Improved Techniques for Training GAN's : A review

Ashutosh Kumar Jha (ME14B148)

March 2017

1 Introduction

Generative Adversarial Networks(GAN's) are a class of methods for learning generative models using game theory. GAN's simultaneously train 2 networks: A generator network G and a discriminator network D. G generates data $\mathbf{x}$ by taking random noise $\mathbf{z}$ as input so as to generate samples from $p_{\text{data}}(\mathbf{x})$. D on the other hand tries to maximize the probability of labelling a sample $\mathbf{x}$ from G as not being drawn from the original data distribution. G's goal(objective) is to fool the discriminator D.
It can thus be thought that D and G are playing a 2 player minimax game. The ideal goal of GAN training is thus to find the Nash Equilibrium of this game.

In the original paper by Goodfellow et al., the authors propose that instead finding the Nash equilibrium (by making updates in the function space which is a hard task and fails to converge since the cost functions are highly non-convex and the parameter space is continuous and large), updates from gradient descent be made rather in the parameter space(Even though there are no convergence guarantees for updates in the parameter space).
Another drawback of the original paper was that there was no evaluation metric provided for how well the images generated by a GAN were. Even though the networks tried to maximize the log likelihood of the samples, a better log likelihood did not necessarily mean better samples.

2 What is this paper about? (TL;DR)

This paper introduces *heuristic* techniques to get around the non-convergence problems of GAN's (as described above).
Apart from better techniques, the paper also provides with an evaluation metric for how well a GAN is.
Beyond both of these, I personally think the true success of the paper is in

the fact that it describes a way of using GAN's in the semi-supervised learning framework and acheives state of the art results on 4 popular image datasets.

3 Details of the techniques

In the subsections below, all the techniques described by the paper are explained. First an idea is added to each subsection and then mathematical details are provided to facilitate ease of reading.

3.1 Feature Matching

Idea: The idea of feature matching is to prevent G from overtraining on D. Instead of directly maximizing the output of D, we try to make G match the features of the data. (Sort of distillation of neural nets where you try to make a student learn the soft-outputs of an expert). To do that, you try to train G to generate outputs with features matching an intermediate layer of D.

Details: Let f($\mathbf{x}$) denote the activations on an intermediate layer of D. The objective function of G is now defined as $||\mathbb{E}_{x\sim p_{\text{data}}} f(\mathbf{x}) - \mathbb{E}_{z\sim p(\mathbf{z})} f(G(\mathbf{z}))||^2$. The discriminator is trained the usual way.

3.2 Minibatch Discrimination

Idea: The problem of single mode collapse from the original paper is addressed in this part of the paper. The idea of the technique is to make the discriminator look at a minibatch of examples in combination. The task of the discriminator is effectively still to classify single examples as real data or generated data, but it is now able to use the other examples in the minibatch as side information.

Details: In the experiments described in the paper, the authors explicity aim to identify generator samples that are particularly close. The closeness is defined as $c_b(\mathbf{x}_i, \mathbf{x}_j) = exp(-||M_{i,b} - M_{j,b}||_{L1})$ where M_i is produced by multiplying a 3D tensor T with $f(\mathbf{x})$. The output of the minibatch layer is thus:

$$\begin{aligned} o(\mathbf{x}_i)_b &= \sum_{j=1}^{n} c_b(\mathbf{x}_i, \mathbf{x}_j) \\ o(\mathbf{x}_i) &= \begin{bmatrix} o(\mathbf{x}_i)_1 & o(\mathbf{x}_i)_2 & \cdots & o(\mathbf{x}_i)_B \end{bmatrix} \end{aligned} \tag{1}$$

Minibatch Discrimination allows us to generate visually more appealing examples however, Feature Matching was found to work much better if the goal is to obtain a strong classifier using the approach to semi-supervised learning

3.3 Historical Averaging

Idea: The original problem with GAN's was that there was no guarantee with gradient descent that a two player game like GAN won't go into extended orbits.

The idea is to add the term $||\theta - \frac{1}{t}\sum_{i=1}^{t}\theta[i]||$ to each player's cost function where $\theta[i]$ is the value of parameters at a past time i. Since Orbits are penalised by always being far from their mean,this approach is able to find the minima of continuous non-convex low dimensional games and thus also helps with GAN's.

Details: No further details. The idea is just a heuristic which helped in GAN training.

3.4 One-Sided Label Smoothing

Idea: The idea derived from the label smoothening where 0 and 1 targets are smoothened to 0.1 and 0.9. Instead here only the positive targets are smoothened and negative labels are left at 0. This seems to make the networks more resistant to adversarial examples.

Details: Replacing positive labels with α and negative labels with β gives the optimal discriminator at $D(\mathbf{x}) = \frac{\alpha p_{\text{data}}(\mathbf{x}) + \beta p_{\text{model}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_{\text{model}}(\mathbf{x})}$. The presence of p_{model} in the numerator is problematic because, in areas where p_{data} is approximately zero and p_{model} is large, erroneous samples from p_{model} have no incentive to move nearer to the data. Thus only the positive labels are smoothened.

3.5 Virtual Batch Normalization

Idea: The idea is similar to batch normalization. However instead of doing vanilla batch normalization, the minibatch is normalized by using statistics (i.e. mean and variance) from a fixed reference batch which is fixed at the start of the training.

Details: The problem with vanilla batch normalization is that it makes the output of the BN layer for a given input highly dependant on different other inputs in the same minibatch which is not desirable as it would increase the variance of the BN outputs. So a fixed reference batch is used.

3.6 Inception Score Evaluation Metric

The idea behind introducing an automated evaluation metric as opposed to human evaluation is that the human evaluation metric varies depending on the setup of the task and the motivation of the annotators.

The authors thus introduce a new evaluation metric called the **inception score**. Here the inception module (from the GoogleNet paper) is applied to every generated image to obtain $p(y|x)$. The inception score is then defined as: $\exp(\mathbb{E}_x \text{KL}(p(y|x)||p(y))$.
The paper further cautions that the Inception score should be used as a rough guide to evaluate models that were trained via some independent criterion;

directly optimizing Inception score will lead to the generation of adversarial examples.

4 Semi-Supervised Learning

This section in my personal opinion is the most brilliant part of the paper.

The problem of semi-supervised learning can be defined as a learning task where small amount of labelled data is available with large amount of unlabelled data and a learning task has to be done (say classification).
The paper proposes a very simple idea towards doing semi-supervised learning. What we do is essentially ask the discriminator D to generate a label for an input as well apart from the task of discriminating whether it comes from true distribution or not.

We can set up D as follows: Take a standard classifier classifying input $\mathbf{x}$ into K different classes. We now introduce a new class $y = K + 1$ as "generated" class. We now add samples from G to our dataset into the K+1^{th} class. We may then use $p_{model}(y = K + 1|\mathbf{x})$ to supply the probability that $\mathbf{x}$ is fake, corresponding to $1 - D(\mathbf{x})$ in the original GAN framework. We can now also learn from unlabeled data, as long as we know that it corresponds to one of the K classes of real data by maximizing $\log p_{model}(y \in 1, ..., K|x)$.

It can now be shown (shown in the paper) that the loss for training the classifier can be given by:

$$L = L_{\text{supervised}} + L_{\text{unsupervised}} \tag{2}$$

where $L_{\text{unsupervised}}$ is same as the GAN loss and $L_{\text{supervised}}$ is the negative log probability of the label, given that the data is real.
An obvious way to minimize L is now to train in the original GAN framework with discriminator as defined above.

5 Experiments and Results

The techniques discussed above were applied to perform semi-supervised learning on MNIST,CIFAR-10,SVHN and ImageNet and produced state of the art results on all the datasets (except ImageNet where it is still bad although is still SOTA). An important implementation point in the experiments is that Minibatch Discrimination performed badly with all the experiments however Feature Matching worked well so Minibatch Discrimination wasn't used. Another side point in the implementation put up on github is that they have not used VBN but simple BN for all the experiments.
Along with the test accuracy for the semi-supervised learning task, it was seen that the inception scores obtained using the methods described in the paper were also better than those using other methods. In the author's opinion, the

Inception score correlates well with our subjective judgment of image quality.

In case of ImageNet where pictures of animals were generated, the above models are able to identify important features such as fur,eyes,nose,etc.(an unprecedented task) however, the pictures do not make any anatomical sense.

6 Directions for Further Research

This paper primarily introduces heuristics towards solving the non-convergence problems of GANs. No theoretical guarantees are provided anywhere (in fact the problem was present in the original GAN paper as well). Finding good theoretical guarantees for convergence of these techniques is definitely a huge direction for further research which the authors suggest.