Encoder/Discriminator-Trained CNN For Adversarial Resistance

Anirudh Jain * **Boyang Dun** *

Abstract

We investigate the possibility of building an adversarial-resistant convolutional neural network using a competing discriminator-encoder model. The discriminator is trained to distinguish between intermediate hidden classifier 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 competition between the discriminator and encoder enforces an activation invariance within the classifier across real and adversarial examples, which helps the rest of the network make accurate predictions on a greater range of input.

1. Introduction

Recent research has indicated that deep neural networks are highly susceptible to attacks from adversarial example. These are visually imperceptible perturbations of regular images that are designed to fool classifiers into making incorrect guesses. There is a rapidly growing body of research on the development of adversary-resistant networks, and here we present our research into encoding adversarial resistance into a convolutional network. Recently, particular adversarial attacks have even been shown to be effective across several different network architectures, demonstrating their generalizability (Papernot et al., 2016). Thus, given the broad ramifications of the problem and the ease of designing ordinary adversarial attacks, it is essential that adversarial-resistant networks be designed and developed.

In our project, we build out an adversarial-resistant convolutional neural network and compare its effectiveness against several baselines. At a high level, our experimental model was inspired by the generator-discriminator competition within a GAN(Goodfellow et al.), except we replace the competing generator with an encoder. We hope that our particular approach would force the network to filter out adversarial noise so that it classifies adversarial examples at a higher accuracy. This concept is novel and has not been applied to a convolutional neural network or a dataset as complex and large as CIFAR-10. We show that the proposed model is able to encode some adversarial resistance compared to baseline models.

The applications of our robust 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 can be maliciously targeted. It is essential that robust, adversarial-resistant networks be designed and developed

2. Related Work

We first review recent work on adversarial examples and approaches to adversarial defense.

2.1. Adversarial Examples

A number of attack strategies to generate adversarial examples have been proposed in the white-box setting, where the adversary has full access to the classifier (Szegedy et al., 2013; Goodfellow I., 2015; Carlini, 2017; Moosavi-Dezfooli et al., 2016; Biggio et al., 2017; Papernot et al., 2016). Goodfellow et al. propose the fast gradient sign method (FGSM), which applies a first-order approximation of the loss function to construct adversarial samples. Formally, given an instance x, an adversary generates adversarial example $xA = x + \eta$ with $L\infty$ constraints in the untargeted attack setting as $n = sign(\nabla_x f(x, y))$, where 'f is the cross-entropy loss used to train the neural network f, and y represents the ground truth of x. Given the gradients of the network, this is a very fast and powerful attack that has been proven to work against a variety of architectures. Optimization based methods have also been proposed to optimize adversarial perturbation for targeted attacks while satisfying certain constraints (Carlini, 2017). Its goal is to minimize the objective function as $||\eta|| + \lambda f(xA, y)$. However, the optimization process is slow and can only optimize perturbation for one specific instance each time.

2.2. Adversarial Resistant Models

A popular approach to defend against adversarial noise is to augment the training dataset with adversarial examples (Szegedy et al., 2013; Goodfellow I., 2015; Moosavi-Dezfooli et al., 2016). Adversarial examples are generated

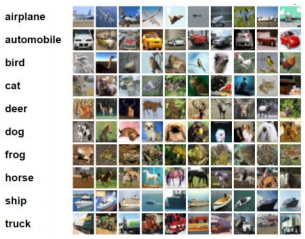
and added to the training set. This often results in increased robustness when the attack model used to generate the augmented training set is the same as that used by the attacker. However, this method can often cause overfitting to the adversarial images in the training set and can also detract from the overall classifier performance since the adversarial images can confuse the network by polluting the learned feature space.

The idea of augmenting a network with a discriminator to enforce hidden representation invariance is explored by Erraqabi et al. (Erraqabi, 2018) for adversarial resistance and Ganin et al. (Yaroslav G., 2016) on domain adaptation, where the network learns features that adapt to different domains for the same task. However, Erraqbi et.als approach was simplistic in that it only applied this concept to a three-layer fully connected network on MNIST with no hyperparameter searching or insertion of the discriminator at different points in the network. We were able to successfully apply this concept to a much more complicated CNN network, expand to a larger and more diverse dataset, and also perform a hyperparameter search for the purposes of determining the optimal model. Metzen et al. (Met) also uses this concept in the context of detection where the discriminator, trained separately from the classifier, is used as a detector of adversarial attacks. However, in our approach, the discriminator is used as a competing network to force the encoder to maintain invariance in the hidden representation.

3. Dataset and Features

3.1. Dataset

We use the CIFAR-10 dataset (Krizhevsky), which consists of 60,000 32x32 color images divided into 10 classes, with 6,000 images per class. We split the dataset 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, and we sample from the training data one mini-batch at a time to train our model. Each minibatch contains 128 images, and we did no preprocessing of the CIFAR-10 dataset.



Representative images of the 10 classes from CIFAR-10

3.2. Adversarial Input Generation

The Fast Gradient Sign Method (FGSM) attack involves perturbing the input data. It uses the sign of the gradient at every pixel to determine the direction with which to change the corresponding pixel value. Given an input x and the true label y, the FGSM attack sets the perturbation δ to be:

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

We augment our entire dataset by including a corresponding FGSM-generated adversarial image for each normal image in training, validation and testing sets (ϵ = 0.2). Thus, each regular image has a corresponding adversarial image:

Real (Left) and corresponding FGSM-image (Right)

4. Methods

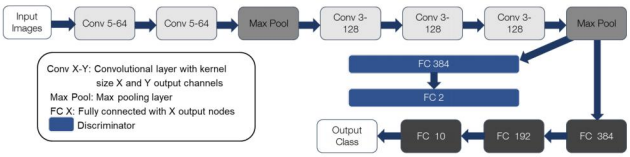
4.1. Model

At the highest level, our model is comprised of a classifier and a discriminator. Our regular CNN represents the classifier and attempts to classify images correctly. The discriminator, however, receives as its input an intermediate output of the classifier and, based on that information, attempts to predict if the original input image was adversarially-generated or real. The final part of our model is the encoder, which comprises the portion of the classifier that generates the output fed to the discriminator.The encoder has two simultaneous objectives – classify images accurately, and fool the discriminator.

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

More specifically, our optimal encoder is built from two convolutional layers, a max-pooling layer, three more convolutional layers, and another max-pooling layer (layers 1..7). The rest of our optimal classifier contains three

fully-connected layers (layers 8..11). The output from the last layer of the encoder (hidden representation) is fed into the discriminator, comprised of two fully-connected layers where the final layer predicts whether the features came from a real or adversarial input. The model is illustrated below:



Encoder/Discriminator CNN

4.2. Encoder

The encoder has two simultaneous objectives: 1) Correctly classify the input image and 2) Fool the discriminator into classifying all images as real. The encoder is then competing with the discriminator to create a hidden representation that is invariant regardless of whether the input image was adversarial or not (i.e. filtering out the perturbations). The loss of the encoder is given by:

$$L(x_{adv}) = -\beta log(D(E(x_{adv})))$$

where $E(x)$ represents the output of the encoder and $D(x)$ represents the output of the discriminator, which is the cross entropy of the output from its final layer. The encoder loss is not calculated separately but rather is included as part of the classifier loss with the parameter β.

4.3. Discriminator

Depending on the discriminator insertion point, either output from the first or second max pooling layer is fed into the discriminator (two fully connected layers) which discriminates between real and adversarial images. The loss of the discriminator is given by:

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

Thus, the goal of the discriminator is to correctly classify the input as adversarial or not.

4.4. Classifier

The classifier classifies images input into the network. The classifier takes into account both the classification loss and the encoder loss and balances them through α and β respectively. The classifier loss is:

$$L(x, x^{adv}, y) = -log(P(y|x)) - (1 - \\ \alpha)log(P(y|x^{adv})) - \beta log(D(E(x_{adv})))$$

For each labeled sample (x, y), the classifier outputs class probabilities P_θ, or cross-entropy.

4.5. Training

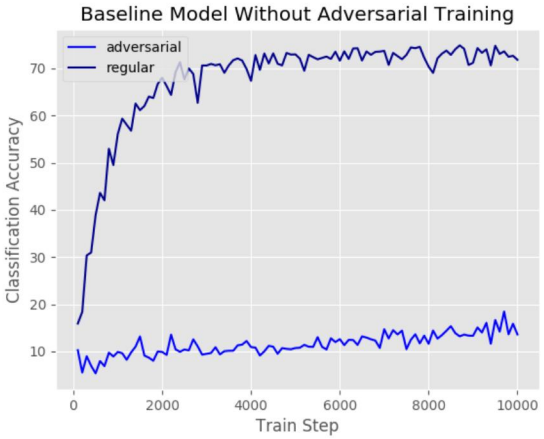
At each iteration, the network was given both a mini-batch of real data (x, y) and the corresponding mini-batch of adversarial examples (x^{adv}, y). The classifier and discriminator were trained simultaneously through backpropagation, and we determined the best optimization algorithm to be RMSPropagation.

5. Experiments and Results

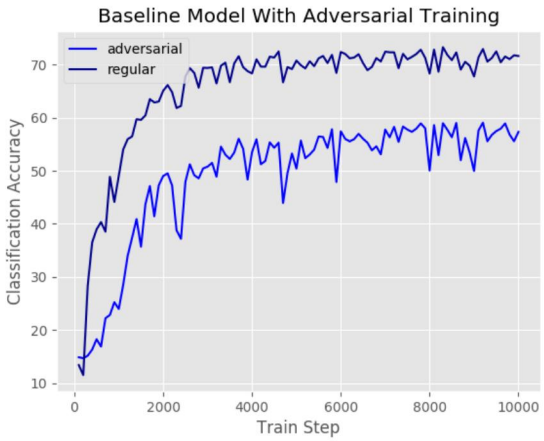
The hyperparameters that we searched over include learning rate, β (amount that discriminator accuracy impacts encoder loss), output layer in classifier that feeds into the discriminator, and train optimizer type. Our primary metric was accuracy because we were interested in seeing how often the model predicted the correct classification for both adversarial and regular examples. Given that CIFAR-10 has 10 classes

5.1. Baseline Models

Our first baseline model is the same CNN architecture trained on only real images. The following graph shows our best-performing baseline model's validation accuracy over 10,000 training steps. Its predictions are significantly less accurate on adversarially-generated images than real images.

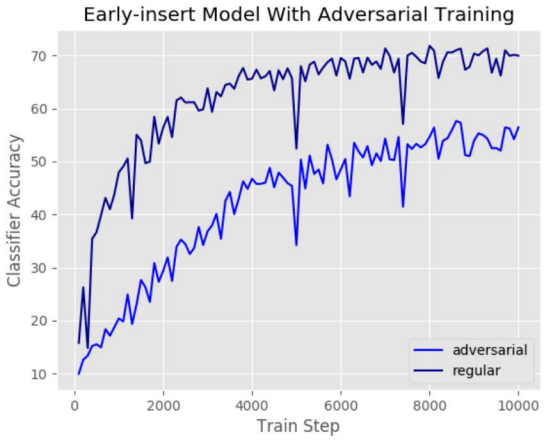


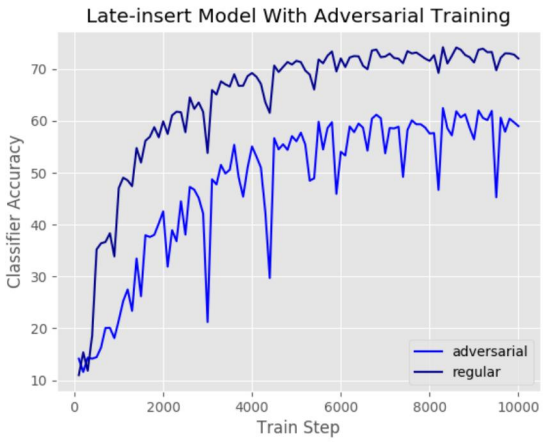
Our second baseline model is the architecture trained on both real and adversarially-generated images (i.e. adversarial augmented training similar to Goodfellow et al.). Since the model has been trained to recognize this specific set of adversarially-generated images, its validation accuracy on adversarial images is much higher than that of the first baseline even though their performance on regular images is comparable.



5.2. Encoder/Discriminator Model

The following two graphs show the best model performance we obtained by feeding the input of the first and second max-pooling layers respectively into the discriminator. The late-insertion model performed significantly better than the early-insertion model, so its corresponding test accuracy is what we report in the next comparison section.

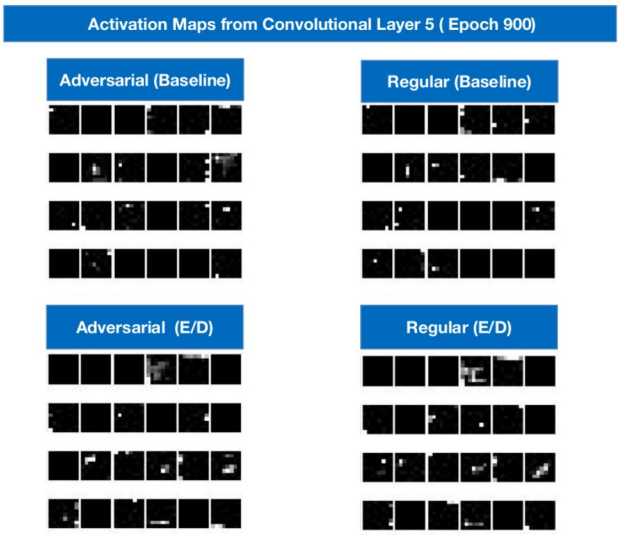




By manually going through 100 adversarial images that our experimental model misclassified, we determined that 64% of those images did not have a clearly defined class. By that, we mean that the image itself was perturbed to the point that even we couldn't classify the image properly. Correspondingly, that means approximately 36% of our error can still be eliminated if we assume human error approximates Bayesian error.

5.3. Comparison

The following representation of the activations output from convolutional layer 5 of the encoder visually show the difference in amount of hidden state invariance between the second baseline and the classifier co-trained with a discriminator. In the four activation maps, the sub-squares in the same relative locations correspond to the same convolutional output channel. We see that the adversarial and regular activations of the baseline are visually more distinct than the adversarial and regular activations of the encoder/discriminator model. Thus, the encoder/discriminator model enforces a greater activation mapping invariance than the adversarially-trained baseline.



Class Activation Maps from convolutional layer 5 (Epoch 900)

The following table reports the test accuracies for each of our best-performing baseline and discriminator/encoder models. There's a 39.75% improvement in adversarial image classification accuracy between the two baselines, and our model has a 4.47% improvement over the second baseline. Thus, training with adversarially-generated images performed significantly better than without. There's no significant variation in the performance of the three models on regular images indicating that adversarial training did not have a large impact on model performance.

	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%

Accuracy reported on regular and adversarial images from test set

Although not shown, our training accuracy on both adversarial and regular images was around 10 percent higher than validation accuracy. To mitigate this problem, we used L2-regularization in both baselines and experimental model.

6. Discussion and Conclusion

Our findings indicate that training the encoder against the discriminator moderately improves the accuracy of the convolutional neural network classifier on adversarial images. Moreover, its accuracy did not significantly change for regular images.

The hyperparameters that we determined to be most effective were $\beta = 0.1$, learning rate $= 1e{-}3$, RMS propagation as the optimizer, and convolutional layer 5 as the insertion point for the discriminator.

Based on varying the value of β, we also saw that classifier and discriminator performances are positively correlated on our convolutional CIFAR-10 network (Figure 1, Appendix). That means that if the encoder starts performing much better than the discriminator, our classification accuracy also goes down. Of the β values we searched over, we found that 0.1 allows the encoder to maintain somewhat of an output invariance without sacrificing classification accuracy.

Although we would have to do further experimentation to validate the following claim, the cause of the strong positive correlation between the discriminator and classifier performances may be due to the nature of the convolutional kernel, which comprises a single set of weights that must iterate across the entire image. Unlike a fully-connected kernel (Erraqabi, 2018), it cannot tune itself on only certain regions of the window, forcing it to maintain too great an output invariance that also leads to it losing valuable features necessary for accurate classification.

Overall, the proposed model showed improvement in adversarial classification, demonstrating promise in maintaining an intermediate activation invariant.

7. Future Work

We hope to build on our initial work by performing a more extensive hyperparameter search that involves the architecture of the convolutional network itself. We also plan on testing our model on larger and more complicated datasets like ImageNet to determine how generalizable this technique is to a greater range of input features. Finally, we hope to investigate other discriminator architectures that may enforce a stronger output invariant from the encoder portion of the image classification network. Discriminator parameters to consider include number of layers and layer type (convolutional or not).

8. Contributions

We believe all members contributed valuably and equally.
Boyang Dun - Network implementation, data collection
Anirudh Jain - Adversarial attack design, data collection

References

Biggio, Battista, Corona, Igino, Maiorca, Davide, Nelson, Blaine, Srndic, Nedim, Laskov, Pavel, Giacinto, Giorgio, and Roli, Fabio. Evasion attacks against machine learning at test time. *CoRR*, abs/1708.06131, 2017. URL http://arxiv.org/abs/1708.06131.

Carlini, N., Wagner D. Towards evaluating the robustness of neural networks. *Security and Privacy, 2017 IEEE Symposium on*, 2017.

Erraqabi, A., Baratin A. Bengio Yoshua. Lacoste-Julien S. A3t: Adversarially augmented adversarial training. *Machine Deception Workshop*, 2018.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative Adversarial Networks. *ArXiv e-prints*.

Goodfellow I., Shlens J., Szegedy C. Explaining and harnessing adversarial examples. *International Conference on Learning Representations*, 2015.

Krizhevsky, Alex. Learning Multiple Layers of Features from Tiny Images.

Moosavi-Dezfooli, Seyed-Mohsen, Fawzi, Alhussein, Fawzi, Omar, and Frossard, Pascal. Universal adversarial perturbations. *CoRR*, abs/1610.08401, 2016. URL http://arxiv.org/abs/1610.08401.

Papernot, Nicolas, McDaniel, Patrick D., Goodfellow, Ian J., Jha, Somesh, Celik, Z. Berkay, and Swami, Ananthram. Practical black-box attacks against deep learning systems using adversarial examples. *CoRR*, abs/1602.02697, 2016. URL http://arxiv.org/abs/1602.02697.

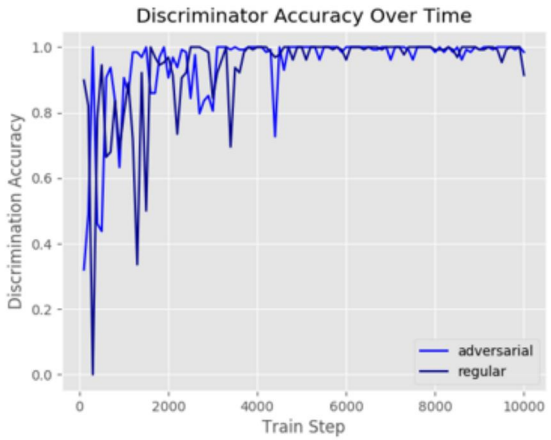
Szegedy, Christian, Zaremba, Wojciech, Sutskever, Ilya, Bruna, Joan, Erhan, Dumitru, Goodfellow, Ian J., and Fergus, Rob. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013. URL http://arxiv.org/abs/1312.6199.

Yaroslav G., Evgeniya U., Hana A. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 2016.

9. Code Repository

https://github.com/BoyDun/adversarial-train-defense

10. Appendix



Discriminator accuracy of best model on adversarial vs regular images