Adversarial Diversity and Hard Positive Generation

Andras Rozsa, Ethan M. Rudd, and Terrance E. Boult*
University of Colorado at Colorado Springs
Vision and Security Technology (VAST) Lab
{arozsa,erudd,tboult}@vast.uccs.edu

Abstract

State-of-the-art deep neural networks suffer from a fundamental problem – they misclassify adversarial examples formed by applying small perturbations to inputs. In this paper, we present a new psychometric perceptual adversarial similarity score (PASS) measure for quantifying adversarial images, introduce the notion of hard positive generation, and use a diverse set of adversarial perturbations – not just the closest ones – for data augmentation. We introduce a novel hot/cold approach for adversarial example generation, which provides multiple possible adversarial perturbations for every single image. The perturbations generated by our novel approach often correspond to semantically meaningful image structures, and allow greater flexibility to scale perturbation-amplitudes, which yields an increased diversity of adversarial images. We present adversarial images on several network topologies and datasets, including LeNet on the MNIST dataset, and GoogLeNet and ResidualNet on the ImageNet dataset. Finally, we demonstrate on LeNet and GoogLeNet that fine-tuning with a diverse set of hard positives improves the robustness of these networks compared to training with prior methods of generating adversarial images.

1 Introduction

Deep neural networks are powerful learning models which have been successfully applied to vision, speech and many other tasks [9, 11, 18, 19, 10, 20, 4, 3]. Szegedy et al. [21] showed that several machine learning models, including state-of-the-art deep neural networks, misclassify small non-random perturbations of correctly classified images. In many cases, these misclassifications are made with high confidence. Szegedy et al. [21] dubbed these perturbed misclassified samples *adversarial examples*. In order to generalize well, deep neural networks are expected to be robust to moderate perturbations to their inputs. Thus adversarial examples, with only small perturbations, are problematic.

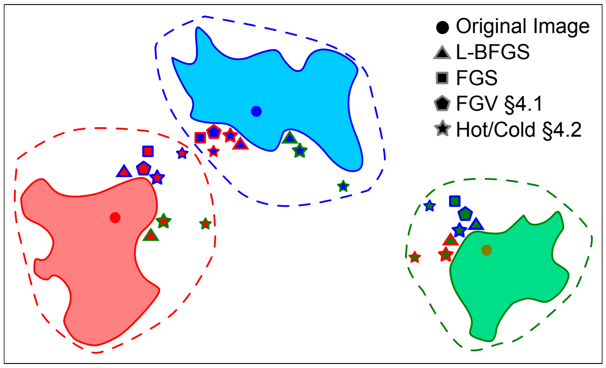


Figure 1: Adversarials and Hard Positives. *This paper demonstrates how to generate a much more diverse set of adversarial examples than the existing L-BFGS [21] or fast gradient sign (FGS) [8] methods. Via a range of perturbation amplitudes along the learnt adversarial directions – not just the closest adversarial sample – we can generate hard positives to fine-tune the class definitions, thereby extending previously overfit decision boundaries to improve both accuracy and robustness. The extended decision boundaries are represented by dashed lines. This simplified schematic uses shapes to depict different types of hard positive examples. Inner colors depict the original class, while outer colors depict the classification by the base network. For better visualization we show only one input image with corresponding adversarial/hard positive examples for each class.*

A deeper problem is that the generated adversarial images are relatively portable across different neural networks, which means that they are consistently misclassified by models of similar network architectures trained on varying training data, with different hyperparameters, or even different numbers of layers or types of activations [21]. While one classification error is a simple practical problem, these highly unexpected recognition errors suggest a more fundamental problem. Namely, the combinations of training samples and algorithms that we use to train our networks are not sufficient.

At their core, adversarial examples are nothing more than perturbed versions of ordinary examples that cause unexpected recognition mistakes in networks. This suggests that the adversarial problem can be addressed by finding

*This work supported in part by NSF#1320956 RI: Small: Open Vision

ways to efficiently augment the training set with representative adversarial examples which increase the diversity and thus generalization of the training set. Enhancing the training set is a common technique in machine learning and has been applied by others to deep neural networks. For example, Bengio et al. [2] demonstrated that deep networks benefit even more from out-of-distribution examples including perturbed versions of the original training examples.

In this paper, we make the following contributions:

1. We introduce a new measure to quantify *adversarial images* using a novel *Perceptual Adversarial Similarity Score* (PASS), which is more consistent with human perception than prior L_p norm measurements.
2. We introduce a new approach which is capable of generating large numbers of diverse adversarial images.
3. While researchers have focused on generating the closest adversarial images, we argue that to augment training sets and thereby improve accuracy and robustness of learning models, it is better to use *hard positives* formed by non-minimal perturbations. We demonstrate that these non-minimal hard positives are more effective for training than existing adversarial models.

2 Related Work

While not called *adversarial examples*, the oldest work related to our research is the use of perturbations and/or hard negative mining for training. Introducing perturbations to inputs to improve learning machines is a long standing method in machine learning, e.g., it was used when the MNIST dataset [12] was introduced and put forward as a training methodology in [17]. Research by Loosli et al. [14] took perturbation-enhanced training to a new level and developed an open source tool called InfiMNIST [14] that produces MNIST examples by applying small affine transformations and noise to the original images.

In detection problems, where the number of negatives can be enormous, a related technique is *hard negative mining* in which naturally occurring but "hard" and hence informative examples are used to improve training, e.g., [6].

Baluja et al. [1] proposed a related approach which generates perturbations and applies them to the inputs, observing how multiple trained networks respond to each input. Their method applies small affine image transformations without having any knowledge about the inner states of the networks. They used peer networks as a control-mechanism to filter out radical perturbations and to identify which perturbations are useful for retraining. However, random perturbations can be an inefficient technique to generate good training data, since a good system will correctly classify the vast majority of such inputs. We generated one million new examples with InfiMNIST and tested them on three differently trained LeNet [13] networks to identify adversarial examples among these images. The results show that the proportion of adversarial examples in the InfiMNIST dataset is very small – 2.199±0.132% – which means that finding the small affine transformations that form adversarial examples via this method is computationally non-trivial. Although capable of finding high-value training examples, these types of *guess and check* approaches, which perform random perturbations and determine if the results are misclassified, can be prohibitively expensive.

The idea of using optimization and internal network state to find adversarial examples in machine learning models was introduced in [21]. The authors also demonstrated the first method to reliably find those perturbations via small adjustments to pixel values. The approach relies on a box-constrained optimization technique (L-BFGS) which, starting from a randomly chosen direction, aims to find the smallest perturbation in the input space that causes the perturbed image to be classified as a predefined target label. Szegedy et al. [21] performed several experiments on a few varying networks and datasets, including MNIST [12], and demonstrated that the same adversarial example is often misclassified by different networks.

In [8], Goodfellow et al. presented a simpler and computationally cheaper algorithm to produce small perturbations causing unexpected recognition errors. Their approach – the fast gradient sign (FGS) method – creates perturbations by using the sign of the gradient of loss with respect to the input, and the required gradient can be effectively calculated using backpropagation. Experiments in the paper demonstrated that FGS reliably causes a wide variety of learning models to misclassify their perturbed inputs. It is important to note that while both L-BFGS and FGS use gradient information, FGS is much faster because the gradient is used only once as an explicit direction for a line-search, as opposed to L-BFGS, which performs many gradient computations. Stepping in the direction of the sign of the gradient of loss with respect to the input image continuously reduces the classification score of the original class until another class obtains a higher score. Assuming that the original classification was correct, this causes a classification error. The resulting perturbation images look like dense random noise. Rather than producing images for training, the paper suggests using an improved objective function which directly incorporates the sign of the gradient of loss. That model reduces the average classification error rate from 0.99% to 0.782% on MNIST.

A fundamentally different approach from [21, 8] was proposed by Sabour et al.[16]. The approach seeks to find not only adversarial images that are misclassified; it also seeks bounded error inputs that have internal representations which are closest to *guide images*. In their work, the authors demonstrate that adversarial examples can be produced which are not only incorrectly classified at the output layer, but are also close to any specified internal representations of the network. They use L-BFGS to find the adversarial images which mimic the internal representations of targeted images, also demonstrating that the adversarial problem is more complex than just mapping output errors.

In all research that uses explicitly optimized adversarial images to improve networks, the authors aim to synthesize the perturbations of smallest difference (by some measure) that cause misclassification. By contrast, we further amplify adversarial perturbations over a diverse range to obtain additional *hard positives* to retrain our networks.

3 PASS

Different measures have been used in the literature to quantify adversarial images; commonly L_p norms [21, 8, 1]. These works implicitly agree that *adversarial images* are modified inputs which cause unexpected recognition errors in machine learning models, yet *are correctly classified by humans*. Sabour et al. concluded that L_p measures are not well matched to human perception [16]. Consistent with [7], this suggests that adversarial images should be quantified in terms of *just noticeable difference*. However, a natural interpretation of *imperceptible* should also allow many transformations, including small translations and rotations, which result in images that are perturbed to noticeable extents compared to their original counterparts, yet still appear to be *plausible* samples of the same images – samples that a network should not get wrong. Consider a biometric face verification system under attack: viewpoint variations result in noticeably different images that a human operator still perceives as a different view of the same input. If the distortions are too large, e.g., a face with reversed aspect ratio or a very noisy image, a human operator would likely notice a clear problem with the data even if he or she could still discern the identity of the person in the image.

To quantify the degree to which an image is adversarial, we therefore seek a psychometric measure which considers not only element-wise similarity but also *plausibility* that the image in question is a different view of the same input. We perform this measurement as a two stage process: aligning the perturbed image with the original, then measuring similarity of the aligned images. Due to potential radiometric and noise differences between the images, simple correlation or feature-based alignments may not work very well. Instead, we maximize the *enhanced correlation coefficient* (ECC) [5] between the adversarial image $\tilde{x}$ and the original image x. Let $\psi(\tilde{x}, x)$ be a homography transform from adversarial image $\tilde{x}$ to the original image x, with 3×3 homography matrix $\mathbf{H}$. We optimize the objective

$$\underset{\mathbf{H}}{\arg\min} \left|\left| \frac{\overline{x}}{||\overline{x}||} - \frac{\overline{\psi(\tilde{x}, x)}}{||\overline{\psi(\tilde{x}, x)}||} \right|\right|, \tag{1}$$

where an overline $(\bar{\cdot})$ denotes the zero-mean version of an image. The x minimizing Eq. 1 will maximize $ECC(\psi(\tilde{x}, x), x)$.

The second stage of our comparison measures similarity. Previous work [21, 16] has predominantly quantified the degree of *adversarial* in terms of element-wise L_2 or L_∞ distances. However, these norms are extremely sensitive to small geometric distortions and do not map well to psychophysical notions of similarity, even under non-geometric perturbations; even after alignment, a single pixel-wise difference along a strong edge is all it takes to maximize the L_∞ distance between two very similar images.

Research by [22] suggests that the human visual system is most sensitive to changes in *structural patterns* and developed the structural similarity (SSIM) index as an objective for lossy image compression. SSIM quantifies similarity of image pairs based on structural and brightness differences.

Given two images, x and y, let $L(x, y)$, $C(x, y)$, and $S(x, y)$ be luminance, contrast, and structural measures, respectively defined as

$$L(x, y) = \left[\frac{2\mu_x\mu_y + C_1}{{\mu_x}^2 + {\mu_y}^2 + C_1} \right]$$

$$C(x, y) = \left[\frac{2\sigma_x\sigma_y + C_2}{{\sigma_x}^2 + {\sigma_y}^2 + C_2} \right]$$

$$S(x, y) = \left[\frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \right],$$

where μ_x, σ_x, and σ_{xy} are weighted mean, variance, and covariance respectively and C_i's are constants to prevent singularity. With these, the regional SSIM index (RSSIM) is defined as

$$\mathit{RSSIM}(x, y) = L(x, y)^\alpha C(x, y)^\beta S(x, y)^\gamma, \tag{2}$$

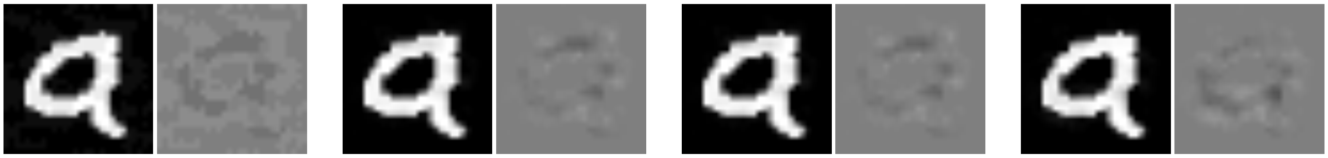
where α, β, and γ are chosen to reflect relative importance of luminance, contrast, and structure respectively. For consistency with [22], we set $\alpha = \beta = \gamma = 1$, and use an 11×11 kernel of $\sigma = 1.5$ for weights. SSIM is then obtained by taking the average of RSSIM over all pixels: for an m-pixel image,

$$\mathit{SSIM}(x, y) = \frac{1}{m} \sum_{n=1}^{m} \mathit{RSSIM}(x_n, y_n). \tag{3}$$

We combine the photometric-invariant homography transform alignment with SSIM to define the *perceptual adversarial similarity score* (PASS) between $\tilde{x}$ and x as

$$\mathit{PASS}(\tilde{x}, x) = \mathit{SSIM}(\psi(\tilde{x}, x), x). \tag{4}$$

PASS then serves as a similarity measure to quantify how *adversarial* a misclassified image is. While previous works quantified adversarial in terms of some similarity or dissimilarity measure, both Szegedy et al. [21] and Goodfellow et al. [8] mention a constraint which they implicitly assume but do not explicitly define: namely, in order to be adversarial perturbations must be *imperceptible*. Mathematical definitions purely in terms of L_p norms do not operationally enforce such a constraint; the perturbation of minimum L_p norm may be quite perceptible for certain images. Insofar as PASS serves as a psychometric, we can use it to make this constraint explicit. Let y be the label of x, let f be the



(a) Fast Gradient Sign
PASS=0.727, L_2=0.990, L_∞=13

(b) Fast Gradient Value
PASS=0.979, L_2=0.196, L_∞=22

(c) Hot/Cold - 1
PASS=0.977, L_2=0.198, L_∞=22

(d) Hot/Cold - 2
PASS=0.965, L_2=0.239, L_∞=24

Figure 2: ADVERSARIALS ON LENET/MNIST. *Adversarial examples and perturbations for an input image classified as digit 9 with metrics shown below images in form (PASS, L_2 norm per pixel, L_∞). (a) Fast Gradient Sign method: adversarial classified as 4 (b) Fast Gradient Value approach: adversarial classified as 4 (c) Hot/Cold approach with the most similar class: adversarial classified as 4 (d) Hot/Cold approach with the second most similar class: adversarial classified as 2. Perturbation-visualization: black=-1, white=+1, gray=no-change.*

classifier, and let $\tau \in [0,1]$ be a threshold on perceptible PASS values. Then an adversarial image is defined as

$$\underset{d(x,\tilde{x})}{\arg\min}\, \tilde{x} : f(\tilde{x}) \neq y \text{ and } \mathit{PASS}(x,\tilde{x}) \geq \tau, \qquad (5)$$

where $d(x,\tilde{x})$ is some dissimilarity measure, e.g., $1 - \mathit{PASS}(x,\tilde{x})$ or $||x - \tilde{x}||_p$ – potentially constrained along the directions learnt by an adversarial generation algorithm. Note that the value of τ may vary depending on the network and the dataset, but for any fixed domain this gives a quantitative adversarial threshold. Hard positives are similarly constrained by a *PASS* threshold, but need not be the samples of minimum dissimilarity.

4 Adversarial Example Generation

In this section, we first provide an overview of the notations that we use. We then discuss an existing method for generating adversarial examples. Finally, we introduce a novel method for adversarial image and hard positive generation and discuss implementation details.

Let θ be the parameters of the model, $x \in [0,255]^m$ the m-pixel image of integer values applied as input to the network, y the label of x, $J(\theta,x,y)$ the cost used to train the neural network, and f be the learnt classifier that maps input images to a discrete set of n labels. Let $B_l(\cdot)$ be the backpropagation operator defined in Sec. 4.3.

For a given input image x classified as y, our goal is to produce perturbation η such that perturbed image $\tilde{x} = x + \eta$ is adversarial according to Eq. 5. To generate hard positives, we simply scale η by a constant ≥ 1.

4.1 Fast Gradient Value

We commence our research using the fast gradient sign (FGS) method introduced in [8], but seek greater adversarial diversity. An obvious extension of FGS is to consider a scaled version of the raw gradient of loss instead of using only the sign of the gradient. We refer to this as fast gradient value (FGV), and we show that it produces notably different adversarial perturbations. Specifically, the direction determined by

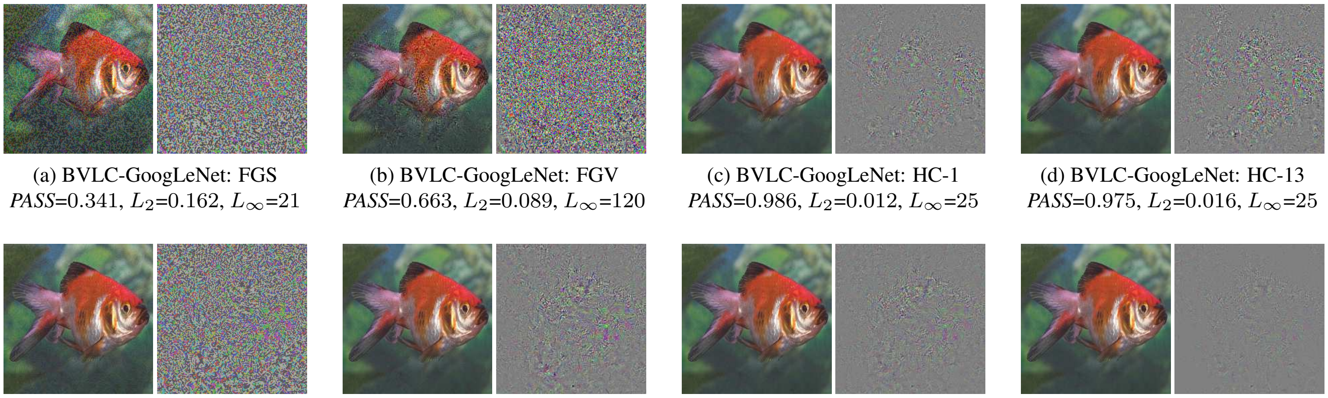
$$\eta_{grad} = \nabla_x J(\theta,x,y) \qquad (6)$$

does not ignore the differences in gradient magnitude between corresponding pixels as FGS does. The intuition here is that to effectively increase the loss with minimal distortion, pixels with larger gradients in terms of L_1 norm need to be changed more radically than others. Compared to FGS, this slight modification leads to different directions where significantly different adversarial examples exist.

As shown in Fig. 2(a) and Fig. 3(a), the adversarial perturbations produced by the FGS method cover almost the entire image, including the "background". By contrast, using raw gradient of loss, shown in Fig. 2(b) and Fig. 3(b), generates more focused perturbations that cause less structural damage resulting in higher PASS as we can see in Fig. 5. Still, FGS and FGV approaches combined produce only two adversarial examples per input image, and it is natural to ask if we can generate more.

4.2 Hot/Cold Approach

The intuition behind FGS and FGV is to decrease the response to the class of interest by following the direction of the gradient of loss. To generate hard positives which *augment* overfit decision boundaries between classes, however, it seems natural to consider derivatives with respect to other layers, and generate adversarial examples which change classifications *while moving toward a specific targeted class*. Our idea is related to [15], which introduced a method called *image inverting*, designed to reconstruct an image which minimizes the loss for given class represented by a *one-hot* feature vector in the penultimate layer. We postulated that inverting a one-hot vector – a vector with one non-zero (hot) element and the remaining elements zeros – at the penultimate layer would eventually create features in lower layers that are exclusive to the selected *hot class*, and other features responsible for neutralizing other classes represented by zeros in the penultimate layer (*non-hot classes*). We extend that concept by adding the "cold" class, to further decrease the role of the current class. Specifically, we craft features for the penultimate layer, the last layer before softmax. At that layer each value is still associated with a particular output class, so we can define directions in the



(a) BVLC-GoogLeNet: FGS
PASS=0.341, L_2=0.162, L_∞=21

(b) BVLC-GoogLeNet: FGV
PASS=0.663, L_2=0.089, L_∞=120

(c) BVLC-GoogLeNet: HC-1
PASS=0.986, L_2=0.012, L_∞=25

(d) BVLC-GoogLeNet: HC-13
PASS=0.975, L_2=0.016, L_∞=25

(e) ResNet-152: FGS
PASS=0.917, L_2=0.031, L_∞=4

(f) ResNet-152: FGV
PASS=0.983, L_2=0.012, L_∞=17

(g) ResNet-152: HC-1
PASS=0.992, L_2=0.008, L_∞=14

(h) ResNet-152: HC-13
PASS=0.997, L_2=0.004, L_∞=7

Figure 3: ADVERSARIAL IMAGES ON IMAGENET. *Adversarial examples and perturbations for a "goldfish" generated by BVLC-GoogLeNet and ResNet-152. Metrics are shown below images in form (PASS, L_2 norm per pixel, L_∞). (a) Fast Gradient Sign (FGS) method with BVLC-GoogLeNet: classified as "starfish", failed adversarial generation due to low PASS (b) Fast Gradient Value (FGV) approach with BVLC-GoogLeNet: classified as "loggerhead turtle", failed adversarial generation due to low PASS (c) Hot/Cold approach with the most similar class (HC-1) on BVLC-GoogLeNet: adversarial classified as "mud turtle" (d) Hot/Cold approach with the 13th most similar class (HC-13) on BVLC-GoogLeNet: adversarial classified as "box turtle" (e) Fast Gradient Sign (FGS) method with ResNet-152: adversarial classified as "cock" (f) Fast Gradient Value (FGV) approach with ResNet-152: adversarial classified as "blowfish" (g) Hot/Cold approach with the most similar class (HC-1) on ResNet-152: adversarial classified as "bee" (h) Hot/Cold approach with the 13th most similar class (HC-13) on ResNet-152: adversarial classified as "cock". For better visualization, perturbations are scaled by a factor of 10 with gray=no-change.*

input image space that help move toward a target (hot) class while moving away from the original (cold) class.

To formalize our *hot/cold* model, let $h(x) \in \mathbf{R}^n$ be the features of the neural network's penultimate layer for input image x, and let y be the label for x. We construct a hot/cold feature vector ω_{hc} based upon $h(x)$ as follows: First, we define a target class $\tilde{y} \neq y$ as hot, i.e., we add features exclusive to this class $\tilde{y}$ to input image by defining its corresponding value as $|h_{\tilde{y}}(x)|$. Second, we identify class y as cold with a value of $-h_y(x)$ as we intend to step away from it by removing its exclusive features from the input image. Thus the constructed penultimate layer feature vector is

$$\omega_{hc}(x) = \begin{cases} |h_j(x)| & \textit{if } j = \tilde{y} \\ -h_j(x) & \textit{if } j = y \\ 0 & \textit{otherwise.} \end{cases} \tag{7}$$

To be able to target both similar and dissimilar classes of input image x, we use scalar value $|h_j(x)|$ for the hot class. Finally, as described below, we use backpropagation to estimate the image changes needed to move according to our constructed feature vector ω_{hc} by computing the image $\eta_{hc} = B_l(\omega_{hc})$. The operator $B_l(\cdot)$ is an approximation to the derivative backpropagated down to the image level. Sec 4.3 explains how this is conducted at intermediate layers. While any positive value for the hot class and any negative value for the cold class have beneficial effects in terms of providing adversarial directions, we find that, in general, values derived from the extracted feature vector of the original image perform very well as they naturally capture the relative scale of the respective class's features.

As we can see in Fig. 2 for MNIST and in Fig. 3 for ImageNet, our hot/cold approach provides comparable results (PASS) to the previously described FGV approach in Sec. 4.1. However, with this new approach we can explicitly move in the direction of targeted classes obtaining several different adversarial directions for each input image. This greatly increases the adversarial diversity available for training.

4.3 Implementation Details

For our adversarial generation implementations and experiments, we use the popular deep learning framework Caffe [10]. Caffe allows us to obtain inner representations of images in neural networks and backpropagate feature representations. However, Caffe's `backward` method is limited to start with features of the top layer. To perform backpropagation of feature representations at deep(er) layers, there are two options. First, we can create truncated networks, make a targeted layer sit at the top of each, and then use the `backward` method to perform the backpropagations of interest. This approach is cumbersome. The second approach is to modify the `backward` method to directly backpropagate from any specified feature representations. By simply eliminating few lines of code responsible for checking whether the specified starting point is the top layer, we can use a single network for our experiments. This is how we implement the backpropagation operator $B_l(\cdot)$.

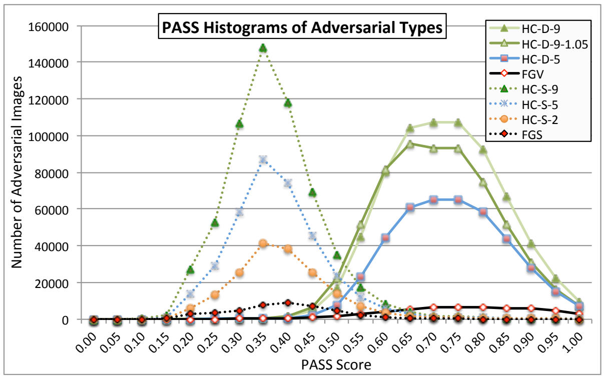


Figure 4: PASS Scores. *Example showing distributions of PASS scores for different types of adversarial and hard positive LeNet/MNIST images. The HC-D-X distributions use the hot/cold approach (Sec.4.2) with X closest scoring classes, while HC-S-X use the sign of the derivatives. HC-D-9-Y uses the derivatives scaled by Y beyond the point of minimal adversarial PASS in the direction of the derivatives. FGS is the fast gradient sign method of [8]. These plots demonstrate that using sign instead of raw directions defined by derivatives significantly reduces PASS scores, e.g., compare HC-S-9 and HC-D-9. They also illustrate that even small extensions beyond the minimal adversarial distance reduces PASS as well. Finally, the plots show the sheer increase in number of adversarials over the basic FGS approach.*

Our novel approach for obtaining adversarial images calculates feature-derivatives at any layer, and allow us to obtain perturbations which define different directions with respect to a given input image. We can search along those directions for the closest perturbations that cause mislabeling to obtain adversarial images or further extend this search to obtain additional hard positives. Since we apply a given direction as a perturbation to the input image, by scaling the perturbation by larger and larger values and adding it to the original image, we move the original image farther and farther along that direction. In order to efficiently discover the closest adversarial point in that direction, we apply a line-search technique with increasing step-sizes. To find adversarials, we search for the smallest possible adversarial perturbation in the last section of line-search by applying a binary search. Finally, we would like to emphasize that all images we generate have discrete pixel-values in $[0, 255]$.

5 Experiments

The focus of our experiments is threefold. First, we generate adversarial examples on LeNet/MNIST and collect metrics to compare different algorithms in terms of L_2 distances, L_∞ distances, and PASS. Second, we demonstrate that while PASS can be applied to complex images, "good" thresholds for adversarial and hard positive examples vary with the domain. Finally, we demonstrate on LeNet/MNIST that retraining with a diverse set of hard positive images can improve the original network's performance more than just training with adversarials generated via the FGS approach.

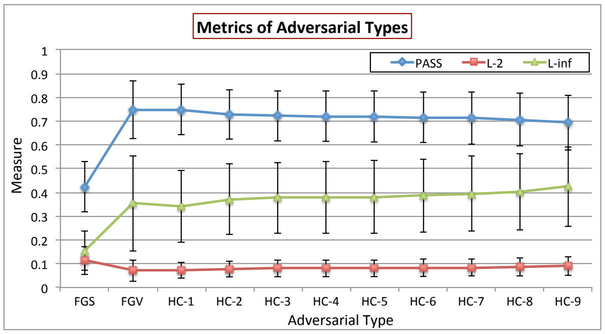


Figure 5: Metrics. *Example showing various metrics (PASS, L_2, L_∞) for different types of MNIST adversarial images generated on three LeNets. FGS, FGV, and HC-X denotes hot/cold approaches targeting the Xth closest scoring classes. While adversarials of FGV and our HC methods have higher L_∞ distances, they also have better PASS scores than FGS. For visualization both L_2 and L_∞ distances are scaled so that max distances are 1.*

5.1 MNIST - Adversarial Metrics

Here we display metrics for various types of adversarial examples generated on LeNets trained with the MNIST dataset. Fig. 4 shows the distribution of PASS scores for different types of adversarial images compared with FGS. As we can see in Fig. 5, L_2 and L_∞ distances do not map well to perceptual similarity as defined by PASS. For example, while adversarial images generated by the FGS method have noticeably lower L_∞ distances than those generated by the FGV method, adversarials of FGV maintain significantly higher PASS.

5.2 MNIST - Training with Hard Positives

We show that fine-tuning with a diverse set of hard positive images enhances LeNet/MNIST's robustness to such examples and also improves its accuracy. We trained 3 different LeNet/MNIST networks with the standard 60K training samples and different random initializations/orderings using the training parameters distributed with Caffe. While many techniques are known to improve performance, e.g. batch normalization and different objective functions, we do not employ those approaches here as the goal is to evaluate the performance gains on a very standard network and make it easy for others to replicate/compare.

For each network, we generated adversarial and hard positive images of different types. We then fine-tuned each network three times using different random orderings of the adversarial training data, giving 9 different networks for testing. Fine-tuning was for 20K iterations; the base networks were initially trained with 10K iterations.

To test, we used both the standard MNIST test data of 10K images and a set of 70K held-out adversarial and hard positive images that were not used for training. The 70K held-out set contained a broad mix of adversarial types, 5K

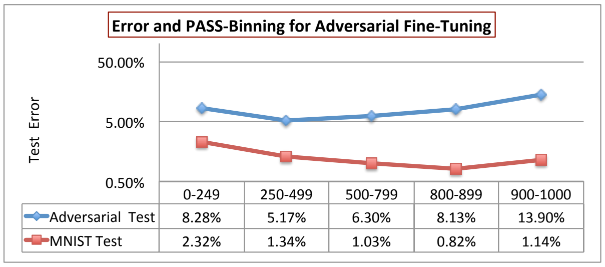


Figure 6: PASS SCORES. *Example showing how error on both original and all adversarial MNIST test data varies for networks trained using generated hard positives with different ranges of PASS scores. Note that networks trained on data with the highest PASS scores have higher MNIST test error and much higher adversarial test error. Even though adversarial testing includes data from all ranges, training using data only with PASS scores in the 800-900 range still does well in adversarial testing.*

each for 14 types: FGS [8], FGV (cf. Sec. 4.1), 9 types of hot/cold approaches (cf. Sec. 4.2) from HC-1 to HC-9, HC-1 scaled by 1.05, HC-1 scaled by 1.10, and adversarials of the hot/cold approach using signs of derivatives for the closest scoring class. Training used adversarial images from only the associated network on a subset of types. To better test generalization, adversarial and hard positive testing uses samples from all networks, thus the basic LeNets do not show 100% error.

Table 1 shows the average accuracies for different test models. These results are consistent both with the hypothesis that increasing diversity improves training and the hypothesis that PASS scores for hard positives should be non-minimal, but not too large. By pushing the boundaries past the minimal adversarial images and closer to the goal class, as graphically depicted as stars in Fig. 1, HC-D-9-1.05, hard positives obtianed by scaling perturbations of HC-1 to HC-9 adversarials (HC-D-9), generalizes the boundaries more and thus increases accuracy over HC-D-9 while significantly improving robustness to adversarial images. The results in Fig. 6 demonstrate that training on images of intermediate PASS values produces better results than training using only images with either the highest or lowest PASS scores. To compare with state-of-the-art adversarial training, we note that HC-D-9, HC-D-9-1.05 and even FGV are significantly better than training with FGS images using the approach of [8]. The line with * contains the results reported in [8] using a modified adversarial-enhanced training objective, which is slightly better than training with a finite set of FGS images. This is likely because the objective function can take advantage of a changing adversarial objective during training.

MNIST	Adversarial	Method
.673% ± .035	14.85%	HC-D-9-1.05
.683% ± .057	21.10%	HC-D-9
.697% ± .021	48.35%	InfiMNIST (1M)
.697% ± .031	29.61%	FGV
.723% ± .047	21.52%	HC-D-5
.733% ± .032	29.29%	HC-S-5
.782% *	17.9%∗	Goodfellow et al. [8]
.790% ± .122	31.06%	FGS
.937% ± .110	64.21%	Basic LeNet

Table 1: ERROR RATES. *Error rates of various adversarial and hard positive trained networks on both MNIST and adversarial test sets. Increasing adversarial diversity clearly improves results. The best performing model, HC-D-9-1.05, using the hot/cold approach with derivatives from all 9 classes to generate hard positives, reduces the MNIST test error 28.18% over the basic LeNet and 14.81% over training with fast gradient sign (FGS) [8]. Note that this model also has the lowest susceptibility to adversarial images with an error rate of 14.85%.*

We also performed comparisons with approaches that use other types of augmented data. We generated 1M images with InfiMNIST [14] and trained three networks for 30K iterations. We see that HC-D-9 and HC-D-9-1.05, each of which are trained with approximately 340K adversarial images, are slightly (but not statistically) better than training on 1M images from InfiMNIST. FGV, trained with 37663 images, which is only 3.7% of the InfiMNIST data, does equally as well. Furthermore, the InfiMNIST trained networks also perform poorly on adversarial examples.

5.3 ImageNet - Adversarial Training

We conducted preliminary experiments using our diverse adversarial approach to enhance the performance of BVLC-GoogLeNet on the ImageNet dataset. Instead of training a new network from scratch, we show that we can improve the existing BVLC-GoogLeNet model by fine-tuning it with diverse adversarial images. Because our focus is on using hard positives to enhance learning, we take only the center crop (which is used by the pre-trained BVLC-GoogLeNet). We started by taking 15 correctly classified images per class, and generated 20 different HC-types – from the most similar (HC-1) to twentieth most similar (HC-20) class as hot, yielding 300K images in total. We filtered out radical perturbations, keeping only the adversarial images with *PASS* scores higher than 0.99. This yields approximately 250K hard positives for training. We then mixed the 250K adversarial images with the original approximately 1.28M center crops of the ImageNet training set. Our fine-tuning procedure relies on original hyperparameters distributed with the BVLC-GoogLeNet model in the Caffe Model Zoo. We fine-tuned the original BVLC-GoogLeNet model for 500K iterations with batch size of 80, i.e. about 25 epochs. We tested on center crops of the ImageNet validation set and report top-1 and top-5 error rates. The top-1 error is 30.552%, and top-5 error rate is 10.604%. While this performance is not state-of-the-art for ImageNet, our improvement in top-1 error rate is 2.07e-4% per added image, and our top-5 gain is 1.01e-4% per added image. As computed from the data in [20], using 10 crops decreased their top-5 error rate with 7.67e-6% per added image, while the use of 144 crops reduced top-5 error by 1.26e-06% per added image. Therefore, using our hard positives in training yields greater improvement for each additional image.

6 Conclusion

In this paper, we have introduced a new perceptual measure for *adversarial images*, which allows us to explicitly quantify the constraint that perturbations must be imperceptible in order to form adversarial images. We have also introduced novel ways of generating diverse adversarial images, and have shown that amplifying adversarial images to create additional *hard positives* can be used to augment training datasets and further enhance accuracies over training with adversarial images alone. Instead of looking like random noise, many of our diverse adversarial perturbations exhibit strong structural artifacts, which are far more likely to occur naturally than perturbations of former adversarial generation methods that generally look like random noise.

Although we could not achieve state-of-the-art performance on MNIST or ImageNet, our results on MNIST are substantially better than those of any prior work leveraging adversarial training or input perturbations on LeNet/MNIST. Our approach also uses far fewer training images and improves robustness to adversarial examples. On ImageNet, we show compelling evidence to suggest that using hard positive images in the training set offers substantially greater performance improvement per image than using translated crops. We have shown adversarial images generated on the state-of-the-art ResNet-152, which suggests that our approach can likely be applied to improve the state of the art.

References

[1] S. Baluja, M. Covell, and R. Sukthankar. The virtues of peer pressure: A simple method for discovering high-value mistakes. In *Comp. Anal. of Images and Patterns*, pp. 96–108. Springer, 2015. 2, 3

[2] Y. Bengio, F. Bastien, A. Bergeron, N. Boulanger-Lewandowski, T. M. Breuel, Y. Chherawala, M. Cisse, M. Côté, D. Erhan, J. Eustache, et al. Deep learners benefit more from out-of-distribution examples. In *Int. Conf. on Artificial Intelligence and Statistics*, pp. 164–172, 2011. 2

[3] L.-C. Chen, A. Schwing, A. Yuille, and R. Urtasun. Learning deep structured models. In *Int. Conf. Machine Learning (ICML)*, pp. 1785–1794, 2015. 1

[4] C. Clark and A. Storkey. Training deep convolutional neural networks to play go. In *ICML*, pp. 1766–1774, 2015. 1

[5] G. D. Evangelidis and E. Z. Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *TPAMI*, 30(10):1858–1865, 2008. 3

[6] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, pp. 1–8, 2008. 2

[7] J. R. Flynn, S. Ward, J. Abich IV, and D. Poole. Image quality assessment using the ssim and the just noticeable difference paradigm. In *Eng. Psy. & Cog. Ergo. Understanding Human Cognition*, pp. 23–30. Springer, 2013. 3

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *Int. Conf. Learning Rep. (ICLR)*, 2015. arXiv:1412.6572. 1, 2, 3, 4, 6, 7

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. arXiv:1512.03385. 1

[10] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proc. Int. Conf. on Multimedia*, pp. 675–678. ACM, 2014. 1, 5

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pp. 1097–1105, 2012. 1

[12] Y. LeCun, C. Cortes, and C. J. Burges. The MNIST database of handwritten digits, 1998. 2

[13] Y. LeCun, L. Jackel, L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. Muller, E. Sackinger, P. Simard, et al. Learning algorithms for classification: A comparison on handwritten digit recognition. *Neural networks: the statistical mechanics perspective*, 261:276, 1995. 2

[14] G. Loosli, S. Canu, and L. Bottou. Training invariant support vector machines using selective sampling. *Large scale kernel machines*, pp. 301–320, 2007. Project infiMNIST available at http://leon.bottou.org/projects/infimnist Accessed 1/28/16. 2, 7

[15] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *CVPR*, pp. 5188–5196, 2015. 4

[16] S. Sabour, Y. Cao, F. Faghri, and D. J. Fleet. Adversarial manipulation of deep representations. In *ICLR*, 2016. arXiv:1511.05122. 2, 3

[17] P. Y. Simard, D. Steinkraus, and J. C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *Int. Conf Document Analysis and Recognition, 2003.*, pp. 958–963. IEEE, 2003. 2

[18] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1

[19] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng. Parsing natural scenes and natural language with recursive neural networks. In *ICML*, pp. 129–136, 2011. 1

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pp. 1–9, 2015. 1, 8

[21] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *ICLR*, 2014. arXiv:1312.6199. 1, 2, 3

[22] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 3