Using DCGAN to generate images of brains

|  |  |  |
| --- | --- | --- |
| Dan Mohler | Jordan Winkler | Prad Ejner |
| damohler@iu.edu | jomawink@iu.edu | prad.ejner@gmail.com |
| Computer and Information Sciences | Computer and Information Sciences | Computer and Information Sciences |
| Indiana University South Bend | Indiana University South Bend | Indiana University South Bend |
| South Bend, IN, USA | South Bend, IN, USA | South Bend, IN, USA |

*Abstract*— *Using a DCGAN (Deep Convolutional Generative Adversarial Network) on Jun Cheng’s brain tumor dataset of brain MRIs we were able to generate fake images of brain MRIs. Then we compare generated images with the real images using Frechet Inception Distance (FID) to measure how well the DCGAN model performed.*

Keywords—GAN, DCGAN, FID, medical, brain

# Introduction

Generative adversarial networks (GANs) are a recent innovation (2014) in machine learning invented by Ian Goodfellow and his colleagues [1]. Given a training set, a neural network will learn to generate new data with the same statistics as the training set. For example, GANs can use a training set of photographs of human faces to create images of human faces. These generated images don't belong to any real person.

GANs consist of two neural networks that contest with each other. The generative network generates plausible candidates while the discriminative network evaluates candidates by distinguishing the generator's fake data from real data. At the beginning of training, the generator will produce data that is obviously fake and the discriminator quickly learns to tell that it is fake data. As training continues, the generator will continue to get closer to producing output that can fool the discriminator. Eventually, if the generator training goes well, the generator will produce data which is difficult to distinguish as fake or real. The discriminator gets worse at distinguishing the differences between real and generated and the generator accuracy decreases.

# Dataset

We used a magnetic resonance imaging (MRI) dataset from Cheng that consists of 3064 images of brain tumors with contrast added [2]. Fradet produced a tool that we modified to download and convert the MATLAB data into 8-bit TIF image files [3]. The images had low contrast because only a small portion of the available range was used in the dataset. Our solution was to scale the images individually to fill the range of unsigned 8-bit integer. From the scaled dataset, we selected images of only the side view. This resulted in 1019 images on which the model trained. The resolution that we used was 128 x 128 pixels.

# Discriminator

The discriminator in a GAN is a classifier. The discriminator tries to distinguish real data from the data created by the generator. During discriminator training, the generator does not train. The weights for the generator remain constant while it produces images for the discriminator to train on.

The data used for training the discriminator comes from two sources: real data and generated data. Real data are considered positive instances during training, and fake data are considered negative instances during training. If the discriminator misclassifies real instance as fake or fake data as real, the discriminator loss penalizes the discriminator. The discriminator will use the discriminator loss to update its weights through backpropagation.

# Generator

The goal of the generator is to fool the discriminator. Using feedback from the discriminator, the generator will learn to produce images that the discriminator will classify as real data.

Typically, with deep learning models, input data is an instance that to classify or make a prediction about. However, with GANs, the desired output that is entirely new data instances. Therefore, the generator will start by taking random noise as its input. The generator will then transform this noise into meaningful output.

Typically, we alter a neural network's weights to reduce the error or loss of its output. However, with GANs, the generator is trained with the loss from the discriminator. This penalizes the generator whenever it produces data that the discriminator classifies as fake.

Backpropagation adjusts each weight by calculating the weight's impact on the output. Backpropagation start with the output from the discriminator and flows back through the discriminator into the generator to obtain gradients. The gradients are used to change the generator weights.

# Implementation

We used a deep convolutional GAN or DCGAN implementation from Jolly [4]. DCGANs were first proposed by Radford and Chintala to overcome the scaling problem of earlier GANs [5]. Unlike a conditional GAN, the model used does not distinguish between data classes. However, the DCGAN has the benefits of using existing techniques to understand the feature maps created by convolutional layers, as shown by Radford and Chintala. This architecture works for our case, as we are only using the side view of brain MRIs. We chose this architecture because of limited training time and ease of implementation.

The architecture of the model was chosen to be held constant and four of the model’s hyperparameters were varied. We evaluated the performance after varying only a single factor over a fixed number of epochs. The number that was chosen was 50,000 epochs due to time constraints and previous indications of performance during testing.

# Measuring Results

In GANs, the generator and the discriminator measure how well they are doing relative to each other. For example, we measure how the generator is fooling the discriminator and how well the discriminator identifies data instances as real or fake. These metrics do not measure image quality or image diversity.

## Fréchet Inception Distance (FID)

The Frechet Inception Distance score, or FID for short, is a metric that calculates the distance between feature vectors calculated for real and generated images using a statistical representation of both. The probability variable for the real image can be represented as a multivariate Normal distribution Xr ~ N(μr,  Σr) and the generated one as Xr ~ N(μr,  Σr). Giving

FID=||*μr*−*μ**g*||2+Tr(Σ*r*+Σg−2(ΣrΣg)1/2)

Tr is the trace of a matrix. And ||.|| is the Euclidean distance between vectors. Lower FID score indicates the two groups are more similar or have more similar statistics. The perfect score would be 0. 0 indicating that the two groups of images are identical. FID scores are used to evaluate the quality of images generated by GANs and lower scores have been shown to correlate well with higher quality images.

## History of FID

FID scores were proposed in 2017 by Martin Heusel, et al. in "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium" [6].

FID score was proposed as an improvement over the existing Inception Score (IS) since the Inception Score did not capture how synthetic images compared to real images. Heusel et al. showed how lower FID scores correlate with better-quality images when systematic distortions were applied such as blur and random noise.

# Results

A selection of eight images from our scaled and filtered dataset is show in Figure 1. Table I shows some of the original values for various hyperparameters that were used by our source. Figure 2 shows a selection of eight images produced by training the model for 50,000 epochs on our scaled and filtered dataset. The images in Figure 2 show significant mode loss and this reflects in the FID score of 193 indicated in Table 1.

Our metric for FID was calculated using Tang’s implementation [7]. It is worth mentioning that FID scores calculated with this method are only useful for direct comparisons as we are not using the same number of images as some models have done to compute FID score. For our evaluation we used 1019 real images and 1019 generated images.

Table II shows our approach to improving the model by varying some of the hyperparameters with a fixed number of epochs. The table shows that, out of the values tested, a dropout of 0.45 resulted in the lowest FID score of 110 (shown in bold). The eight images generated using these hyperparameters are shown in Figure 3. While the images in Figure 3 do have greater diversity than in Figure 2, there also appears to be a defect in the right top corner of a majority of the images.

# Further Explorations

As the number of epochs was held constant and the hyperparameters chosen were only changed individually, it is likely that the model could be further improved by a more exhaustive search. An optimizer could also be used in place of a grid search or a stochastic approach.

Another approach that may warrant further study is to change the model to have two time-update rates, one for the generator and one for the discriminator. This could potentially allow longer training and more learning before detrimental effects such as mode loss occur.

A further exploration of the concept of generating MRIs could be to change the architecture to a conditional GAN. Each type of tumor could be a label and the view and location of the tumor could also be a label. This could be useful for training medical professionals and for developing brain tumor classifiers that are less brittle.

1. Original Values

| Dropout | Batch Normalization Momentum | Adam Learning Rate | Adam Beta 1 | FID Score |
| --- | --- | --- | --- | --- |
| 0.25 | 0.80 | 0.0002 | 0.50 | 192.77 |

1. Modified Values

| Dropout | Batch Normalization Momentum | Adam Learning Rate | Adam Beta 1 | FID Score |
| --- | --- | --- | --- | --- |
| 0.27 | 0.80 | 0.0002 | 0.50 | 205.93 |
| 0.30 | 0.80 | 0.0002 | 0.50 | 144.54 |
| 0.33 | 0.80 | 0.0002 | 0.50 | 164.24 |
| 0.36 | 0.80 | 0.0002 | 0.50 | 161.72 |
| 0.39 | 0.80 | 0.0002 | 0.50 | 161.21 |
| 0.42 | 0.80 | 0.0002 | 0.50 | 117.77 |
| **0.45** | **0.80** | **0.0002** | **0.50** | **110.27** |
| 0.25 | 0.70 | 0.0002 | 0.50 | 187.83 |
| 0.25 | 0.75 | 0.0002 | 0.50 | 173.23 |
| 0.25 | 0.82 | 0.0002 | 0.50 | 156.22 |
| 0.25 | 0.80 | 0.00015 | 0.50 | 189.31 |
| 0.25 | 0.80 | 0.00025 | 0.50 | 201.41 |
| 0.25 | 0.80 | 0.0003 | 0.50 | 222.07 |
| 0.25 | 0.80 | 0.0001 | 0.50 | 184.46 |
| 0.25 | 0.80 | 0.0002 | 0.40 | 186.82 |
| 0.25 | 0.80 | 0.0002 | 0.45 | 176.28 |
| 0.25 | 0.80 | 0.0002 | 0.55 | 190.34 |
| 0.25 | 0.80 | 0.0002 | 0.60 | 217.35 |

IMAGE RESULTS

A picture containing photo, different, posing

Description automatically generatedFig. 1. Side view images from dataset

A picture containing photo, beverage, food

Description automatically generatedFig. 2. Images generated without modifications FID: 193

A picture containing photo, beverage, show

Description automatically generatedFig. 3. Images generated with optimized parameters FID: 110

##### References

1. Goodfellow, Ian, et al. (2014). “Generative Adversarial Networks”, ArXiv Preprint, arXiv:1406.2661
2. Cheng, Jun. (2017) “brain tumor dataset v5”, Figshare dataset. https://figshare.com/articles/brain\_tumor\_dataset/1512427
3. Fradet, Guillaume. (2019). “brain-tumor-mri-dataset”, GitHub repository, https://github.com/guillaumefrd/brain-tumor-mri-dataset
4. Jolly, Mitchell. (2019). “brain-tumor-mri-dataset”, GitHub repository, https://github.com/mitchelljy/DCGAN-Keras
5. Radford, L. Metz, and S. Chintala, (2016). “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks,” ArXiv Preprint, ArXiv:1511.06434
6. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). “Gans trained by a two time-scale update rule converge to a local nash equilibrium”, Advances in Neural Information Processing Systems, (pp. 6626–6637)
7. Tang, Shichang. (2018). “Frechet-Inception-Distance”, GitHub repository, https://github.com/tsc2017/Frechet-Inception-Distance