deep-

ak Dagar1 , Dinesh Kumar Vishwakarma2,* Biometric Research Laboratory, Department of Information Technology, Delhi Technological University, Bawana Road, Delhi-110042, India

12. L. Deng, D. Yu, and J. Platt, “Scalable stacking and learning for building deep architectures,” in Proc. ICASSP, Mar. 2012., Sebastian Berg4 , Nathaniel J. Smith12, Robert Kern13, Matti Picus4 , Stephan
13. Gulrajani et al., 2017,Chen & Koltun (2017),Odena et al., 2017
14. Instance Normalization: The Missing Ingredient for Fast Stylization Dmitry Ulyanov, Andrea Vedaldi, Victor Lempitsky
15. Martin Arjovsky and Leon Bottou. Towards principled methods for training generative adversarial networks. ´ In ICLR, 2017.
16. Don't Decay the Learning Rate, Increase the Batch Size Samuel L. Smith, Pieter-Jan Kindermans, Chris Ying, Quoc V. Le
17. Delving Deep into Label Smoothing Chang-Bin Zhang† , Peng-Tao Jiang† , Qibin Hou, Yunchao Wei, Qi Han, Zhen Li, and Ming-Ming Cheng arXiv:2011.12562v2 [cs.CV] 22 Jul 2021
18. Empirical Evaluation of Rectified Activations in Convolution Network Bing Xu Wang winsty Tianqi Chen tqchen@ Mu Li muli@c arXiv:1505.00853v2 [cs.LG] 27 Nov 2015
19. Adam: A Method for Stochastic Optimization Diederik P. Kingma, Jimmy Ba
20. Zhiwei Guo, Yang Li, Zhenguo Yang, Xiaoping Li, Lap-Kei Lee, Qing Li, Wenyin Liu, "Cross-Modal Attention Network for Detecting Multimodal Misinformation From Multiple Platforms", IEEE Transactions on Computational Social Systems, vol.11, no.4, pp.4920-4933, 2024.
21. G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor, “Converting Static Image Datasets to Spiking Neuromorphic Datasets Using Saccades,” Frontiers in Neuroscience, vol. 9, pp. 1–15, nov 2015