Assignment 3, On Generative Models, Adversarial Examples and Defense Mechanisms

Prateek Kumar Agnihotri(2016BB10033)

***Abstract*—Generative modes hae turned the whole regime of Deep learning paradigm with more than 2000 papers published in this domain just within last 2 year. It shows how much promise this area hold in the current Deep Learning society and through this assignment we try to get us acquainted with these state of the art mythologies. The assignment has 3 part.**

***Index Terms*—GANs (Generative Adversarial Netwroks), VAE (Variational Auto Encoder),Adversarial Example, Carnigi Wagner Attack, FGSM Attack.**

I. Introduction

First is creation of a Generative Model. We tried both GAN and VAE and will present the result of both on data set such as MNIST, Fashion-MNIST and CIFAR10. Second part is to train a classifier on these dataset and then break the classifier using generative adversarial examples. We tried FGSM and Carnigi Wagner attack on the classifier. The last part is to make a defense mechanism to make our classifier robust against such adversarial examples. For that we use a novel technique which we will describe later in this paper. The technique is based on the fact that Lipshitz criteria helps a classifier to generalize well and makes the model insensitive to small change in the input space. So if an adversarial example is too close to a real example in the true data manifold the classifier will remain insensitive to the adversarial noise and will classify it correctly.

II. Datasets

We tried this assignment on 3 datasets. MNIST, Fashion-MNIST and CIFAR-10. It will be good if we take a look at them before going through the complex details of our models and training. MNIST dataset consists of 28×28 grayscale

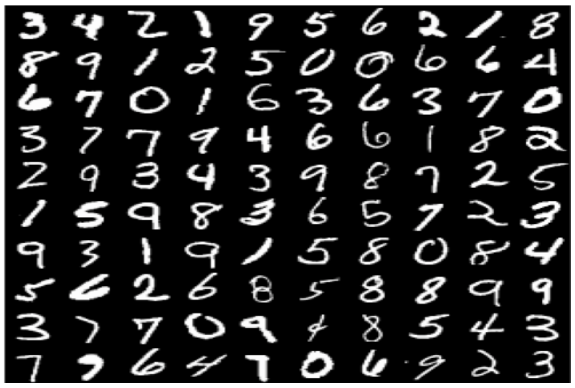

Fig. 1. MNIST Image dataset

image of hand written digit containing 60,000 images in total. There are total 10 classes. Fashion-MNIST dataset consists of 28×28 grayscale image of Fashion Accessories like handbags

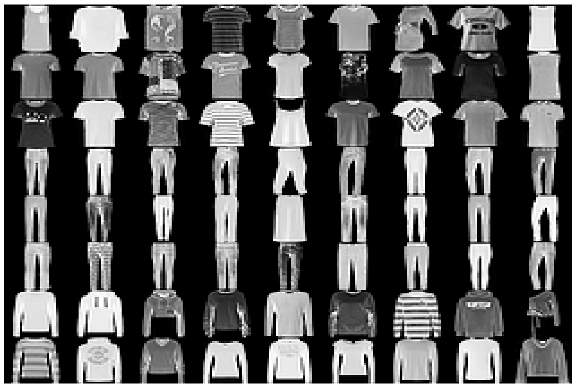

Fig. 2. FashionMNIST Image dataset

and heels containing 60,000 images in total. There are total 10 classes. The CIFAR-10 dataset consists of 60000 32×32

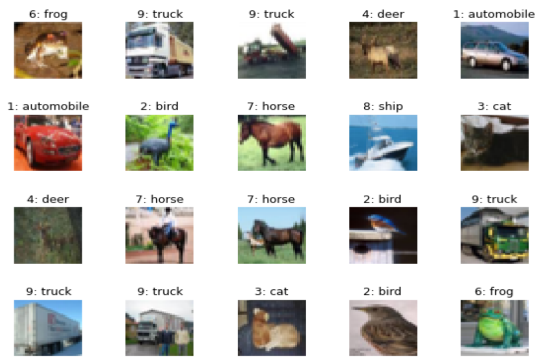

Fig. 3. CIFAR10 Image dataset

colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

III. Part 1, Generative Models

Here we describe and present the result of some of our generative models that we used during the experimentation. As it seems GAN training is one heck of a task andwegot to know it pretty well. Most of the times they don't converges and when they do, the output is sometimes not good, sowehad to try a lot of architectures for generating a good image. We tried both VAE and GAN methodology which we will describe now.

A. VAE as Generative Model

Before trying GANs we thought of trying VAE on Fashion-MNIST dataset to see how well it performs. The idea of VAE is to minimize KL between a desired latent space representation and current distribution. As mentioned in many literature the ELBO can be written as,

$$\log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)}\right) \geqslant \mathcal{L}\left(\boldsymbol{\theta}, \boldsymbol{\phi} ; \mathbf{x}^{(i)}\right) \tag{1}$$

$$E_{q_{\phi}(\mathbf{z}|\mathbf{x})}\left[-\log q_{\phi}(\mathbf{z}|\mathbf{x})+\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})\right] \tag{2}$$

After further modification,

$$\mathcal{L}\left(\boldsymbol{\theta}, \boldsymbol{\phi} ; \mathbf{x}^{(i)}\right) =$$

$$-D_{KL}\left(q_{\phi}\left(\mathbf{z}|\mathbf{x}^{(i)}\right) \| p_{\boldsymbol{\theta}}(\mathbf{z})\right)+E_{q_{\phi}\left(\mathbf{z}|\mathbf{x}^{(i)}\right)}\left[\log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)}|\mathbf{z}\right)\right]$$

VAE tries to maximize this lower bound using SGVB. We present the result now.

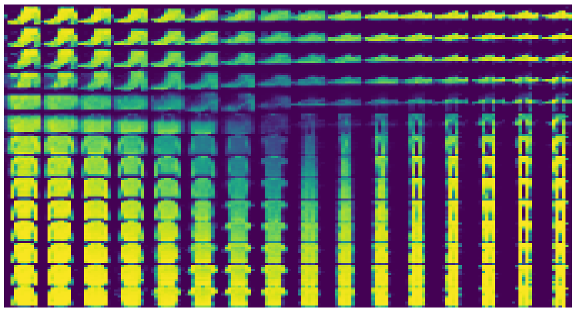

Fig. 4. VAE output on FMNIST

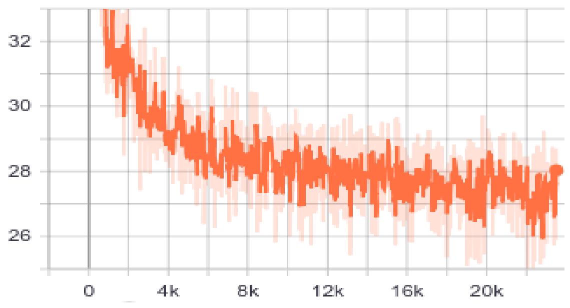

Fig. 5. Loss vs Number of epoch During Training

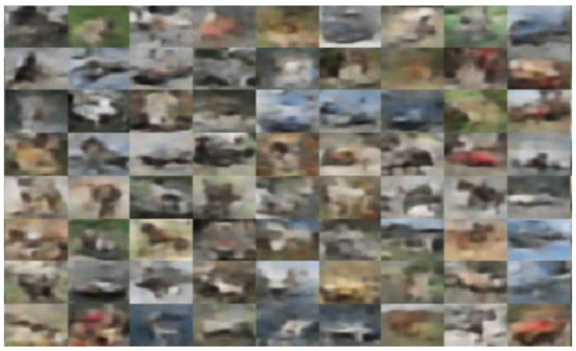

Fig. 6. CIFAR output on VAE

We also observed the latent encoding in the ten dimensional manifold. Which is as followed.

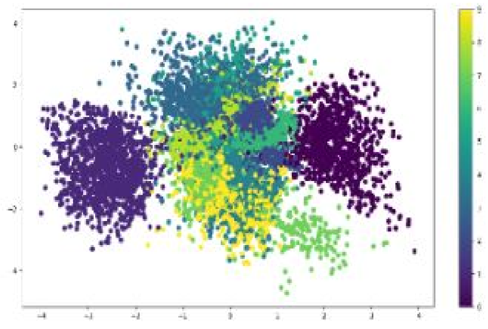

Fig. 7. Latent Space Encoding in 2D

B. GANs

The general cost function of a typical GAN is as follow as given in Goodfellow's original paper

$$\min_{G} \max_{D} V(D, G)=E_{\boldsymbol{x} \sim p_{\text{ data }}(\boldsymbol{x})}[\log D(\boldsymbol{x})]+E_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log (1-D(G(\boldsymbol{z})))\tag{3}$$

$$D_{G}^{*}(\boldsymbol{x})=\frac{p_{\text{data}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x})+p_{g}(\boldsymbol{x})} \tag{4}$$

Represents the optimal discriminator. The cost function under such setting becomes,

$$C(G)=-\log (4)+K L\left(p_{\text{ data }} \| \frac{p_{\text{ data }}+p_{g}}{2}\right)+K L\left(p_{g} \| \frac{p_{\text{ data }}+p_{g}}{2}\right) \tag{5}$$

Which is nothing but,

$$C(G)=-\log (4)+2 \cdot J S D\left(p_{\text{ data }} \| p_{g}\right) \tag{6}$$

We tried a lot of variation of different GANs and will show their outputs here. The first idea was to use a simple DCGAN and see if it works. But contrary to our believes DCGAN failed miserably on basic data set such as MNIST. As given in the paper Self Attention GANs this follows the result that only very deep Convolution Networks can efficiently represent the relationship between long term variables in a particular image. Output of our GAN.

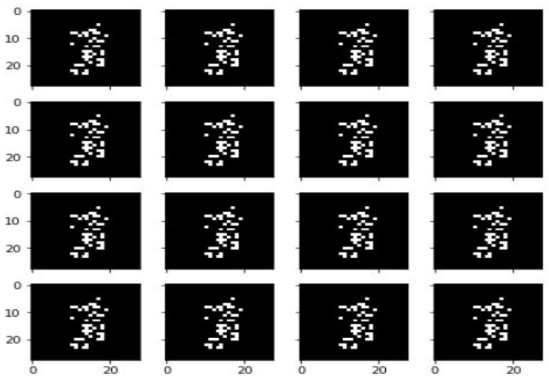

Fig. 8. DCGAN output on MNIST

Which is pretty much noisy. So we tried a fully connected neural netwrok too, and the results were very good. Not only it did converge fast, the output was less noisy and more real. Now since our model was working we thought of doing the

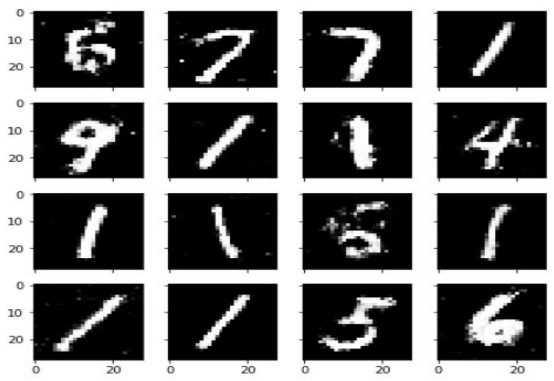
Fig. 9. Fully connected GAN output on MNIST after only 100 epochs

experiment with other datasets too. So we ran the same GAN of Fashion-MNIST and on CIFAR10 too. Here we present the output of the same. As the training progressed we got better

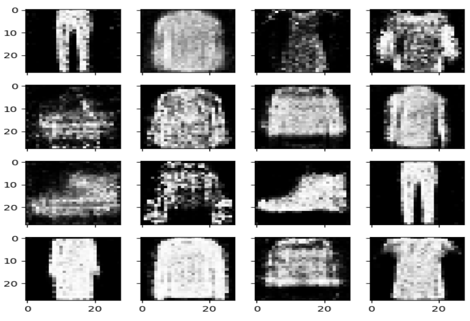
Fig. 10. Fashion-MNIST output after 100 epochs only

and better results.

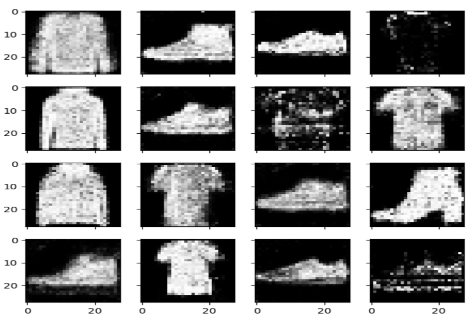
Fig. 11. Fashion-MNIST output after 200 epochs

Now comes the turn of CIFAR10, which is by far was most difficult to train. Because of its complex presentation

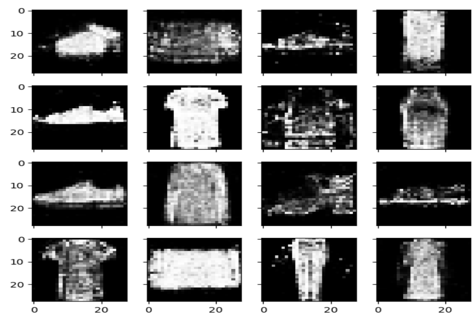
Fig. 12. Fashion-MNIST output after 300 epochs

and structure. For training a GAN on this netwrok we used again a DCGAN with deeper architecture. The outputs were good this time contrary to MNIST trained earlier on DCGAN.

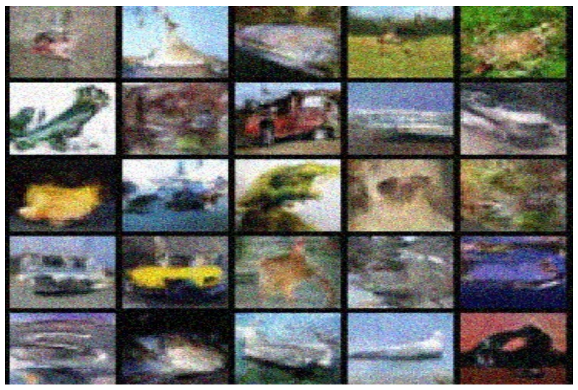
Fig. 13. CIFAR10 images generated by DCGAN after 1000 epochs

IV. Part 2, Training a Classifier

We trained two classifier, one for Fashion-MNIST and one for CIFAR10. Fashion-Mnist Classifier: 2-Convolutional layers having 64 and 128 filters with kernel size 2 2, followed by a dense layer with 256 neurons and then a dense layer with softmax to predict the probability of belonging to a certain class. Dropout is used to regularize the model. The best accuracy attained by the classifier on the test data is 92.36%.

CIFAR10 Classifier: 3-Convolutional layers having 32, 64 and 128 filters with kernel size 2 2, followed by a dense layer with 256 neurons and then a dense layer with softmax to predict the probability of belonging to a certain class. Dropout is used to regularize the model. The best accuracy attained by the classifier on the test data is 79.82%. The classifier are CNN based and hence perform pretty good even on complex tasks as such.

The learning curves obtained

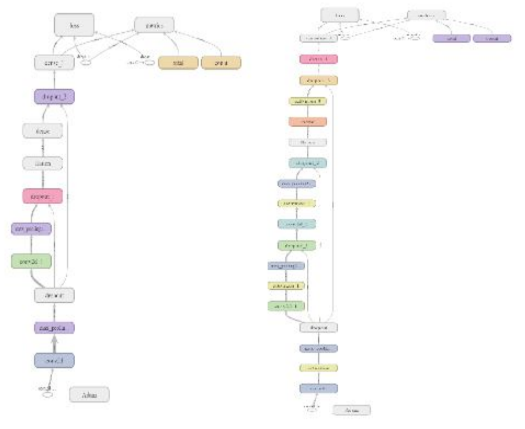

Fig. 14. Architecture of FMNIST(left) and CIFAR10(right) classfier

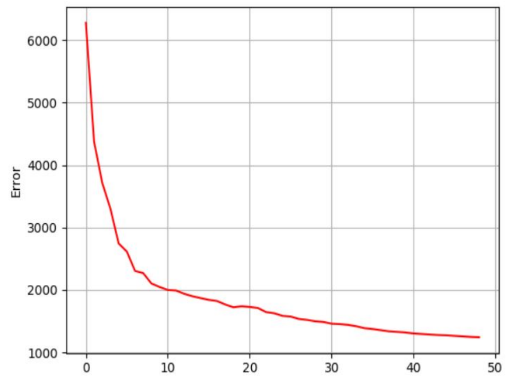

Fig. 15. Error vs Number of epochs for FMNIST classifier

V. Adversarial Example Generation

We choose Fast Gradient Sign Method (FGSM) and Carlini-Wagner (CW) attack to break our classifiers as one is fast and other is accurate.

A. FGSM attack

Fast Gradient Sign Method (FGSM): Goodfellow et al. proposed a fast and simple method that relies on taking a first-order Taylors series approximation on the loss term. This performs a one step gradient update along the direction of sign of gradient at each pixel to make the sample. Let L(x,y) be loss function for an image x with correct label y.

$$\delta = \epsilon \cdot \text{sign}\left(\nabla_x (L(\mathbf{x}, y))\right) \tag{7}$$

where ϵ is a hyper parameter.

B. Carlini-Wagner Attack

CW is an optimization based attack that used logits-based objective function instead of commonly used cross-entropy loss. The perturbation is found by optimizing the function:

$$\begin{aligned} \min\nolimits_{\delta \in R^n} \mathcal{D}(\mathbf{x}, \mathbf{x}+\delta) + c \cdot h(\mathbf{x}+\delta) \\ \text{s.t} \quad \mathbf{x}+\delta \in [0,1]^n \end{aligned} \tag{8}$$

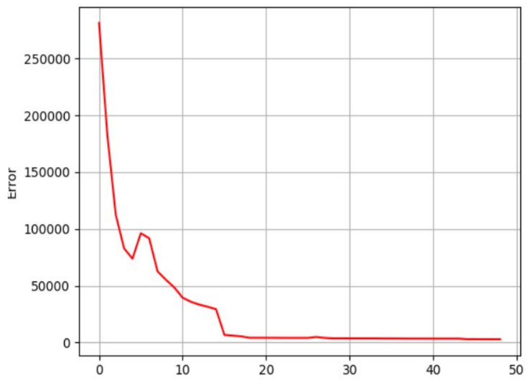

Fig. 16. Error vs Number of epochs for CIFAR10 classifier

Results on Fmnist:

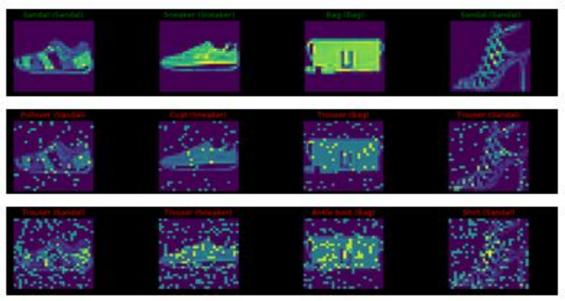

Fig. 17. 1-Before Attack 2-After CW-attack 3-After FGSM-attack

Results on CIFAR10:

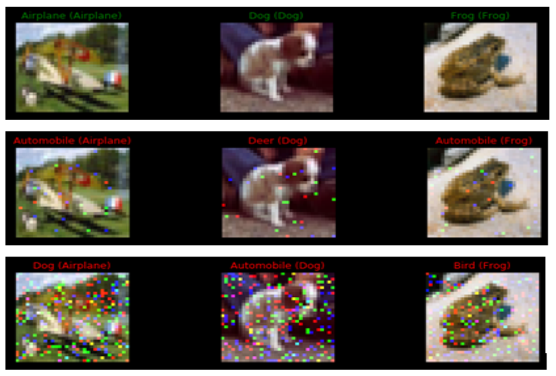

Fig. 18. 1-Before Attack 2-After CW-attack 3-After FGSM-attack

The first row represent normal CIFAR images, row 2 represents CW attack generated adversary, row 3 represents FGSM attack generated adversary.

VI. Defence Mechanism

We tried a new technique to make our model robust against adversarial examples. We used the idea of Lipshitz constraint and its definition to train a classifier with inherent insensitivity towards small variations in the input images which makes an image an adversarial example. We give a proper background before telling what exactly we did and also elaborate our intuitions.

A. Background

Let S be an open set in some R^n. A function f :$R^n \rightarrow R^m$ is called Lipschitz continuous on S if if there exists a non-negative constant L_f R0, called a Lipschitz constant of function f on S, such that the following condition holds:

$$\|f(x) - f(y)\| \leqslant L_f \|x - y\| \tag{9}$$

or,

$$\frac{\|f(x) - f(y)\|}{\|x - y\|} \leqslant L_f \tag{10}$$

So, we imposed lipschitz constraint on our classifier(a NN) excluding the softmax layer to see if it works better. And as proved by *Muhammad Usama and Dong Eui Chang* in their paper *Towards Robust Neural Networks with Lipschitz Continuity* in **Theorem 1**, it should. And it did work in our experimentation.
The adversarial examples which were earlier classified as wrong now are classified right and we need more noise to actually break this classifier which is actually a good thing. Hence we ac hived our purpose of defence.

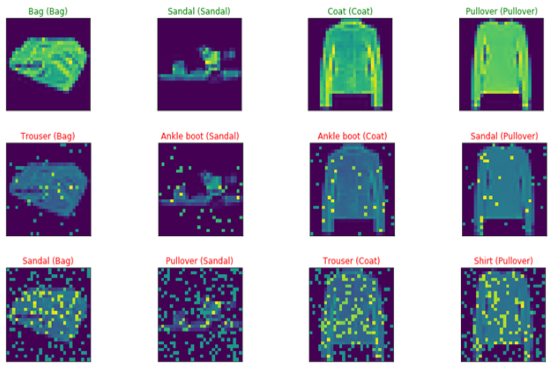

Fig. 19. 3rd Row shows noise required by CW attack to break a lipschitz FMNIST classifier

So we observed in this case that the noise needed to break a Lipschitz classifier is increased significantly and the earlier adversarial examples which were getting misclassified due to some added noise are now getting correctly classified because of the fact that lipschitz classfier is insensitive to small changes in input. Basically lipschitz constraint denotes that small changes in input will give small changes in output. So for very small added adversarial noise, the classifier output will change slightly and hence better will be the classification.

VII. Conclusion

This assignment got us acquainted with the epitome of deep learning, the generative models and their usages. We got to know about GANs, their usage and how to train them, We learned how dreadful GAN training is and how difficult it is to stabilize a GAN. We use $K = 4$ iteration to train the discriminator before training the generator and it gave us optimal results. It took us 2 days to train a GAN but it was worth it. VAE on the other hand is easy to train and still gives good results. Then we learned about adversarial examples generation which are nothing but points with added noises in the data manifold which are similar looking to a target image x with class c but when we feed in this image to our trained classifier the output is not c. The reason why they arise is still unknown but non generalizability of model is blames for such occurrences. Then we learned about Lipschitz constraint and its benefit in defence mechanism of a classifier against adversarial examples. Overall this assignment was the best in terms of learning scope and difficulty level. W

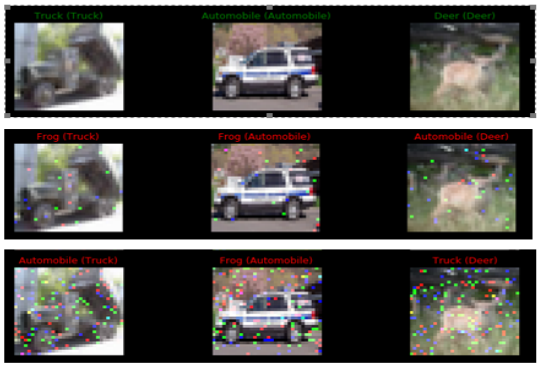

Fig. 20. 3rd Row shows noise required by CW attack to break a lipschitz CIFAR classifier