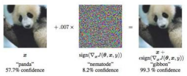
Encoder/Discriminator-Trained CNN for Adversarial Resistance

Anirudh Jain, Boyang Dun
{anirudhj, bodun}@stanford.edu

Motivation

- An **adversarial attack makes small perturbations to input images** that result in a highly confident misclassification by the neural network. There is a rapidly growing body of research on the development of adversary-resistant networks, and here we present our research into **encoding robustness into the network**
- Models with different architectures often misclassify the same adversarial examples, showing that adversarial examples expose **fundamental blind spots in our algorithms**. Thus, a **new architecture has to be developed** that takes into account adversarial attacks



- The **applications of this new network are significant** given the ubiquity of convolutional neural networks in computer vision applications and their **vulnerability to adversarial attacks**. Several applications such as self-driving cars and facial recognition are can be maliciously targeted. It is **essential that robust, adversarial-resistant networks be designed** and developed

Problem Definition

Design and train a robust convolutional neural network model to correctly classify both normal and adversarially generated images in the CIFAR-10 dataset

Solution

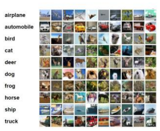
- We built an **adversarial-resistant convolutional network using a competing discriminator-encoder model**. The discriminator is trained to distinguish between intermediate hidden representations of real and adversarial examples while the classifier is trained to both correctly classify the data and fool the discriminator on adversarial examples

The purpose of this technique is to enforce an activation invariance across real and adversarial examples. This means the encoder successfully filters out adversarial noise, which leads to better classification on adversarial data

Data and Features

Dataset

- We used the **CIFAR-10 dataset**, which consists of 60,000 32x32 color images in 10 classes, with 6,000 images per class
- The dataset was split into **50,000 training images, 5,000 validation images, and 5,000 test images**. The test and validation sets contain 500 randomly selected images from each class



Adversarial Generation

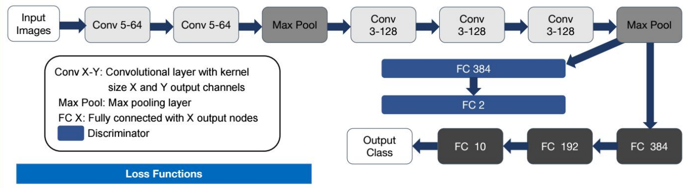
- The **fast gradient sign method (FGSM) attack** uses the sign of the gradient to determine which direction to change the corresponding pixel value. Given an input x and true label y, the perturbation δ is:

$$\delta = \epsilon * \text{sign}(\nabla_x J(x,y))$$

- We augmented our entire dataset by including a corresponding FGSM-generated adversarial image for each normal image in training, validation and testing sets (epsilon = 0.2)

Real vs Adversarial

Model and Loss Functions



Loss Functions

- **E(x)** represents the output of the encoder
- **D(x)** represents the output of the discriminator, which is the cross entropy of the output from its final layer

Discriminator Loss

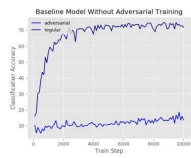
$$\mathcal{L}_{\theta_{disc}}(x, x^{adv}) = -\log P_{\theta_{disc}}(real|x) - \log P_{\theta_{disc}}(adv|x^{adv})$$

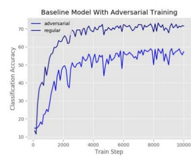
Encoder Loss

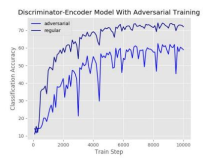
$$\mathcal{L}_{\theta_{class}}(x, x^{adv}, y) = -\alpha \log P_\theta(y|x) - (1-\alpha)\log P_\theta(y|x^{adv}) - \beta \log D(E(x^{adv}))$$

- The non-blue nodes comprise the **classifier**, which classifies images input into the network
- Depending on our settings, either output from the first or second max pooling layer is fed into the **discriminator**, which discriminates between real and adversarial images
- The portion of the classifier before the discriminator's insertion point is the **encoder**, which is simultaneously trained to fool the discriminator

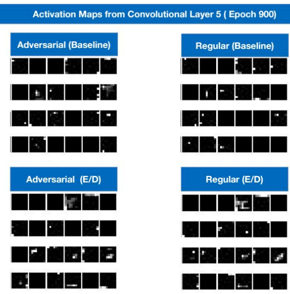
Results







Activation Maps from Convolutional Layer 5 (Epoch 900)



- **Validation accuracy** was reported on adversarial and regular examples over 100 epochs
- Baseline without adversarial training has **poor performance on adversarial examples**
- Models trained on adversarial examples perform **comparably on regular inputs** and **significantly better on adversarial inputs**

- The encoder/discriminator model enforces a greater **activation mapping invariance** than the adversarially-trained baseline
- This can be seen visually on the left, where activation mappings of a select image were taken from the last convolutional layer after the 900th training epoch
- The **upper row (baseline model) varies more across channels than than the lower row** (encoder/discriminator model)

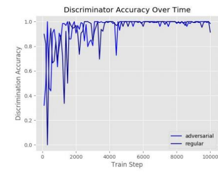
Analysis

Hyperparameter Search

- For our model, we **conducted a hyperparameter search** over learning rate, beta, optimization method, and discriminator insertion point
- The optimal hyperparameters were found to be a **learning rate of 1e-3, beta value of 0.1, RMSProp optimizer, and convolutional layer 5 as the discriminator insertion point**

Discriminator Analysis

- The **classifier performed worse with discriminator inserted at conv. layer 2** than at conv. layer 5
- Discriminator steadily improved over time, which means it **out-trains the encoder**



Test Set Results

	Reg. Images	Adv. Images
Baseline w/o adv. training	76.23%	17.92%
Baseline w/ adv. training	72.93%	57.67%
Discr/Enc Model	73.84%	62.14%

- **Training with adversarially-generated images performed significantly better** than without
- Our model provided a **4.47% increase in adversarial accuracy** over the baseline with adversarial training and a **44.2% increase** over the baseline without adversarial training
- Similar accuracy on regular examples indicates that **adversarial training did not have a large impact on model performance**

The proposed model showed improvement in adversarial classification, demonstrating promise in maintaining an intermediate activation invariant

Future

Hyperparameters
Perform a more extensive hyperparameter search that involves the architecture of the convolutional network itself

Datasets
Test our model on more complicated datasets like CIFAR-100 and ImageNet

Discriminators
Investigate other discriminator architectures that may enforce a stronger output invariant from the encoder portion of the image classification network. Parameters to consider include number of layers, layer type (convolutional or not), etc

References

Carlini, N., Wagner, D. (2017). Towards Evaluating the Robustness of Neural Networks. *Security and Privacy, 2017 IEEE Symposium on.*
Erraqabi, A., Baratin, A., Bengio, Yoshua., Lacoste-Julien, S. (2018). A3T: Adversarially Augmented Adversarial Training. *Machine Deception Workshop*.
Goodfellow, I., Shlens, J., Szegedy, C. (2015). Explaining and Harnessing Adversarial Examples. *International Conference on Learning Representations.*
Mytrovtsiy, S. (2017). Cifar-10 CNN Implementation Using TensorFlow library. https://github.com/exelban/tensorflow-cifar-10