An Empirical Analysis of GAN-based DATA Augmentation for Classification

Romina Abadi [1]

Abstract

In recent years, Generative Adversarial Networks (GANs) have been applied in various areas in Machine Learning, including data augmentation. This project examines the extent to which GAN-based data augmentation enhances a classifier's performance. The experiments on MNIST data-set show that GAN-generated data can increase a classifier's performance, especially when limited data is available. However, based on this project's results, GAN-generate data can not completely replace the real data, as the classifier's performance decreased being trained only on the GAN-generated data compared to the real data.

1. Introduction

Recent advances in Generative Models have allowed creating realistic-looking synthetic images that are comparable to the real data-sets. The similarity between the synthetic and real images suggests that using artificially created images can be beneficial in training classifiers, either to augment the available data or even to replace the initial data set for security or privacy reasons.

GANs have significantly gained popularity among generative models. They are mainly of interest because of their ability to create high-quality, real-looking images. This project aims to evaluate the extent to which using GAN-generated data for augmenting a data-set can benefit a classifier's performance. As the total information available to the classifier and GAN does not change, it is believed that there must be a limit to which the GAN generated data increases the classifier's accuracy. This project's goal is to examine the effectiveness and limit of GAN-based data augmentation for the classification task.

[1]Department of Electrical Engineering and Computer Science, York University, Toronto, Ontario, Canada. Correspondence to: Romina Abadi <rominaabadi1994@gmail.com>.

Project Report for EECS6327 (Fall 2020), Department of Electrical Engineering and Computer Science, York University.

2. Literature Review

GANs were introduced in 2014 as a framework to generate samples from a data distribution (Goodfellow et al., 2014). They have been proven to be potent models for generating samples from a data distribution. Since 2014, various additions have been applied to GANs for different purposes. The similarity of all GANs is training two networks in an adversarial procedure. One of the usages proposed for the GAN generated data is data-augmentation in settings where the available data is limited or unbalanced, or it is from a slightly different domain. The GAN architectures used for data augmentation to improve classification vary based on the problem's context.

DC-GAN, introduced by (Radford et al., 2016), is a GAN designed for generating images, and can be used to enlarge image data-sets when the available data is balanced but limited. (Frid-Adar et al., 2018) use DC-GAN to create high quality synthetic CT images of liver lesions. They use this approach to enhance the accuracy of a Convolutional Neural Network (CNN) trained to detect three different types of liver lesion. Their results indicate using GAN-based augmentation approach increase both sensitivity and specificity for all three classes.

To ensure the diversity of synthetic images, (Shi et al., 2018) add a diversity enhancement regularization to the Generator and evaluate the effect of the generated data on classification when limited balanced data is available. Their method is shown to be effective when tested on three different MNIST, SVHN, and CIFAR-10 data-sets.

BA-GAN is proposed by (Mariani et al., 2018) for balancing imbalanced data-sets by adding synthetic images to the minority classes. They create the images by using all the available data for training a GAN. An Auto-encoder is used along with the GAN to initialize the Generator and to determine the distribution of the different classes of the data. Moreover, the Discriminator, in addition to determining if the data is fake, is trained to learn the classes of the data set. This approach is used by (Chatziagapi et al., 2019) to improve Speech Emotion Recognition.

Variations of CycleGAN (Zhu et al., 2017) and Pix2Pix translation (Isola et al., 2018) are used to adapt a data set to a slightly different context. (Lee et al., 2020) use this

approach to convert daytime images to nighttime images in order to improve the object detection of the blind-spot camera at night. Same approach is employed by (Zhu et al., 2018) to alter images of faces to represent different emotions, in order to enhance emotion classification.

In addition to using GAN for data augmentation, the effectiveness of the GAN generated images for training a classifier is proposed as a new measure for validating the quality of GAN generated data (Shmelkov et al., 2018).

The main focus of this project is examining the limitations of GAN-based data augmentation, and the effect of using excessive amount of GAN generated data on classification.

3. Methodology

To evaluate the effect of data generated with GAN on classifier's performance in limited data situations, a GAN is trained on different small proportions of the MNIST data set. Then, the GAN generated images are used to train a classifier. In the remaining parts of this section, the models and training process used for the GAN is provided along with the architecture of the classifier used in the experiments.

In all of the experiments DC-GAN architecture (Radford et al., 2016) is used. The reason for using DC-GAN is the smoother training and faster convergence of the networks for generating images, compared to the initial GAN architecture. DC-GANs are trained in an unsupervised manner. To create labelled images, GAN is trained on different classes separately. For evaluating GAN-based augmentation, a fully connected classifier's accuracy is calculated and compared in different scenarios. To evaluate the accuracy, the whole MNIST test set is used in all the experiments.

In the remaining parts of this section, more detail of the models is provided.

3.1. Classifier

The classifier used for all the experiments is a fully connected Deep Neural Network with one hidden layer of size 300 and ReLU activation function, with batch normalization and a drop-out layer with rate 0.1. The accuracy of the model trained on all the data is 0.97. For optimization, Adam optimizer with learning rate of 0.001 is used.

3.2. DC-GAN

In this section the details of the GAN implementation is discussed.

3.2.1. Training

For training the GAN, two different deep networks, Generator and Discriminator compete against each other. The overall GAN objective can be formulated with a "value" function (Goodfellow et al., 2014):

$$\min_D \max_G V(D,G) = \mathop{\mathbb{E}}_{x\sim p_{data}}[\log(D(x))] + \mathop{\mathbb{E}}_{z\sim p_z}[\log(1-D(G(z)))] \quad (1)$$

Where B is the batch size, and D(.) is the probability of input data being fake calculated by the Discriminator. In the above equation z is driven from a normal distribution (referred to as the latent space), and x is the data driven from the real data distribution. Note that the above formulation is a bit different from the formulation presented in (Goodfellow et al., 2014), because in this project's implementation, the objective functions of the Generator and Discriminator are minimized as opposed to the paper.

The objective of the discriminator is to maximise the accuracy of classifying real and synthetic images, the discriminator's loss function can be formulated as:

$$\frac{1}{2B}\Big(\sum_{i=1}^{B}\log D(x_i) + \sum_{i=1}^{B}(1-\log D(G(z_i)))\Big) \quad (2)$$

On the other hand, the Generator tries to minimize the probability the Discriminator assigns to the synthetic data being fake. The Generator's loss function can be written as:

$$L_G = \sum_{z=i}^{B}\log D(G(z)) \quad (3)$$

Each training step consists of three stages:

- Fake images are created from random noise using the Generator (the number of fake images created in each step equals to the batch size).
- The Discriminator is trained on the same number of real and fake images. The Discriminator's loss (equation 2) is calculated using binary cross entropy loss function. The true labels for real and fake data are set to be one and zero respectively.
- The Generator is trained on the Discriminator's predictions for the fake images. The Generator's loss (equation 3) is calculated using binary cross entropy loss function for the Discriminator's output on the fake images, using 0 as true labels.

For both Discriminator and Generator Adam optimizer is used with learning rate of 0.001. Each training step is performed using a batch size of 64. Section 3.2.2 and 3.2.3 provide details of the Discriminator and Generator models used in the DC-GAN. In both models a ratio α is used for fine-tuning the number of filters. More details of fine tuning is provided in section 4.1.

3.2.2. Discriminator Architecture

The Discriminator consists of two Convolutional layers with 32α and 64α filters respectively. Each layer uses Leaky-ReLU with negative slope 0.3 as activation function, followed by a Drop-out layer with rate 0.3. The final layer is a Fully Connected layer with output size one, using a Sigmoid activation function.

3.2.3. Generator Architecture

The Generator creates images using a 100 dimensional input noise drawn from a normal distribution. The first layer is a Fully Connected layer with output size of $7 \times 7 \times 128\alpha$. The output is then reshaped to $[7, 7, 128\alpha]$ and connected to three Deconvolution layers with 128α, 64α, and 1 filters. The last two layers use stride two, creating the output shape of $[28, 28, 1]$. The activation function for the first two layers is leaky ReLU with negative slope 0.3, and the last layer uses Tanh activation that creates the outputs in -1 and 1 range, as suggested in (Goodfellow et al., 2014).

4. Experiments and results on the MNIST data-set

This section provides details of three experiments on the MNIST data-set and the observed results.

The experiments are performed by choosing different amounts of data from the MNIST data-set as the initial available sample set. The number of samples used in the experiments are 500, 1000, 2000, and 4000 to simulate different scenarios with limited available data. Additionally, evaluations are done on all the available training data (60000 images) as well, to examine the effect of the GAN generated data when all the MNIST data set is available. GAN generated images for different sample sizes can be found in Figure 1. All the samples are chosen randomly.

In all the settings, GANs are trained only once for 2000 epochs. For all the reported classifier's accuracy, the accuracy is observed three times and the average of the three observations is provided as the result. The classifier is trained for 15 epochs in all the settings. Also, in all the experiments, the training accuracy has reached 0.98 or above. In order to prevent over-fitting, the reported accuracy is the maximum observed accuracy over all epochs.

4.1. Fine Tuning the GAN

For fine tuning, three GAN models with three different α rates (1, 2, and 4) are trained on 500 MNIST images. Then, the classifier is trained on 500 images generated by GAN. The best α rate is chosen based on the classifier's performance when trained only on the synthetic data. The optimal α found in this experiment is 2. This value is used for the rest of the experiments. To get the best results, it is best to find the optimal α for different initial training sizes. However, in this project we used the same α for all different training sizes due to lack of time for evaluating different scenarios. The results are summarized in Table 1.

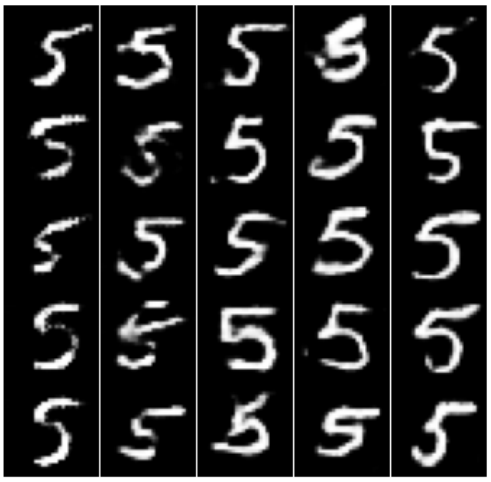

Figure 1. DC-GAN generated images with different sample sizes. Columns from left to write show images produced by training the GAN on 500, $1k$, $2k$, $4k$, and $60k$ samples respectively.

Table 1. The classifier's accuracy trained with 500 GAN generated images using different $alpha$ values.

α	1	2	4
Accuracy	0.6489	**0.6801**	0.6753

4.2. Experiments

4.2.1. Experiment 1: Comparing GAN generated Data with Real Data

This experiment is to evaluate the classifier's performance when trained on synthetic data compared to the real data. Five different sample sizes are chosen. The GAN is trained on the same samples that are given to the classifier when it is trained on the real data. The results are summarized in Table 2 and Figure 2.

As shown in Figure 2, the classifier's accuracy trained on fake data becomes closer to its accuracy trained on real data as the sample set size increases. However, the accuracy for real data is always higher than the fake data. This suggests,

Table 2. The classifier's accuracy trained on synthetic (Fake) and real (Real) data for different sample sizes (Size).

SIZE	500	1K	2K	4K	60K
REAL	0.8092	0.8700	0.9139	0.9421	0.9723
FAKE	0.6801	0.7623	0.8524	0.9243	0.9512

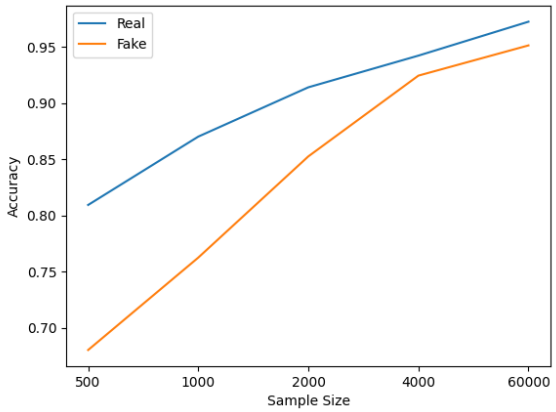

Figure 2. The classifier's accuracy trained on synthetic and real data for different sample sizes

in our setting, the GAN generated data cannot replace the real data.

4.2.2. EXPERIMENT 2: TRAINING THE CLASSIFIER ONLY ON GAN GENERATED DATA

The second experiment examines whether increasing the amount of GAN generated data affects the performance of the classifier or not. The fact that the classifier's accuracy improves when it is trained on more real data, shows it is capable of achieving higher accuracy if trained with more real images. Assuming GAN has been able to learn the data distribution, providing more GAN generated data must give the classifier more information about the distribution and increase its accuracy. However, the results, summarized in Table 3, show that increasing the amount of data generated by GAN does not affect the accuracy of the classifier. This might be due to mode collapse, which causes the Generator to only generate images from specific parts of the distribution and reduces the diversity of the generated data. As a result, the Generator does not produce various images, which makes increasing the GAN generated data ineffective.

Table 3. The classifier's accuracy trained on GAN generated data only. R is the ratio of the GAN generated data provided to the classifier to the size of the sample (Size) used to train the GAN.

SIZE	500	1K	2K	4K	60K
R=1	0.6801	0.7623	0.8524	0.9243	0.9512
R=2	0.6801	0.7622	0.8523	0.9241	0.9516
R=4	0.6802	0.7623	0.8525	0.9243	0.9511

Table 4. The classifier's accuracy trained with real and GAN generated data at the same time. R is the ratio of the GAN generated data to the real data. "Size" refers to the initial sample size.

SIZE	500	1K	2K	4K	60K
R=0.00	0.8092	0.8700	0.9139	0.9421	0.9723
R=0.25	0.8495	0.8883	0.9175	0.9439	**0.9834**
R=0.50	0.8566	0.8958	0.9248	**0.9495**	0.9768
R=0.75	0.8583	0.8942	0.9287	0.9407	0.9732
R=1.00	0.8662	**0.9061**	**0.9303**	0.9386	0.9727
R=2.00	**0.8680**	0.8943	0.9175	0.9305	0.9716
R=4.00	0.8623	0.8635	0.9140	0.9231	0.9707

4.2.3. EXPERIMENT 3: USING GAN GENERATED DATA FOR DATA AUGMENTATION

In the final experiment, GAN generated data is added to the initial sample set as a data augmentation technique. For each sample size, different amounts of GAN generated data are added to the real data set (sample set) and provided to the classifier for training. The ratio of the number of GAN generated images to the initial sample images is referred to as "R". The results are summarized in Table 4 and Figure 3.

It is observed that adding GAN generated data improves the classifier's performance for all different sample sets; this shows using GAN effectively enhances the classifier's performance. Moreover, it is noticed that the optimal ratio decreases for the larger sample sets. The maximum increase in the accuracy is 0.0588 for the sample size 500.

5. Using GAN generated Data for CIFAR-10 data Augmentation

As an additional evaluation, a similar GAN used in section 4 was adjusted and trained using the whole CIFAR-10 data-set. In this experiment each GAN is trained for 3000 epochs. The classifier used for classification is a two layered CNN. The classifier's accuracy trained on all the CIFAR-10 train images is 0.74. This result is not comparable to the state-of-the art results (> 0.9), but it is suitable to evaluate the effect of data augmentation.

Despite the generated images not being visually satisfactory

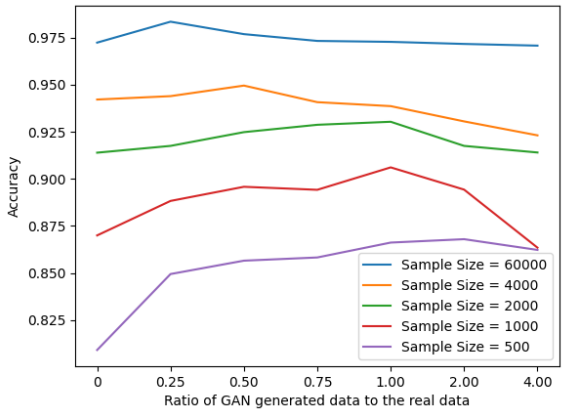

Figure 3. The classifier's accuracy trained with real and GAN generated data at the same time plotted against the ratio of GAN generated data to the real data for different sample sizes.

Table 5. The classifier's accuracy trained on synthetic (fake) and real data. 50k CIFAR-10 images are used as the initial sample set.

RATIO	ACCURACY
R=0.00	0.7411
R=0.01	**0.7492**
R=0.025	0.7484
R=0.05	0.7261
R=0.075	0.7204
R=0.10	0.7063

(as depicted in Figure 4 and Figure 5), adding them to the CIFAR-10 data-set by $R = 0.01$ ratio increased classifier's performance by 0.8% on the CIFAR-10 test images. The results are summarized in Table 5.

As Figure 6 shows, increasing the ratio causes a steep decline in the accuracy; this is probably due to the high amount of noise in the generated images.

6. Conclusions

Experiments performed in this project showed GAN generated data can be helpful for augmenting a data-set. Adding the GAN generated data was most effective when the initially available data set was significantly small. However, observations indicated that there is a limit to which the GAN generated data increases the accuracy of the classifier, and adding more than the limit might decrease the classifier's performance. This could be due to noise in the GAN generated data, or due to the fact that the GAN has not been able to provide diverse data.

This projects' results indicated that the classifier trained only on the GAN generated data could not achieve the accuracy of the classifier trained on the real data. Adding more GAN generated data did not change the classifier's accuracy. This suggests replacing the initially available data with GAN generated data might cause a decline in the classification accuracy. However, different GAN architectures should be used and evaluated to confirm this observation.

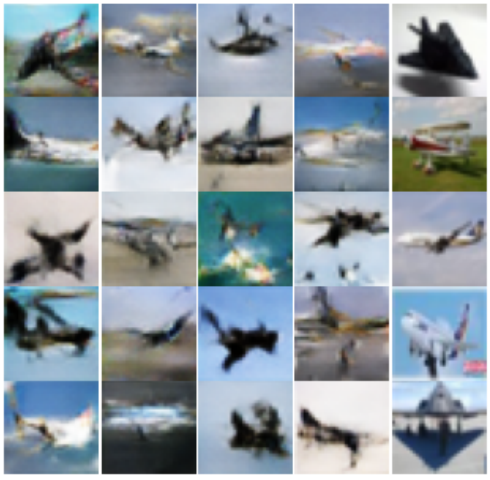

Figure 4. DC-GAN generated CIFAR-10 images for class "airplane". The images in the right-most column are from the data set. All the images are chosen randomly.

Finally, it was shown that the accuracy of the CIFAR-10 classification increased by adding GAN generated data to the initial data-set, even though the GAN generated images did not have high quality.

7. Discussion and Future Work

This projects findings indicate there is an optimal ratio for adding GAN-generated data, and to ensure the effectiveness of this approach, the ratio must be calculated in different problems.

For evaluating GAN-based data augmentation, DC-GAN architecture is used. The models are fine-tuned based on the desirable results, however, due to time limits, fine tuning has been done for limited model features and in limited scenarios. To get more reliable results the models should be fine tuned thoroughly for each task separately, with the objective to increase the classifier's performance trained on the GAN generated data and tested on the real data.

Figure 5. DC-GAN generated CIFAR-10 images for class "horse". The images in the right-most column are from the data set. All the images are chosen randomly.

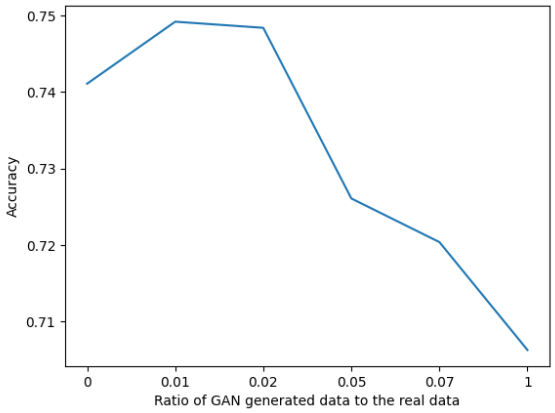

Figure 6. The classifier's accuracy trained with real and GAN generated CIFAR-10 image plotted against the ratio of GAN generated data to the real data for sample size 50k.

Additionally, to have a better understanding of the effect of GAN-based data augmentation, the experiments performed on MNIST data-set should be performed on more complex data sets such as CIFAR-10.

To the extent of the author's knowledge, no comparison has been performed between different GAN architectures for data augmentation. Many different adjustments to the initial GAN implementation have been proposed to create high quality images. It is of interest to investigate how using different GANs for data augmentation effects the final accuracy.

Finally, future work must be done to examine the possibility of generating samples that achieve the same accuracy as real samples. If such samples are feasible, it is interesting to examine if similar to this project's observations, adding a large ratio of such images to the initial training set decreases the classifier's performance.

8. Code

All the codes are implemented using Tensorflow.Keras API. Google Colab GPU is used for training the GANs and the Classifiers.

References

Chatziagapi, A., Paraskevopoulos, G., Sgouropoulos, D., Pantazopoulos, G., Nikandrou, M., Giannakopoulos, T., Katsamanis, A., Potamianos, A., and Narayanan, S. Data augmentation using gans for speech emotion recognition. pp. 171–175, 09 2019. doi: 10.21437/Interspeech.2019-2561.

Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., and Greenspan, H. Synthetic data augmentation using gan for improved liver lesion classification, 2018.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems*, volume 27, pp. 2672–2680. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. Image-to-image translation with conditional adversarial networks, 2018.

Lee, H., Ra, M., and Kim, W. Nighttime data augmentation using gan for improving blind-spot detection. *IEEE Access*, 8:48049–48059, 2020. doi: 10.1109/ACCESS.2020.2979239.

Mariani, G., Scheidegger, F., Istrate, R., Bekas, C., and Malossi, A. C. I. BAGAN: data augmentation with

balancing GAN. *CoRR*, abs/1803.09655, 2018. URL http://arxiv.org/abs/1803.09655.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016.

Shi, H., Wang, L., Ding, G., Yang, F., and Li, X. Data augmentation with improved generative adversarial networks. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 73–78, 2018. doi: 10.1109/ICPR.2018.8545894.

Shmelkov, K., Schmid, C., and Alahari, K. How good is my gan? In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y. (eds.), *Computer Vision – ECCV 2018*, pp. 218–234, Cham, 2018. Springer International Publishing.

Zhu, J., Park, T., Isola, P., and Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2242–2251, 2017. doi: 10.1109/ICCV.2017.244.

Zhu, X., Liu, Y., Li, J., Wan, T., and Qin, Z. Emotion classification with data augmentation using generative adversarial networks. In Phung, D., Tseng, V. S., Webb, G. I., Ho, B., Ganji, M., and Rashidi, L. (eds.), *Advances in Knowledge Discovery and Data Mining*, pp. 349–360, Cham, 2018. Springer International Publishing.