Explaining Adversarial Examples by Local Properties of Convolutional Neural Networks

Hamed H. Aghdam, Elnaz J. Heravi and Domenec Puig
Computer Engineering and Mathematics Department, Rovira i Virgili University, Tarragona, Spain
{*hamed.habibi, elnaz.jahani, domenec.puig*}*@urv.cat*

Keywords: Adversarial Examples, Convolutional Neural Networks, Lipschitz Constant.

Abstract: Vulnerability of ConvNets to adversarial examples have been mainly studied by devising a solution for generating adversarial examples. Early studies suggested that sensitivity of ConvNets to adversarial examples are due to their non-linearity. Most recent studies explained that instability of ConvNet to these examples are because of their linear nature. In this work, we analyze some of *local* properties of ConvNets that are directly related to their unreliability to adversarial examples. We shows that ConvNets are not locally isotropic and symmetric. Also, we show that Mantel score of distance matrices in the input and output of a ConvNet is very low showing that topology of points located at a very close distance to a samples might significantly change by ConvNets. We also explain that non-linearity of topology changes in ConvNet are because they apply an affine transformation in each layer. Furthermore, we explain that despite the fact that global Lipschitz constant of a ConvNet might be greater than 1, it is locally less than 1 in most of adversarial examples.

1 INTRODUCTION

Despite their success in various tasks of computer vision, Convolutional Neural Networks (ConvNets) suffer from sensitivity to *adversarial examples*. In general, an adversarial example is an example which is generated by slightly perturbing the original sample. Sensitivity of ConvNets to adversarial samples was first discovered by (Szegedy et al., 2014b). Researchers further studies adversarial samples by creating perturbation vectors using various objective functions. Recently, (Goodfellow et al., 2015) suggested that vulnerability of ConvNets to adversarial samples is due to their linear nature.

To our knowledge, previous works have not analyzed local properties of ConvNets that are directly related to their stability against adversarial examples. In this paper, we study some of these properties in order to better explain the reason that ConvNets might be sensitive to small perturbations. Specifically, we conduct various data-driven studies and show that ConvNets are likely not to be isotropic and symmetric around original samples. We support these hypothesis by analyzing the convolution operation in the frequency domain and showing that permutation of input can change the output of the convolution. For this reason, a ConvNet might compute different scores for two adversarial examples located at the same distance from original sample. In addition, we explain why a ConvNet might not be isotropic. Besides, we show that although adversarial examples are very close to the original sample it is highly probable that their topology changes greatly by ConvNets. This behavior is also explained in terms of affine transformation and distance matrices. Our empirical Lipschitz analysis reveals that the global Lipschitz constant can be high (greater than 1) but it is usually less than 1 when we study the Lipschitz constant in a small region around each clean sample.

2 EMPIRICAL STUDY

In general, an adversarial example x_a is defined as:

$$x_a = x + \nu \tag{1}$$

where $\nu \in [-\varepsilon, \varepsilon]^{H \times W \times 3}$ is the perturbation vector and $x \in \mathbb{R}^{H \times W \times 3}$ is the original image. Representing the classification score of a ConvNet by $\Phi : \mathbb{R}^{H \times W \times 3} \rightarrow [0,1]^K$, we can find ν using two different approaches including *optimization-based* and *data-driven*. Given the original image x and its actual class label k, the former approaches try to minimize a regularized objective function. The objective function can be minimizing the score of the actual class regularized by

Aghdam H., Heravi E. and Puig D.
Explaining Adversarial Examples by Local Properties of Convolutional Neural Networks.
DOI: 10.5220/0006123702260234
In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017)*, pages 226-234
ISBN: 978-989-758-226-4

$|\nu|$ in order to find perturbations which are not easily perceivable to human eye (Szegedy et al., 2014b). (Aghdam et al., 2016) also proposed another objective function to find ν such that x_a is misclassified but its distance from decision boundary is minimum.

In contrast, the latter approach finds ν by generating many candidates which satisfy the condition $|\nu| \leq T$ where T is a threshold value. (Goodfellow et al., 2015) compute $sign(\nabla\Phi(x))$ and generate x_a by setting $\nu = \varepsilon sign(\nabla\Phi(x))$ and applying a line search over ε. The optimization-based approaches help us to quickly study stability of a ConvNet to small perturbations. However, they do not provide detailed information about adversarial samples and response of ConvNets to small perturbations. Besides, distribution of the values in the perturbation vector ν found by these techniques may not follow a specific distribution. Moreover, the data-driven technique in (Goodfellow et al., 2015) is mainly used for regularizing a ConvNet and it has the same issues as the optimization-based techniques.

In this work, we have conducted a data-driven technique for studying *local* properties of ConvNets. Specifically, we are mainly interested in properties which are related to stability of ConvNets against small perturbations. We study these properties on AlexNet(Krizhevsky et al., 2012), GoogleNet(Szegedy et al., 2014a), VGG Net(Simonyan and Zisserman, 2015), Residual Net(He et al., 2015) trained on ImageNet dataset as well as the ConvNets in (Ciresan et al., 2012) and (Aghdam et al., 2015) trained on the German Traffic Sign Benchmark (GTSRB) dataset(**?**).

2.1 Isotropic

A zero-centered function is isotropic if it returns an identical value for all points located at specific distance from origin. We say $\Phi(x)$ is locally isotropic at point $\mathbf{a} \in \mathbb{R}^W \times H \times 3$ if:

$$\forall_{\nu_1,\nu_2 \in [-\varepsilon,\varepsilon]^{W\times H\times 3} \wedge \|\nu_1\|=\|\nu_2\|=R} \Phi(\mathbf{a}+\nu_1) = \Phi(\mathbf{a}+\nu_2) \quad (2)$$

where ν_1 and ν_2 are the perturbations vectors and $\mathbf{a}$ is the original image. In other words, the output of the function at all points located at distance R from $\mathbf{a}$ must be identical. Mathematically speaking, we can approximate $\Phi(x_a)$ using the Taylor theorem. Formally:

$$\Phi(x_a) = \Phi(x+\nu) = \Phi(x) + \nabla\Phi(x)\nu + \frac{1}{2}\nu^T H(\Phi(x))\nu \quad (3)$$

where ∇ and $H(.)$ are the gradient and Hessian of $\Phi(x)$. Based on this equation, a ConvNet is locally isotropic at x if elements of ∇ are identical and $H(.)$ is a diagonal matrix where the non-zero elements are equal. Therefore, isotropic property can be measured during backpropagation by computing the pairwise difference between elements of $\nabla\Phi(x)$. However, results obtained by this way might not be promising. This is due to the fact that (3) approximates the output using only the first and second gradients. Theoretically, if $\Phi(x)$ is flat near x_a both ∇ and $H(.)$ will be zero showing that $\Phi(x)$ is isotropic in a very small region close to x.

To analyze a larger region around x, we need higher order terms in (3). Since approximating using higher order terms is not trivial in (3), we analyze isotropic property of different ConvNets empirically. To be more specific, given original image x, we compute:

$$\begin{aligned} \forall_{r\in[\varepsilon,1,\ldots,R]} \forall_{i\in\{1,\ldots,T\}} \;\; s_i^r = \Phi(x + r\frac{\nu_i}{\|\nu_i^r\|}) \\ s.t. \;\; \nu_i^r = \mathcal{U}(-1,1). \end{aligned} \quad (4)$$

In this equation, $\mathcal{U}$ indicates the uniform distribution. According to this equation, we generate T perturbations that all of them are located at distance r from x and compute the classification score of x_a. We set $T = 100$ and $R = 20$ and computed the above equation on 300 samples for each ConvNet and its corresponding dataset. It is worth mentioning that we pick the samples that are classified correctly by ConvNet with more than 99% confidence. Figure 1 illustrates the results.

The horizontal axe shows the radius and the vertical axe shows the range of score (in logarithmic scale) for each radius and each sample obtained by computing $range(r) = \max(\forall_{i\in\{1,\ldots,T\}} s_i^r) - \min(\forall_{i\in\{1,\ldots,T\}} s_i^r)$. Ideally, if $\Phi(x)$ is isotropic around x, $range(r)$ must be zero for all adversarial examples located at distance r from x. In addition, color of each circle in this figure shows the mean score of the adversarial samples. Finally, square markers shows that there was at least one adversarial example at that particular radius that has been misclassifed by the ConvNet.

We observe that none of the ConvNets are perfectly isotropic even at distance ε from a sample. However, their score does not significantly change at distance ε. By increasing the radius to 1 pixel, all ConvNets become more non-isotropic. Finally most of ConvNets become very non-isotropic at distance 10 pixels.

2.2 Symmetricity

Mathematically, multivariate function $f(\mathbf{X}) = f(x_1,\ldots,x_n)$ is symmetric if its value for any permutation of input arguments is identical. For instance,

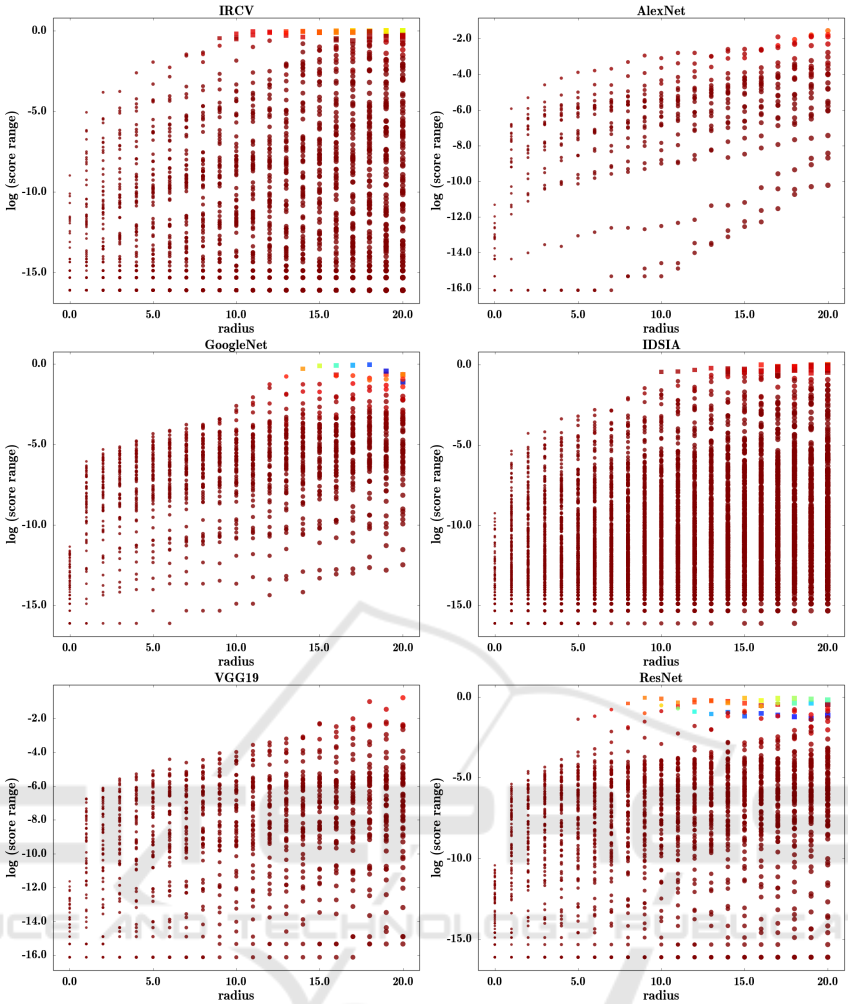

Figure 1: Isotropic property of different ConvNets. Refer to text for detailed information.

$f(x_1,x_2)$ is symmetric if $f(a_1,a_2) = f(a_2,a_1)$ for all values of a_1 and a_2. Also, $f(x_1,\ldots,x_n)$ is locally symmetric at point $[a_1,\ldots,a_2]$ if

$$f(\mathbf{a}+\nu), \quad [\nu_1,\nu_2,\ldots,\nu_n] \in [-\varepsilon,\varepsilon]^n \tag{5}$$

is identical for all permutations of perturbation vector ν. In terms of images and a ConvNet, $\Phi(x_a)$ must be identical for all permutations of ν.

This means that re-ordering the elements of ν must not change the output. This property is described on Figure 2. The background shows the value of $\Phi(x)$ in the region nearby the illustrated image in this figure. It is clear that $\Phi(x)$ is maximum given the clean image x. Assume two perturbation vectors ν_1 and ν_2 where ν_2 is obtained by re-ordering the elements of ν_1. It is expected that $\Phi(x+\nu_1) = \Phi(x+\nu_2)$ since probability density function of elements of ν_1 and ν_2 are identical and $\|\nu_1\| = \|\nu_2\| = \varepsilon$. Note that the perturbation vectors are not perceivable on the perturbed images to human eye in this figure. Furthermore, $\Phi(x)$ is not symmetric at the given image in this figure. Hence, one of them is classified as another class since it falls into a region where the classification score is low. Notwithstanding, if $\Phi(x)$ was symmetrical at the given image both perturbed images would be classified correctly. Consequently, symmetricity is an important property for being tolerant against small perturbations.

Note that an isotropic function is also symmetric. In addition, if a function is not isotropic, it is still possible that the function possess the symmetrical property. To empirically study local symmetricity of $\Phi(x)$, we performed the following procedure on each sample in dataset:

$$\forall_{r\in[\varepsilon,R]}\forall_{i\in\{1,\ldots,T\}} s_i^r = \Phi(x + permute(r\frac{\nu^r}{\|\nu^r\|})) \quad s.t.\ \nu^r = \mathcal{U}(-1,1). \tag{6}$$

Configuration of the parameters in this equation is

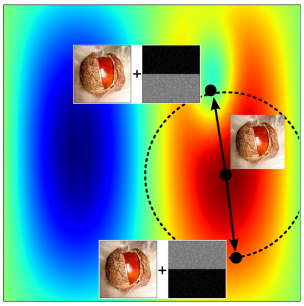

Figure 2: ConvNets must be symmetry at x.

similar to Section 2.1. Figure 3 shows the results on different ConvNets. The results suggest that except for adversarial example in distance less than ε from x, none of the ConvNets are symmetric.

We argue that being *locally* isotropic is an important property for having a more tolerant ConvNet to adversarial examples. To be more specific, we expect that all adversarial samples that are located at the same distance from the original samples to have identical scores. Radial Basis Function networks intrinsically possess this property since features that are located in an equal distance from the basis canters will have identical values. However, feature extraction in ConvNets is mainly based on convolution operations. Assume a convolution kernel $K = [k_{ij}] \in \mathbb{R}^{Q \times P}$ and two *different* adversarial examples $x_a^1, x_a^2 \in [-\varepsilon, \varepsilon]^{H \times W}$ where $\|x_a^1\| = \varepsilon$ and $x_a^2 = permute(x_a^1)$. Denoting the convolution operation by $*$, it is provable that $K * x_a^1 \neq K * x_a^2$ if $\exists_{ij} k_i > 0 \wedge k_j > 0$. To verify this, we study the convolution operation in the frequency domain. Convolution in spatial domain equals to multiplication in frequency domain. In other words, $K * x_a^1 = \mathcal{F}(K).\mathbb{F}(x_a^1)$ and $K * x_a^2 = \mathcal{F}(K).\mathbb{F}(x_a^2)$ where $\mathcal{F}(.)$ transforms the input into frequency domain. The term $K * x_a^1$ will be equal to $K * x_a^2$ if $\mathbb{F}(x_a^1) = \mathbb{F}(x_a^2)$. Since x_a^1 and x_a^2 are two different inputs, their Fourier transform will not be identical. Then, $\mathbb{F}(x_a^1) \neq \mathbb{F}(x_a^2)$ which shows that convolving the same filter with permuted inputs does not produce identical results. Notwithstanding, if $\|x_a^1\|$ is close to zero, the results of convolution operation becomes more comparable.

Extending this fact to ConvNets, we realize that output of the first convolution layer in a ConvNet will not be similar (except very few cases such as setting values of all weights to zero) given two inputs x_a^1 and x_a^2 where $x_a^2 = permute(x_a^1)$. Then, the output of the first layer may pass through a MAX-pooling layer where the outputs become more dissimilar. This is one explanation that why ConvNets in Figure 1 and Figure 3 are not isotropic and symmetrical.

Based on (3), one may argue that we can add regularization terms to the objective function in order to minimize the norm of gradient vector and Hessian matrices at each training sample. However, it should be noted that, this can make a function isotropic and symmetric in a very small region since we do not take into account higher order derivatives. Results in Figure 1 and Figure 3 shows that ConvNets are reasonably locally isotropic in very small region. As the result, regularizing by the aforementioned terms might not improve the stability significantly.

2.3 Topology Preservation

From one point of view, a ConvNet transforms a D_{input} dimensional input vector to a D_{output} dimensional vector in the layer just before the classification layer. For example, AlexNet transforms a $256 \times 256 \times 3$ dimensional vector to a 4096 dimensional vector in layer fc2.

Assume $X_{input} = \{X_{input}^1, \ldots, X_{input}^N\}$ is a set of D_{input} dimensional vectors each representing raw pixel intensities. Also, considering that $\Phi_L(X) : \mathbb{R}^{D_{input}} \rightarrow \mathbb{R}^{D_{output}}$ is the output of the L^{th} layer in a ConvNet, $\Phi_L(X_{input})$ returns set $X_{output} = \{X_{output}^1, \ldots, X_{output}^N\}$ where each element is obtained by applying $\Phi_L(x)$ on the corresponding element in X_{input}.

By defining a metric such as Euclidean distance, we can view X_{input} and X_{output} as two different topological spaces. While the topology of points in X_{input} is not suitable for the task of classification, topology of points in X_{output} has been adjusted such that the classes become linearly separable in this space. It is clear that topology of these two spaces are likely to be very different.

Now, assume set $x_{input}^{perturbed} = \{x + \nu_1, \ldots, x + \nu_N\}$ including perturbed examples of x where $\nu_i \in [-\varepsilon, -\varepsilon]^{D_{input}}$. While it is clear that topology of scattered points in X_{input} changes greatly using $\Phi_L(X)$ (because classification accuracy of raw points in X_{input} is usually much lower that points in X_{output}), we are not sure how $\Phi_L(X)$ affects the topology of points in $x_{perturbed}$. Note that points in $x_{input}^{perturbed}$ are very close together before applying $\Phi_L(X)$ on them.

Lets assume the simplest scenario where $\Phi_L(X) = XW$ and $W \in \mathbb{R}^{D_{input} \times D_{output}}$ is a weight matrix. In other words, we assumed that $\Phi_L(X)$ transforms the points to a new space by using a linear transformation. One way to show topology of X_{input} and X_{output} is to compute a distinct *distance matrix* for each of them where element ij in this matrix is obtained by computing $\|X^i - X^j\| = \|d_{ij}\| = d_{ij} d_{ij}^T$. Assuming $\Phi_L(X) = XW$, the element ij in distance matrix of the transformed space will equal to $\|X^i W - X^j W\| = \|(X^i - X^j)W\| = \|d_{ij} W\| = d_{ij} W (d_{ij} W)^T$.

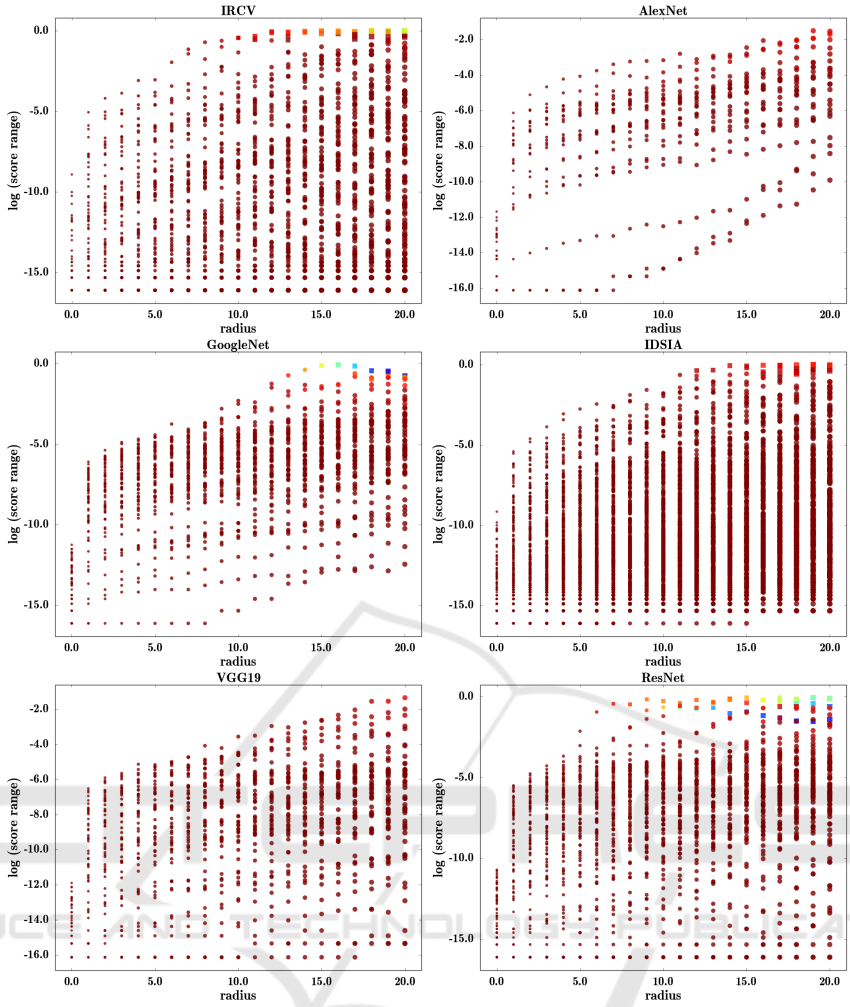

Figure 3: Symmetricity property of different ConvNets. Refer to text for detailed information.

Using the properties of matrix transpose, we obtain $d_{ij}W(d_{ij}W)^T = d_{ij}WW^T d_{ij}^T$. This means that the relation between distance matrix of X_{input} and distance matrix of X_{output} *is not* necessarily linear even when $\Phi_L(X)$ is a linear function. We say a transformation preserves the topology of X_{input} when the distances matrix of X_{output} is a linear function of distance matrix of X_{input}. For example scaling a set of two dimensional vectors does not change their topology since the distance matrix of the transformed points is a scaled version of the distance matrix of the original points. But, applying an affine transformation on them can change their topology.

Changing topology means that the distances between different points are manipulated nonlinearly. In other words, if the closest point to A is point C in the original space, the closest point to A might be point B in the transformed space. Our aim is to determine how a ConvNet affects topology of points in $x_{input}^{perturbed}$. Denoting the distance matrix of $x_{input}^{perturbed}$ with $\mathcal{D}_{input}$ and distance matrix of $x_{output}^{perturbed} = \{\Phi_L(x+\nu_1), \ldots, \Phi_L(x+\nu_N)\}$ with $\mathcal{D}_{output}$, we can compute:

$$\alpha = \mathcal{D}_{input}^{-1}\mathcal{D}_{output}. \quad (7)$$

If applying $\Phi_L(X)$ does not change the topology of $x_{input}^{perturbed}$, matrix $\alpha \in \mathbb{R}^{N\times N}$ will be *diagonal with identical values*. Even though α tells us how topology of points exactly changes after applying $\Phi_L(X)$ but it is not trivial to compute a score using α representing degree of non-linearity of topology changes. For this reason, we utilized Mantel test for comparing two distance matrices. Specifically, Mantel test compute the Pearson product-moment correlation coefficient ρ using many permutations of element of distance matrices. We say the relation between two matrices is linear when $|\rho| = 1$. To empirically study

this property of ConvNets we followed the procedure in (4) to generate adversarial examples in specific radii. Then, we computed the Mantel score between $x_{input}^{perturbed}$ and $x_{output}^{perturbed}$ for each ConvNet separately. Figure 4 shows the results.

We observe that topology of point does not linearly change even when they are very close to x. This is due to the fact that the Mantel score for all of ConvNets is $-0.1 < \rho < 0.1$. As we mentioned earlier, a simple linear transformation such as affine transformation changes the topology of points. If we think of ConvNets as fully-connected networks with shared weights, we realize that every neuron in this network applies the affine transformation $f(XW+b)$ on its inputs where $f(.)$ is an activation function. This affine transformation changes the topology of points. Considering a deep network with several convolution layers, the input passes through multiple affine transformations which greatly changes the topology of inputs. As the result, points located at distance ε from the original sample will not have the same topology at the output of a ConvNet.

2.4 Lipschitz

The method discussed in Section 2.3 takes into account all pair-wise distances between samples in order to compare topology of points before and after applying $\Phi_L(X)$. *Lipschitz* analysis is an alternative method to study non-linearity of a function. Specifically, given $X_1, X_2 \in \mathbb{R}^{D_{input}}$ and function $\Phi_L(X) : \mathbb{R}^{D_{input}} \rightarrow \mathbb{R}^{D_{output}}$, Lipschitz analysis finds a constant L called Lipschitz constant such that:

$$\begin{gathered} \|\Phi_L(X_1) - \Phi_L(X_2)\| \leq L\|X_1 - X_2\| \\ for\ all\ X_1, X_2 \in \mathbb{R}^{D_{input}}. \end{gathered} \tag{8}$$

This definition studies the global non-linearity of a function. Szegedy *et.al.* (Szegedy et al., 2014b) showed how to compute L for a ConvNet with convolution, pooling and activation layers. Notwithstanding, Lipschitz constant L found by applying (8) on whole domain of a ConvNet does not accurately tell us how output of the ConvNet changes *locally*. This problem is shown in Figure 5. We see that the purple function is more non-linear than the yellow function. This is due to the fact that its non-linearity is less when $|x| > 5$. Notwithstanding, degree of non-linearity of both function are similar when $-5 < x < 5$. The Lipschitz analysis in (8) does not take into account local non-linearity of a function. Instead, it find L which equals to greatest gradient magnitude in whole domain of the function.

Our aim is to study behaviour of function on adversarial examples. Therefore, we must compute Lipschitz constant L locally. To be more specific, denoting an adversarial sample with $x_a = x + \nu$ and a clean sample with x we find L_x such that:

$$\begin{gathered} \|g(\Phi_L(x_a)) - g(\Phi_L(x))\| \leq L_x\|h(x_a) - h(x)\| \\ for\ all\ \nu \in [-\varepsilon, \varepsilon]^{D_{input}}. \end{gathered} \tag{9}$$

where $g(.)$ and $h(.)$ are two function to normalize their input. From topology point of view, the above equation studies how adversarial examples are transformed by a ConvNet with respect to the original sample. If $L_x < 1$ for all adversarial samples, this means that $\Phi_L(X)$ attracts the adversarial examples toward the clean sample (They become closer to the clean sample after being transformed to D_{output} dimensional space by the ConvNet). However, the distance between adversarial examples and the clean example remains unchanged when $L_x = 1$ for all adversarial samples. Finally, $\Phi_L(X)$ repels the adversarial examples from the clean sample when $L_x > 1$.

A ConvNet will be more tolerant against adversarial samples when $L_x < 1$. This is due to the fact that when adversarial samples get closer to the clean sample, it is more likely that they have classification scores close to the clean sample. To empirically study the Lipschitz constant, we generated the samples using (4) and computed $\|\Phi_L(x+\nu) - \Phi_L(x)\|$ as well as $\|\nu\|$. It is worth mentioning that the clean samples as well as ν_i^r in (4) are the same for all the ConvNets trained on the same dataset. In addition, $g()$ and $h()$ are two separate min-max normalizers in which their parameters are obtained by feeding thousands of samples to each ConvNet and collecting the minimum and maximum value in the input and output of the ConvNet. Finally, each sample has a unique seed for the uniform noise function. This means that if we run the algorithm many times on different ConvNets for the sample i, the same adversarial examples will be generated in all the cases. By this way, we can compare the results from the ConvNets trained on the same dataset.

Figure 6 shows the relation between these two factors. In addition, the black and blue lines are obtained by fitting a first order (linear regression) and second order polynomial on data. Color of each point corresponds to the radius to which the adversarial sample is located. The colder color shows a smaller radius.

Even though (Szegedy et al., 2014b) mentioned that the **global** Lipschitz constant on AlexNet is greater than 1, our empirical analysis revealed that all of the ConvNets in our study are in general **locally** contraction. In other words, the Lipschitz constant on is less than 1 in *most* of the cases meaning that adversarial examples become closer to the original sample despite the fact that their topology changes by $\Phi_L(x)$. This suggests that that although ConvNet are

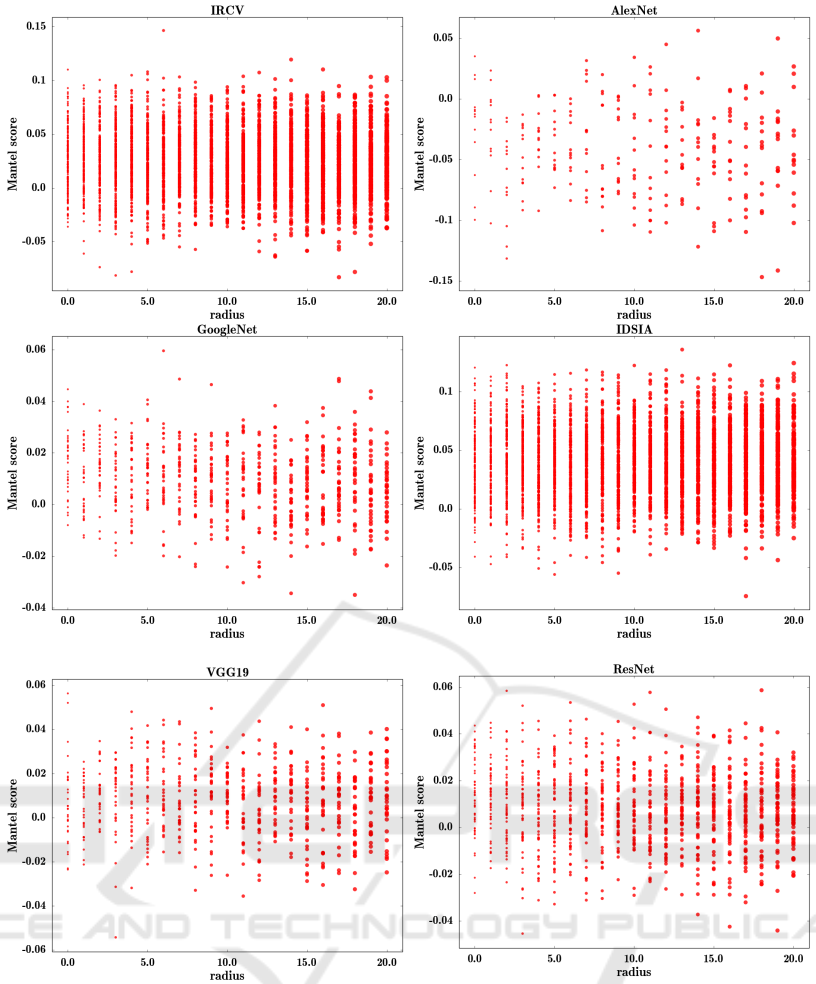

Figure 4: Topology preservation in ConvNets. Refer to text for detailed information.

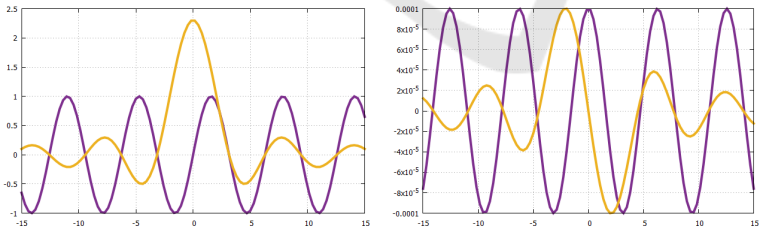

Figure 5: Two functions with identical Lipschitz constants. Left) Plot of two functions and Right) Derivative of two functions.

non-linear functions, they are generally locally contraction. As the result, explaining adversarial examples with global properties related to non-linearity of ConvNet might not be accurate.

3 CONCLUSION

In this paper, we empirically studied local properties of various ConvNets that are related to their vulnerability to adversarial examples. Specifically, we showed that state-of-art ConvNets trained on ImageNet and GTSRB datasets are not isotropic and symmetric around original samples. This means when we add two noise vectors with identical magnitudes to the clean sample, classification score of the adversarial examples might not be similar. We explained the reason in frequency domain. In addition, we studied how topology of adversarial examples located around the clean samples are affected by the ConvNet. We found

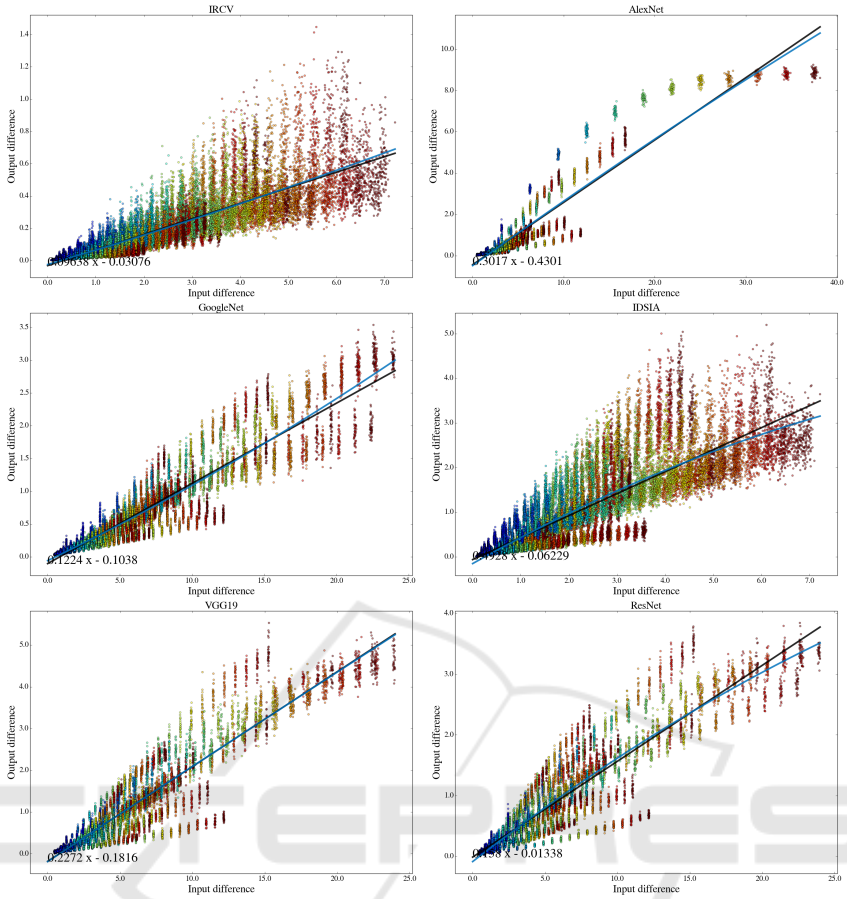

Figure 6: Topology preservation in ConvNets. Refer to text for detailed information.

that ConvNets change the topology of adversarial examples even when they are very close to clean samples. Finally, we analyzed the distance of adversarial examples in the input domain and the output of ConvNets. We found that adversarial examples are very likely to become closer to clean samples after being transformed by a ConvNet to a new space.

ACKNOWLEDGEMENTS

Hamed H. Aghdam and Elnaz J. Heravi are grateful for the supports granted by Generalitat de Catalunya's Agècia de Gestió d'Ajuts Universitaris i de Recerca (AGAUR) through the FI-DGR 2015 fellowship and University Rovira i Virgili through the Marti Franques fellowship, respectively.

REFERENCES

Aghdam, H. H., Heravi, E. J., and Puig, D. (2015). Recognizing Traffic Signs using a Practical Deep Neural Network. In *Robot 2015: Second Iberian Robotics Conference*, pages 399–410, Lisbon. Springer.

Aghdam, H. H., Heravi, E. J., and Puig, D. (2016). Analyzing the Stability of Convolutional Neural Networks Against Image Degradation. In *Proceedings of the 11th International Conference on Computer Vision Theory and Applications*.

Ciresan, D., Meier, U., and Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, number February, pages 3642–3649. IEEE.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). Explaining and Harnessing Adversarial Examples. *Iclr 2015*, pages 1–11.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep Residual Learning for Image Recognition. In *arXiv prepring arXiv:1506.01497*.

Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105. Curran Associates, Inc.

Simonyan, K. and Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recogni-

tion. In *International Conference on Learning Representation (ICLR)*, pages 1–13.

Szegedy, C., Reed, S., Sermanet, P., Vanhoucke, V., and Rabinovich, A. (2014a). Going deeper with convolutions. In *arXiv preprint arXiv:1409.4842*, pages 1–12.

Szegedy, C., Zaremba, W., and Sutskever, I. (2014b). Intriguing properties of neural networks.