**Abstract**:

Deep learning algorithms are being used today in most real-time applications and they are found to perform extremely well with best accuracy and performance as never before. With its rapid growth, it’s critical to ensure the Security and reliability of such Algorithms. However,like Machine learning algorithms, Deep learning too succumbs to the class of Adversarial attacks.

Schezdy et al showed that they are vulnerable to Adversarial samples which are specially created and perturbed data which aims to fool or wrong the algorithm. This poses a serious problem because these perturbed data are perceived benign by Humans and so it escapes inside the mysterious Black box of Neural Networks.

Adversarial attacks have been implemented in real-time and it enforces its practicality and importance.Hence,Adversarial attacks and the ways to defend or make the model roboust towards these attacks have been a increasingly important research topic with a surge of work going on both sides.

It is seen as an whack-a-mole or cat and mouse game,as Attacks and defenses are introduced to combat each other. But unfortunately,the attacks have had a upper hand till now and the nature of adversarial attacks innately make it easier to create adversaries than to defend it.

In this paper, we take an Image Classification problem of classifying Retinal fundus images into 2 classes based on the presence of the disease Diabetes Retinopathy(DR).

Importance of the prob:

A CNN Deep learning model is created to classify the images using state-of-art image classification methods and its vulnerability towards various existing Adversarial attacks are studied. The attacks considered can be grouped into 2 classes,

1)Optimization based -L-BFGS ,Carlini and Wagner

2) Gradient based -Fast Gradient Sign method(FGSM)

The success rate of the attacks and the trade-off between producing less distorted and more confident adversaries is studied.Comparisons based on the Confidence level,distortion produced,time taken, effect on the model, are made between the attacks.

To test the success of attacks, defense mechanisms are used to improve the robustness of the model and the attacks are tested on the robust model. Again 2 classes of defense mechanisms are tried,

1)Adversarial Training

2)Defensive distillation -Gradient masking

The results about the defenses and their reliability against the attacks are tested and also an attempt is made to enforce the best direction of these 2 classes to increase the robustness of the models.

Finally, Ideas to improve the robustness of the model, GANs are studied which we see as potential topics to counteract the currently attack-dominated Deep learning world.

**Introduction:**

With advancements in computation power, the usage of Deep Learning to perform tasks such as Natural Language Processing, Virtual Assistants, pixel restoration, visual recognition etc., has increased manifold. But each innovation comes with its own threats. Such a security threat to the existing deep learning algorithms has been discovered by the researchers. Such threats are called the adversarial attacks. Often such attacks are so small that they are imperceptible. Still, they can easily fool deep-learning algorithms to misclassify such models with high confidence.

Adversarial attacks:

Adversarial attacks are malicious attempts to perturb a datapoint x0 to x where x, x0 ∈ R such that x may or may not belong to a certain defined class.

For example:

x0 is a feature vector of Class C1

and

x is a feature vector of Class C2 (after perturbation).

The original datapoint belonging to C1 is misclassified to be of class C2 after perturbation. Here, if the target class C2 is specified beforehand explicitly, then such adversarial attacks are called targeted adversarial attacks. If they aren’t specified, i.e., the purpose is only to push the datapoint away from C1, then the attacks are called untargeted adversarial attacks. Untargeted attacks are strictly less powerful than the targeted ones.

Types of adversarial attacks based on the knowledge of adversaries:

White-box attack:

In this attack, the adversaries are assumed to have full knowledge of the algorithm, parameters of the model and architecture and can craft adversarial samples on the target model by any means.

Gray-box attack:

In this attack, the adversaries are assumed to have the knowledge of only the structure of the target model.

Black-box attack:

In this attack, the adversaries can depend only on query method to generate adversarial samples.

**Diabetic Retinopathy:**

Diabetic Retinopathy is a complication of diabetes mellitus, which causes lesions on the Retina and is the leading cause of blindness in developed countries. It causes the blood vessels in the retina to swell and to leak fluids and blood. It affects 80% of the people who have diabetes for more than 20 years. Treatment of Diabetic Retinopathy is the early stage reduces the risk of blindness. During a comprehensive dilated eye exam, a special kind of dilation is done to dilate the pupils which allows the ophthalmologists to examine the back of the eye clearly. The ophthalmologist manually looks for abnormalities in the retina, but such a process is time-, cost- and effort-consuming and can also lead to misdiagnosis. Now, with increased computational power and the increase in the usage of deep – learning models, better performance has been achieved in the medical diagnosis field.

DR is detected by the appearance of different kinds of lesions on the retinal fundus image. The different types are Microaneurysms (MA), haemorrhages (HM), soft and hard exudates (EX).

Microaneurysms (MA) is the earliest sign of DR and it appears as small red round dots on the retina due to the weakness of vessel’s walls. The size is less than 125 μm with sharp margin.

Haemorrhages (HM) appear as larger spots on the retina, where its size is greater than 125 μm with irregular margins. There are 2 types of HM, which are flame (superficial) and blot (deeper).

Exudate is fluid that leaks out of blood vessels into nearby tissues. The fluid is made of cells, proteins, and solid materials. Exudate may ooze from cuts or from areas of infection or inflammation. It is also called pus.

**Background:**

Threat Model- Notion and explaining how Neural nets classify in general.

Transferability

Example attack on MNIST input:

Left to right, perturbation increases.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

Example on Imagenet dataset:(Cat to WaterOuzel)



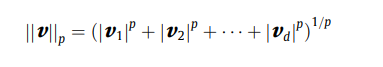
This visually imperceptible image was classified as Waterouzel with 99% confidence with the help of the Targeted CW attack(discussed below).

**How to measure perturbation?**

To estimate and measure the amount of perturbation on the input image, various distance metrics are proposed. This is used as a measure to quantify the similarity between two images.

Lp norm is used throughout the literature for this purpose. 3 commonly used metrics are L0, L1 and Linf norms. However,there is still an argument regarding the effectiveness of these metrics and the need for better distance metrics that quantify human perception more closely. Carlini et al mentioned about thi in their paper.

The distance between x and x’ is computed as ||x-x’||p where ||.||p is defined as



p is any real number and d is the dimension of the distance vector v.

Based on values of p, the norms are differentiated as,

1)L0 norm-

This norm measures the number of pixels modified in x’ wrt to x. Formally, it counts all i such that x’i not equal to xi.

This L0 distance metric is used to explain the defense method of defensive distillation by Papernot.

2)L2-

L2 norm measures the Euclidean distance between x’ and x. This norm can be low when many pixels are perturbed to a small amount, Initial attacks used this norm primarily.

Add formula

3)Linf-

It is the most popular and widely used metric which fixes a maximum amount that each pixel is allowed to change and there is no limit on the number of pixels that can change. It can be seen as the maximum change of any pixel.



Goodfellow et al argued ,this metric to be the most optimal and is used in most Gradient-based attacks.

**Deep Residual Neural Network Training:**

**Datasets:**

Kaggle - It consists of 35,126 retina scan images to detect diabetic retinopathy that have been resized into 224 x 224 pixels. The original dataset is available at [Diabetic Retinopathy Detection](https://www.kaggle.com/c/diabetic-retinopathy-detection).

IDRiD - It consists of 512 retina scan images captured at an Eye Clinic located in Nanded, Maharashtra, India. The images were resized into 224 x 224 pixels.

The datasets were originally divided based on the severity of Diabetic Retinopathy. The levels of DR along with its associated retinal findings are listed below. These classes were combined during training and the CNN model can classify samples into two classes only, namely, DR and No DR.

|  |  |
| --- | --- |
| **Diabetic Retinopathy Level** | **Retinal Findings** |
| Mild NPDR | Microaneurysms only |
| Moderate NPDR | MAs, Retinal Hemorrhages, Venous beading |
| Severe NPDR | Intraretinal hemorrhages, Definite venous beading, Prominent IRMA |
| Proliferative DR | Neovascularization, Preretinal hemorrhage |

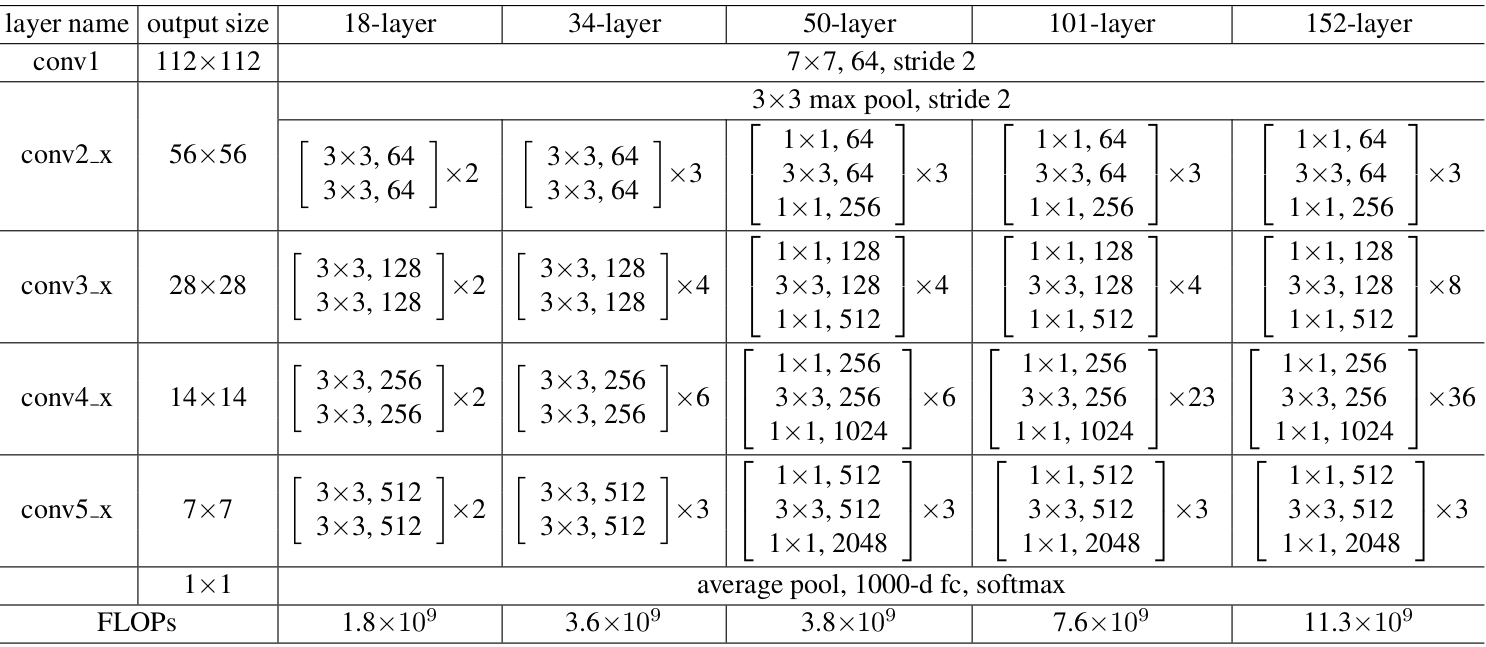
The aforementioned datasets were combined and manually class-balanced. This was done by truncating the excess amount of No DR samples from the dataset. The remaining samples were divided into training, validation and test sets as shown below.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Dataset** | **Number of Images** | **No DR** | **DR** | **Training Set** | **Validation Set** | **Test Set** | **Image Size** |
| Kaggle | 18966 | 9556 | 9400 | 14956 | 3720 | 290 | 224 x 224 |
| IDRiD | 554 | 184 | 370 | 444 | - | 110 | 4288 x 2848 |

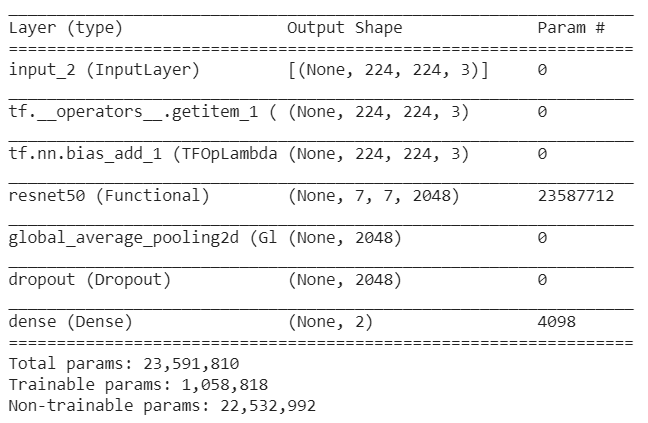
**Network Architecture**

The primary model of use was Resnet50, a variant of ResNet model with 48 Convolution layers, 1 MaxPool and 1 Average Pool layer. It had been widely used in medical transfer learning applications. The model was initialized with pretrained weights from ImageNet and then trained on our retinal fundus dataset i.e., transfer learning from ImageNet. The ImageNet dataset contains 14,197,122 annotated images and is used in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), a benchmark in image classification and object detection. ResNet was the winner of ILSVRC 2015.

The general architecture of the ResNet family is shown below.



The final network architecture after adding a global average pooling layer, dropout layer and the final classifying dense layer is shown below.



**Hyperparameters -**

Batch Size - A batch size of 16 was used for training since it provided better generalization and was small enough to fit in memory.

Learning Rate Schedule - A learning rate of 0.0001 was found to give the best results based on experimentation.

Loss function - Categorical cross entropy and binary cross entropy loss functions were both experimented with. In the end, all models were trained with categorical cross entropy loss function since it was decided that all training was to be done with the model acting as a multi-class classifier instead of a binary classifier to maintain uniformity.

Optimizer - Several optimizers were tested for stochastic gradient descent parameter updates namely Adam, RMSprop and SGD. No optimizer showed significant increase in performance over the other two. But it was found that Adam optimizer performed the best overall. The results are summarized below.

**Evaluation -**

The metrics used to evaluate the models are -

Mean Average Precision (MAP) - It is a measure of how relevant the predictions of a classifier are with respect to the ground truth labels.

Area Under the ROC Curve (AUROC) - It measures the expectation that a positive data point drawn at random is ranked higher than a negative datapoint drawn at random.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Optimizers** | **MAP** | **AUROC** | **MAP - finetune** | **AUROC - finetune** |
| Adam | 0.63 | 0.69 | 0.67 | 0.72 |
| RMSprop | 0.61 | 0.67 | 0.67 | 0.68 |
| SGD | 0.65 | 0.70 | 0.66 | 0.72 |

Other models tested -

Apart from Resnet, other pre-built models were trained on the retinal fundus dataset too. The results obtained are summarized below.

|  |  |  |
| --- | --- | --- |
| **Model Architecture** | **Accuracy** | **Accuracy/After fine-tune** |
| VGG16 | 68 | 69.5 |
| EfficientNetB0 | 64.5 | 69.7 |
| DenseNet121 | 73 | 71.2 |

**Adversarial Attacks:**

**Attacks based on Iterative Optimization of the Objective function:**

This class of attacks searches through the input space of the images to find an adversarial input with minimum perturbation. This is modeled as an Optimization problem where the aim is to optimize a chosen objective function over inputs, i.e the images.

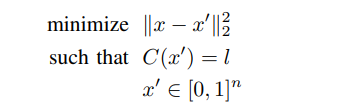
This allows to include additional criteria to be followed while finding adversaries inside the objective function which will be optimized together.

The first adversarial attack method proposed by Schezdy et al in 2014.follows this procedure.

**1)L-BFGS attack:**

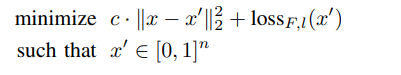
Given an input image x,the aim is to find an adversarial point x’ that minimises

L2(Euclidean distance) norm and also f(x’) not equal to y.(Change formula)



Here, x’ is chosen such that it is the closest image to x that is classified as l by the model f.

But this optimization problem is intractable and this problem can be hard to solve directly as the constraint is highly nonlinear. So the authors use a hybrid loss function and optimize over it.



The loss function computes the loss given input x’ with the target label l. Cross-entropy loss is one plausible loss function

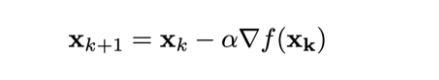
This problem is solved for multiple values of constant c and one-dimensional optimization,either line search or binary search is used to find the least constant c>0 that provides an adversarial input of minimum distance x’.

The Optimization algorithm used to solve this Limited Memory BFGS(L-BFSG) algorithm, which is a Quasi -Newton, 2nd order optimization method. BFGS algorithm imposes the closeness constraint which helps our goal of finding minimum norm x’. This gives an exact solution for convex functions, however, Neural nets are Non-convex in general and so an approximate solution is provided.

To impose the condition or bounds for perturbing the input x’, a box-constraint is used and x’ is found within this space. Hence the method used is a Box-constrained L-BFGS algorithm.

Due to Second order optimization process,which involves computing Hessain matrix,this iterative optimization problem is long and computationally expensive. But if the objective function is approximated as linear,in which case a simple gradient descent optimization can be used, is not efficient as it leads to very limited local information available at each step and so at each iteration only small steps(determined by the learning rate) should be taken,which can again make the process slow to converge to an optimal adversary.

Additionally, using second-order optimization computes the step size by itself and converges better than the first order case.But this method is still computationally expensive and Adversarial training was limited with L-BFGS for this reason.



Alpha-learning rate

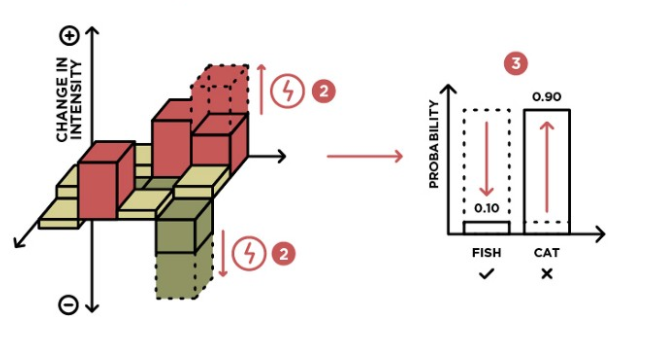
**Attacks by perturbing the input based on dL/dx**

**2)Fast Gradient Sign method(FGSM)**

Intuition: Perturb the input to maximally change the loss function of the model.

This is accomplished by finding the optimal direction in the input space where the loss function ascends/descends steeply and so pushing the inputs in that direction will find an input that is close to the given image and also is misclassified.

Gradient descent can be used to find the steepest direction where loss minimizes, by following the gradient of the loss with respect to input( usually done with weights, during training).We then push the given input towards this direction by a small amount (eps) which determines the Linf norm. Therefore eps specifies the maximum perturbation allowed for any single pixel. As it doesn’t account for the number of pixels changed, even perturbing all the pixels within the eps range will account for minimal Linf norm and also gives a reasonable push in the gradient direction. It finds the amount to perturb for each pixel and changes them simultaneously.



If the attack is Untargeted, then the goal is to increase loss and so the perturbation is added to input x, whereas if it is targeted, we reduce the loss with target and so perturbation is subtracted to push the input in opposite direction.

Formally,



The gradient of loss function J as a function of model parameters.is computed wrt to input x using Backpropagation, and its sign is used to identify the direction to move the input. It is multiplied by eps, which is Linf norm and specifies the maximum perturbation allowed in any chosen direction. Thus eta specifies the perturbation or noise which must be added or subtracted to the original image, to push it in the desired direction.



Thus this is a one step update algorithm and is known as the Fast Gradient Sign method. This method was primarily designed to be fast and not efficient and so it creates less perturbed adversarial examples quickly,which can be used for purposes like Adversarial Training.

Why does this method work?

We observed that simple gradient descent was one dimensional and it used very little local information and also was linear. This brings us back to the question of why Deep learning models are vulnerable to Adversarial perturbations. Schezdy et al and earlier papers assumed the overly non-linear nature and overfitting of the model is the reason. But Goodfellow et al showed that it is the opposite and the Linear nature of the Neural networks is the cause.

To add value to this Linearity Hypothesis, they found that adversarial examples which were initially thought to be in random pockets spread in the input space, was actually a continuous subspace. This nature can be explained with the argument that Neural networks are excessively linear and so it classifies data that are far from decision boundary with confidence even though it hasn’t seen it before. These subspaces that are victim to the linear decision boundary serve as places to find adversarial inputs.

(Add image)

To argue Neural nets are linear despite their Non-linear structure and activations, Goodfellow et al argue that Optimization functions used like ReLU, Maxout are nearly linear and even Sigmoid is mostly operated around the linear region. Therefore, Neural nets are seen as piecewise linear with not many linear units. This surprising result comes from the fact that Neural nets are made linear to make them easy to Optimize over, and strong non-linear networks like RBF are seen to resist these properties.

The relation between the inputs and logits is almost linear as opposed to between the parameters/weights and the outputs. This makes optimizing over the input space easier to solve using gradient descent.

How one step update is efficient?

Goodfellow et al, also noted that the direction of perturbation found using gradient descent is more important than the specific points in space. This was confirmed by plotting 2D maps of the input space and perturbation towards that direction irrespective of the starting point moves into adversarial subspace. The amount of pushing is determined by epsilon, but due to the high dimensionality of the perturbation(Linf) and adversarial subspace, every reasonable perturbation gets into the adversarial subspace easily.

The high dimensionality of subspaces also supports the Transferability property which allows attacks to be used on other models, which is discussed in detail in a later section.

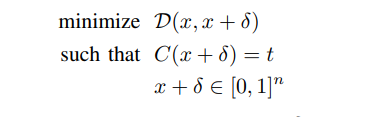
This idea of Gradient-based attacks has been improved by doing this iteratively and methods like BIM(Iterative FGSM), Kurakin, and R+FGSM were proposed. Projected Gradient descent which clips the gradient, is shown to produce the worst-case adversaries among the first order Linf attacks and so it is seen as the Universal first-order loss.It is also currently in 3rd place in Madrylab Whitebox attacks leaderboard

[MadryLab/mnist\_challenge: A challenge to explore adversarial robustness of neural networks on MNIST. (github.com)](https://github.com/MadryLab/mnist_challenge)

**3)Carlini-Wagner Attack:**

This attack belongs to the Optimization class, and it modifies the L-BFGS method, by using a custom objective function instead of the standard cross-entropy loss. The authors Carlini et al compared the efficiency of various objective functions to test their efficiency(higher confidence) and effectiveness(lower perturbation and cost).

Similar to L-BFGS, they formulate the problem as follows, given input x and target class t,

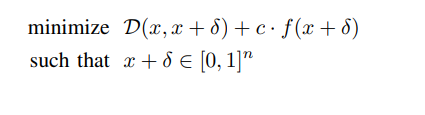


here, D is the Lp distance metric and C is the model classifier.Again, as this problem is hard to solve directly, they use another custom function f(x+del) such that



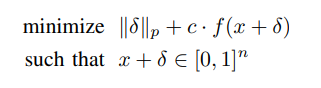
Informally, f(x+del) goes less than 0, only when the model classifies the image as t.

Now, the optimization function is reduced to,



Where ‘c’ is a suitably chosen constant >0, and there exists a value of c such that these two formulations give identical results.

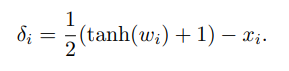
Here D is the Lp distance metric and the goal is to find optimal perturbation del, such that it is the minimum perturbation that classifies the input as t.



Constant c is chosen as the smallest number which gives x’ such that f(x’)<=0 and it is done using a modified binary search.

Box-constraint:

L-BFGS inherently included the constraint x+del in its box-constrained variation, but to allow other optimization algorithms to hold this constraint, the authors suggest three methods, Projected gradient descent, Clipped gradient descent which has problems for complex iterative functions and might get stuck in a local minimum. To overcome these, they suggest a change of variable method, using tanh function.



It introduces a new variable w (in tanh space) and instead of optimizing over del, now the problem optimizes over w. Since tanh lies between -1 and 1, this makes the condition x+del between 0 to 1, hold.

This method is seen as smoothing of clipped gradient, to solve the problem of getting stuck in a local minimum.

Adam optimizer is used along with this changed variable w to impose the box constraint. The authors found that the standard cross-entropy loss function was a poor choice for the objective function, as it uses a large value of ‘c’ initially which resulted in overly greedy gradient steps and prevented the attack from finding the optimal adversary.

From the seven objective functions they tested, they found the following function to be the most optimal



Here, Z(.)i denotes the logits for class i.

The intuition behind this function is that it optimizes for the distance between class t and the most likely class other than t. For targeted attacks, this minimizes the gap between the highest class logit(i) and target class(t).

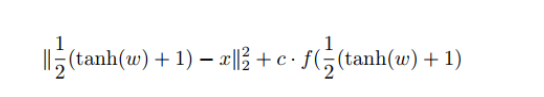
To control the optimization, they introduced a constant k. The optimization will stop when the logit difference between t and runner-up class is at most k. Therefore, to get highly confident adversaries, a large value of k should be used.

Formally this modified objective function is,



L2:

Putting these ideas together, the problem is formulated as, given input x, and a target class t, we search over and find optimal w , that minimizes



Where f() is the best objective function found.

The attack was highly successful compared to previous attacks of its time and produced effective and minimum distorted adversaries than L-BFGS,JSMA and Deepfool.

This attack also proved to break most of the earlier proposed defenses, most importantly Defensive distillation, which was the best attack then. The idea that the authors enforced is that Gradient masking or smoothing of gradients which most of those failed attacks followed, isn’t the right direction and showed that it can be easily circumvented through their attacks. They also argued that stronger attacks like those proposed by them should be used to evaluate the robustness of the defense and the model.

**Adversarial Defenses:**

Will there be a defense?

The **Universality theorem** by Kurt Hoknik, states that, with enough Neurons and enough training points, one can approximate any continuous function with arbitrary precision. This property of Neural nets enforces that, we can make the Neural nets learn these adversaries too, but the cost and practical implications are unknown. This is better than for Linear models, which Goodfellow et al,proved can’t resist the adversarial attacks as they can’t model Non-linear functions alike Neural nets. So, Deep learning has the possibility of defenses.

How to defend?

It was seen that attacks use a framework that exploits gradients computed to estimate the sensitivity of networks to their input dimensions.

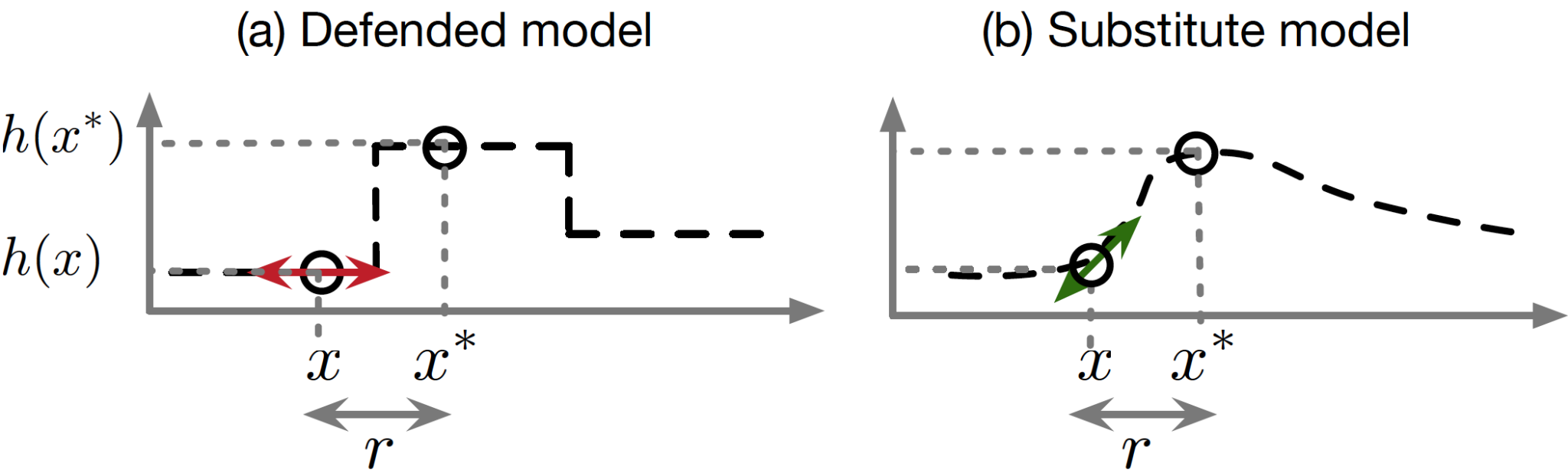
According to Paper not et al, it can be seen as two steps,

1. Evaluate the sensitivity of class change to each input feature to find the most sensitive direction
2. Use the sensitivity information to select an effective perturbation del x among the input dimensions.

As an example, the FGSM method computes gradients and uses the loss function with the target label to find the sensitive direction.

Therefore, to circumvent this we need to prevent the attacker from measuring gradients and this is known as Gradient masking or smoothing. Therefore, a probable defense method should start to smooth the gradients.

Also, Carlini Wagner mentioned that the defense methods should also try to break the transferability property to provide general robustness.

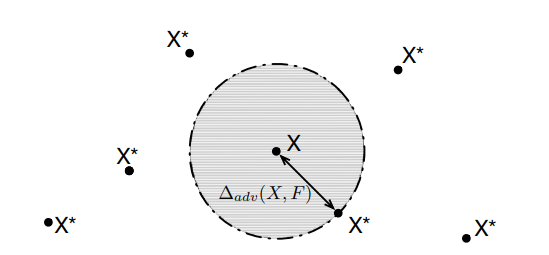


Gradient masking(it is hard to compute the gradients in the defended model on left)

How to measure the Robustness of a model?

Papernot et al explain ways to measure the robustness of a model and the requirements that a defense method should satisfy.

The defense method should smooth the classifier function F and make it classify inputs consistently around a given sample, within a radius. i.e. it should ensure the right classification in a closed neighborhood, which means perturbations within this radius will not be misclassified.



Here, X is the sample considered and all samples within the circle shown should be classifed the same as X. Adversaries(X\*) if any lies only outside of this circle.

The higher the perturbation needed to misclassify a sample, the more robust the model is.

Defense requirements:

1)Low impact on architecture- the defense method shouldn’t manipulate the architecture, as it might be practically infeasible and time-consuming

2)Maintain accuracy-

Making the model robust towards Adversaries shouldn’t decrease its original accuracy, and few methods proposed earlier like L1,L2 regularization and weight decay didn’t satisfy this requirement.

3)Maintain speed of network-

The defense method won’t be useful practically if it consumes a lot of time and is computationally expensive. This is more important to defense methods applied in Test time after deployment.

**Defense methods**:

Various defense methods have been proposed, the first being Adversarial Training which still proves to be an efficient method that can be improved, another important method named Defensive distillation, was successful on attacks like FGSM, JSMA and opened up the ideas on Gradient masking.

**Defensive distillation**:

Hinton et al, introduced Knowledge distillation,a process to extract knowledge from one model and apply it to a compressed model and make it perform with the same accuracy.Papernot et al, exploited this process and modified it to act as a defense that met all the above requirements.

Distillation in short, is training a Teacher model by introducing a Temperature parameter T in the final Softmax activation, applied to logits.The softmax output of the teacher model(soft labels) is used for training another model with same Temperature T.

Intuition:

The intuition behind this is that the knowledge acquired by the model during training is encoded not only in its weights, but also in the output probaility vector,which holds the relative information about classes. Informally, in case of Hard labels, the model only gets to know that it belongs to one particular class and no information about its similarity with remaining classes. All samples belonging to a class are treated with the same weights,even though some samples might be less similar to that class. Soft labels, uses a vector of probabilities and the similarities of the given sample, with all classes, are learned by the model.This helps the model understand the relative difference between the classes.

Formally,

The knowledge extraction is controlled by the parameter T,and increasing T, rises all the probabilty values thereby reducing the relative difference between them. There are more large vlaues than single class with a higher probability. As T->infinity the value for each class approaches 1/N and it becomes a Uniform distribution.This can be seen as, it helps to generalize instead of becoming overly confident on a single class and avoids overfitting.

Also, it was found that Temperature T is inversely proportional to the gradients of the output and so increasing temperature made the gradients smaller, making it hard for the attacks to find sensitive directions as mentioned earlier.

Therefore, this method prevented attacks that relied on gradients like FGSM, JSMA successfully.The authors noted that increasing temperature reduced the success rate of the attacks and made the model more robust.

Drawbacks:

However, Carlini et al showed in their paper the drawbacks of defense methods that try to smooth the gradient and showed that gradient masking isn’t a reliable defense and can be broken. They also broke the distillation defense using their CW attack.

Is it hard to defend?

Attacking has an upper hand over defense here, as the attackers have many options to exploit. They can find methods to attack any given model, and it’s the job of the defense person to circumvent each and every attack. As an analogy it can be compared to Physical conflict, creating new nuclear bombs is easy as opposed to building a city that resists the attack.(from cleverhans blog).

According to Cleverhans blog,few other reasons for the dominance of attacks are

1”*Adversarial examples are hard to defend against because they require machine learning models to produce good outputs for every possible input”*

Even though Neural nets work on a large number of inputs, they can’t assure to work on every possible input and the input size is also very large.

2)”*Adversarial examples are hard to defend against because it is hard to construct a theoretical model of the adversarial example crafting process. Adversarial examples are solutions to an optimization problem that is non-linear and non-convex for many ML models, including neural networks.”*

We still don’t have theoretical tools to completely model these Optimization problems and so creating a universal or general defense mechanism is harder.

Carlini broke 7 out of 8 defenses accepted by ICLR 2018 and showed the vulnerability or weak nature of the defenses existing. Currently, there is no universal defense mechanism that provides defense against all adversarial examples.

Among the classes of defenses proposed, only Adversarial training satisfied the requirements but it is computationally expensive to be used practically and the search for new defense methods is on.

**Proposed Work:**

Attack framework:

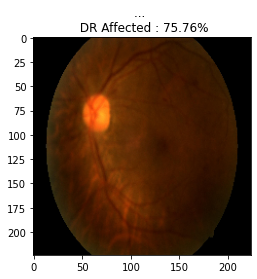
100 original retina image samples, that are rightly classified by the proposed model are used for implementing the attacks. As it is a Binary classification problem,only targeted attaks have been tested, and we believe untargeted attacks should also provide simialr results.

Whitebox attack is assumed,it is a reasonable assumption,as due to transferability property, unseen models can also be attacked similarly.The norm or distance metric is taken as l2 norm.

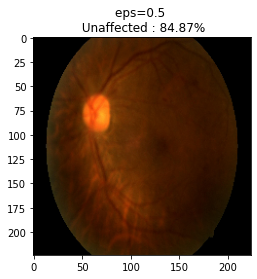
**1)L-BFGS**

Squared L2-norm is used as distance metric, and the constant c,which measures the relative importance of the norm and loss terms in the objective function,is computed using Binary search.

To control the perturbation, an input parameter epsilon is used. High values of epsilon give,large perturbation and confidence of adversaries is increased.It is the only input parameter to control the result of L-BFGS.



For this input image, L-BFGS failed with epsilon less than 0.5 and for epsilon=0.5,it gave an adversary with 84% confidence with norm 184.45.

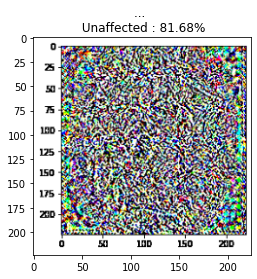


|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Eps | Success rate | Avg.conf | Max.conf | Avg.norm | Max.norm | Strong adversaries |
| 0.07 | 61% | 66.891% | 92.01% | 29.1 | 31.4 | 36 |
| 0.2 | 90% | 76.71% | 96.23% | 70.7 | 75.1 | 71 |
| 0.5 | 100% | 99.43% | 99.86% | 175 | 189.76 | 99 |
| 1 | 100% | 99.43% | 99.86% | 346.35 | 360.99 | 100 |

Neglecting norm constraint, optimizing only the loss function gave,

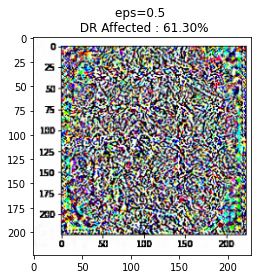
100% success rate with Average confidence of 99.67% and Average norm of 231.38.

Perturbation(created by FGSM to push towards Unaffected direction)

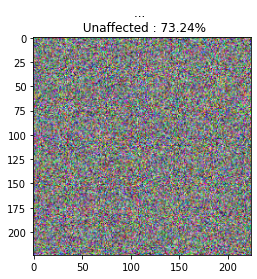


Interestingly,it was also classified as Unaffected with 81.68%.

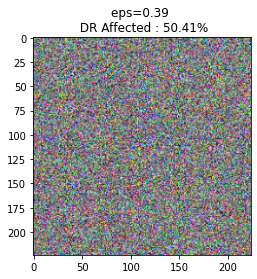
We tried applying L-BFGS method to this,and it needed a high epsilon value of 1 to misclassify it as DR affected with 60% confidence and very high norm of 350.5.So, now we have created a synthetic image sample of an unaffected retina.



The same experiment was repeated with random noise,



This image was easier to attack and it produced an adversarial image with lesser epsilon value of 0.39,with a lesser norm of 130.5



This observation signifies that, the perturbation noise added in FGSM is not just a random noise but carefully structured to push a benign sample towards Unaffected direction and so it captured the features of Unaffected image, which made it hard to misclassify it as seen above.

TODO:c value range

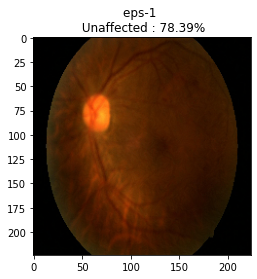
**2) FGSM**

The step size in the direction of gradient is controlled by epsilon,with smaller values of epsilon giving less distortion and adversaries with low confidence.

It was observed that, very high epsilon values also didn’t help,as it prevented the gradient descent to converge to an optimal perturbation and so success rate decreased for very high values of epsilon.(Started decreasing from epsilon=3)

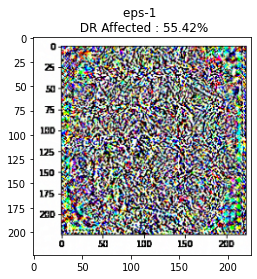
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Epsilon | Success rate | Max.conf | Avg.conf | Max.norm | Avg.norm | Strong adversaries |
| 0.1 | 85 | 69.78 | 94.37 | 1505.24 | 1505.32 | 64 |
| 0.25 | 95 | 83.54 | 98.78 | 96.9 | 97 | 71 |
| 0.5 | 99 | 91.62 | 99.9 | 193.98 | 193.98 | 86 |
| 1 | 99 | 94.77 | 99.98 | 387.97 | 388 | 92 |
| 2 | 99 | 95.12 | 99.97 | 775.95 | 776 | 92 |

The test sample given to FGSM,with epsilon=1,gave a successful adversarial image



The confidence is less than L-BFGS,and the norm is 387.97 which is high perturbation. It is generally inferred that FGSM is not an efficient attack,owing to its single step update and it is mainly used for its speed. These observations prove the results.

The noise sample was given to FGSM and the following was observed at epsilon=1



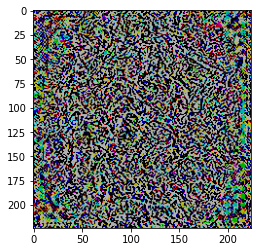
It successfully completes the attack on the Noise sample but the low confidence value signifies again that the orignal noise had the properties of unaffected retina image strongly as expected.

The Noise perturbation added to transform a DR affected sample to Unaffected sample for different epsilon values were recorded.

|  |  |  |  |
| --- | --- | --- | --- |
| eps=0.1 | eps=0.7 | eps=1 | eps=4 |
|  |  |  |  |

To validate the hypothesis given by Goodfellow et al, that only the pertubation direction is important, and is same for all samples of a given distribution, we added the perturbaiton obtained from one sample on a different sample,and tested if it behaved as an adversary.

Noise sample(used to push towards Unaffected,eps=1)



This Noise sample is added to a different benign sample that is classified as DR affected.



Adding noise with this benign sample, gave the following result,



The confidence is observed to be reduced, eventhough the added noise didn’t misclassify it has unaffected, it has pushed the sample towards Unaffected directiona and we believe a stronger noise sample will perform the misclassification too, proving the claim that the adversarial directions stay the same for all samples within a training set.

A random noise was given as input with a target label, to see the adversarial image created from the noise. This is also to prove that, the initial starting point isn’t important and perturbation in the right direction,will provide the target classification.

TODO:

1)Models easy to optimize,are easy to perturb

2)Visualise the optimal gradient direction for a sample and compare with other samples,visualise adversarial subspace.

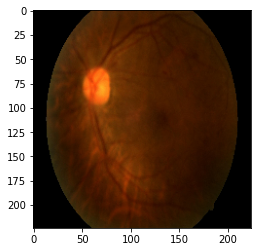
3)Find dimensioanlity or number orthogonal attack directions(plausible directions to perturb the input), the more the dimension, the easy it is to perturb.

4)Show that transferability property can be overcome by reducing the adversarial subspace dimension.

**3)Carlini Wagner attack**

This attack used Adam optimizer and optimized an objective function similar to L-BFGS. It was also computationally expensive and slower than previous attacks. The reasons for which are, the large number of iterations it takes each time and to find an optimal constant value ‘c’.

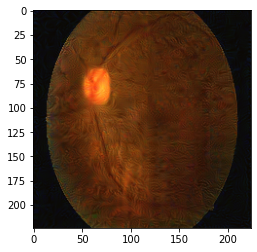
The attack produced adversaries with low distortion and the attack was designed to chose the least perturbed adversary that also misclassifies, so it produced low confidence adversaries as expected compared to other methods.



Confidence-56%, Norm=45

This method gave the minimum perturbation among the 3 attacks.

If we increase the number of iterations and k value(objective function) to increase the confidence, we get a highly perturbed image.



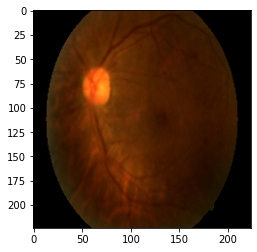
100% confidence

The perturbation of this image is visually perceptible.

Modified CW attack:

We modified CW attack to strike a balance between both and to choose an adversary with high confidence and the lowest possible perturbation with that confidence. So, the goal is shifted towards producing confident adversaries, but a check is made to choose an optimal perturbation.

This modified CW attack produced the following adversary.



77% confidence, norm=95.3

As expected it resulted in a more confident adversary and the perturbation also didn’t increase much.

Result:

CW attack (aims for minimum distortion)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Confidence | Success rate | Avg.conf | Max.conf | Avg.norm | Max.norm | Strong adversaries | Min.norm |
| 100 | 100% | 97.63% | 100% | 46.4 | 94.8 | 18% | 0.022 |
|  |  |  |  |  |  |  |  |

Modified CW attack(aims for maximum confidence)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Confidence | Success rate | Avg.conf | Max.conf | Avg.norm | Max.norm | Strong adversaries | Min.norm |
| 100 | 99% | 70.32% | 72.96% | 90.6 | 99.9 | 50% | 73.4 |
| 0 | 99% | 56.64 | 66.18 | 42.4 | 75.3 | 10% | 0.002 |

Observations:

1)Lower perturbations compared to other attacks and Success rate was also seen to consistently higher(almost 100%) for even random samples.

2)Lower confidence value, as the attack aims to choose lowest possible perturbation

3)Less number of stronger adversaries, this seems reasonable as the attack was producing lower confidence adversaries.

4)Modified CW attack gave higher confidence than the original at the cost twice the perturbation approximately.

Defense:

1)Adversarial Training

Framework:

Created Adversarial examples with FGSM and retrained the model with the correct labels. We observed that training with adversarial examples decreased the Success rate of attacks and decreased the confidence of the adversaries. However,it couldn’t still prevent attacks and misclassifications happened.The more number of sampels we trained,the lesser the confidence of the adversaries.

We also observed that Adversarial training increased the original accuracy of the model which proves the argument made by Goofellow et al that it is a good regularizaiton step better than Dropout.

Madry et al and Athalye et al showed that doing Adversarial training with PGD attacks is better(worst case exmaples) and is the only defese that met their claims.

2)Defensive distillation

**Conclusion:**

This paper aimed to demonstrate the applicability of adversarial attacks and their implications on an important Image classification task of classifying Retina images.This enforces that importance of making Neural networks robust and the need to combat these attacks,as Neural networks are relied on in critically important tasks as above.

We tried to test the results and arguments related to the proposed attacks and to get a general overview of the different properties of these attacks. We also briefly explained about the problems in each of the attacks and general ideas that were discussed to solve them.We tested particularly the arguments related to FGSM attack and verified the importance of Adversarial directions.

On the defense side, we tested a popular defense method, Defensive distillation and explained its vulnerability with Carlini attack as proposed in the papers.We also discussed general requirements, a framework for an ideal defense mechanism.

We discussed about abstract questions like Why Adversarial examples exist, Can it be defended and Why is it easier to attack than defend.

References:

1)Schezdy et al

2)Goodfellow et al

3)Carlini and Wagner et al

4)Papernot et al

5)Alex Madry et al

6)Athaleye et al

7)First paper

8)Retina paper

9)Universailty theorem paper

10)Hinton et al distillation

11)Cleverhans blog