Adversarial Examples for Eye-State Classification

Şefika EFEOĞLU

University of Potsdam

January 3, 2021

Motivation

- A robot named Chubby broke and hit the booth glasses without any instructions at Shenzhen Hi-tech Fair, 2016.
- Knightscope of Slicon Valley Robotics knocked down and injured a 16-month old boy in 2016.
- Uber autonomous test vehicle hit the 49-year-old woman and she died in 2018.

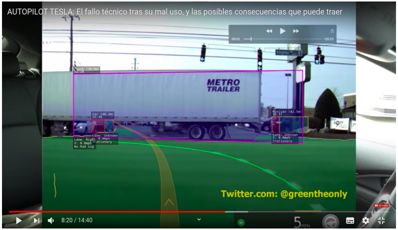
AUTOPILOT TESLA: El fallo técnico tras su mal uso, y las posibles consecuencias que puede traer
METRO TRAILER
Twitter.com: @greentheonly
8:20 / 14:40
5 MPH

Figure: Autonomous Driving problem of Tesla due to perturbation.

Motivation Continue..

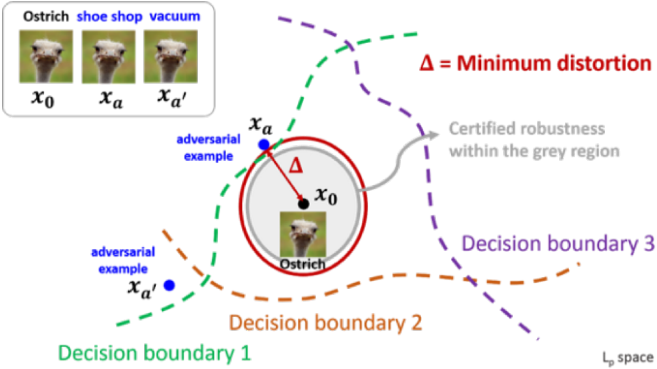

Figure: Adversarial Examples and Decision Boundary

Outlines

- Introduction
- Background Knowledge
- Project
- Conclusion

Introduction

- Deep Neural Networks achieve extreme accuracy on image classification tasks
- However, vulnerable to adversarial examples.
- Regularization is ineffective against to perturbation
- ***Approach***: regularization-based approach to adversarial examples using Parseval Networks and Adversarial Training.
- ***Objective* :** Improve the robustness of Eye-State Classifier using Adversarial Examples with Adversarial Training

Eye-State Dataset

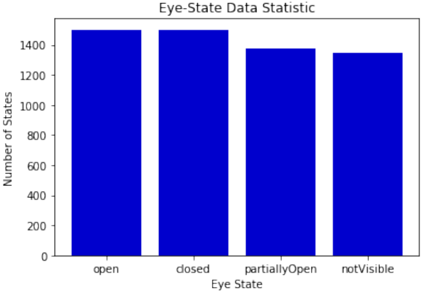

Figure: Eye-State data set consists of 5400 images, and the distribution of each eye state on the histogram

Background Knowledge

Road Map

1. Adversarial Examples
2. Fast Gradient Sign Method
3. Wide Residual Networks
4. Parseval Networks
5. Signal to Noise Ratio (SNR)

Adversarial Examples

1. specialised inputs created with the purpose of confusing a neural network
2. cause misclassification
3. fools the networks identifying a given input.

Types of adversarial attack

1. Blackbox attack
2. WhiteBox Attack
   - Fast Gradient Sign Method
   - Projected Gradient Descent
   - Deepfool etc..

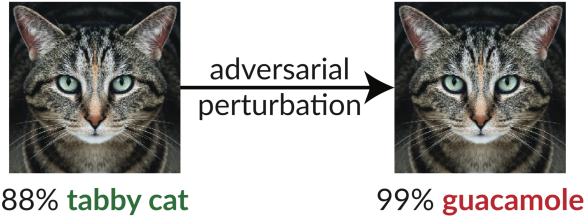

Figure: adversarial example of the cat image

Fast Gradient Sign Method

Definition

$$adv\_x = x + \epsilon \cdot sign(\bigtriangledown_x J(\theta, x, y))$$

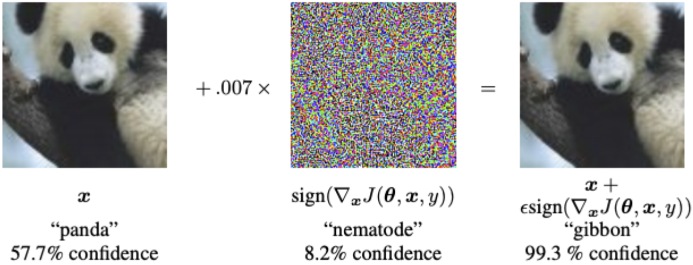
$\boldsymbol{x}$
"panda"
57.7% confidence
$+ .007 \times$
$\mathrm{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$=$
$\boldsymbol{x} + \epsilon \mathrm{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

Figure: the example of that how adversarial example of an image is obtained using Fast Gradient Sign Method

Wide Residual Network

Problem of Deep Neural Networks

- improving accuracy costs is expensive.
- training is a problem of diminishing feature reuse.
- very slow to train

Wide Residual Network

Problem of Deep Neural Networks

- improving accuracy costs is expensive.
- training is a problem of diminishing feature reuse.
- very slow to train

Proposed solutions

- decrease depth
- increase width of residual networks (Wide ResNet).

Parseval Networks

Objectives

Using the advantages of orthogonality and convexity constraints, improve the accuracy of the deep neural networks.
Additionally, it provides faster converges on learning curves.

Parseval Networks

Objectives

Using the advantages of orthogonality and convexity constraints, improve the accuracy of the deep neural networks.
Additionally, it provides faster converges on learning curves.

2 constraints below and parseval training are considered

- Orthogonality constraint
- Convexity constraint in aggregation layer
- Parseval Training

Orthogonality Constraint

Definition

- an optimization algorithm on the manifold of orthogonal matrices
- another name is Stiefel Manifold

$$R_\beta\left(W_k\right) \leftarrow \frac{\beta}{2}\left|\left|W_k^T W_k - I\right|\right|_2^2$$

  is expensive after each gradient update step.
- after every main gradient update, second update is applied to make the algorithm more efficient

Orthogonality Constraint

Definition

- an optimization algorithm on the manifold of orthogonal matrices
- another name is Stiefel Manifold

$$R_{\beta}\left(W_k\right) \leftarrow \frac{\beta}{2}\left|\left|W_k^T W_k - I\right|\right|_2^2$$

  is expensive after each gradient update step.
- after every main gradient update, second update is applied to make the algorithm more efficient

$$W_k \leftarrow (1+\beta)W_k - \beta W_k W_k^T W_k$$

Convexity Constraint in Aggregation Layer

Definition

- In Parseval Networks, aggregation layers output a convex combination of their inputs.
- To ensure that Lipschitz constant at the node n is such that

$$\Lambda_p^n \leq 1$$

euclidean projection is applied below

$$\alpha^* = \arg \min_{\gamma \in \Delta^{K-1}} ||\alpha - \gamma||_2^2$$

Parseval Training

Algorithm 1: Parseval Training

```
$\Theta = \{W_k, \alpha_k\}_K^{k=1}$ , $e \leftarrow 0$
while $\{e \leq E\}$ do
    Sample a minibatch $\{(x_i, y_i)\}_{i=1}^{B}$ .
    for $k \in \{1, ..., K\}$ do
        Compute the gradient ;
        $G_{W_k} \leftarrow \bigtriangledown_{W_k} l(\Theta, \{(x_i, y_i)\})$
        $G\alpha_k \leftarrow \bigtriangledown_{\alpha_k} l(\Theta, \{(x_i, y_i)\})$
        Update the parameters:
        $W_k \leftarrow W_k - \epsilon \cdot G_{w_k}$
        $\alpha_k \leftarrow \alpha_k - \epsilon \cdot G_{\alpha_k}$
        if hidden layer then
            Sample a set $S$ of rows of $W_k$
            Projection:
            $W_s \leftarrow (1 + \beta) W_s - \beta W_s W_s^T W_s$.
            $\alpha_k \leftarrow argmin_{\gamma \in \Delta^{K-1}} \left|\left|\alpha_{K-\gamma}\right|\right|_2^2$
        end
    end
    $e \leftarrow e + 1$
end
```

Wide Residual Network vs Parseval Network

Properties Name	Wide Residual Networks	Parseval Networks
Kernel Initializer	Gaussian	Orthogonal
Orthoganality Constraint	X	✓
Convexity constraint	X	✓

Table: shows that the properties of two different networks

Signal to Noise Ratio (SNR)

1. abbreviated as SNR or S/N)
2. a measure used in science and engineering
3. the ratio of useful information to false or irrelevant data in a conversation or exchange.

$$SNR(x, \delta_x) = 20 \log_{10} \frac{\|x\|_2}{\|\delta_x\|_2}$$

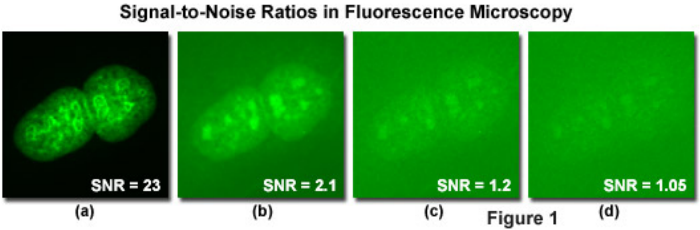

Figure: Example

Project

Methodologies

1. **Neural Network Models** : Convolutional Neural Network, Residual Network, and Parseval Network.
2. Train the models without Adversarial Examples.
3. Train the models using Adversarial Example with Adversarial Training algorithm.
4. Evaluate the models using **transferability** of Adversarial Examples.
5. Evaluate the effect of weight decay on the accuracy of CNN against adversarial examples.

Hyperparameter Tuning

Learning Rate: 0.1, 0.01
Regularization Penalty: 0.01, 0.001, 0.0001
Batch Size: 64, 128, 256
Epochs: 50, 100, 150

Model Name	**Width(k)**	**Accuracy**	**Loss**	**Recall**	**Precision**
Baseline of Simple ResNet	1	0.667830	0.942042	0.634336	0.650836
Baseline of Wide ResNet	2	0.656195	1.077292	0.597981	0.635004
WideResNet16-4	4	0.641070	1.374967	0.614458	0.668218

Table: shows the effect of width factor on deep neural networks which has 16 layers.

Hyperparameter Tuning-Box Plot for Model Loss

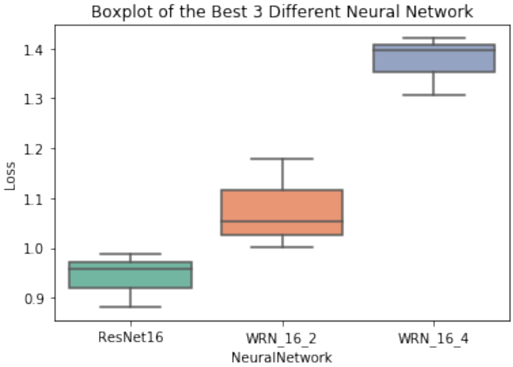

Figure: Model Selection with 3 Fold Cross Validation

Non-Adversarial Training Results

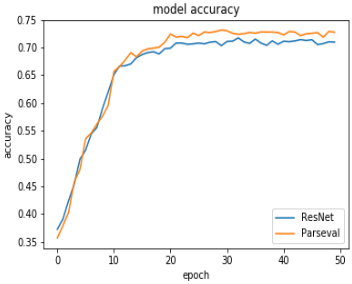

Figure: Simple ResNet and Parseval

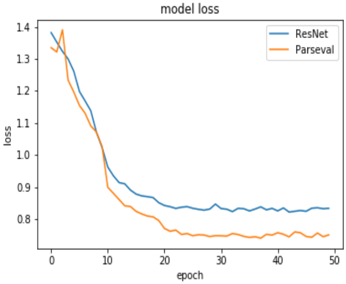

Figure: Simple ResNet and Parseval

Attack the Model with Different Noise Levels

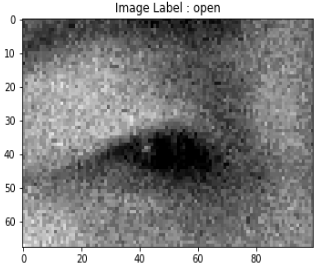

Figure: Label: Open

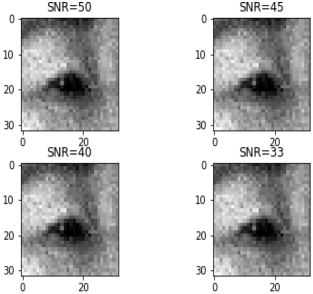

Figure: Label: Partly Open

Signal To Noise Ratio Results

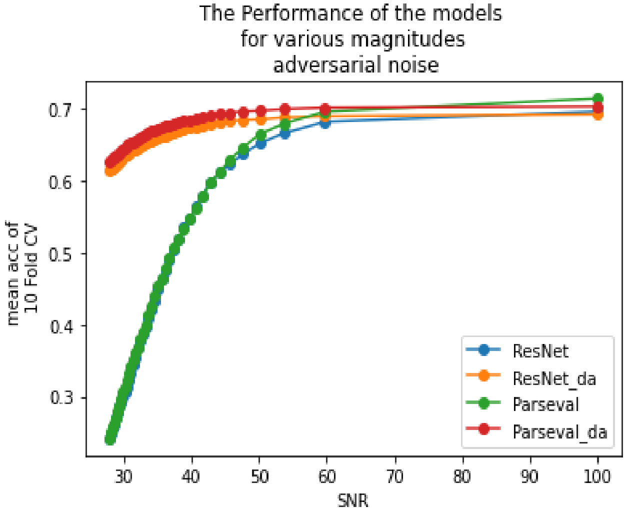

Figure: The accuracies of the models against different Signal to Noise Ratio (SNR)

Summary of SNR Results

Model Name // SNR	**Clean**	**50**	**45**	**40**	**33**
Parseval	0.714	0.665	0.629	0.562	0.4
ResNet	0.696	0.652	0.623	0.563	0.396
Parseval(Adversarial)	0.703	0.697	0.695	0.687	0.664
ResNet(Adversarial)	0.692	0.685	0.682	0.674	0.652

Table: The results show the mean accuracies of the models applied 10 fold CV against test dataset with different SNRs

Convolutional Neural Networks

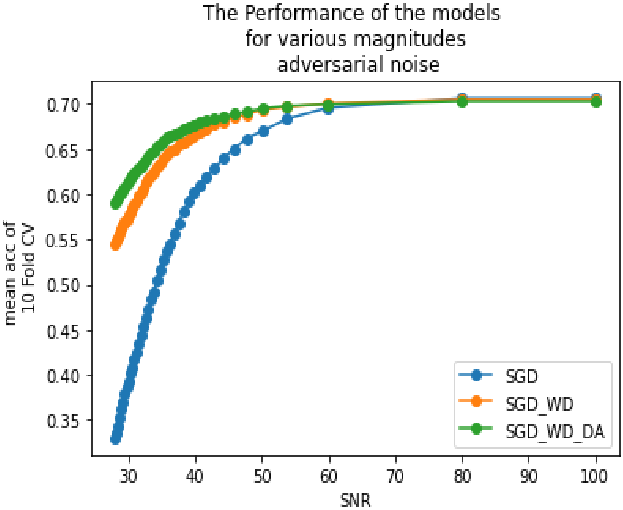

Figure: The accuracies of the Fully Connected models against different Signal to Noise Ratio (SNR). Weight decay(WD) = 0.0001.

Effect of Weight Decay on CNN

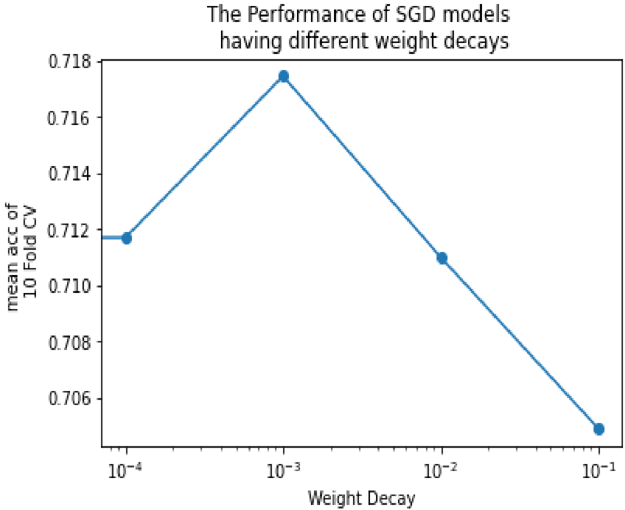

Figure: shows the effect of weight decay on model performance. L2 Regularization was used.

Summary of SNR Results for Convolutional Neural Networks

Model//SNR	**Clean**	**50**	**45**	**40**	**33**
SGD	0.706	0.67	0.65	0.602	0.472
SGD_WD	0.705	0.694	0.685	0.665	0.618
SGD_WD_DA	0.703	0.695	0.689	0.677	0.642

Table: shows the accuracies of fully connected models against the different SNR levels.

Conclusion

- Basic Residual Network is enough for this classification model.
- The model can be made smooth using adversarial training.
- Robustness was improved using adversarial training
- However, adversarial training is expensive.
- The result of SNR attacks to the models shows that Parseval Networks are more accurate than its vanilla corporate.
- CNN model Using weight decay outperformed CNN models without weight decay.
- Adversarial training approach outperformed CNN models with/without weight decay on experiments of adversarial examples.

Repository:
https://github.com/sefeoglu/adversarial_examples_parseval_net

Bibliography

1 Cisse, Bojanowski, Grave, Dauphin and Usunier, Parseval Networks: Improving Robustness to Adversarial Examples, 2017.

2 Zagoruyko and Komodakis, Wide Residual Networks, 2016.

3 Zhang, Jiliang, and Li, Chen, Adversarial Examples: Opportunities and Challenges,2018