Curse of Dimensionality in Adversarial Examples

Nandish Chattopadhyay*†, Anupam Chattopadhyay†, Sourav Sen Gupta† and Michael Kasper*
*Fraunhofer Singapore, Nanyang Technological University, Singapore
Email: nandish.chattopadhyay@fraunhofer.sg, michael.kasper@fraunhofer.sg
†School of Computer Science and Engineering, Nanyang Technological University, Singapore
Email: nandish001@e.ntu.edu.sg, anupam@ntu.edu.sg, sg.sourav@ntu.edu.sg

***Abstract*—While machine learning and deep neural networks in particular, have undergone massive progress in the past years, this ubiquitous paradigm faces a relatively newly discovered challenge, adversarial attacks. An adversary can leverage a plethora of attacking algorithms to severely reduce the performance of existing models, therefore threatening the use of AI in many safety-critical applications. Several attempts have been made to try and understand the root cause behind the generation of adversarial examples. In this paper, we try to relate the geometry of the high-dimensional space in which the model operates and optimizes, and the properties and problems therein, to such adversarial attacks. We present the mathematical background, the intuition behind the existence of adversarial examples and substantiate them with empirical results from our experiments.**

***Index Terms*—Deep learning, Neural networks, Adversarial attacks, High-dimensional space, Geometry, Norm.**

I. Introduction

In the recent past, there has been a steady growth in the proliferation of machine learning as a paradigm, spearheaded by advancements in the field of deep learning, with neural network architectures reaching human accuracy level, and in some specific tasks, achieving super-human performances [1]. More and more areas of science and technology have adopted the data driven way, giving rise to increasing enthusiasm in AI and related applications, among the industry and academia alike. Traditionally, machine learning or statistical learning methods involved a considerable amount of data pre-processing and feature engineering, and the hand-crafted data was provided to the models [2]. Arguably, this lacked generalization to some extent, and model performances in terms of accuracy measures were also reaching a saturation level.

Deep learning models provided the much needed breakthrough. While the perceptron algorithm was an old one [3], and neural networks being in the scene for a long time, their true potential was realized only after the advent of the required processing capacity, in terms of hardware resources. The inherent linear nature of the neural network architectures, arising from the sequences of linear transformations, was capitalized heavily to parallelize operations among multiple GPU cores, therefore speeding up the training process using backpropagation algorithm. These deep networks, with their massive set of tuning parameters, possess enough learning capacity to learn features and patterns in the data, to extents which were not reachable before.

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore, under its International Research Centres in Singapore Funding Initiative.

A. Motivation : Adversarial Examples

Soon after the boom in flourishing AI, also popularly referred to as the "AI revival", a new discovery proved to a newly found problem. In 2015, it was shown that the high performing neural networks were vulnerable to adversarial attacks [4]. The models, which would perform the task of classification very accurately on a test image dataset, would perform very poorly on a slightly tweaked dataset of those images, which the adversary could generate. Introduction of little structured perturbations, unobservable to the human eye, can bring about an unprecedented degradation in model performance, with a lot of misclassification [5]. Later, it was also shown that these adversarial examples were transferable between models. That is, an adversarial example generated for a particular neural network architecture, would also act like an adversarial example to a seemingly different model, for example a support vector classifier [6]. Subsequently, a plethora of attack mechanisms were developed by different groups of researchers working in this particular domain, and each of them cause different extents of damage to the performance of the trained model [7].

Naturally, there was effort to understand why such a phenomenon happens. Goodfellow et al. [8] attributed the linear nature of the neural networks to be the primary reason behind such attacks. Other works suggested otherwise [9], putting the blame on dimensionality [10]. It is worth noting that the models trained for classifying images work in a very high dimensional landscape. The properties of high dimensional spaces are quite counter-intuitive and the geometry is different from what one would normally expect in low dimensions, behaviorally. While some of it has been mathematically modeled quite rigorously, there are some gaps as well in the literature.

B. Contribution : Curse of Dimensionality

In this paper, we study the effect of data dimensionality in case of adversarial examples, and strongly second the connection proposed by Gilmer et al. [10]. In fact, we go one step further, and hypothesize that adversarial examples are *easier* to generate on a dataset with *higher* dimension. We explore some interesting and relevant properties of the high dimensional space and present our empirical study on some models and datasets to test our hypothesis.

978-1-7281-2009-6/$31.00

In particular, we address the issue of why it is easy to generate such adversarial examples, how that behaves with different dimensionality of input feature vectors, and why it is difficult to measure adversarial perturbation at high dimensions using the standard distance measures like L_1 and L_2 norms.

Our contributions primarily comprise of two components:

- *Theoretical Justification*: Mathematical and statistical formulation relevant in the context of classification of high-dimensional images to justify the effects of dimensionality on the generation of adversarial examples.
- *Experimental Verification*: Empirical study, with extensive experimentation on image datasets with varying dimensions to understand the effect of dimensionality on the generation of adversarial examples.

C. Organization of the Paper

In Section II, we address the adversarial attack in a formal way to introduce the mathematical setup. We follow it up, in Section III, with the theoretical formulation needed for our analysis, including the relevant mathematical results and statistical properties. In Section IV, we present the experimental setup and observations in support of our hypothesis, finally culminating in some summative remarks in Section V.

II. Adversarial Examples in Machine Learning

With the evolution of standard machine learning into the realm of deep neural networks, the task which has seen maximum progress is supervised image classification [11]. Ironically, adversarial attacks have been first observed for images as well. Given a image classifier model, for a correctly classified test data point, a corresponding adversarial example would be a hand-crafted image created by introducing a very little perturbation imperceptible to the human eye, which would be wrongly classified by the same model. This notion holds good for any machine learning system.

Typically, an image classification system would have four key components. First, is the model. It could be some neural network architecture or some other machine learning model like a maximum margin classifier (SVM) [12]. Second, there should be a training dataset, which is of course a sample drawn from the population of all imaginable images of a particular selected resolution. The training dataset is labelled for the purpose of supervised learning. Third, there is a test dataset, which is part of the population that doesn't belong to the training set. And finally, after the model is trained on the training dataset, we obtain the trained manifolds for the different classes, which are separated by the classifier [13].

As an illustration of how adversarial examples are generated, let us consider a relatively simple binary classification problem. Figure 1 shows a 2D representational projection of the setup. The actual truth is the real world around us. In the context of this problem, that is, for the specific classification task, we have the notion of the *population*, which is exhaustive in nature, and is simply a 3D to 2D approximation of the real world. One may note that it is not possible to obtain its realization. For the task of image classification, as mentioned

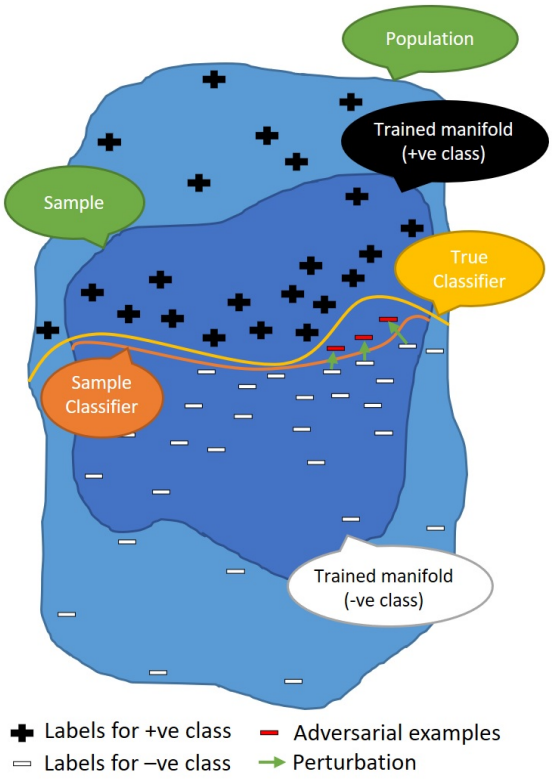

Fig. 1: Adversarial Examples in context of a Classifier.

earlier, the population is the set of all images of the specific set resolution in which we are capturing the images. We assume that there exists a *true classifier*, with respect to the population, that correctly classifies all images. A decent manifestation of this true classifier would be the human vision system. Had this classification model been known, there would not have been any requirement for machine learning.

In an attempt to approximate the true classifier, we have to consider a sample from the population, and try to build models to classify the sample space into trained manifolds corresponding to the classes. The so-developed classifier is the *sample classifier*. The sample space could be realized as a vast collection of data points, a dataset of images that is, for image classification. The sample classifier would typically be a deep neural network or a support vector machine (SVM) or some other hyperplane classifier. For simplicity of nomenclature we assume that the true classifier for the population is indistinguishable from the true classifier within the sample, although it may so happen in practice, that the restriction of the classifier to the sample is dependent on the sample itself.

As evident from Figure 1, there might naturally exist a "gap" between the two classifiers or separating hyperplanes, arising due to the approximation, made on an in-exhaustive sample. This gap results in generating an *adversarial space*, wherein any point would naturally belong to two different manifolds, with respect to the two aforementioned classifiers,

and therefore be *adversarial* in nature. It is interesting to note that if the points are close to the boundary of the individual trained manifolds, a little perturbation may shift some of the points across the sample classifier into the adversarial space, wherein they will be misclassified by the model, but not by the human annotator (true classifier). If the perturbation is too much, then the data point may move across the true classifier as well. That is, if the perturbation is greater than a particular threshold, the human annotator will misclassify it too, and that wouldn't be an adversarial example any more.

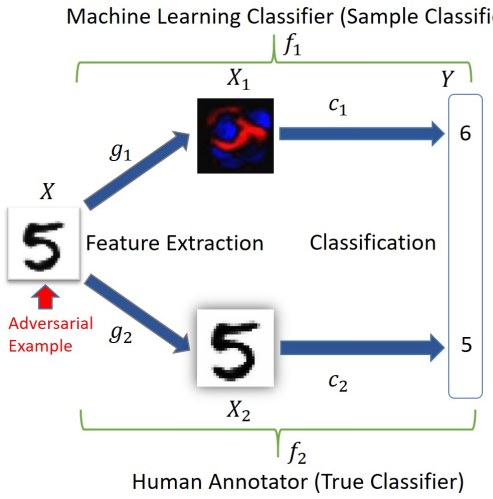

Fig. 2: Generation of Adversarial Examples.

As shown in Figure 2, X is the space of input samples. For images as input to the system, X would trivially be the vector space of dimensions equaling the number of pixels in the image. The two classifiers that we are considering here are f_1 (sample classifier) and f_2 (human annotator). The classifiers have two components each, feature extraction and classification. X_1 is the feature space for the sample classifier and X_2 is the feature space for the human annotator, where d_1 and d_2 are norms defined in the spaces X_1 and X_2 respectively. As shown, $f_1 = c_1 \circ g_1$ and $f_2 = c_2 \circ g_2$ [14].

Let us consider $x \in X$, a training sample. Given x, the corresponding adversarial example x^*, for a norm d_2 defined on the space X_2, and a predefined threshold $\delta > 0$, satisfies:

$$\begin{aligned} f_1(x) \neq f_1(x^*) \quad &\text{and} \quad f_2(x) = f_2(x^*) \\ \text{such that} \quad &d_2(g_2(x), g_2(x^*)) < \delta \end{aligned} \tag{1}$$

We have two key takeaways from the above formulation. First, the generation of adversarial examples with little perturbation is facilitated by the existence of many data points near the boundary of the trained manifolds of the classes, and second, that the adversarial perturbation is bounded in practice.

III. Theoretical Formulation

In this section, we concentrate on the core mathematical and statistical properties of two crucial ingredients in understanding adversarial examples – the trained manifolds of a classifier, and the measure for adversarial perturbation.

A. The Trained Manifolds

Since we wish to try and understand why bounded perturbations to data points cause adversarial behaviour, it is natural to focus our attention on the geometry of the high-dimensional trained manifolds for the respective classes. In particular, it is important to consider the distribution of data points on the manifold. Certain properties of high-dimensional objects, like most of their volume being near their surface, could be important in understanding how small perturbations can shift data points across the classifier hyperplane.

Let us consider an object A in a d-dimensional space R^d. If we shrink A by a little amount ϵ to produce a new object $(1-\epsilon)A = \{(1-\epsilon)x \mid x \in A\}$, we have the following:

$$\text{volume}((1-\epsilon)A) \;=\; (1-\epsilon)^d \;\text{volume}(A) \tag{2}$$

To show that this holds, we partition A into infinitesimal *cubes*. Then, $(1-\epsilon)A$ is the union of a set of cubes obtained by shrinking the cubes in A by a factor of $(1-\epsilon)$. When one shrinks each of the $2d$ sides of a d-dimensional cube by a factor of $(1-\epsilon)$, its volume naturally shrinks by a factor of $(1-\epsilon)^d$. Thus, for any object A in R^d, we have:

$$\frac{\text{volume}((1-\epsilon)A)}{\text{volume}(A)} \;=\; (1-\epsilon)^d \;\leq\; e^{-\epsilon d} \tag{3}$$

If we fix ϵ and let $d \to \infty$, the right hand side of the inequality in Eq. (3) rapidly approaches zero. This means that nearly all the volume of A must be in the portion of A that doesn't belong to the region $(1-\epsilon)A$. That is, nearly all the volume of A is close to its surface in a high-dimensional setting.

We will use this idea to make some distribution assumptions of the data points on the manifold. We present the relevant mathematical properties of two most popular distributions, the Uniform distribution and the Gaussian distribution, at high-dimensions, as is available in the literature [15]. It may be mentioned here that there is scope of extending this mathematical formulation for other distributions as well.

Uniform Distribution: Let us consider that the points are uniformly distributed within the manifold. If we consider a unit ball in d dimensions A, from Eq. (3), it follows that at least a $1 - e^{-\epsilon d}$ fraction of the volume of the unit ball is concentrated in $A \setminus (1-\epsilon)A$, which means in a small annulus of width ϵ at the boundary. It can be mentioned here that, most of the volume of the d-dimensional unit ball is contained in an annulus of width $O(1/d)$ near the boundary. Generalizing to a d-dimensional ball with radius r, the width of the annulus would be $O(r/d)$. Another very interesting fact about the ball in high dimensions is that most of its volume is concentrated near the equator. One can note that for any unit vector defining "north", most of the unit ball's volume lies in the thin slab of points which have a dot-product with that vector in the magnitude of $O(1/\sqrt{d})$. In particular, it can be shown that at least a $1 - \frac{2}{c}e^{-c^2/2}$ fraction of the volume of the d-dimensional

unit ball has $|x_1| \leq \frac{c}{\sqrt{d-1}}$, as shown in Figure 3, for any $c \geq 1$ and $d \geq 3$. Thus, if we consider the d-dimensional ball as the geometry of the manifold (which minimizes surface area), most of the points will be near the boundary.

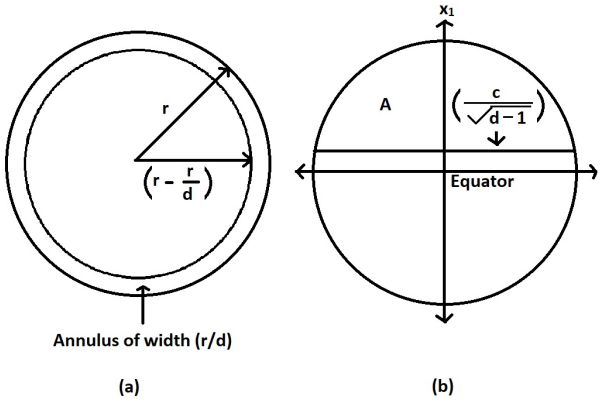

Fig. 3: Volume Distribution in a Uniform d-Dimensional Ball.

It may also be noted that apart from the uniform d-dimensional ball, for any other arbitrary geometry of the manifold, the surface area would only be greater. Thus, under the assumption of uniformly distributed data points, the points will be nearer to the boundary of the manifold, thereby facilitating the generation of adversarial examples.

GAUSSIAN DISTRIBUTION: The typical 1-dimensional Gaussian has most of its mass near its mean (say, the origin). But at high-dimensions, like in the case of the manifolds in discussion, the behaviour changes significantly. The *Gaussian Annulus Theorem* [15] states that for the d-dimensional spherical Gaussian with unit variance in each direction, for any $\beta \leq \sqrt{d}$, all but at most $3e^{-c\beta^2}$ of the probability mass lies within the annulus $\sqrt{d} - \beta \leq |x| \leq \sqrt{d} + \beta$, where c is a fixed positive constant. Though the density is maximum at the origin, there is very little volume there. Nearly all of the probability is concentrated in a thin annulus of width $O(1)$ at a radius of $\sqrt{d}$. Thus, even under the assumption of Gaussian distribution of data points, the points will mostly be nearer to the boundary of the manifold, thereby facilitating the generation of adversarial examples.

INTUITION: There is not much theoretical treatment of behaviour of other distributions at high-dimensions. However, though not complete enough to establish causality rigorously, from what we have seen, it is fair to derive the intuition that the concentration of data points near the boundaries or surfaces of the trained manifolds, away from the respective centres, irrespective of exact distribution assumption, is a key ingredient in the process of adversarial behaviour of slightly perturbed data points, which are misclassified by the model. It is therefore natural to check whether this notion holds good in practice or not, by using adversarial attacks on models trained on different datasets of varying dimensions. We have reported our empirical findings in Section IV, later in this paper.

B. Measuring Adversarial Perturbation

The other important aspect about adversarial attacks is the measurement of perturbation. As evident from the formal definition of adversarial examples in Eq. (1), the structured perturbation introduced to a data point to convert it into an adversarial example, is bounded by a small quantity δ, measured with respect to a norm defined in that space. We are studying the relevance of dimensionality in the context of adversarial attacks, and it is therefore fair to be interested in knowing how the extent of perturbation vary with different dimensions of the trained manifolds in which the data points are in. In the previous subsection, we presented the theoretical basis behind the ease of generation of adversarial examples, when operating in high-dimensions. Quite naturally, one should be keen to look at the correlation (if any) between the variation of perturbation and varying dimensions. Intuitively, one might expect that as we move to higher and higher dimensions, with the ease of generation of the attacks, the perturbations might decrease. But, this is difficult to test, in practice. The reason behind that, lies in the failure of standard metrics of distance, like the L_p class of norms, at high dimensions [16]. Ironically, this is also an effect of the curse of dimensionality. We argue that L_p norms are quite meaningless as a measure of adversarial distance at high dimensions.

L_2 NORM AT HIGH DIMENSIONS: To begin with, let us make an assumption on the distribution of the data points, which we will relax later. If a number of random points are generated in a d-dimensional space using a Gaussian distribution, then the distance between all pairs of points will essentially be the *same* when d is sufficiently large. Consider the L_2 norm, the square of the distance between two data points, that is, two d-dimensional images in our case. Let the two images be the original test image y, and the corresponding adversarial example, z. Their distance, in L_2 norm, satisfies

$$\|y - z\|_2^2 \;=\; \sum_{i=1}^{d} (y_i - z_i)^2, \tag{4}$$

which can be seen as the sum of d independent samples of some random variable x that is essentially a distribution of the square of difference of two Gaussian distributions. More precisely, the square of the distance is the sum of independent samples $x_i = (y_i - z_i)^2$ of some random variable x of bounded variance. The Law of Large Numbers [17] dictates that with high probability, the average of the samples are close to the expectation of the particular random variable. Formally, it states that, if we have $x_1, x_2, x_3, \ldots, x_n$ to be n independent samples of any random variable x with finite variance, then

$$\Pr\left(\left|\frac{x_1 + x_2 + \ldots + x_n}{n} - E(x)\right| \geq \epsilon\right) \leq \frac{Var(x)}{n\epsilon^2} \tag{5}$$

In the d-dimensional space, $n \approx d$ is sufficiently large for this particular convergence of the notional 'distance metric' random variable to the expectation of the random variable. This provides the basis for the failure of distance metric L_2.

L_p NORM AT HIGH DIMENSIONS: A more generic result, on the failure of the L_p class of norms in its ability to act a metric of distance and perform the basic task of discriminating between the nearest and the furthest points, at high dimensions, is presented below [18]. Note that the Law of Large Numbers is more generic and hence we can relax our previous Gaussian assumption on the distribution.

Let us consider a d-dimensional data space, as earlier. Let N be the number of data points, F be a 1-dimensional data distribution in $(0, 1)$ and X_d be a particular data point from F^d with each coordinate drawn from F. In the context of images as the data points, X_d would be a d-dimensional image (the dimension being fixed based on resolution) with scaled pixel values. Let the distance between $(x^1, \ldots, x^d)$ and $(y^1, \ldots, y^d)$ using the distance metric $L_p = \{\sum_{i=1}^{d} |x_1^i - x_2^i|^p\}^{1/p}$ be denoted as $dist_d^p(x, y)$. Let $||\cdot||_p$ be the distance of a vector to the origin (our reference point for analysis) using the function $\text{dist}_d^p(\cdot, \cdot)$. Let $\text{E}[x]$ and $\text{var}[X]$ be the expected value and the variance of the random variable X.

Define $D_{max}^{p,d} = \max\{||X_d||_p\}$ and $D_{min}^{p,d} = \min\{||X_d||_p\}$. Under the assumption that the distribution behaves a certain way as d increases, we have:

$$\lim_{d\to\infty} \text{var}\left(\frac{||X_d||_p}{E[||X_d||_p]}\right) = 0 \quad \Rightarrow \quad \frac{D_{max}^{p,d} - D_{min}^{p,d}}{D_{min}^{p,d}} \xrightarrow{p} 0,$$

where the probabilistic convergence $Z_d \xrightarrow{p} c$ means that a sequence of vectors $Z_1, \ldots, Z_d$ converges in probability to a constant vector c if $\forall \epsilon > 0$, $\lim_{d\to\infty} \Pr[\text{dist}_d(Z_d, c) \leq \epsilon] = 1$.

This gives a clear indication that at high-dimensions, the L_p norm is unable to discriminate between two data points (test image and its corresponding adversarial example, in this context), which are nearest and furthest from a reference point (origin of the manifold, in this case). It is quite meaningless therefore, to use the L_p norm as a metric of distance in adversarial analysis. We substantiate this intuition through experiments on different datasets, as in Section IV.

IV. EXPERIMENTS AND OBSERVATIONS

In this section, we present our experimental findings and empirical results that support the hypothesis that we built up thus far. In order to make the assertion that the "generation of adversarial examples is easier in higher dimensions", we wish to show that under otherwise similar experimental setup, the adversarial attacks work better at situations which involve higher dimensions of the input feature vectors.

A. Experimental Setup

MODEL: We used a neural network model, some standard datasets in the context of image classification and two popular adversarial attack methods. The deep convolutional neural network that we have used is the typical implementation of the VGG network [19], which was proposed by the Visual Geometry Group at University of Oxford. The network has four feature extraction blocks comprising of convolution and pooling layers, followed by a multi-layer perceptron of three fully connected layers. The first three blocks designed for the purpose of learning the features include two 2-dimensional convolutional layers, followed by a maximum-pooling layer. The fourth block consists of one convolutional layer and one maximum-pooling layer. This is followed by the dense layers, which is a fully connected multi-layer perceptron with three linear layers. Little modifications were made from time to time, depending upon the dataset being used. Figure 4 shows a schematic diagram of the neural network model.

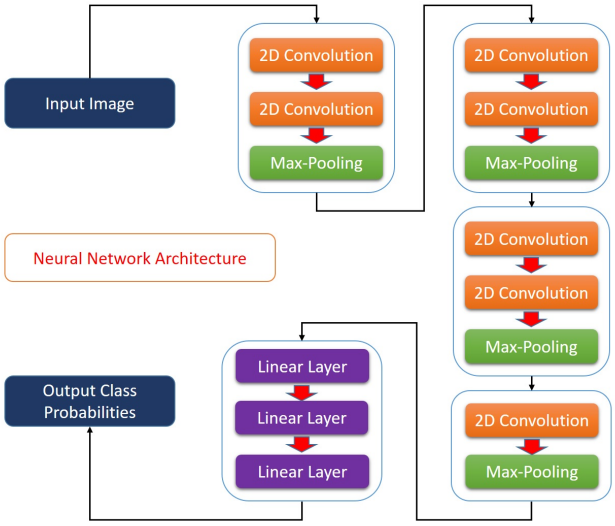

Fig. 4: Architecture of the VGG [19] Neural Network.

DATASET: We used three datasets, which are widely used in the study of image classification tasks. The first one is the MNIST dataset [20], a collection of images of handwritten digits. Each image has a resolution of 28×28 pixels, and are gray-scale in nature with 10 classes corresponding to the 10 digits, and therefore is quite a low-dimensional feature vector (784 pixels) in the context of our analysis. The second dataset is the CIFAR-10 dataset [21], which is a collection of coloured images of everyday objects. Each image has a resolution of 32×32 pixels, across the three fundamental colours (red, green, blue) and is a comparatively higher dimensional input feature vector (3072 pixels). Here also, there are 10 classes. It is worth noting here, that to conduct extensive tests to understand the behaviour of the models at different dimensions, we modified the datasets as required and this is explained in further details in the next segments.

ADVERSARIAL METHOD: The two methods of introducing perturbations into images to create adversarial examples, used in this analysis, are the Fast Sign Gradient method and the Momentum Iterative Fast Sign Gradient method. The Fast Sign Gradient method was first introduced by Goodfellow et al. [22] in 2014. This is an effective adversarial generation technique, that uses L_∞-norm as the distance measure d_2 in Eq. (1). For this adversarial attack method, the natural choice is to make the attack strength the same at every feature dimension. The

perturbation r to be introduced to a clean sample x, to turn it into an adversarial one, is the solution of:

$$\arg\min_{r}(c\ d_2(x, x+r) - \text{Loss}(f_1(x+r), f_1(x))), \quad (6)$$

where $x+r \in [0,1]^p$, with p being the total number of features and c being the constant for the Lagrange multiplier. In other words, the adversarial example x^* can be obtained from a clean sample x by maximizing the loss function $J(x^*, y)$, where J is usually cross-entropy loss and y is the class label. The Fast Sign Gradient method satisfies the L_∞ norm bound of $||x^* - x||_\infty \leq \epsilon$ and is therefore obtained as:

$$x^* = x + \epsilon \cdot sign(\nabla_x J(x, y)), \quad (7)$$

where $\nabla_x J(x, y)$ is the gradient of the loss function w.r.t x.

Apart from this one-step gradient based approach, it is also possible to create adversarial attacks in an iterative way, using the similar concept of attack. These methods typically apply the fast sign gradient many times, with a small step size α. An iterative version of FSGM is:

$$x_0^* = x, \quad x_{t+1}^* = x_t^* + \alpha \cdot sign(\nabla_x J(x_t^*, y)) \quad (8)$$

In order to ensure that the adversarial examples generated in this process are restricted within the ϵ bound with respect to the L_2 or L_∞ norm, one could clip X_t^* in the ϵ neighborhood of x or set $\alpha = \epsilon/T$, with T being the number of iterations.

The Momentum Iterative Fast Gradient method was introduced by Dong et al. [23], in 2018 and belongs to this class of attacks. The momentum method in general is a technique for accelerating gradient descent algorithms by accumulating a velocity vector if the gradient direction of the loss function across the iterations. The memorization of the previous gradients helps to traverse efficiently though the landscape. This idea is used to generate adversarial examples efficiently.

B. Experimental Design

In an attempt to support the hypothesis discussed earlier, we carried out extensive experiments to understand the behaviour of adversarial attacks with dimensionality. For each of the individual experimental settings, the performance metric for the models and the procedural scheme were kept constant, to facilitate comparability wherever needed. The measure of performance used throughout, is the classification accuracy (used synonymously as performance hereafter), that is the percentage of correct classifications made by a particular model on a particular set of samples, be it the train-set, test-set or a set of adversarial examples created by some attack methods (each combination is referred to as a setting).

The procedure adopted for the analysis on any given setting is three-fold. Firstly, the neural network is trained on a training dataset, and the hyper-parameters are tuned to obtain the best in class performance. Thereafter, the model is put to test against the test dataset, and the performance is noted. Finally, the same test dataset is used to create the adversarial examples, with respect to the trained model (using the gradients) using the two attack mechanisms mentioned earlier [24]. It must be noted that the hyper-parameters of the attack mechanisms are kept fixed throughout, for comparability. The tuned neural network model is then tested against the artificially created adversarial examples set. The performance of the model is recorded. In a given experimental setting, the comparison of the trained model's performance on the particular test-set and its corresponding adversarial examples set would give us an indication of whether the setting is favorable or not for adversarial attacks. The datasets used have been mentioned earlier, the hand-written character recognition MNIST dataset, the CIFAR-10 dataset and the ImageNet dataset. We carried out a few different kinds of studies for the analysis, as follows.

VARYING DIMENSIONALITY: To understand whether or not greater number of features and higher inherent dimensionality of the datasets affect the generation of adversarial examples from a particular sample of data points, as is the hypothesis, we carried out the three-fold procedure on three datasets of very different individual size of features. The comparison of performance of the model should therefore be indicative of empirical results to substantiate the claimed hypothesis.

UP-SAMPLED RESOLUTION: To understand whether or not the raw resolution of an input image has any impact on the generation of adversarial examples, we experimented with upsampling the resolution of images without adding any new information, that is without changing the inherent dimensionality of the dataset. This was carried out using a nearest neighbor interpolation approach of the pixels in the image, by doubling, quadrupling the resolution of the images. Following that, the same three-fold procedure was carried out on the images.

DIMENSION REDUCTION: To understand whether or not the inherent dimensionality of the dataset has an impact on the creation of adversarial examples, we experimented with reducing the dimension of the respective datasets, using standard techniques like the singular value decomposition. While accomplishing this task, we created four different settings, that is combinations of train-set, test-set and adversarial set, from the original dataset, which contained 99%, 95%, 90% and 80% of the actual information. One may note here that, for singular value decomposition, the information content of each of the data slices or components corresponding to each of the singular values can be obtained from its surrogate, the proportion of the singular values themselves. Following this decomposition, we carried out the same three-fold process.

MEASUREMENT OF DISTANCE: In order to understand whether the standard metrics of distances, like the L_2 norm, can be used as a good metric of the notion of distance in the high-dimensional spaces, we performed some experiments. We studied the distribution of the individual pair-wise distances between the images belonging to a particular class. We also looked at the distribution of the adversarial perturbation, meaning the pair-wise distances between the clean samples and the adversarial samples. We obtained a superimposed plot of the distributions to look at the efficacy of the distance measure in question, the L_2 norm.

C. Experimental Results

In this section, we present the observations of the various experiments mentioned in the previous subsections. For each of the settings, which is a combination of a train-set, a test-set and two adversarial sets created using two types of attack methods Fast Sign Gradient method (FSGM) and the Momentum Iterative Fast Sign Gradient method (MI-FSGM), we have reported performances of the trained model.

Table I shows the performances of the individually trained neural network on three datasets of differing sizes. For each dataset, the corresponding number of pixels in each image is provided within parenthesis. The performance of the trained model is tested on the test set and the two adversarial datasets, created using the two types of attacks mentioned above. It is clearly evident that with the growing inherent dimensionality of the datasets, the performance of the neural network on the adversarial datasets reduces significantly. That is, adversarial examples are generated better for higher dimensionality.

TABLE I: Adversarial Attacks and Dimensionality.

Dataset	Performance on Test Set	Performance on Adversarial Set (FSGM)	Performance on Adversarial Set (MI-FSGM)
MNIST (28x28)	98.8%	55.8%	51.7%
CIFAR-10 (3x32x32)	84.6%	14.2%	13.8%
ImageNet (3x224x224)	78.2	9.5%	6.8%

Table II shows how the generation of adversarial examples is affected by up-sampling the resolution of the images. Like earlier, for each of the datasets, the number of pixels in each of the images is provided in parenthesis. The performances of the individually trained models are presented corresponding to each of the settings mentioned earlier, on the test set and the two adversarial datasets, created using the two types of attacks mentioned above. From the results of the experiments, one can see that for the two datasets used, with three variants for each (in terms of resolution), the performance of the attacks observed, was very close to each other consistently for both the datasets. The conclusion that follows this observation is that there is no significant effect of changing the mere resolution of the images (without adding any new information, therefore not affecting the inherent dimensionality of the data) on the performance of the neural network on the adversarial dataset.

Table III presents the observations in the study of generating adversarial examples from the dimension reduced versions of the datasets, as explained in the earlier segment. For each of the datasets, only those components are considered which make up for a particular proportion of the overall information, as indicated in parenthesis in the table. The performance of the trained model is tested on the test set and the two adversarial datasets, created using the two types of attacks mentioned above. One can observe from the empirical findings that with dimension reduction, the performance of the neural network in both the datasets have improved, indicating that adversarial attacks become less effective with dimension reduction.

TABLE II: Adversarial Attacks on Up-Sampled Images.

Dataset	Performance on Test Set	Performance on Adversarial Set (FSGM)	Performance on Adversarial Set (MI-FSGM)
MNIST (28x28)	98.8%	55.8%	51.7%
MNIST (56x56)	98.6%	55.8%	51.5%
MNIST (112x112)	98.8%	55.6%	51.7%
CIFAR-10 (3x32x32)	84.6%	14.2%	13.8%
CIFAR-10 (3x64x64)	85.1%	13.9%	13.8%
CIFAR-10 (3x128x128)	84.9%	14.1%	13.7%

TABLE III: Adversarial Attacks and Dimension Reduction.

Dataset	Performance on Test Set	Performance on Adversarial Set (FSGM)	Performance on Adversarial Set (MI-FSGM)
MNIST (99%)	99%	56.1%	54.3%
MNIST (95%)	97.8%	58.3%	57.1%
MNIST (90%)	97.5%	59.1%	57.5%
MNIST (80%)	96.8%	64.2%	61.1%
CIFAR-10 (99%)	83.2%	14.2%	13.7%
CIFAR-10 (95%)	80.1%	15.9%	14.8%
CIFAR-10 (90%)	77.3%	17.8%	15.2%
CIFAR-10 (80%)	74.8%	19.8%	17.6%

Figure 5 shows the distributions of distances superimposed with the distribution of adversarial perturbation (plotted in blue). For this study, we considered the CIFAR-10 dataset, which has 3072 pixels per image. We plotted the distribution of the pair-wise L_2 norm distances of all points belonging to a particular class (L_2 norm distance in the X-axis and the probabilities in the Y-axis) and then repeated the process for all the ten classes in the dataset, shown in different colours. Then, we superimposed that plot with another plot of the distribution of adversarial perturbation, that is the pair-wise L_2

norm distances between clean samples and their corresponding adversarial samples, shown in blue. From the figure, one can observe that the distribution of the pair-wise distances of the images within the classes are highly overlapping, and is centred around the value of $\sqrt{d}$ (where d is the number of pixels). Interestingly, the plot in blue, which is the distribution of the adversarial perturbations, and is also found to have a peak at around $\sqrt{d}$ (numerically around 56). This indicates that as suggested in the earlier sections, the distance metric fails to provide meaningful measures at high dimensions as all measures of distance tend to converge to the numeric value of the square root of the dimensionality.

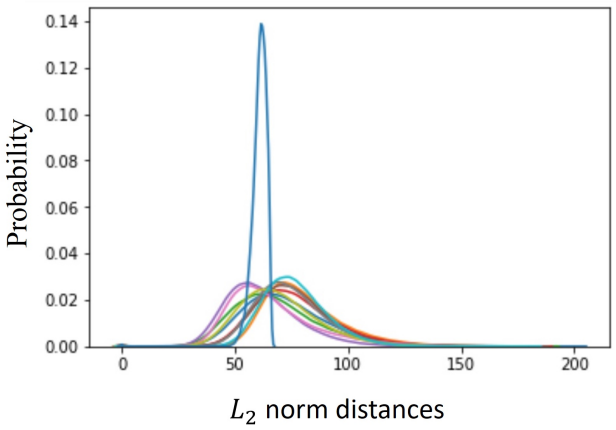

Fig. 5: The Distribution of Pairwise Distances.

V. Conclusion

In this paper, we hypothesize that generation of adversarial attacks in case of neural networks benefits from higher dimensionality in data. To support our hypothesis, we present the theoretical formulation of the problem, and present our intuition in terms of high-dimensional geometry. We also present extensive experimental results in support of our hypothesis. We also argue the futility of using standard L_p norms as a distance measure in case of high-dimensional manifolds (the feature space of neural networks), especially while analyzing the generation of adversarial examples.

We would like to view this work as an initial attempt at connecting adversarial attacks to the inherent dimensionality of data, both in terms of the theoretical intuition, as well as through empirical evidence. It will be quite interesting to extend the idea to obtain a comprehensive understanding of adversarial examples and their exact relationship with the inherent dimension and geometry of the data.

Acknowledgments

The authors would like to thank the anonymous reviewers of IJCNN 2019 for their valuable feedback and constructive criticism that helped in significantly improving the technical and editorial quality of the paper.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, NY, USA:, 2001, vol. 1, no. 10.

[3] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[4] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[5] I. Goodfellow, P. McDaniel, and N. Papernot, "Making machine learning robust against adversarial inputs," *Communications of the ACM*, vol. 61, no. 7, pp. 56–66, 2018.

[6] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *arXiv preprint arXiv:1605.07277*, 2016.

[7] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "Adversarial attacks and defences: A survey," *CoRR*, vol. abs/1810.00069, 2018. [Online]. Available: http://arxiv.org/abs/1810.00069

[8] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples (2014). arxiv preprint," *arXiv preprint arXiv:1412.6572*.

[9] S. Dube, "High dimensional spaces, deep learning and adversarial examples," *arXiv preprint arXiv:1801.00634*, 2018.

[10] J. Gilmer, L. Metz, F. Faghri, S. S. Schoenholz, M. Raghu, M. Wattenberg, and I. Goodfellow, "Adversarial spheres," *arXiv preprint arXiv:1801.02774*, 2018.

[11] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[12] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[13] C. M. Bishop *et al.*, *Neural networks for pattern recognition*. Oxford university press, 1995.

[14] B. Wang, J. Gao, and Y. Qi, "A theoretical framework for robustness of (deep) classifiers under adversarial noise," *arXiv preprint*, 2016.

[15] J. Hopcroft and R. Kannan, "Foundations of data science," 2014.

[16] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the surprising behavior of distance metrics in high dimensional space," in *International conference on database theory*. Springer, 2001, pp. 420–434.

[17] D. Freedman, R. Pisani, and R. Purves, "Statistics. 2007," *ISBN: 0-393970-833*, 1978.

[18] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is nearest neighbor meaningful?" in *International conference on database theory*. Springer, 1999, pp. 217–235.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[20] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, vol. 2, 2010.

[21] A. Krizhevsky, V. Nair, and G. Hinton, "The cifar-10 dataset," *online: http://www. cs. toronto. edu/kriz/cifar. html*, 2014.

[22] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.

[23] Y. Dong, F. Liao, T. Pang, H. Su, X. Hu, J. Li, and J. Zhu, "Boosting adversarial attacks with momentum," *arXiv preprint arXiv:1710.06081*, 2017.

[24] N. Papernot, N. Carlini, I. Goodfellow, R. Feinman, F. Faghri, A. Matyasko, K. Hambardzumyan, Y.-L. Juang, A. Kurakin, R. Sheatsley *et al.*, "cleverhans v2. 0.0: an adversarial machine learning library," *arXiv preprint arXiv:1610.00768*, 2016.