Adversarial Attack on Deep Neural Networks using GANs

CS772A: PROBABILISTIC MACHINE LEARNING
SUPERVISOR: PROF. PIYUSH RAI

TEAM MEMBERS

Aadil Hayat	17111001
Sarthak Mittal	150640
Someshwar Jain	150723

ABSTRACT

In this project we wish to explore the research problem of Adversarial Attacks on Deep Neural Networks. Deep learning takes advantage of the concept of learning complex mapping about a dataset by building upon simpler mappings. It takes advantage of large datasets and computationally efficient training algorithms to outperform other approaches at various machine learning tasks. However this very powerful method has imperfections which makes it vulnerable to adversarial samples. These samples are crafted by adversaries with the intent of causing deep neural networks to misclassify. In this project we will look at a training method where we will use the concept of Generative Adversarial Networks to learn a Deep Neural Network function that generates adversarial samples and also a Deep Neural Network based classifier that will be robust to such adversarial samples.

OBJECTIVE

The main objective of this project is to explore a method of training a generator network and a classification network in a minmax game setting very similar to Generative Adversarial Network. In this setting the generator would be a deep neural network pre-trained on the given dataset to generate the realistic looking images from the dataset given the input class. While the classifier would be deep neural network based classifier trained on the given dataset. These networks will then be trained to increase each other's error similar to minmax game setting of GANs.

PREVIOUS WORK

DeepFool (1511.04599v3)

DeepFool worked on the basic technique of modifying the input image a little so that the classifier misclassifies it. The modification is done using a noise matrix whose values are added to the image matrix element by element. The noise matrix is found using a loss function that minimizes the sum of squares of all the elements of the matrix given the condition that the classifier mis-classifies.

Andrej Karpathy Blog (Breaking Linear Classifiers on ImageNet)

Andrej uses the technique that given an image, we feed forward through the classifier network and on backpropagating, instead of doing a parameter update of the model, we do the pixel update of the image to increase the probability score of whichever class we want to make the model predict. In this way, we change the image just a little at each step, where the updates are not differentiable, but the classifier ends up failing to classify the image correctly.

PLANNED WORK

Datasets

In this project we will work with mainly with 3 major datasets: MNIST Handwritten Digits, CIFAR-10 and ImageNet.

MNIST Dataset

The MNIST dataset (Modified National Institute of Standards and Technology dataset) is a dataset of handwritten digits containing 60,000 training images and 10,000 testing images.The digits have been size-normalized and centred in a fixed-size image.

CIFAR-10 Dataset

CIFAR-10 is a subset of the 80 million tiny images dataset and consists of 60,000 32*32 color images containing one of 10 object classes, with 6000 images per class.

ImageNet Dataset

ImageNet is an image database organized according to the WordNet hierarchy in which each node of the hierarchy is depicted by hundreds and thousands of images. On an average each node has more than five hundred images.

Implementation

We train a normal conditional GAN, with more complicated architectures as we shift to more complex datasets, with a Generator and a Discriminator to generate images that are similar to the ones present in the dataset. Then we train a separate classifier on the dataset that is able to predict with good accuracy.

Once we have trained the conditional GAN and classifier separately initially, we then train them together in such a way that the Generator of the GAN is trained against the classifier instead of the original Discriminator. The Generator is made to create images of certain class and then trained with the Classifier in such a way that it creates the images of that class only but in such a way that the classifier mis-classifies. In a way, we want our GAN's Generator to learn how to minimally morph the images of a certain class in such a way that the Classifier, which generalized well over the dataset, fails to classify the image correctly.

To train the Generator with the Classifier, we use a cross entropy loss function where the label to the Classifier will be the targeted class, and the label provided to the Generator will be the original class. We regularize this process by pairing this Generator with the initial Discriminator too so that it continues to keep producing images that resemble to the class, that is, doesn't morph the images too much.

Related Papers

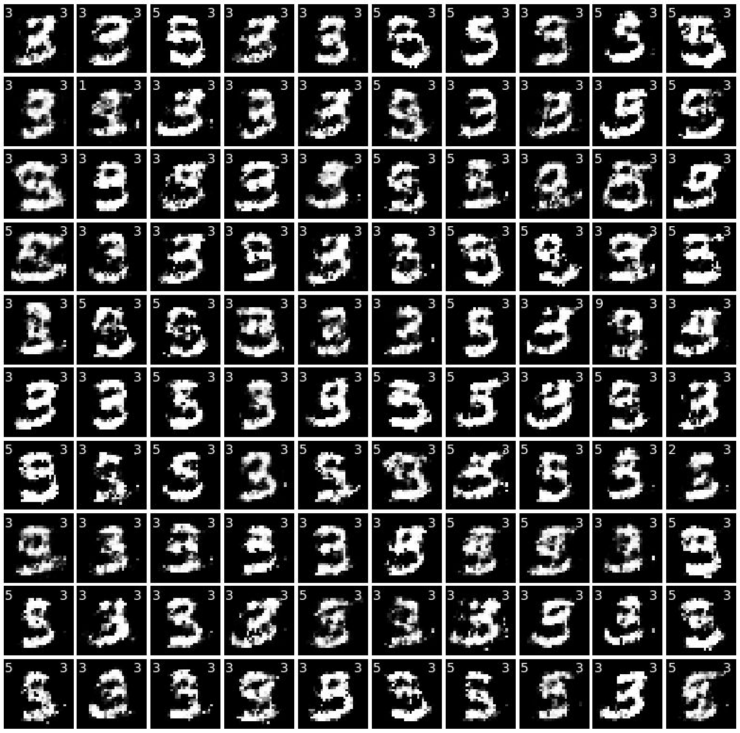
- Conditional Generative Adversarial Nets (1411.1784v1) by Mehdi Mirza and Simon Osindero
- Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks (1511.06434v2) by Alec Radford, Luke Metz and Soumith Chintala
- DeepFool: a simple and accurate method to fool deep neural networks (1511.04599v3) by Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi and Pascal Frossard
- Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images (1412.1897v4) by Anh Nguyen, Jason Yosinki and Jeff Clune

MILESTONES

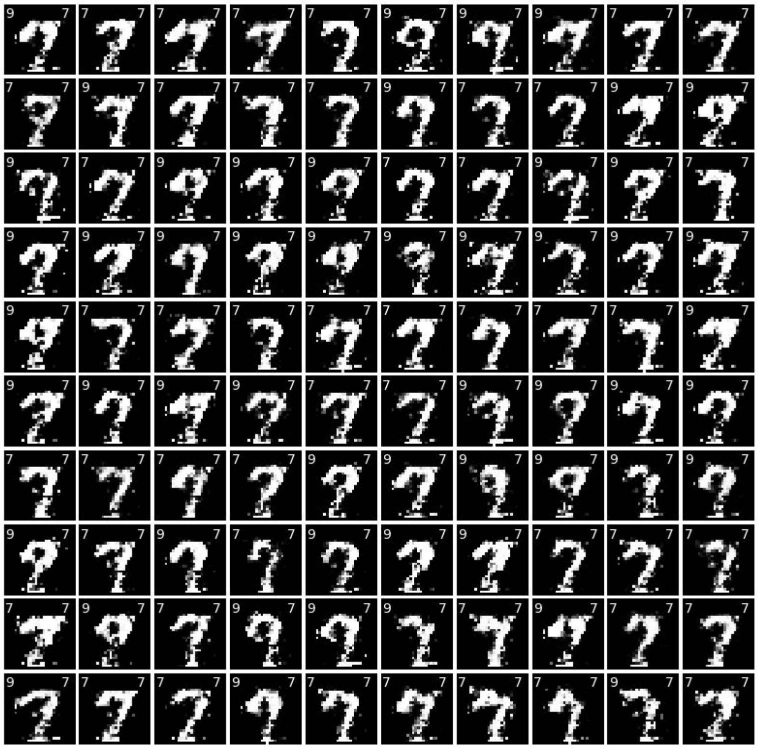
MNIST Dataset with Dense Layers 10 September

Images Generated

Images of 3 being classified as 5

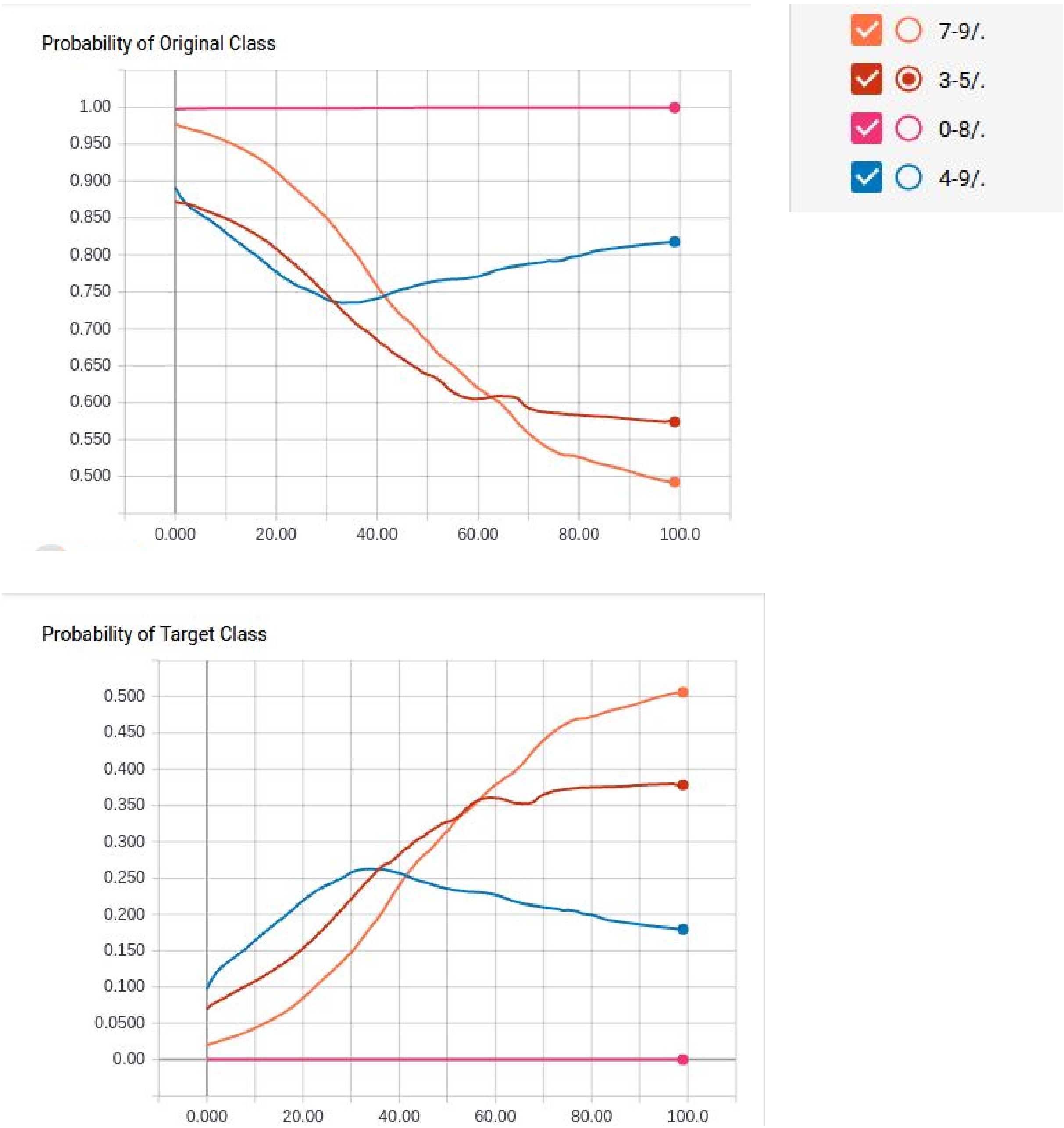


Images of 7 being classified as 9



Softmax Probability Logits

Implement the above described model for MNIST dataset with simple conditional Generator, Discriminator and Classifier comprising only of 2 dense layers. We have some results related to this as shown below.



Probability of Original Class

Probability of Target Class

Timeline

MNIST with Convolutional Network	**25 September**
CIFAR-10 with Convolutional Network	**5 October**
ImageNet with Convolutional Network	**1 November**