Winter 2019 UCLA CS260 Machine Learning Algorithm Final Project Report: Data Augmentation by using GAN

Jiachen Zhong *julightzhong10@cs.ucla.edu*
Zixiang Liu *zliu46@g.ucla.edu*

March 19, 2019

1 Introduction

Data augmentation usually means to generate more meaningful data points based on the current dataset in order to improve the training performance of machine learning model. As a kind of generative machine learning method, Generative Adversarial Network (GAN) (Goodfellow et al., 2014) is able to learn the mapping from a given distribution to the interested data space and produce fake but good quality data. Therefore, intuitively, GAN can be used for producing more fake data to perform data augmentation. In this project, we did the survey, implementation, and comparison on several methods to utilize GAN for data augmentation. We also analyzed the result and perform some improvement based on what we observed.

2 Related Works

2.1 Generative Adversarial Network

GAN was firstly introduced by Goodfellow et al. (2014) , which can be used to learn the distribution of a dataset and produce fake data through a two players zero-sum game. Mathematically, it can be described as the optimization problem on the following objective function:

$$\min_G \max_D V(D,G) = L_1 + L_2, \tag{1}$$

where G and D are the two players, Generator and Discriminator. Also, L_1 and L_2 are defined as:

$$\begin{aligned} L_1 &= \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_{data}}[\log D(\boldsymbol{x})] \\ L_2 &= \mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{x})))]. \end{aligned} \tag{2}$$

The equations can be interpreted as: Generator learns a mapping from an arbitrarily given space to the interested data distribution, while discriminator analyzes them, try to distinguish between true data and fake data from generator. The two sides in this game are Multilayer Perceptron(MLP) in the original paper but can be replaced in Convolutional version for image data (Radford et al., 2015).

2.2 Conditional GAN

Since 2014, there have been many modified versions of GAN, one of those types is adding condition function to GAN (Mirza and Osindero, 2014) to form a Conditional Generative Adversarial Network (CGAN). It allows GAN to produce data from the specific given class rather than randomly. Generating data based on given labels means CGAN can generate more information than traditional GAN, making CGAN more suitable to do data augmentation. Mathematically, for CGAN, the criterion is slightly different from GAN. With condition term $\boldsymbol{y}$ representing the class label added to the equation,

the two parts now become:

$$L_1 = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_{data}}[\log D(\boldsymbol{x}|\boldsymbol{y})]$$
$$L_2 = \mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{x}|\boldsymbol{y})))]. \quad (3)$$

2.3 ACGAN

More recent research, called ACGAN (Odena et al., 2017), adds an auxiliary classifier to CGAN's discriminator network to increase the model to produce data with better quality and more accurate label than CGAN. Mathematically, it can be seen as the add a classification part on the objective function:

$$\max_D V(D,G) = L_D + L_C$$
$$\max_G V(D,G) = L_C - L_D \quad (4)$$

where L_D and L_G are:

$$L_D = \mathbb{E}[\log \Pr(X = real|X_{real})]+ \mathbb{E}[\log \Pr(X = fake|X_{fake})]$$
$$L_C = \mathbb{E}[\log \Pr(Y = y|X_{real})]+ \mathbb{E}[\log \Pr(Y = y|X_{fake})]. \quad (5)$$

We will later show the improvement of using ACGAN over CGAN.

3 Experiments

3.1 GAN Implementation and Training

We implemented the GAN using PyTorch and trained it on GPU. In order to reduce the computational cost, both Generator and Discriminator of GAN in this project are MLP with 3 layers. The hidden units of layers in both networks are 512 to 1024 to 2048 from first to the last layer respectively. ReLU and Batch Normalization (Ioffe and Szegedy, 2015) were applied except for the last layer. In real training implementation, we trained G for one step with D's parameters fixed first, and then train D for one steps with G's parameters set to not trainable, iteratively using Adam optimizer (Kingma and Ba, 2014) with *batch size*=128, *learning rate*= 0.00005, β_1=0.5, and β_2=0.999.

3.2 Data Augmentation Method

A Data Augmentation GAN proposed by Antoniou et al. (2017) is the first attempt to use GAN to augment classification process with limited training set. To perform supervised learning with those augmented data, effective labeling methods are needed. Overall, there are three major types of labeling techniques in GAN augmentation methods.

Pseudo Label, proposed by Lee (2013), assigns a pseudo label to each image generated by GAN that has the max probability by classifier in each iteration. And feed it as a training sample in the next iteration. This means the image may have a different label in each iteration. We adopted a naive way of implementing it. Since CGAN can generate fake images conditioned on class label, we assign this label to fake image and does not change it during training of classifier. Therefore from now on, pseudo label refers to the method of simply combining training set and the GAN images with pseudo label to form a larger training set, and then training the classifier on this larger set.

All-in-one labels all GAN generated images with a new class label different from any existing labels in training set (Odena, 2016). And appends images and labels to training set to form the larger set to do classification.

LSRO, which expends to Label Smoothing Regularization for Outliers, combines the feature from both Pseudo Label and All-in-one methods. Introduced by Zheng et al. (2017), it labels each GAN output with a uniform distribution over all class labels so that it is less likely to over-fit. Let $p(y)$ be the predicted probability for the actual class of real image, we use the same LSRO version of cross entropy loss, l_{LSRO}, as proposed in the original paper:

$$l_{LSRO} = -(1 - Z)log(p(y)) - \frac{Z}{K}\sum_{k=1}^{K} log(p(k)), \quad (6)$$

where $Z = 0$ for all real images, and $Z = 1$ for all fake images. It can be interpreted as two parts. For

Table 1: Accuracy after Data Augmentation using CGAN.

Labeling	Fashion	Improvement	CIFAR-10	Improvement	CIFAR-100	Improvement
No GAN image	0.8562	N/A	0.4881	N/A	0.2663	N/A
Pseudo Label	0.8838	+0.0276	**0.5147**	+0.0266	0.2648	-0.0015
LSRO	**0.8954**	+0.0392	0.4642	-0.0239	**0.2777**	+0.0114
All-in-one	0.8846	+0.0284	0.4811	-0.007	0.2417	-0.0246

real images, it is the same as original cross entropy loss. For fake images, it is the average of log predicted probability.

3.3 Datasets

We use three common image datasets to implement our method.

Fashion MNIST is a 10 classes data set by Xiao et al. (2017) with 60,000 training images and 10,000 testing images. Images are 28×28 grayscale pictures, which are the same size and data format as the original digit MNIST data set. The distribution of samples in each class is uniform, meaning in training set each class has 6,000 samples. Fashion MNIST serves as a new benchmark data set for image classification algorithms. In our experiment, GAN generates the best output on Fashion MNIST because of its simplicity.

CIFAR-10 provided by Krizhevsky and Hinton (2009) has 60,000 32×32×3 images, where 50,000 are training set and 10,000 are testing set. The images are colorful, recorded in RGB fashion, and each belongs to one of 10 classes. Similar to Fashion MNIST, the distribution of class is also uniform. Each class has exactly 5,000 samples in training set. Because of its larger size and more channels, it is much more complex than Fashion MNIST. Our GAN result is poor on CIFAR than on Fashion MNIST.

CIFAR-100 also provided by Krizhevsky and Hinton (2009) has the same training and testing structure as CIFAR-10. The image size and data arrangement are also same as CIFAR-10. The only difference is the That CIFAR-100 has 100 classes. More classes and same among of total data means less images per class, only 500 samples per class. Thus this data set is the hardest both for GAN training and classification among the three data sets.

3.4 MLP classifier

We implemented a simple 3 hidden layers Multilayer Perceptron(MLP) classifier. The network first flattens the image into 1-D array, and then the size of each layer is respectively 512 to 1024 to 2048 to the number of classes. All layers use ReLU as activation function and Batch Normalization except the last layer, which uses softmax as activation. Between each fully connectted layer, a dropout rate of 0.2 is added to avoid over-fitting. We used Adam optimizer with *learning rate*=0.001, β_1=0.9, and β_2=0.999. We firstly finish the training of GAN model and randomly sample 50% extra augmented data generated from GAN under uniform given conditions. Then we combine augmented data and original data to train the classifier.

4 Result and Analyses

4.1 Augmentation Result

The results are given in Table 1. All result are obtained from the best result among multiple running attempts because the training and generation of GAN requires sampling random noise which leads the generated result in every running different. For Fashion MNIST and CIFAR-100, LSRO produces the maximum increase in accuracy. For CIFAR-10, Pseudo label has the best output. It is possible that GAN generated the best images for CIFAR-10, so Pseudo label could improve classification result a lot. Not all methods can bring improvement by using extra GAN data especially in CIFAR-10 and

Table 2: Accuracy after Data Augmentation using ACGAN.

Labeling	Fashion	Improvement	CIFAR-10	Improvement	CIFAR-100	Improvement
No GAN image	0.8562	N/A	0.4881	N/A	0.2663	N/A
ACGAN+Pseudo Label	**0.9061**	+0.0499	**0.5797**	+0.0916	0.2901	+0.0238
ACGAN+LSRO	0.9018	+0.0456	0.5796	+0.0915	0.288	+0.0217
ACGAN+All-in-One	0.9008	+0.0446	0.5795	+0.0914	**0.2905**	+0.0242
Best using CGAN	0.8954	+0.0392	0.5147	+0.0266	0.2777	+0.0114
Best using ACGAN	0.9061	+0.0499	0.5797	+0.0916	0.2905	+0.0242

CIFAR-100, we think that is because our GAN is too simple to produce high quality data according to some sample outputs which we randomly selected some in Appendix A and we will provide some analyses in the following section 4.2.

4.2 Intuitive Analyses for results of CGAN

Since training GAN is a quite hard procedure, we did not pay much effort to the quality of images produced by GAN. And, since we used simple MLP in the GAN instead of using DCGAN, the ability of GAN to produce good quality on complex images is limited. According to the properties of 3 data sets and some generated sample, which we randomly selected some in Appendix A, from GAN, we believe the quality order from good to bad of the data generated by GAN is Fashion, CIFAR-10, and CIFAR-100. Moreover, by checking some generated samples, the CGAN can not properly produce the right image under given label. Therefore, we believe the result in Table 1 is under a different quality and poor-labeled data augmentation situations. That explains why the three labeling methods performed inconsistently on three data sets. We also made the following assumptions on three labeling methods:

Pseudo Label should perform best when Conditional GAN outputs are sufficiently related to the given conditions(labels) if the overall quality of augmented data are good. Then classifier would learn true feature from generated image and its label.

LSRO is used when the labels of augmented data are not accurate but uniform. In this case, since the distribution of labels are uniform, LSRO can better describe the label of fake images as a combination of all classes. Also, since LSRO still assumes that the augmented data belong to existing class, the quality of augmented images should not be too bad.

All-in-one ought to be used when label for augmented data are neither accurate nor uniform. The results cannot represent existing classes well and should be labeled as counter example for existing classes. And since all augmented data are dealt as outliers, this method may be the best option when the generated data are bad on quality.

To verify the the assumptions, we trained a ACGAN in order to improve the images produced with more accurate label and the detail is in the section 4.4.

4.3 Improvement

Based on the above assumptions, we implemented and trained ACGAN to generate better labeled data. The results are shown in Table 2. We still produced extra 50% augmented data under uniform given conditions. The accuracy by using data from ACGAN is consistently better than the corresponding accuracy obtained using CGAN. Since the original ACGAN paper (Odena et al., 2017) point out that adding auxiliary classifier may improve the generated image quality, and that is the possible reason why the performance of classifier got boost on all labeling methods under ACGAN's data. At the same time, Pseudo Label with ACGAN fake images performed better on Pseudo label than LSRO, and this observation supports the first and second assumption we made in section 4.2 since ACGAN improves the accuracy of labeling. For CIFAR-100, the All-in-one method performs best. The reason may be that

the CIFAR-100 is too complicated for simple MLP ACGAN to provide real meaningful augmentation data, and most augmented data can be dealt as outliers.

5 Conclusion and Future Works

In this project, we surveyed, implemented, compared, and analyzed three GAN data augmentation labeling methods on three simple datasets under simple MLP GAN structure. Based on the result, we also proposed some reasonable assumptions and made an improvement based on the assumptions. For future works, we may perform perform fine tune on hyper parameters since did not in this project. In addition, the overall project is controlled within very simple cases due to the time limitation, but we believe those methods could also be verified on much more complicated situations. Therefore, we can use more complicated GAN structure(e.g. DCGAN) and test on larger scale data sets as the future works after this course. Another possible future work is to improve the ability of GAN to produce data with diversity. Since we want to use GAN to do data augmentation, we need to make sure GAN is able to generate data which is correct corresponding to label but different from the original data. To the best expectation, we may work on finding a good way to manipulate GAN to produce adversarial examples which are the worst cases to fail a machine learning model. If we can make GAN to learn how to effectively generate adversarial examples, it will be very helpful to improve the robustness of machine learning models.

References

Antoniou, A., Storkey, A., and Edwards, H. (2017). Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical report, Citeseer.

Lee, D.-H. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, page 2.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Odena, A. (2016). Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*.

Odena, A., Olah, C., and Shlens, J. (2017). Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org.

Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

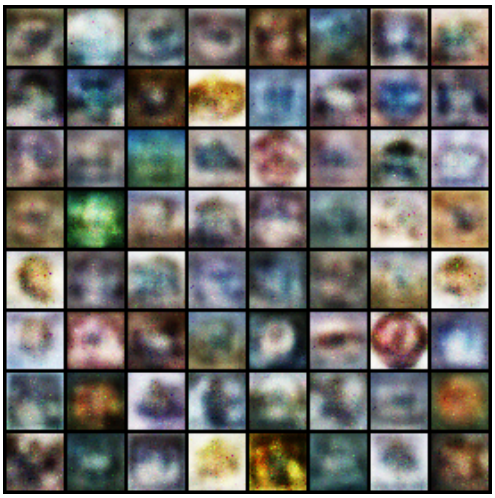
Zheng, Z., Zheng, L., and Yang, Y. (2017). Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3754–3762.

A Appendix: Sample GAN outputs

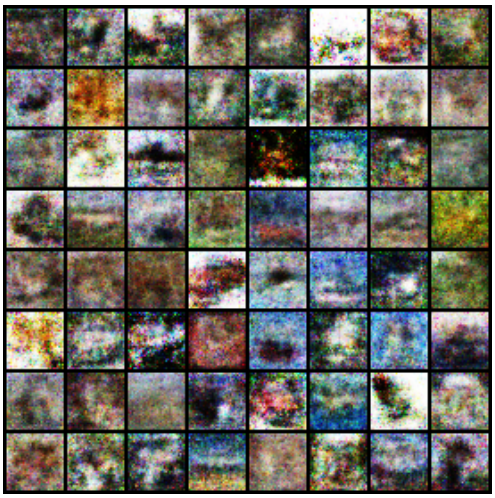
(a) CGAN

(b) ACGAN

Figure 1: Sample generated Fashion images from CGAN v.s. ACGAN

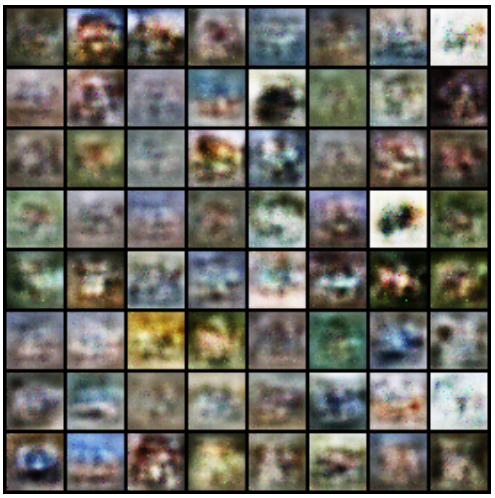


(a) CGAN

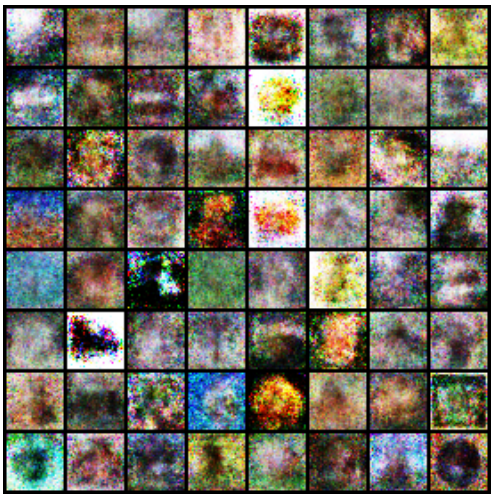


(b) ACGAN

Figure 2: Sample generated CIFAR-10 images from CGAN v.s. ACGAN



(a) CGAN



(b) ACGAN

Figure 3: Sample generated CIFAR-100 images from CGAN v.s. ACGAN