Are Neural Networks Excessively Invariant?

Sriram Balasubramanian Gaurang Sriramanan Vinu Sankar Sadasivan

Abstract

Despite the remarkable success of deep neural networks in a myriad of settings, several works have demonstrated their vulnerability to imperceptible perturbations, known as adversarial attacks. On the other hand, prior works have also demonstrated the excessive insensitivity of deep networks to large-magnitude perturbations in input space. In light of these observations, we aim to identify techniques to study the extent of excessive invariance displayed by deep neural networks. Towards this, we propose a novel Null Space Projected Gradient Descent (NSPGD) attack, that iteratively refines image perturbations without affecting network activations. Further, we study the efficacy of confidence calibration of classification networks as a mode to mitigate excessive invariance, in order to train models that are robust to such attacks, while simultaneously being robust to standard adversarial attacks.

1. Introduction

Deep neural networks have been successfully used to achieve state of the art performance in several fields such as computer vision and natural language processing, often achieving a quantum leap in performance over classical algorithms. However, the vulnerability of such networks to adversarial attacks (Carlini et al., 2019) has spurred immense interest in developing techniques to reduce the excessive sensitivity of such models (Madry et al., 2019; Zhang et al., 2019). In contrast, in this work we seek to investigate the phenomenon of excessive invariance in deep networks, and analyse this conjugate mode of failure.

We illustrate the phenomenon of excessive sensitivity and excessive invariance in Figure-1. Standard adversarial attacks accentuate the existence of misclassified inputs despite being imperceptibly different from clean samples, and thus induce island-like pockets in the decision landscape as seen in Figure-1(a). In sharp contrast, models that are excessively invariant fail to recognise large-magnitude semantic changes to the input, that would be conspicuous to humans. Thus, for a model that is both excessively invariant and excessively sensitive, we obtain extended branch-like decision boundaries, wherein high-confidence predictions are retained despite substantial changes in the input domain.

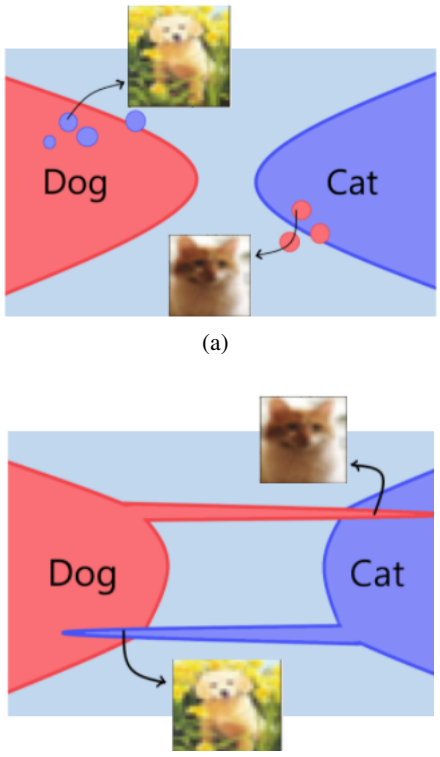


Figure 1. (a) Decision boundaries of a model that is excessively sensitive but not excessively invariant ("island-like"). (b) Decision boundaries of a model that is excessively sensitive and excessively invariant ("branch-like").

We hypothesize that these two phenomenons are distinct from each other and solving them will require distinct methods. While adversarial training helps in reducing the number of adversarial samples within an ϵ ball, adversarial samples can possibly be present just outside the ϵ ball (as we shall show). Therefore, while adversarial training solves the problem of "island-like" decision boundaries, we will have to consider the region outside the ϵ ball to address the problem of "branch-like" decision boundaries.

2. Related work

The problem of excessive invariance in neural networks is first pointed out by Jacobsen et al. (2018) who show that neural networks can be excessively invariant to semantically meaningful changes for ImageNet (Deng et al., 2009) and MNIST (LeCun, 1998) datasets using invertible ResNets. Tramèr et al. (2020) show that models that are adversarially trained with robustness guarantees for large epsilon bounds can be "overly" smooth. However, this result is for provably robust models with relatively large norm of around 0.4 on MNIST. Also, image pairs are restricted to be within epsilon distance, therefore the true extent of invariance of model is unknown. However, the prior works do not explicitly establish the presence of an equi-confidence path from the source image to the target image.

Stutz et al. (2020) propose Confidence-Calibrated Adversarial Training (CCAT) for training neural networks to generalize to unseen threat models. CCAT trains network to reject adversarial examples with low confidence predictions. In this manner, a CCAT model trained only on L_∞ based threat models generalize well to other unseen attacks. While CCAT achieves robustness through sample rejection, recent works (Laidlaw et al., 2020; Kireev et al., 2021) use a neural perceptual threat model to train networks against the set of all imperceptible adversarial examples. Zhang et al. (2018a) propose Learned Perceptual Image Patch Similarity (LPIPS) measure that correlates well with human perception. Laidlaw et al. (2020) propose Lagrange Perceptual Attack (LPA) by designing adversarial examples bounded by the neural perceptual LPIPS measure.

3. Problem statement

Consider an image classification problem with n dimensional images and d classes. Let f denote a neural network, with softmax prediction $f(\boldsymbol{x})$ for an image $\boldsymbol{x} \in [0,1]^n$. Given a source image $\boldsymbol{x}_s$, a target image $\boldsymbol{x}_t$, and model softmax prediction $f(\boldsymbol{x}_s) = \hat{\boldsymbol{y}}_s, f(\boldsymbol{x}_t) = \hat{\boldsymbol{y}}_t$ such that $\hat{\boldsymbol{y}}_s, \hat{\boldsymbol{y}}_t \in \mathbb{R}^d$, argmax $\hat{\boldsymbol{y}}_s \neq$ argmax $\hat{\boldsymbol{y}}_t$, we can pose the following question:

Find an image $\boldsymbol{x}$ which minimizes $||\boldsymbol{x} - \boldsymbol{x}_t||$ such that there exists a "path" of images $\boldsymbol{x}_s = \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n = \boldsymbol{x}$ where:

1. $\forall i, ||\boldsymbol{x}_i - \boldsymbol{x}_{i+1}|| \leq \epsilon$,
2. $[\hat{\boldsymbol{y}}_i]_j \geq [\hat{\boldsymbol{y}}_s]_j - \delta$ where $\hat{\boldsymbol{y}}_i = f(\boldsymbol{x}_i), j =$ argmax $\hat{\boldsymbol{y}}_s$,.

for some thresholds $\epsilon, \delta > 0$ which are small enough.

Note that if not for the requirement that a "path" of images has to exist, this problem would reduce to the standard adversarial perturbation problem as posed in Szegedy et al. (2014)

We declare the attack a "success" if $\boldsymbol{x}$ is "indistinguishable" from $\boldsymbol{x}_t$, or $d_{\text{perceptual}}(\boldsymbol{x}, \boldsymbol{x}_t) < d$ for some $d > 0$. We use the LPIPS distance as proposed by Zhang et al. (2018a) to measure $d_{\text{perceptual}}$.

4. Analyzing a two-layer neural network

We first present nullspace analysis of a fully connected two-layer neural network using a non-linear ReLU activation function. Consider a network with such an architecture with trained weights $W_1 \in \mathbb{R}^{h \times n}$ and $W_2 \in \mathbb{R}^{d \times h}$. The neural network can then be represented as:

$$f(\boldsymbol{x}) = S\left(W_2\left(ReLU(\, W_1 \boldsymbol{x}\,)\right)\right)$$

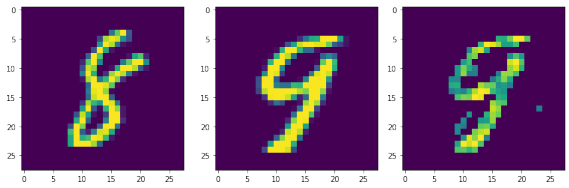
where for simplicity, the bias terms have been absorbed into the weight matrices W_2 and W_1, and S is the Softmax function. It is evident that if $\boldsymbol{\delta}$ is a perturbation such that $\boldsymbol{\delta}$ is in the nullspace of W_1, then $f(\boldsymbol{x} + \boldsymbol{\delta}) = f(\boldsymbol{x})$ for all images x, irrespective of the magnitude of $\boldsymbol{\delta}$.

Thus, in order to avoid trivial solutions such as these, we now assume that W_1 is a full-rank matrix, with a trivial nullspace. This is commonly seen in practice as well: if MNIST images are considered as 784-dimensional vectors, networks with the above architecture are common wherein the hidden activations are of dimension $h = 1000$, with the resultant W_1 matrix generally being full rank. In sharp contrast, we observe however that nullspace of matrix W_2 is expected to be significant, given that the number of distinct classes $d \ll h$.

To analyse this network, we define the following matrices expressed as a function of $\boldsymbol{x}$. $W_1^+(\boldsymbol{x})$ is a submatrix of W_1 which contains the maximal subset of rows of W_1 such that $W_1^+(\boldsymbol{x})\boldsymbol{x}$ is element-wise greater than $\boldsymbol{0}$. $W_2^+(\boldsymbol{x})$ is a submatrix of W_2 which contains the subset of columns of W_2 corresponding to the subset of rows in $W_1^+(\boldsymbol{x})$. $W_1^{thresh}(\boldsymbol{x}, t)$ is a submatrix of W_1 which contains the maximal subset of rows of W_1 such that all elements of $W_1^{thresh}(\boldsymbol{x}, t)\boldsymbol{x}$ are in the interval $[-t, t]$ for some $t > 0$.

Suppose we have source image $\boldsymbol{x}_s$ and target image $\boldsymbol{x}$. Then, the change in the activations after the first layer is $W_1(\boldsymbol{x} - \boldsymbol{x}_s)$. If we assume that $|[W_1]_{ij}| \leq w$ for all i, j, then $||W_1(\boldsymbol{x} - \boldsymbol{x}_s)||_1 \leq w||\boldsymbol{x} - \boldsymbol{x}_s||_1$. This means that $W_1^+(\boldsymbol{x})$ will remain the same as $W_1^+(\boldsymbol{x}_s)$ for all $\boldsymbol{x}$ provided that $\boldsymbol{x} - \boldsymbol{x}_s$ lies in the nullspace of $W_1^{thresh}(\boldsymbol{x}, w||\boldsymbol{x} - \boldsymbol{x}_s||_1)$.

Now, we observe that locally, $f(\boldsymbol{x}) = S(W_2^+(\boldsymbol{x})W_1^+(\boldsymbol{x})\boldsymbol{x})$. Let the dimension of $W_1^{thresh}(\boldsymbol{x}, w||\boldsymbol{x} - \boldsymbol{x}_s||_1)$ be $h_{thresh}(\boldsymbol{x}, w||\boldsymbol{x} - \boldsymbol{x}_s||_1) \times n$. This implies its nullspace has a dimension of at least $n - h_{thresh}(\boldsymbol{x}, w||\boldsymbol{x} - \boldsymbol{x}_s||)$. The nullspace of $W_2^+(\boldsymbol{x})W_1^+(\boldsymbol{x})$ is at least $n - d$. Therefore, the intersection of the two nullspaces has a dimension of at least $n - d - h_{thresh}(\boldsymbol{x}, w||\boldsymbol{x} - \boldsymbol{x}_s||_1)$. This implies that if $\boldsymbol{x} - \boldsymbol{x}_s$



(a) (b) (c)

Figure 2. (a) Original image of an '8' from MNIST test set (b) Original image of a '9' from MNIST test set (c) '8' transformed to '9' by a single-step null space vector addition

is projected onto this intersection of the two nullspaces to obtain $(\boldsymbol{x} - \boldsymbol{x}_s)_p$, then $f(\boldsymbol{x}_s) = f(\boldsymbol{x}_s + (\boldsymbol{x} - \boldsymbol{x}_s)_p)$. For computing the projection of a vector $\boldsymbol{v}$ onto the nullspace of a matrix M, we minimize $||\boldsymbol{v} - \boldsymbol{v}_p||_2$ subject to the constraint that $M\boldsymbol{v}_p = 0$. Solving the Lagrangian optimization , we get $\boldsymbol{v}_p = \boldsymbol{v} - M^T(MM^T)^{-1}M\boldsymbol{v}$.

To ensure that $N(\boldsymbol{x} - \boldsymbol{x}_s)$ is close to $\boldsymbol{x}$, the dimension of N which is

$$n - d - h_{thresh}(\boldsymbol{x}, w||\boldsymbol{x} - \boldsymbol{x}_s||)$$

should be as high as possible, or $h_{thresh}(\boldsymbol{x}, w||\boldsymbol{x} - \boldsymbol{x}_s||)$ should be as low as possible. Thus, if $\boldsymbol{x}$ or $\boldsymbol{x}_s$ are close, or if w is low, dimension of N will be high.

We train a two-layer network on the MNIST dataset using the Adam optimizer for 50 epochs using the standard cross-Entropy loss. Using the above method, we can easily transform any image to any other image without changing the confidence values using a single step vector addition. An example transformation is shown in Figure-2.

5. Proposed Methods

5.1. Null Space Projected Gradient Descent attack

We can describe the Null Space Projected Gradient (NSPGD) attack as follows:

1. Start with the current image as the source image ($\boldsymbol{x} = \boldsymbol{x}_s$)
2. Compute the difference vector between the target image and the current image ($\boldsymbol{x}_t - \boldsymbol{x}$), and project it onto the null space of the gradient of the confidences w.r.t the image $\boldsymbol{x}$ and add it to the current image.
3. Repeat till the current image $\boldsymbol{x}$ is close to $\boldsymbol{x}_t$ in terms of perceptual distance.

The algorithm is described using pseudocode in 1. By construction, the algorithm produces an image path $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_n$ where $\boldsymbol{x}_i$ is the image $\boldsymbol{x}$ at iteration i. Since the update rule is $\boldsymbol{x}_{\text{new}} = \boldsymbol{x} + \epsilon\boldsymbol{u}/||\boldsymbol{u}||$, we can see that $||\boldsymbol{x}_i - \boldsymbol{x}_{i+1}|| \leq \epsilon$. At each iteration, the check if statement ensures that if $f(\boldsymbol{x}_s)[j] - f(\boldsymbol{x}_i)[j] > \delta$, the attack halts and the final image $\boldsymbol{x}$ is evaluated to check if $d_{\text{perceptual}}(\boldsymbol{x}, \boldsymbol{x}_t) < d$.

Algorithm 1 Null Space Projected Gradient Descent (NSPGD)

Input: Source image $\boldsymbol{x}_s$, target image $\boldsymbol{x}_t$, max iterations m, model f, step size ϵ, confidence threshold δ, perceptual distance threshold d
Initialize $\boldsymbol{x} = \boldsymbol{x}_s, j = \text{argmax } f(\boldsymbol{x}_s)$
for $i = 1$ **to** $m - 1$ **do**
  $\boldsymbol{x}_{\text{diff}} = \boldsymbol{x}_t - \boldsymbol{x}$
  $\boldsymbol{g} = \nabla f(\boldsymbol{x})[j]$
  $c = \min(\boldsymbol{g} \cdot \boldsymbol{x}_{\text{diff}}/(||\boldsymbol{g}|| \cdot ||\boldsymbol{x}_{\text{diff}}||), 0)$
  $\boldsymbol{u} = \boldsymbol{x}_{\text{diff}} - c\boldsymbol{x}_{\text{diff}}$
  $\boldsymbol{x}_{\text{new}} = \boldsymbol{x} + \epsilon\boldsymbol{u}/||\boldsymbol{u}||$
  if $f(\boldsymbol{x}_s)[j] - f(\boldsymbol{x}_{\text{new}})[j] > \delta$ **then**
    if $d_{\text{perceptual}}(\boldsymbol{x}, \boldsymbol{x}_t) < d$ **then**
      Return $\boldsymbol{x}$ {Success}
    else
      Return null {Failure}
    end if
  else
    $\boldsymbol{x} = \boldsymbol{x}_{\text{new}}$
  end if
end for
if $d_{\text{perceptual}}(\boldsymbol{x}, \boldsymbol{x}_t) < d$ **then**
  Return $\boldsymbol{x}$ {Success}
else
  Return null {Failure}
end if

5.2. Analyzing Confidence Calibrated Adversarial Training

Stutz et al. (2020) propose an adversarial training technique called Confidence Calibrated Adversarial Training (CCAT) to make networks robust to unseen attacks through sample rejection. While training, they perturb the target label y of adversarial sample x as follows:

$$\tilde{y} = \lambda(\boldsymbol{\delta}) \text{ one\_hot(y)} + (1 - \lambda(\boldsymbol{\delta}))\frac{1}{K} \quad (1)$$

$$\lambda(\boldsymbol{\delta}) = (1 - \min(1, \frac{\|\boldsymbol{\delta}\|_\infty}{\epsilon}))^\rho \quad (2)$$

where λ is the power transition, K is the total number of classes, $\boldsymbol{\delta}$ is the adversarial noise added to $\boldsymbol{x}$, and ϵ and ρ are constants. Typically, ϵ is set to be 0.03 for CIFAR-10 and $\rho = 10$.

While CCAT was not intentionally designed to address the problem of excessive invariance in neural networks, it en-

forces the constraint that confidences should fall to $\frac{1}{K}$ (10% in case of 10 classes) in case the adversarial perturbation δ is too large through their power transition rule, which addresses the problem of excessive invariance to some extent.

We observe that while the NSPGD attack is seen to be extremely effective on normally trained models and adversarially trained networks (Section-6), it does not break CCAT, as confidences fall rapidly with increasing perturbation size.

5.2.1. Issues with CCAT

There are two major drawbacks in the CCAT method: 1) CCAT rejects adversarial samples with low perturbations which is not desirable as shown in 3, and 2) CCAT uses a fixed L_∞ ϵ bound in its training which may not be aligned with perceptual differences. In Section-5.4, we propose LPIPS-CCAT to mitigate these issues.

Algorithm 2 Logit Space Attack with NSPGD

Input: Source image $\boldsymbol{x}_s$, target image $\boldsymbol{x}_t$, max iterations m, model with logit predictions L, scalar function h, step size ϵ
Initialize $\boldsymbol{x} = \boldsymbol{x}_s + \boldsymbol{\delta}$, where $\delta \sim U(-1, +1)$
for $i = 1$ **to** $m - 1$ **do**
  $L_1 = ||L(\boldsymbol{x}) + h(\boldsymbol{x}) - L(\boldsymbol{x}_s)||^2 - \lambda \cdot ||S(L(\boldsymbol{x})) - S(L(\boldsymbol{x}_s))||^2$
  $L_{smooth} = TV(\boldsymbol{x} - \boldsymbol{x}_s)$
  $\boldsymbol{\delta} = \boldsymbol{\delta} + \epsilon \cdot \text{sign}(\nabla_{\boldsymbol{x}}(L_1 + L_{smooth}))$
  $\boldsymbol{x} = Clamp(\boldsymbol{x} + \boldsymbol{\delta}, 0, 1)$
end for
Initialize NSPGD $\leftarrow \boldsymbol{x}$

5.3. Logit space attack

In order to make the NSPGD attack more effective, we propose to identify explicit invariances in the classification network.

Here we make the observation that the softmax function is translation invariant. Let L denote the logit predictions for a given neural network under consideration, so that $f(\boldsymbol{x}) = S(L(\boldsymbol{x}))$, where S is the Softmax function. Then,

$$S(L(\boldsymbol{x})) = S(L(\boldsymbol{x}) + h(\boldsymbol{x}) \cdot \mathbf{1})$$

where $h(\cdot)$ is an arbitrary scalar function, and $\mathbf{1}$ denotes the all-one vector. That is, if the vector of logits predicted for an image x is shifted uniformly by a scalar function h, the confidences are left unchanged. However, the Jacobian of these quantities with respect to the image can be wildly different. We thereby incorporate this observation into the attack objective and consolidate the NSPGD attack using the same, as shown in Algorithm-2. To improve the quality of images so produced, we also use the Total Variation metric (using L_1 norm) to obtain smoother perturbations in practice.

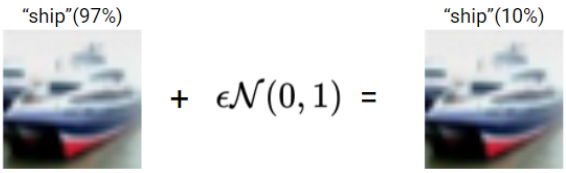


Figure 3. CCAT is seen to be extremely brittle in practice, wherein confidences are seen to dramatically fall with minute perturbations of the input. Here, we present one such example with $\epsilon = 0.01$, which induces random confidence of 10% for its prediction.

By initializing NSPGD with logit space attack, we are slightly relaxing constraint 1 described in section 3 since $||\boldsymbol{x}_1 - \boldsymbol{x}_2||$ need not be less than ϵ, however, the confidences assigned by the model to $\boldsymbol{x}_2$ are exactly the same as that of $\boldsymbol{x}_1$ (or $\boldsymbol{x}_s$)

Algorithm 3 LPIPS-CCAT

Input: Perceptual ϵ_{LPIPS} bound, L_∞ ϵ bound, power transition parameter ρ, dataset $\mathcal{D} = (\mathcal{X}, \mathcal{Y})$, neural network f with $k = 1, ..., K$ output logits, loss function $\mathcal{L}$
while True **do**
  choose random batch $(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_B, y_B) \in \mathcal{D}$
  for $b = 1, ..., B/2$ **do**
    $d(\boldsymbol{\delta}) := \text{LPIPS}(\boldsymbol{x}_b, \boldsymbol{x}_b + \boldsymbol{\delta})$
    $\boldsymbol{\delta}_b := \text{argmax}_{\,d(\boldsymbol{\delta}) \le \epsilon_{\text{LPIPS}}} \max_{k \neq y} f_k(\boldsymbol{x}_b + \boldsymbol{\delta})$
    $\tilde{\boldsymbol{x}}_b := \boldsymbol{x}_b + \boldsymbol{\delta}_b$
    $\epsilon := \|\boldsymbol{\delta}_b\|_\infty \cdot \frac{\epsilon_{\text{LPIPS}}}{\text{LPIPS}(\boldsymbol{x}, \boldsymbol{x} + \boldsymbol{\delta}_b)}$
    $\lambda(\boldsymbol{\delta}_b) := \sigma(\rho \cdot (\epsilon - \|\boldsymbol{\delta}_b\|_\infty))$
    $\tilde{y}_b = \lambda(\boldsymbol{\delta}_b)\ \text{one\_hot}(y_b) + (1 - \lambda(\boldsymbol{\delta}_b))\frac{1}{K}$
  end for
  update parameters using loss
  $\Sigma_{b=1}^{B/2} \mathcal{L}(f(\tilde{\boldsymbol{x}}_b), \tilde{y}_b) + \Sigma_{b=B/2}^{B} \mathcal{L}(f(\boldsymbol{x}_b), y_b)$
end while

5.4. LPIPS-CCAT

LPIPS-CCAT uses LPIPS measure to define sample-wise L_∞ epsilon-bounds. Further, it also uses a sigmoid based power transition function to ensure that adversarial samples with perceptual perturbations less than ϵ_{LPIPS} are classified with high confidence. The power transition function for LPIPS-CCAT is defined as follows:

$$\lambda(\boldsymbol{\delta}) = \sigma(\rho \cdot (\epsilon - \|\boldsymbol{\delta}\|_\infty)) \tag{3}$$

$$\epsilon = \|\boldsymbol{\delta}\|_\infty \cdot \frac{\epsilon_{\text{LPIPS}}}{\text{LPIPS}(\boldsymbol{x}, \boldsymbol{x} + \boldsymbol{\delta})} \tag{4}$$

where ϵ_{LPIPS} for CIFAR-10 is typically 0.03, and $\sigma(.)$ is the sigmoid function. In our experiments, we use a pretrained

Alexnet to compute the LPIPS measure. Algorithm 3 shows the pseudo-code for our proposed method. Initial results for LPIPS-CCAT is discussed in subsection 6.2.

5.5. Softmax temperature control

Temperature scaling for softmax scales the output logits z_i for $i \in 1, ..., K$ by a constant T. Temperature scaling helps neural networks to control the confidence of predictions in neural networks (He et al., 2018; Jang et al., 2016; Zhang et al., 2018b). We propose to make the temperature factor T a sample-wise trainable parameter. This might help the network to inherently learn to classify adversarial examples with high perturbations with low confidence. For a data point x, class probability i after temperature scaling is given as:

$$p_i(\boldsymbol{x}) = \frac{e^{\frac{z_i}{T(\boldsymbol{x})}}}{\sum_{k=1}^{K} e^{\frac{z_k}{T(\boldsymbol{x})}}}$$

6. Experimental results and Analysis

We use ResNet-18 (He et al., 2015)as the base model. We compare against normally trained models, adversarially trained models (Madry et al., 2019), and confidence calibrated adversarially trained (CCAT) models (Stutz et al., 2020) trained on the CIFAR-10 and MNIST datasets. For multi-threat adversarially trained models (MSD) (Maini et al., 2019), we use their pre-trained models which are PreActResNets (He et al., 2016) for CIFAR-10 and LeNet (LeCun et al., 1998) for MNIST which are different from our models.

We use step size $\epsilon = 0.01$ and confidence threshold $\delta = 0.05$. The LPIPS distance utilizes features from AlexNet (Krizhevsky et al., 2012), and the distance threshold for a successful attack is set at $d = 0.03$. We randomly choose 10 images from each class of CIFAR-10 and MNIST as the target images and report the average success rates of NSPGD attack on each source class for every target image in Figure-7. The columns correspond to the source class while the rows corresponding to the randomly chosen target images of each class. We also show the final images after running the NSPGD attack on these randomly chosen images in Figure-4 (CIFAR-10) and 5 (MNIST) for normally trained and adversarially trained models. The diagonal contains the target images for each class, each row corresponds to a different target class and each column corresponds to a different source class. The confidence that the model assigns to each image is shown below each image.

We can observe a few interesting differences in the example images: NSPGD attack on normally trained models is able to produce images very close to the target images, almost indistinguishable to the human eye. However, NSPGD on adversarially trained models produce images which are close to the target image but have certain residual artifacts from the source image. For example, when the ship is transformed into a deer, while the colours are completely changed and the image can no longer be called a ship, we can see some outlines of the ship visible in the image. This gives some evidence to believe that the adversarially trained model is more resistant to these attacks in human interpretable ways. We can observe the same pattern in the MNIST dataset as well.

Datasets	Normal	AT	MSD	CCAT
CIFAR-10	0.9369	0.5839	0.4478	**0.1188**
MNIST	0.8858	0.6826	0.2145	**0.0287**

Table 1. Success rate of NSPGD attack averaged over classes. Pretrained models with different architecture used for MSD.

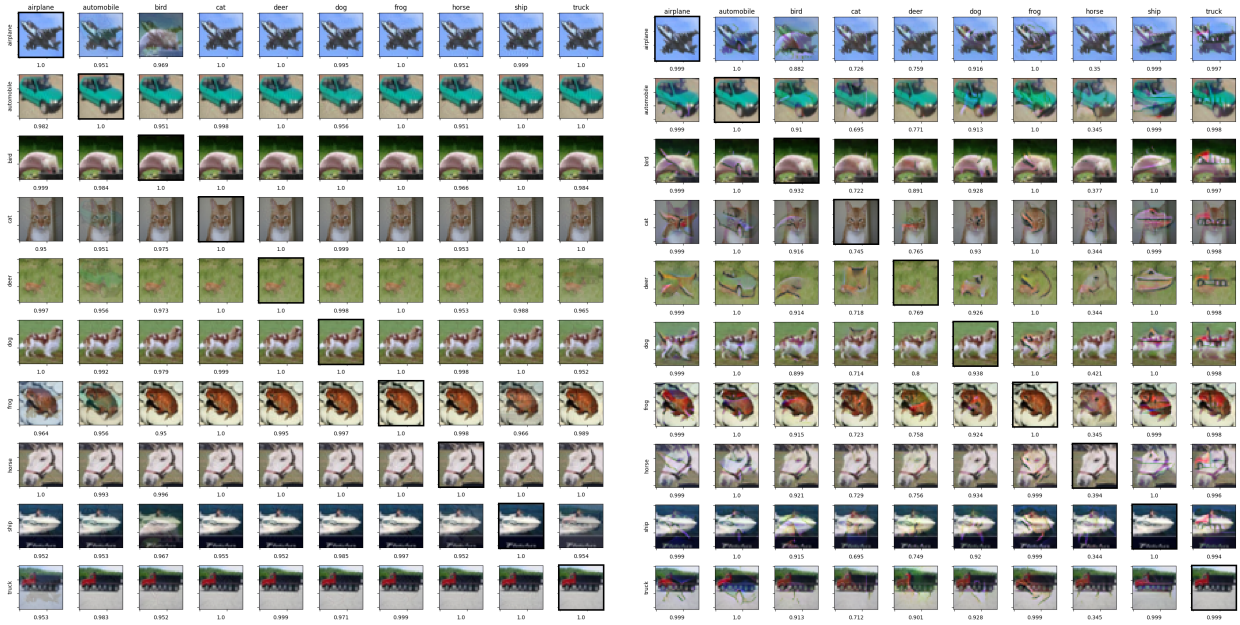
Datasets	Normal	AT	MSD*	CCAT
CIFAR-10	1.000	**0.451**	0.532	0.684
MNIST	1.000	**0.017**	0.391	0.074

Table 2. Success rate of untargeted L_∞ attack with $\epsilon = 0.03$. Pretrained models with different architecture used for MSD.

In the success rate matrix in Figure- 7, we can observe that for the normally trained models, it is easy to transform any image into any other image, and success rates are uniformly high across classes. However, for adversarially trained models, there are certain source classes which are easy to transform into certain target images with success rate of more than 70 %, while on others the success rate is below 30 %. Also, the target images have higher influence on the success rate than the source classes, and we can see that some targets are "easy" while others are "hard". This influence is much higher on MNIST as compared to CIFAR-10. Exactly which targets are easy or hard is sample dependent.

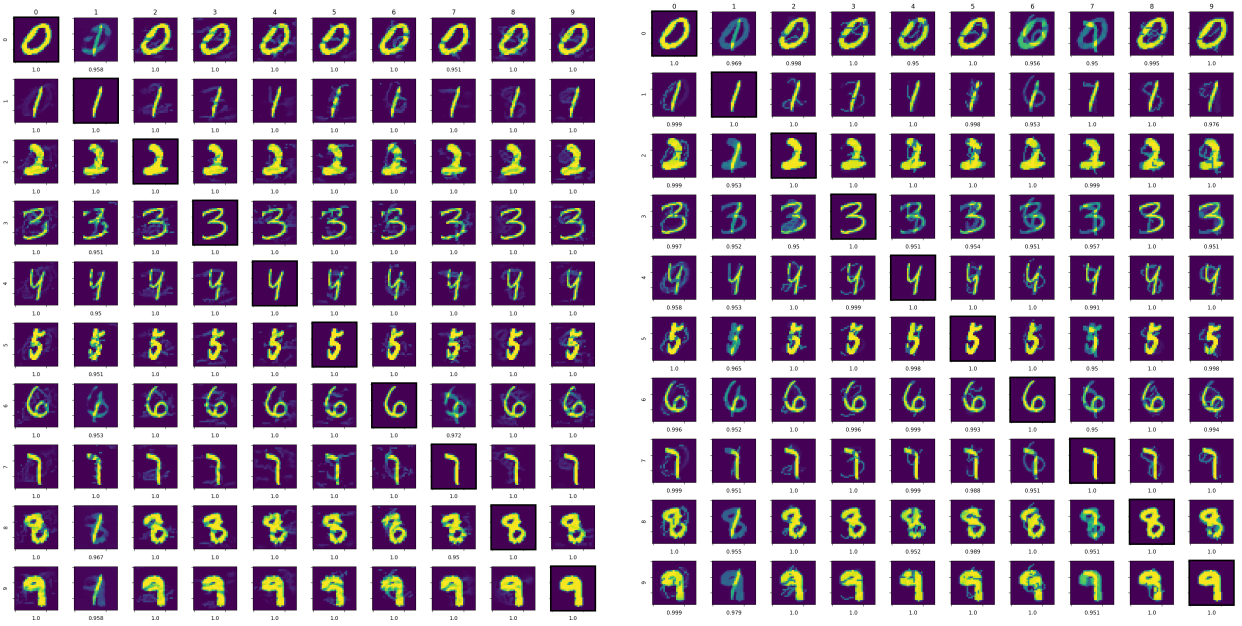
The success rate of NSPGD averaged over all classes are shown in Table-1. Because of the special training of CCAT using the power transition rule, it is highly resistant to any large magnitude changes in the image, which means that confidence drops drastically as we move further away from the image manifold. Thus, the NSPGD attack terminates early and cannot change the images substantially.

We further evaluate consolidated Logit space attack with NSPGD attack on the CCAT model trained on CIFAR-10. We present sample images so resulting in Figure-8. We thus observe that the consolidated attack is indeed effective on CCAT, with high confidences produced for images that are semantically very different from the corresponding source images. We do however observe that the visual fidelity of images so produced could be improved greatly. We note however that as the perturbations produced are made smoother by increasing the weighting of the Total Variation



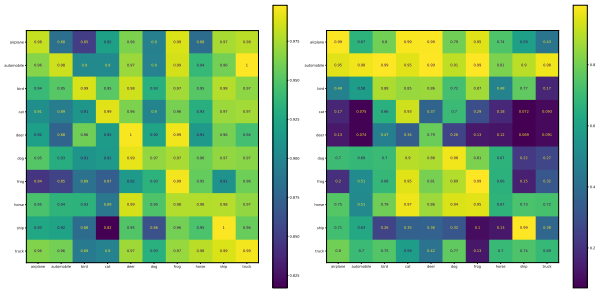
(a) Normally trained model (b) Adversarially trained model

Figure 4. NSPGD attack on CIFAR-10. Confidences are shown below each image. Rows correspond to target classes and columns correspond to the source classes.



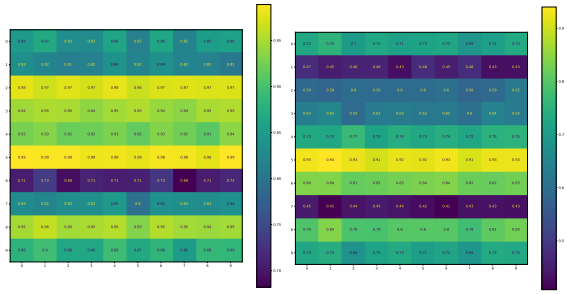
(a) Normally trained model (b) Adversarially trained model

Figure 5. NSPGD attack on MNIST. Confidences are shown below each image. Rows correspond to target classes and columns correspond to the source classes.



(a) Normal model (b) AT model

Figure 6. Percentage of successful attacks on CIFAR-10



(a) Normal model (b) AT model

Figure 7. Percentage of successful attacks on MNIST

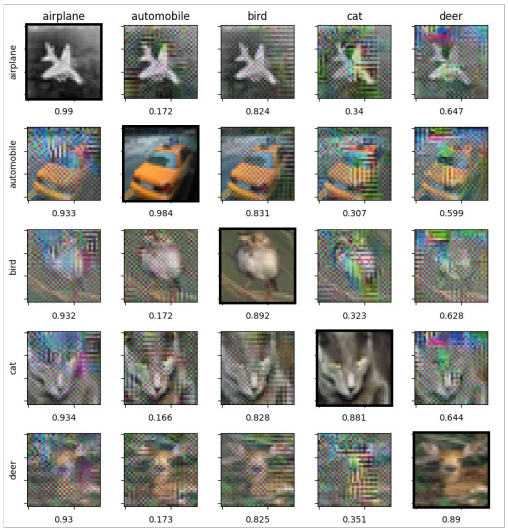


Figure 8. Consolidated Logit space attack with NSPGD attack on a CCAT model on the CIFAR-10 dataset.

metric, prediction confidences generated by CCAT are seen to drop once again.

In Table-2, we can observe that adversarially trained models with $\epsilon = 0.03$ are more robust to seen attacks (L_∞ with $\epsilon = 0.03$) but this does not imply that they are also less excessively invariant, as the Table-1 shows.

6.1. Visualization of the equi-confidence path

We also visualize the equi-confidence path returned by NSPGD by considering an L_2 sphere S with radius r around each intermediate image $\boldsymbol{x}_i$ such that for any image $\boldsymbol{x} \in S(\boldsymbol{x}_i, r)$, $(\boldsymbol{x} - \boldsymbol{x}_i) \cdot \nabla f(\boldsymbol{x}_i)[j] = 0$. We then estimate the 95th and 5th percentile confidences $f(\boldsymbol{x})[j]$ by randomly sampling $\boldsymbol{x} \in S(\boldsymbol{x}_i, r)$. We then visualize the paths by plotting confidences of the source class for different radii (L_2 distances in multiples of 5) and different intermediate images $\boldsymbol{x}_i$ for all i which are a multiple of 10 (see Figure-9).

We can see that the confidences for the adversarial model drop off to zero (at L_2 distances around 30) much slower than the confidences for the normally trained model (at L_2 distances around 20).

This means that although adversarial training reduces the number of adversarial examples within the ϵ ball, the equi-confidence paths are much wider for adversarial models. This could be because adversarial models are smoother as compared to normally trained models, thus they are more invariant in directions normal to the tangent vector at that point.

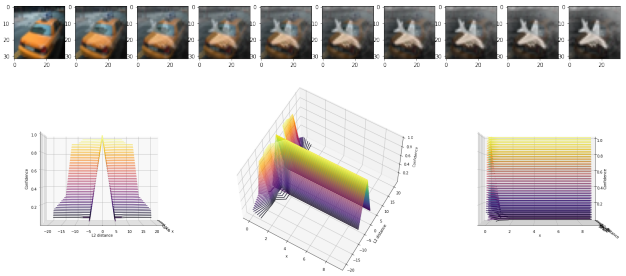
6.2. LPIPS-CCAT

For LPIPS-CCAT, we use the code base of Stutz et al. (2020) from here[1]. We make the modifications to CCAT as shown in Algorithm 3. We use $\rho = 100$ and $\epsilon_{\text{LPIPS}} = 0.03$ for CIFAR-10. Rest of the hyperparameters are left unchanged. We use a pre-trained Alexnet for computing the LPIPS measure as given in Zhang et al. (2018a). We compare our LPIPS-CCAT with CCAT in Figure-10 against L_∞ and LPA attacks (Laidlaw et al., 2020). For LPA, we use the codes from here [2]. Note that CCAT models take around three weeks to fully train. Hence, we present comparisons between fully trained CCAT (200 epochs) and partially trained LPIPS-CCAT (40 epochs) in Figure-10.

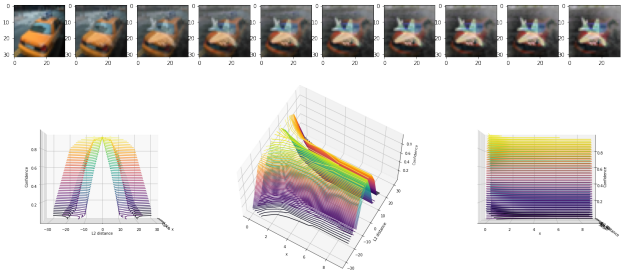
We find that CCAT is better than the partially trained LPIPS-CCAT. However, CCAT rejects samples with very low L_∞ perturbations which is not desirable. We aim to fully train our LPIPS-CCAT model to fairly compare it against the fully trained CCAT model.

[1] https://github.com/davidstutz/confidence-calibrated-adversarial-training

[2] https://github.com/cassidylaidlaw/perceptual-advex



(a) Normally trained model



(b) Adversarially trained model

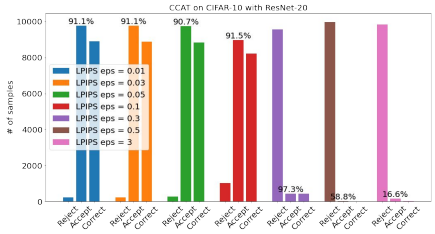
Figure 9. Visualization of the equi-confidence paths. The 95th (purple-orange) and 5th (blue-green) percentile confidences are plotted for each image and each radius.

7. Conclusion

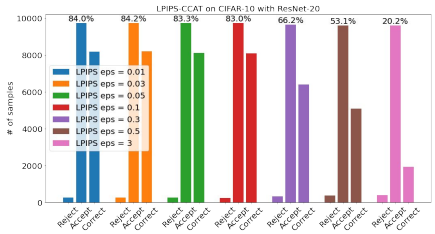
In this work, we explored the phenomenon of excessive invariance in deep neural networks to study a mode of failure conjugate to that of standard adversarial attacks, wherein network activations remain unaltered despite conspicuous changes in the input domain. We first present a theoretical analysis of a two-layer neural network, and verify our predictions empirically on the MNIST dataset. Further, we propose a novel Null Space Projected Gradient Descent (NSPGD) attack, that iteratively updates image perturbations without altering network activations. Furthermore, as a remedial measure for this phenomenon, we propose a training method, LPIPS-CCAT, to simultaneously mitigate excessive invariance and excessive sensitivity.

8. Future Work

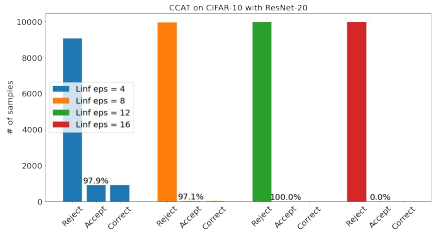
Due to the immense computational requirements incurred in training CCAT, the analysis thus presented was limited to a model trained for only 40 epochs. Hence, this work could be improved by performing the entire 40-step adversarial training for 200 epochs to enable fair comparison, together with better hyperparameter choices for the proposed LPIPS-CCAT model. Further, this work could be extended by exploring the use of Generative Adversarial Networks to generate the NSPGD and logit space attack, which could potentially improve image fidelity and quality. In addition, an adversarially trained network could be used to compute LPIPS distances, since robust models are known to offer better separability properties for large-magnitude perturbations. Towards this, the robust model being trained could itself serve as the LPIPS source model, and is expected to induce better convergence during adversarial training of LPIPS-CCAT.



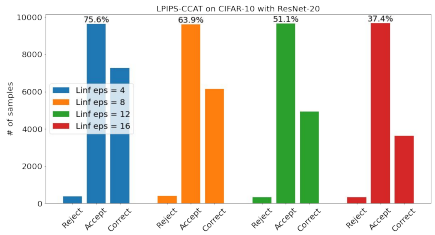
(a) LPA attack on CCAT



(b) LPA attack on LPIPS-CCAT



(c) L_∞ attack on CCAT



(d) L_∞ attack on LPIPS-CCAT

Figure 10. Comparisons between partially trained LPIPS-CCAT and fully-trained CCAT against LPA and L_∞ attacks with increasing ϵ budgets

References

Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I., and Madry, A. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. *CoRR*, abs/1603.05027, 2016. URL http://arxiv.org/abs/1603.05027.

He, Y.-L., Zhang, X.-L., Ao, W., and Huang, J. Z. Determining the optimal temperature parameter for softmax function in reinforcement learning. *Applied Soft Computing*, 70:80–85, 2018.

Jacobsen, J., Behrmann, J., Zemel, R. S., and Bethge, M. Excessive invariance causes adversarial vulnerability. *CoRR*, abs/1811.00401, 2018. URL http://arxiv.org/abs/1811.00401.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Kireev, K., Andriushchenko, M., and Flammarion, N. On the effectiveness of adversarial training against common corruptions. *arXiv preprint arXiv:2103.02325*, 2021.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pp. 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc.

Laidlaw, C., Singla, S., and Feizi, S. Perceptual adversarial robustness: Defense against unseen threat models. *CoRR*, abs/2006.12655, 2020. URL https://arxiv.org/abs/2006.12655.

LeCun, Y. The mnist database of handwritten digits. *Technical report*, 1998.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pp. 2278–2324, 1998. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.42.7665.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks, 2019.

Maini, P., Wong, E., and Kolter, J. Z. Adversarial robustness against the union of multiple perturbation models. *CoRR*, abs/1909.04068, 2019. URL http://arxiv.org/abs/1909.04068.

Stutz, D., Hein, M., and Schiele, B. Confidence-calibrated adversarial training: Generalizing to unseen attacks, 2020.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks, 2014.

Tramèr, F., Behrmann, J., Carlini, N., Papernot, N., and Jacobsen, J.-H. Fundamental tradeoffs between invariance and sensitivity to adversarial perturbations, 2020.

Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. I. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, 2019.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric, 2018a.

Zhang, X., Yu, F. X., Karaman, S., Zhang, W., and Chang, S.-F. Heated-up softmax embedding. *arXiv preprint arXiv:1809.04157*, 2018b.