DS-GA 1008: Using Generative Models for Image Augmentation

Bharathi Priyaa T
Department of Computer Science
Courant Institute of Mathematical Sciences
New York University
bt978@nyu.edu

Shivam Verma
Department of Mathematics
Courant Institute of Mathematical Sciences
New York University
sv1239@nyu.edu

1 Introduction

We focus on using deep generative models, specifically Generative Adversarial Networks (GAN) to perform data augmentation for computer vision tasks. Data augmentation is a technique that is used for pre-processing small datasets to improve model performance.The advent of deep generative models in recent years have opened a number of research avenues and machine learning applications. Thus, we investigate the use of generative models to augment small training sets to be used in machine learning problems like image classification.

2 Why is Data augmentation important

Deep learning methods have received widespread attention over the past few years due to state-of-the-art results on image classification tasks like ImageNet and CIFAR. However, these methods require sizable datasets to achieve good accuracy and avoid overfitting. One way of artificially expanding the dataset is using data augmentation, where simple transforms such as translation, rotation, color jittering, as well as PCA are applied to produce new images, which are then merged with the training dataset as the model is trained. Augmentation can especially help with skewed datasets, which are quite common in real-world datasets which have to be labelled. Data augmentation is widely used in most real-world computer vision models.

3 What are Generative Models

3.1 Deep Generative Models

Generative models, which have recently gotten extensive attention as well as adoption for computer vision problems, are built on the principles of Probabilistic Inference and Deep Learning. The idea behind generative models, as opposed to discriminative models, is to learn the distribution from which the training data is assumed to be sampled from, using KL divergence or cross-entropy as the training loss. These methods are especially useful in semi-supervised approaches, where there is a non-trivial amount of unlabelled data. Some of the most successful generative models are discussed below:

- Variational Auto-Encoders [10] - Tries to learn a low dimensional representation of an image/video from high dimensional data using Variational inference. The graphical model in such a model is a Bayesian network with latent priors (Z) which explain the observed data X. VAEs have been previously used for language modelling problems in text.
- Generative Adversarial Networks [9] - Generative model which uses two networks, a Generator and a Discriminator - which are trained against each other in an adversarial manner.

The Generator generates images in an unsupervised manner and the discriminator tries to learn if an image is real or fake in a semi supervised manner. These two networks are pitted against each other, until the discriminator can no longer distinguish between generated images. GANs have previously been used in Text-to-Image synthesis problems.

- PixelRNN [11] - Models the joint distribution of pixels in an image based on conditional probabilities over neighbouring pixels. PixelRNNs were one of the earliest generative model to model images. These are one of a class of autoregressive models.

4 Related Work

Data augmentation is a widely used technique in training deep learning models for computer vision. Work from Meng et al. [7], Vaibhav. et al. [8] discusses the affect of augmentation on accuracy in much detail. Autoencoders have also been used to sample the learned distributed of images. Machine learning practitioners have used Data Augmentation to boost training accuracies. These are observed to produce highly correlated outputs and are thus a weak form of data augmentation [1]. Improving it by a significant factor (at a reasonable computational cost) would result in broad adoption and significant gains across the entire field.

5 Task

5.1 Traditional Approaches to Data Augmentation

Traditional approaches include data transformations such as rotation, scaling, cropping, sheering, image flipping and scalar translations, as well as color jittering. Examples of augmented images can be seen in fig. 1.

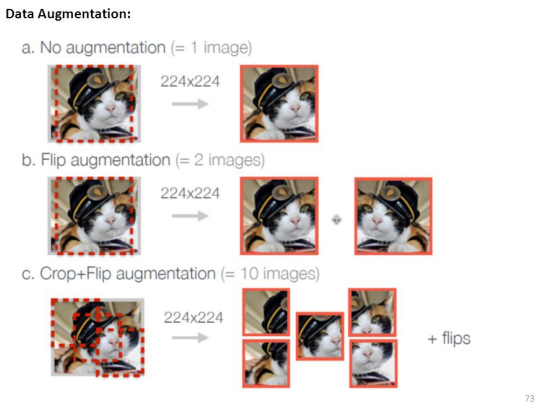


Figure 1: Example of Data augmentation transforms)

5.2 Deep generative models in Augmentation

One of the main applications for Deep generative models is to augment for data scarcity. While a lot of recent applications use generative models to augment training datasets, we did not come across any comparative or investigative study of their effectiveness.

We attempt to provide such a study.

We propose to use deep generative models like GANs to generate images to be later used in place of data augmentation. There are two steps to our problem:

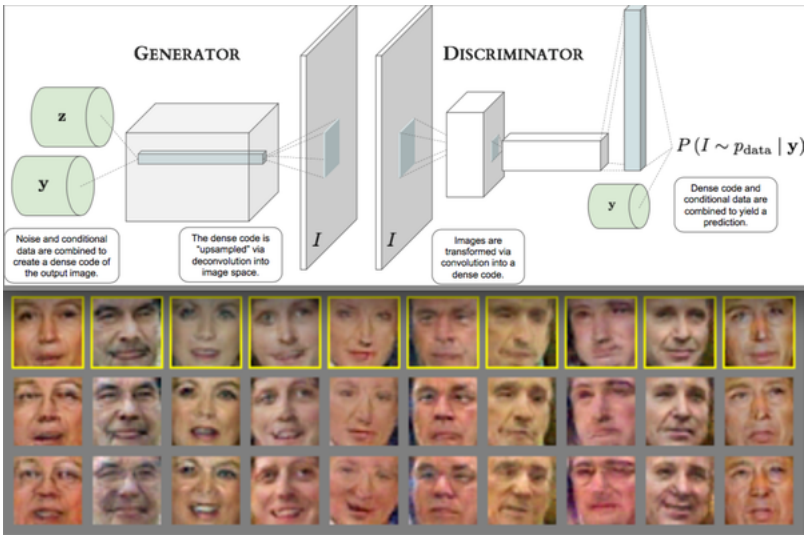
- Image Generation : The input to our generative model is the training dataset of images from 10 classes . The output from our generative model are ”fake” images which resemble ”true” images from each class.
- Image Classification : Both the ”true” images from the original dataset and ”fake” images from generator are input to our classifier. The output from the classifier is a tensor mapping each test image with its corresponding class.

5.3 Generative Adversarial Networks

In general, there are two main kinds of models - generative and discriminative. While a discriminative model discriminates between two or more classes of data, a generative model does not know anything about these classes. Instead, it tries to generate new data which fits the distribution of the training data. More specifically, a generative model G trained on training data X sampled from some true distribution D is one which, given some standard random distribution Z, produces a distribution D' which is close to D according to some closeness metric(a sample $z \sim Z$ maps to a sample $g(z) \sim D'$).

Generative Adversarial Networks (GANs) are a relatively new class of generative models. They are neural networks that are trained in an adversarial manner in order to generate data which mimics the underlying distribution.

Figure 2: GAN Architecture)



6 Experimental Evaluation

6.1 Data

We use the standard CIFAR-10 dataset for both the stages of our experiment. We chose CIFAR owing to the size of the dataset which facilitates fast training on a CPU/GPU.

The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images

from one class than another. Between them, the training batches contain exactly 5000 images from each class.

6.2 Methodology

We aimed to experiment with generative models, specifically Generative Adversarial Networks and Variational Auto-encoders. Each of our experiments have two stages :

6.3 Baseline

To illustrate the effectiveness of two types of Augmentation we used a simple baseline model without any data augmentation.

6.3.1 With Augmentation

We implemented a simple convolutional neural network with the below structure to construct our classifier. To perform data augmentation, we chose to rotate a fixed set of images from each by a random angle in the range $\{0, 45\}$.

```
local net1 = nn.Sequential()
net1:add(nn.SpatialConvolution(3, 16, 5, 5))
net1:add(nn.Tanh())
net1:add(nn.SpatialMaxPooling(2,2,2, 2))
net1:add(nn.SpatialConvolution(16, 128, 5, 5))
net1:add(nn.Tanh())
net1:add(nn.SpatialMaxPooling(2,2,2, 2))
net1:add(nn.View(128*5*5))
net1:add(nn.Linear(128*5*5, 64))
net1:add(nn.Tanh())
net1:add(nn.Linear(64,10))
net1:cuda()
```

Assumptions : We experimented between different learning rates, Activation functions and more wider networks. We chose the above structure as it converged well and also trained fast.

6.3.2 GAN Experimentation

We first trained a GAN on the celebrity-faces dataset [6] which are scaled to 64x64. The generator of the model generated the below images after couple of epochs. Note that even after Epoch = 8, our generator produced images with pretty good facial features. Refer Figure 5.

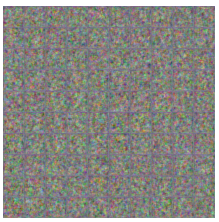


Figure 3: After Epoch = 1

Figure 4: After Epoch = 4

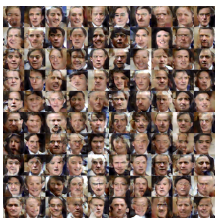


Figure 5: After Epoch = 8

6.3.3 Class conditional GANs

For the generative model, we specifically use a special kind of GAN, known as the class-conditional Generative Adversarial Network (cc-GAN) [12]. A conditional generative model works with the conditional probability $(x \mid c)$ which can be obtained by adding class c as input to both the Generator and Discriminator. We implemented a cc-GAN in Torch, and trained it over the entire CIFAR-10 dataset with parameters [13].

6.3.4 Evaluation

The classification metric serves as an indicator for how well our Generative models performed. We also looked into the Confusion matrix while training the GAN model. We benchmark our results against the baseline CIFAR model. We chose error rates instead of accuracies to compare our models.

6.4 Results

The results at the 100th epoch for the various kinds of augmentations are as follows:

Model	Train loss	Valid loss
Baseline	0.55	2.25
CIFAR-Rotation	0.45	2.28
CIFAR-GAN-500	0.6	2.19
CIFAR-GAN-1500	0.69	1.99
CIFAR-GAN-2000	0.26	2.3
CIFAR-GAN-3000	0.59	2.12

A plot for the cross-entropy loss vs. epochs is in fig. 6.

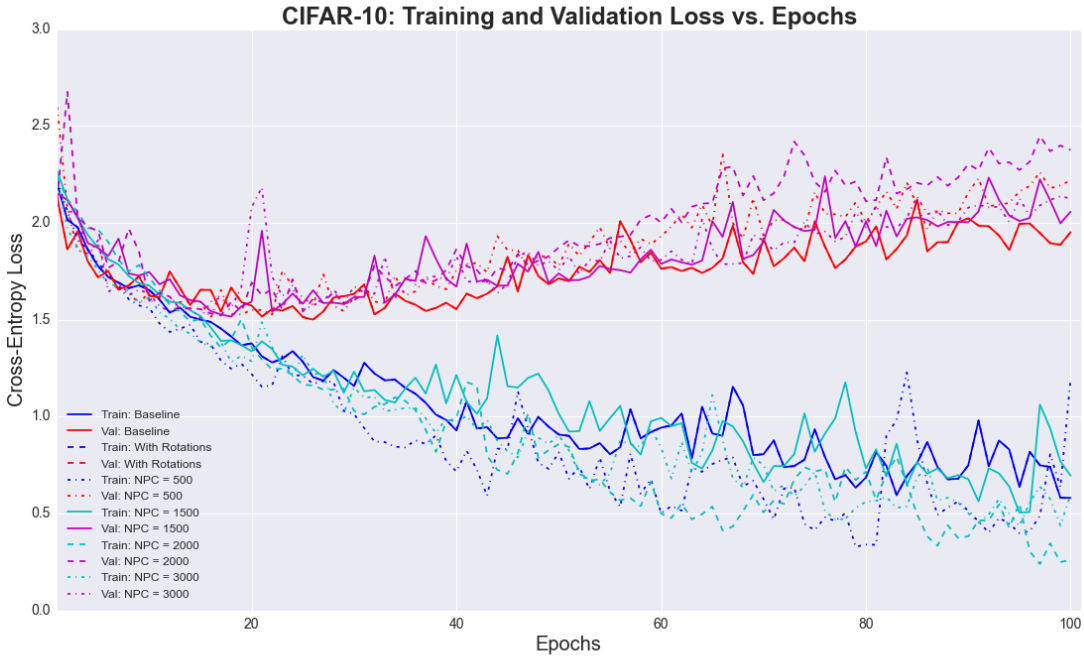


Figure 6: Training and validation losses vs. epochs for simple as well as GAN-based data augmented model.

We also attach below the images generated from the GAN at various epochs. Note that the CIFAR classes are {'airplane', 'automobile', 'bird', 'cat','deer', 'dog', 'frog', 'horse', 'ship', 'truck'}.

Figure 7: Sampled images for 10 classes. Epoch = 10.

Figure 8: Sampled images for 10 classes. Epoch = 50.

Figure 9: Sampled images for 10 classes. Epoch = 100.

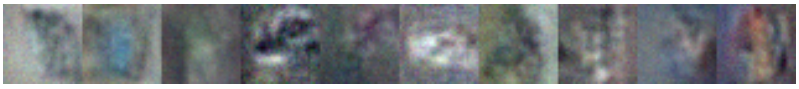


Figure 10: Sampled images for 10 classes. Epoch = 400.

We note how the initial epochs do not lead to sensible images for the classes. However, by the 400th epoch, the generated images begin resembling the actual objects.

We also sample randomly from the trained cc-GAN, and obtain the following 100 images for a few classes (frog, airplane, frog, horse, bird, deer, truck, airplane, automobile, frog).

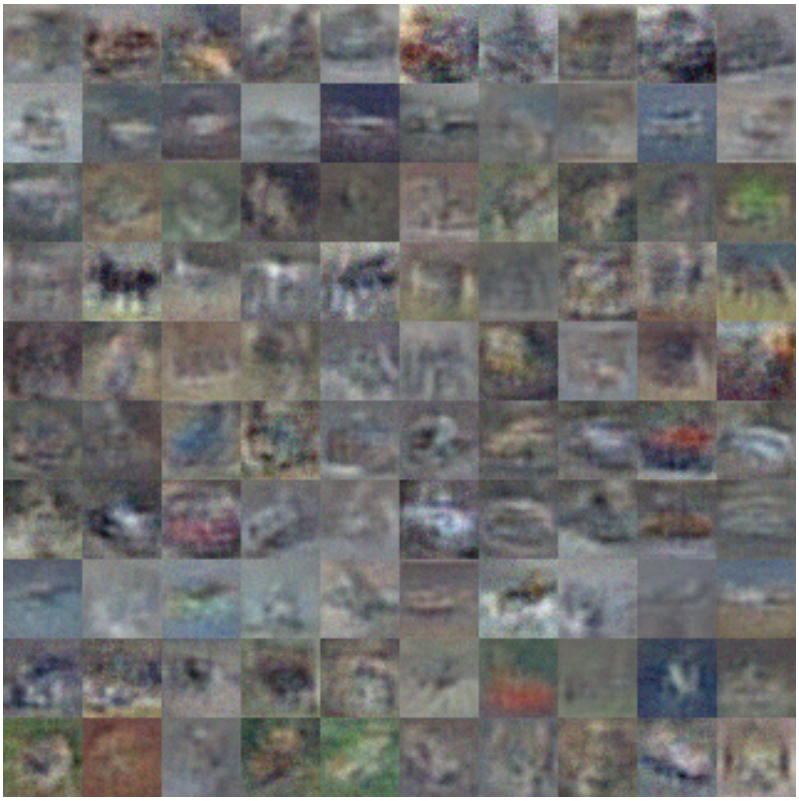


Figure 11: 100 randomly sampled images from the trained GAN.

6.5 Discussion

While traditional approaches for data augmentation are popular, these transformations do not help beyond a point, since they involve rotations, translations, scaling and color jittering. Images augmented with these transformations thus tend to be highly correlated with each other. A stronger form of augmentation would involve learning the manifold of the data distribution using generative models, which include traditional and variational autoencoders, Restricted Boltzman Machines (RBM) as well as Generative Adversarial Networks (GAN). The latter have been known to perform exceptionally well, and are easier to train than the former. We therefore explore using adversarial networks on the CIFAR-10 dataset, consisting of 50,000 images from 10 object classes.

The main advantage of using GANs and other deep generative models for data augmentation is to be able to train models using less labeled data, since the generative networks can be trained in an unsupervised manner. Moreover, by enforcing that the model is exposed to such non-linear "transformations", it tends to regularize the computer vision models.

For our given dataset, we find that while the extra images help decrease the training loss below the baseline and rotation-augmented models, they do not improve the validation accuracy by much. This can be explained by the fact that we restrict our models to train for 100 epochs (due to computing issues), and thus, the validation loss for the GAN-augmented image dataset will further increase for higher number of epochs. Secondly, from Fig. 11, we note that while the images generated maintain a rough shape similar to the original class, they do not resemble the object exactly, and thus, due to the CIFAR-10 dataset containing sharp, non-pixelated images of the objects, does not assist in the prediction very much.

Thus, we suggest that using GAN augmentations for real-world datasets, where the images may be grainy and pixelated. This would lead to better performance over standard data augmentations.

7 Conclusion and Future Work

We experiment with Generative Adversarial Networks on the CIFAR-10 dataset, to facilitate fast training and comparison across other standard models. As an extension, we would like to work on other generative models, such as the Variational Auto-encoder, to compare the performance of our classifiers, as well as combine them with existing augmentations. Our current convolutional network would be fine-tuned and further experimented with different parameters, networks to improve convergence. We would like to continue our experimentation along the above lines before the poster presentation.

In future work, as an extension of this project, we also wish to test better vision models, such as AlexNet and VGGNet. We would like to further visualize how the generator tries to approximate the data distribution during training, by visualizing the hidden states and train/validation loss for the discriminative/generative models.

Our code can be found at the following repository [14].

8 Bibliography

1. http://ai-on.org/projects/smart-data-augmentation-with-generative-models.html
2. https://www.cs.toronto.edu/~kriz/cifar.html
3. https://github.com/facebook/eyescream/blob/master/cifar/scripts/train_cifar_classcond.lua
4. http://torch.ch/blog/2015/11/13/gan.ht
5. http://vis-www.cs.umass.edu/lfw/
6. https://www.researchgate.net/publication/220655764_A_Generative_Data_Augmentation_Model_for_Enhancing_Chinese_Dialect_Pronunciation_Prediction
7. http://www.stat.harvard.edu/Faculty_Content/meng/JCGS01.pdf

8. http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6854671
9. I.J. Goodfellow et al. Generative Adversarial Networks. NIPS 2014.
10. C. Doersch. Tutorial on Variational Autoencoders. 2016.
11. A. van den Oord et al. Pixel Recurrent Neural Networks. 2016.
12. Mehdi Mirza, Simon Osindero. Conditional Generative Adversarial Nets.
13. https://github.com/reachbp/CIFAR-Classification/blob/master/gan/cifar/scripts/train_cifar_classcond.lua
14. https://github.com/reachbp/CIFAR-Classification