

A Comparison of Different GANs for Generating Handwritten Digits on MNIST*

Jeffery B. Russell[†]

*Fourth Year Computer Science Student at RIT
CUBRC Research Assistant
RITlug President*

Ryan Missel[‡]

*Fifth Year Computer Science Student at RIT
CASCI Research Assistant*

Kyle Rivenburgh[§]

Fifth Year Computer Science Student at RIT

(Dated: April 11, 2020)

Generative Adversarial Networks have emerged as a powerful and customizable class of machine learning algorithms within the past half a decade. They learn the distribution of a dataset for the purposes of generating realistic synthetic samples. It is an active field of research with massive improvements yearly, addressing fundamental limitations of the class and improving on the quality of generated figures. GANs have been successfully applied to music synthesis, face generation, and text-to-image translation.

Within this work, we will look at a variety of GAN architectures and how they compare qualitatively on the popular MNIST dataset. We will explore how differing architectures affect time of convergence, quality of the resulting images, and complexity in training. The theoretical justifications and shortcomings of each methodology will be explored in detail, such that an intuition can be formed on choosing the right architecture for a problem.

Keywords: Computer Vision, Generative Adversarial Networks, Machine Learning, MNIST

I. BACKGROUND

Neural networks (NN) were first developed by Bernard Widrow and Marcyan Hoff of Stanford in 1959 under the name of MADALINE (Multiple Adaptive Linear Element) [5]. Neural networks were designed with inspiration taken from biological neurons in human brains. Artificial neurons aggregate information from other neurons and fire off a signal depending on the strengths of previous inputs, which is analogous with how human neurons operate. Neural networks falls into the categorization of supervised learning in artificial intelligence (AI). Under supervised learning, the algorithm needs to be fed in labeled data in order to make future classifications/predictions. This is opposed to unsupervised learning which needs no training data – an example of unsupervised learning would be clustering.

GANs were first proposed by Ian J. Goodfellow in his PhD dissertation in 2014 at the Université de Montréal [2]. The proposed architecture is a dual neural network system, in which a generative model learns to generate realistic samples from a distribution in order to compete against a discriminator that classifies fake images. These models are trained in tandem with one another, both

learning from random initialization how to best one another. A successful result of training is when the Nash Equilibrium between the two models is found. This occurs when the generator has learned the distribution of the data well enough to the point that the discriminator is only as good as random chance.

GAN Architecture

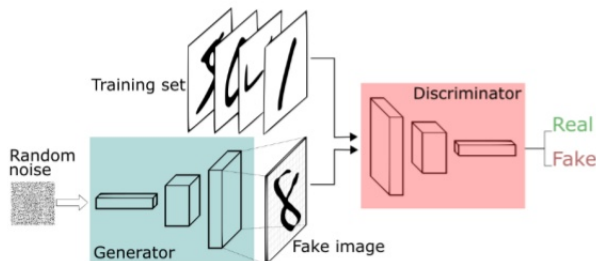


FIG. 1. Architecture of a GAN

* Submitted as a CSCI-431 assignment at RIT

[†] jeffery@jrtechs.net, jxr8142@rit.edu

[‡] rxm7244@rit.edu

[§] ktr5669@rit.edu

Since the advent of GANs in 2014, they have vastly improved and have blown up in the AI research field. State of the art research in GANs is currently focusing on applications in video and voice data.

A. Applications

GANs have been applied to many problems [3]. A sampling of some of the problems are listed below.

- Image Generation
- Music Generation
- Style Transfer
- Video Prediction
- Super Resolution
- Text to Image

B. Deep Convolutional Generative Adversarial Network

Deep Convolutional Generative Adversarial Networks, DCGAN for short, is an architectural modification on the original GAN, in which the generator and discriminator models are reflections of one another. The makeup of each network is a multi-layer, deep convolutional neural network. The idea behind this architecture is that by reflecting the network structure between the two, the computational capacities of each network to learn their respective tasks is equal [4]. In doing this, it should stabilize competitive learning between the two agents and result in a smoother learning, avoiding cases of one network dominance.

C. Wasserstein Generative Adversarial Networks

Wasserstein Generative Adversarial Networks, or WGANs for short, were an improvement on the vanilla GAN proposed by Martin Arjovsky, et al in 2017 [1]. The motivation behind this work is modifying the task of the discriminator in order to stabilize the training between the networks. Instead of having a simple binary classifier that predicts whether an image is real or fake, the discriminator is modified to output the likelihood estimate

of the "realness" or "fakeness" of an image. The theoretical idea is that this continuous estimation incentivizes the generator to minimize the distance between the distribution of its generated images and the real images more than the standard discriminator design. Empirically, this design has shown greater results over the standard GAN architecture in terms of training and architecture stability, as well as being more robust to hyper-parameter configurations.

II. GOALS

This project applies three different GAN architectures to generating handwritten images from the MNIST dataset. We are going to compare: vanilla GANs, DCGANs, and WGANs. Using the results of the three different architectures we wish to judge the performance based on three performance criteria:

- Perceived Quality of Images
- Time required to train
- Training data required

The Modified National Institute of Standards and Technology database (MNIST database) is a dataset comprising of seventy thousand handwritten digits. Sixty thousand of those images are partitioned for training and the remaining ten thousand are left for testing and validation. We are using the MNIST dataset because it is the de facto standard when it comes to machine learning on images.

A. Research Questions

- Which GAN architecture performs best on the MNIST dataset?
- What are the quantitative differences between these architectures in terms of stability of training, and quality of the results?
- How does required training time and convergence rate differ between GAN architectures?

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein GAN. arXiv:stat.ML/1701.07875

[2] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks. arXiv:stat.ML/1406.2661

[3] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng. 2019. Recent Progress on Generative Adversarial

ial Networks (GANs): A Survey. *IEEE Access* 7 (2019), 36322–36333. <https://doi.org/10.1109/ACCESS.2019.2905015>

[4] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv:cs.LG/1511.06434

[5] Bernard Widrow. 1962. Generalization and Information

Storage in Networks of ADALINE Neurons. Self Organiz-

ing Systems. *Yovitz, MC, Jacobi, GT, and Goldstein, GD*
editors (1962), 435-461.