Improved Techniques for Training GANs

GANs.-

Class of methods for learning generative models based on game theory.

Goal: train a generator network G that produces samples from the data distribution p\_data(x).

G is provided by a discriminator network D(x) trained to distinguish samples from the generator distribution p\_model(x) from real data.

Training GANs requires finding a Nash equilibrium of a non-convex game with continuous, highdimensional parameters.

GANs are typically trained using gradient descent techniques.

In this Work: **several techniques to encourage convergence of the GANs game.**

**2. Related Work**

Focus: Improving he stability of training and the resulting quality of GAN samples.

They used some of the “DCGAN” proposed in Radford et al.

*Proposed Techniques : feature matching, minibatch features and batch normalization.*

**Primary goal :** improve the effectiveness of GAN for semi-supervised learning (improving the performance of a supervised task [classification] by learning on additio2. nal unlabelled examples).

**3. Toward Convergent GAN Training**

Training GANs consists in finding a Nash equilibrium to a two-player non-cooperative game.

Each player wishes to minimize its own cost functions:

* JD for the discriminator.
* JG for the generator.

A Nash equilibrium is a point (thetaD, thetaG) such that

* JD is at a minimum with respect to thetaD and
* JG with respect to thetaG.

! We are not aware of any algorithms feasible to apply to the GAN game where:

* Cost functions are non-convex,
* Parameters are continuous,
* The parameter space is extremely high-dimensional.

Intuition: using traditional gradient-based minimization techniques to minimize each player’s cost simultaneously. **BUT** there is a compromise between JD and JG, a modification to thetaD that reduces JD can increase JG and vice versa.

**3.1 Feature matching**

Deals with GANs instability by

* Specifying a new objective for the generator. – **prevents it from overtraining** on the current discriminator.
* The new objective requires the generator to generate data that matches the statistics of the real data.

Definition of the generator: norm 2 difference between expectance)

• Minibatch Discrimination: Helps avoid the collapse of the generator to a single mode by allowing the discriminator to consider multiple examples together.

• Historical Averaging: Involves modifying the cost function to include a term related to the historical average of parameters.

• One-sided Label Smoothing: Improves the discriminator's performance by smoothing only the positive labels.

• Virtual Batch Normalization: Normalizes each example based on a fixed reference batch and itself, reducing the dependency on other inputs in the same minibatch.

**Assessment of Image Quality**

* **Evaluating GANs**: Discusses the challenge of objectively assessing GANs due to the lack of an objective function.
* **Human Annotators and Inception Score**: Uses human annotators and a new metric called the Inception score to evaluate the quality of generated images.

**Semi-supervised Learning**

* **Methodology**: Describes how GANs can be used for semi-supervised learning by adding a new class for generated samples and training a classifier on both real and generated data.
* **Improving Classifier Performance**: Shows that these techniques can improve the performance of a classifier in a semi-supervised learning scenario.

**Experiments**

* **Datasets**: Conducts experiments on various datasets (MNIST, CIFAR-10, SVHN, ImageNet) to test their methods.
* **Results**: Presents results demonstrating the effectiveness of their techniques in improving image quality and classification accuracy.

**Conclusion**

* **Contributions**: Summarizes their contributions in stabilizing GAN training and introducing the Inception score for evaluation.
* **Future Work**: The authors express a desire to develop a more rigorous theoretical understanding of these techniques.

1. **Dataset Preparation**
   * The MNIST dataset, consisting of 60,000 training images and 10,000 testing images of handwritten digits, is collected.
   * The images are preprocessed if necessary (e.g., normalization, resizing).
2. **GAN Architecture Setup**
   * Two neural networks are defined: a Generator (G) and a Discriminator (D).
   * The Generator generates images from random noise.
   * The Discriminator differentiates between real images from the MNIST dataset and fake images generated by G.
3. **Training Process**
   * The Discriminator is trained with real images labeled as "real" and fake images from the Generator labeled as "fake".
   * The Generator is trained to create images that the Discriminator will classify as real.
   * This training is an iterative process, often involving backpropagation and gradient descent optimization methods.
4. **Model Evaluation**
   * During or after training, the generated images are evaluated to see if they resemble real handwritten digits.
   * Techniques like visual inspection or the Inception score (as mentioned in the paper) can be used for evaluation.
5. **Image Generation**
   * Once the GAN is sufficiently trained, the Generator can be used to create new images.
   * These images are novel creations that resemble the handwritten digits in the MNIST dataset.
6. **Application in Semi-supervised Learning**
   * The trained GAN can be used for semi-supervised learning tasks on the MNIST dataset, where it enhances the performance of a classification model by providing additional training data (generated images).