On the Structural Sensitivity of Deep Convolutional Networks to the Directions of Fourier Basis Functions

Yusuke Tsuzuku[1,2], Issei Sato[1,2]
[1]The University of Tokyo, [2]RIKEN
tsuzuku@ms.k.u-tokyo.ac.jp, sato@k.u-tokyo.ac.jp

Abstract

Data-agnostic quasi-imperceptible perturbations on inputs are known to degrade recognition accuracy of deep convolutional networks severely. This phenomenon is considered to be a potential security issue. Moreover, some results on statistical generalization guarantees indicate that the phenomenon can be a key to improve the networks' generalization. However, the characteristics of the shared directions of such harmful perturbations remain unknown. Our primal finding is that convolutional networks are sensitive to the directions of Fourier basis functions. We derived the property by specializing a hypothesis of the cause of the sensitivity, known as the linearity of neural networks, to convolutional networks and empirically validated it. As a by-product of the analysis, we propose an algorithm to create shift-invariant universal adversarial perturbations available in black-box settings.

1. Introduction

Malicious perturbations on inputs can easily change predictions of deep learning models [36]. These perturbations are called adversarial perturbations or adversarial examples. They have been intensively studied concerning deep convolutional networks for object recognition tasks [4, 7, 19, 24, 36, 39]. They are attracting attention because they are potential security issues. One of the intriguing aspects of adversarial perturbations is their universality. Szegedy *et al*. [36] observed transferability of the perturbations between classifiers. Papernot *et al*. [28, 29] exploited the transferability to attack black-box models. Some adversarial perturbations transfer not only between classifiers but also between inputs. Goodfellow *et al*. [7] first discovered the universality, and Moosavi-Dezfooli *et al*. [22] studied this phenomenon in more detail. They found that a single perturbation can change models' predictions for a significant portion of data points. Such input-agnostic perturbations are called universal adversarial perturbations (UAPs). The perturbations also generalize between different networks to some extent.

Figure 1. Examples of images perturbed by single Fourier attack. Added perturbation is the same as in Figure 3. The size of perturbations is $10/255$ in ℓ_∞-distance for the first row and $20/255$ for the second. In Sec. 5.7, we show that the single $10/255$ and $20/255$ perturbations could change predictions for around 40% and 70% of inputs for various architectures, respectively.

We are primarily concerned with UAPs because of their relation to statistical generalization guarantees of deep learning models. For example, studies using PAC-Bayes [27], compression [1], and minimum description length [10] are all concerned with how perturbation propagates networks.[1] In these analyses, how each perturbation changes accuracy on training data and true data distribution matters. In other words, we have an interest in perturbations transferable between inputs. These are nothing else but UAPs. We try to shed lights on the tendency of UAPs

[1]In these studies, we consider perturbations on weights, not inputs. However, perturbations on weights become noises on inputs of the network's subnetworks and investigating UAPs is still useful.

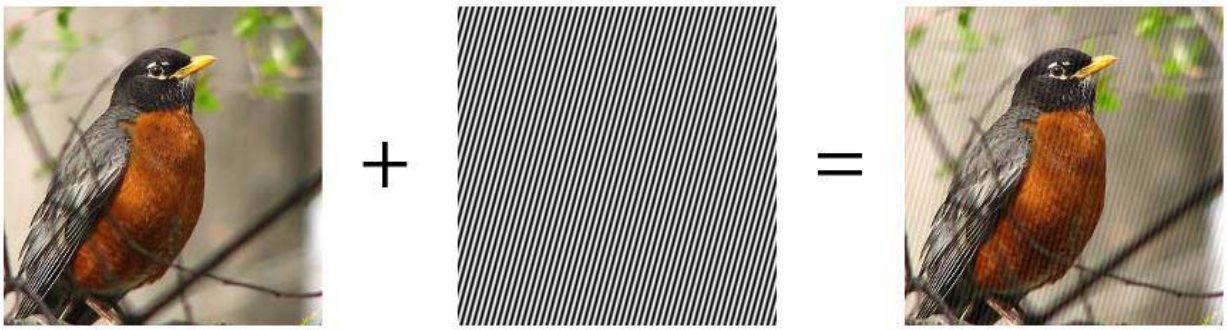

Figure 2. Illustration of our UAPs creation algorithm. We only tune frequency of noise. We do not need access to model parameters, output logits, or training data.

and how they propagate in convolutional neural networks.

Several prior studies tried to understand the properties of the universality and transferability of adversarial perturbations. Goodfellow *et al*. [7] explained the existence of adversarial examples, their transferability, and their universality using linearity of deep neural networks. Tramèr *et al*. [38] investigated the transferable subspace of adversarial perturbations and suggested that it will consist of a high-dimensional continuous subspace. Moosavi-Dezfooli *et al*. [23] showed that the existence of universal adversarial perturbations is inevitable given strong geometrical assumptions on the decision boundaries of models.

Given the transferability and the universality of adversarial perturbations, it is natural to expect the existence of a set of directions to which most networks are input-agnostically sensitive. If we can characterize such directions, it enables us to improve robustness against such perturbations in principled manners. Additionally, we may design better posteriors, weights, or compression algorithms to achieve empirically better generalization bounds. However, prior work can only generate such perturbations by sequential optimization and lacks their useful characterization. We provide a missing characterization of directions by analyzing Fourier basis functions.

The motivation of our analysis comes from two parts. The first is the linear hypothesis of vulnerability, and the second is a property of linear convolutional layers that the singular vectors of which are Fourier basis functions. The property indicates that sensitive directions of convolutional networks are a combination of a few Fourier basis functions. Through extensive experiments on various architectures and datasets, we found networks are sensitive to the directions of Fourier basis functions of some specific frequencies. In other words, we could characterize at least a subset of universal and transferable adversarial perturbations through Fourier basis functions. We also observed that some adversarial perturbations exploit the sensitivity to Fourier basis functions. These findings not only provide a new characterization of adversarial perturbations with benefits described in the preceding paragraph but also suggest a possibility that some known properties of the universality of adversarial perturbations might be due to the structure of convolutional networks.

As a by-product of our analysis, we also developed a method to create shift-invariant universal adversarial perturbations, which is available in black-box settings. Figure 1 shows examples of perturbed images created by our algorithm, which is explained in Sec. 4. Our perturbations have simple and shift-invariant patterns, yet achieved high fool ratio on various pairs of architectures and datasets.

Our contributions are summarized below.

1. We characterized spaces UAPs lie using Fourier basis functions.

2. We evaluated our hypothesis in extensive experiments.

3. We proposed a black-box algorithm to create shift-invariant universal adversarial perturbations.

2. Related work

2.1. Adversarial perturbations

One of the most famous algorithms for creating adversarial perturbations is the fast gradient sign method (FGSM) [7]. Let $J(\theta, x, t)$ be a loss with parameter θ, an input x, and a target label t. Then, FGSM uses $\epsilon \cdot \mathrm{Sign}\left(\nabla_x J(\theta, x, t)\right)$ as the perturbation, where ϵ is a scaling parameter. Another popular approach is performing gradient ascent on some loss $J(\theta, x, t)$. Depending on the choice of the loss and the optimization methods, there are numerous variants for attacks [4, 24]. Adversarial training [7] is a current effective countermeasure against adversarial perturbations. Kurakin *et al*. [19] conducted a large-scale study on adversarial training, and Tramèr *et al*. [37] extensively studied the transferability for defended and undefended models. Evaluations of defense methods are noto-

riously difficult [2, 40]. Thus, some studies have provided theoretically grounded defense methods [17, 41].

2.2. Universal adversarial perturbations

Moosavi-Dezfooli *et al*. [22] showed that some input-independent perturbations can significantly degrade classifiers' prediction accuracy. Such perturbations are called universal adversarial perturbations (UAPs). Moosavi-Dezfooli *et al*. [22] created UAPs by sequentially optimizing perturbations until we achieve the desired fool ratio. During the creation, they did not need access to test data. They showed that UAPs could change over 80% of the predictions of various networks trained on ILSVRC2012 [30]. UAPs also generalize between network architectures to some extent. Recently, Mopuri *et al*. [25] and Khrulkov *et al*. [16] proposed activation-maximization approaches for the creation of UAPs. UAPs degrade the average performance of systems and have different nature from other kinds of adversarial examples.

2.3. Analysis of transferability and universality

Goodfellow *et al*. [7] explained the existence of adversarial examples, their transferability, and their universality by linear hypothesis. In their explanation, the directions of perturbations are the most important in adversarial examples. The hypothesis is based on the following three factors: (1) modern networks behave like linear classifiers, (2) adversarial perturbations are aligned with the weight vectors of models, (3) different models learn similar functions. Thus, adversarial perturbations generalize between clean examples, and also different models. Tramèr *et al*. [38] analyzed the dimensionality of the subspace that adversarial examples lie in. Using first-order approximation, they found that adversarial examples lie in a high-dimensional subspace, suggesting overwrap of the subspace between classifiers. However, the structure of the subspace is unknown except for its estimated dimensionality. Moosavi-Dezfooli *et al*. [23] analyzed the existence of UAPs using strong geometrical assumptions. They also proposed an algorithm to find UAPs using Hessian on input, while it is prohibitively slow with large inputs.

We explain the existence of UAPs on the basis of the linear hypothesis of Goodfellow *et al*. [7]. We push forward the analysis concerning convolutional networks.

2.4. Fourier basis

Jo and Bengio [14] examined whether CNNs learn high-level features by using Fourier features. Some prior work used eps compression or other transformations as defenses against adversarial examples [15, 8, 33]. They remove high-frequency features from images and relates to this paper. However, connections to universality have not been explored. Also, the effects of each frequency have not been studied. In a later section (5), our experiments show that adversarial perturbations do not necessarily lie in high-frequency spots.

3. Preliminary

In this section, we describe the relationship between convolutional layers and Fourier basis. Notations are summarized in the supplementary material.

3.1. Fourier basis and discrete Fourier transformation

Let us define $\omega_N^{i,j} = \omega_N^i \omega_N^j \in \mathbb{C}^{N\times N}$, where $\omega_N = \exp(2\pi\sqrt{-1}/N)$ is the N-th root of an imaginary number. We define F_N be a matrix such that colums are n fourier basis functions with different frequencies. In other words, F_N is a matrix such that

$$(F_N)_{u,v} = \frac{1}{\sqrt{N}} \exp(-2\pi\sqrt{-1}(u+v)/N). \qquad (1)$$

We notate the i-th row of F_N as $(F_N)_i$. Let us define a transformation $S : \mathbb{C}^{N\times N} \to \mathbb{C}^{N\times N}$ as follows.

$$S(x)_{u,v} = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x_{m,n} \exp(-2\pi\sqrt{-1}(um+vn)/N) \qquad (2)$$

This transformation S is called discrete Fourier transformation (DFT). Both the transformation and its inverse can be calculated in the running time of $O(N \log N)$ by using fast Fourier transformation [5].

3.2. Decomposition of convolution operator

We define $Q_N := \frac{1}{N} F_N \otimes F_N$, where $\otimes$ is a Kronecker product. The eigenvectors of a doubly block circulant matrix are known to be Q_N [12]. Since Q_N is unitary, a doubly block circulant matrix can be decomposed as $Q_N D Q_N^{\mathrm{H}}$, where Q_N^{H} is an adjoint matrix of Q_N, and D is a complex diagonal matrix. In a case where channel size is one, since convolution is a doubly circulant matrix when the padding is "wraps around"[6, 31], the above analysis is directly applicable. We can extend the result to multi-channel cases, i.e., $m \geq 1$.

Proposition 1. *Let M be a matrix which represents a convolutional layer with input channel size m_{in}, output channel size m_{out}, and input size $m_{\mathrm{in}} \times N \times N$. Then, M can be decomposed as*

$$M = (I_{m_{\mathrm{out}}} \otimes Q_N)\, L\, (I_{m_{\mathrm{in}}} \otimes Q_N)^{\mathrm{H}}, \qquad (3)$$

where L is a block matrix such that each block is a $N^2 \times N^2$ diagonal matrix.

4. Fourier analysis

In this section, we show that the most sensitive direction of linear convolutional networks is a combination of a few Fourier basis functions. The analysis pushes forward the linear hypothesis of the cause of adversarial examples in Goodfellow *et al*. [7]. The linear approximation may not hold well for deep non-linear networks. However, we can still expect that adding some Fourier basis functions to inputs can largely disturb hidden representations of networks. We assume that the padding of convolutional layers are "wraps around." Notations are summarized in the supplementary material. Proofs of propositions are deferred to the supplementary material.

4.1. Sensitivity of stacked convolutional layers

We first consider stacked stride-1 convolutional layers without activation functions. In the case, we can show that the singular vectors of the whole layers can be represented by a linear combination of single Fourier basis functions between input channels.

Proposition 2. *Let $M^{(i)}$ be a convolitional layer with input channel size $m^{(l)}$, output channel size $m^{(l+1)}$, input size $m^{(l)} \times N \times N$, and stride 1. Let M be a stacked convolutional layers with linear activation, i.e., $M(X) = \left(M^{(1)} \circ M^{(2)} \circ \cdots \circ M^{(d)}\right)(X)$. Then, the right singular vectors of M can be represented by $\vec{a} \otimes (F_N)_i \otimes (F_N)_j$ for some $i, j \in \{0, \ldots, N-1\}$ and $\vec{a} \in \mathbb{R}^{m^{(1)}}$.*

In other words, the most sensitive directions of linear convolutional neural networks without reduction layers is a single Fourier basis function. We can further extend the result to cases when there are normalization layers or skip connections.

Proposition 3. *Let $M^{(i)}$ be a convolitional layer with input channel size $m^{(l)}$, output channel size $m^{(l+1)}$, input size $m^{(l)} \times N \times N$, and stride 1. Let M be a stacked convolutional layers with linear activation plus a skip connection, i.e., $M(X) = \left(M^{(1)} \circ M^{(2)} \circ \cdots \circ M^{(d)}\right)(X) + X$. Then, the right singular vectors of M can be represented by $\vec{a} \otimes (F_N)_i \otimes (F_N)_j$ for some $i, j \in \{0, \ldots, N-1\}$ and $\vec{a} \in \mathbb{R}^{m^{(1)}}$.*

Proposition 4. *A convolutional layer followed by a normalization layer such as batch-normalization or weight-normalization can be rerepresented as another convolutional layer without normalization at test time. Thus, Props. 2 and 3 also hold when normalization layers exist.*

These propositions show that manipulating a single Fourier basis function on input can be most effective to disturb internal representations of convolutional neural networks.

4.2. Reduction layers

In this section, we show that the singular values of the convolutional layers can be written by a combination of a few Fourier basis functions even when there are reduction layers, such as convolutional layers with stride > 1 or average pooling layers.

Proposition 5. *Let M be a convolutional layer with stride $s > 1$ where $N = 0 \pmod s$. Then, the singular value of the layer can be represented by a linear combination of Fourier basis functions $\{(F_N)_{i'} \otimes (F_N)_{j'} | i' = i \pmod s, j' = j \pmod s\}$ for any i and j.*

Since the average pooling layer is a special case of convolutional layers, we can apply the above theorem to the layer.

4.3. Single Fourier attack

We propose an algorithm to find universal adversarial perturbations using Fourier basis functions. The attack exploits the sensitivity of convolutional networks to the Fourier basis directions analyzed in the previous section. While the linear approximation in the analysis might not hold well in deep networks, we can still expect that the directions will disturb hidden representations.

A sketch of the algorithm is as follows. We select one Fourier basis function and use it as a UAP. The method to select the frequency is described later in this section. The sketch of the algorithm is incompatible with the restriction that the inputs must be real. To satisfy the condition, we have the following proposition.

Proposition 6. *$S(x)_{i,j} = S(x)^*_{N-i,N-j}$ iff the input x is real-valued, where $S(x)^*$ is a conjugate of $S(x)$.*

Thus, we make $S(x)_{i,j} = S(x)^*_{N-i,N-j}$ satisfied to meet the real-value constraint. Algorithm 1 shows the pseudocode of the algorithm, which is named single Fourier attack (SFA). Figure 3 shows a visualization of Fourier basis in 8×8 space and an example of perturbations created by SFA. Figure 1 shows examples of perturbed images. It seems that this attack does not change human's predictions, and models should be robust against the attack.

To perform the attack, we need to find effective frequencies of the target classifiers. To test the sensitivity, first, we query a pair of an original image and its perturbed version. Next, we check whether the classifier's output differs or not. We repeat the procedure and solve a black-box optimization problem formulated as follows.

Problem 1. *Given a data distribution $D \subset \mathbb{R}^{N \times N \times 3}$, target function $f : \mathbb{R}^{N \times N \times 3} \times (\{1, \ldots, N\} \times \{1, \ldots, N\}) \to \{0, 1\}$, find a frequency $w \in \{1, \ldots, N\} \times \{1, \ldots, N\}$ which maximizes*

$$\int_D f(x, w) dx. \tag{4}$$

Algorithm 1: Single Fourier attack

hyperparam : i, j: frequency, ϵ: size of perturbation
input : x : image
foreach *c in channel*:
$x_c \leftarrow x_c + \epsilon((1+i)(F_N)_i \otimes (F_N)_j + (1-i)(F_N)_{N-i} \otimes (F_N)_{N-j})$;
$x_c \leftarrow$Clip$(x_c, 0, 1)$;

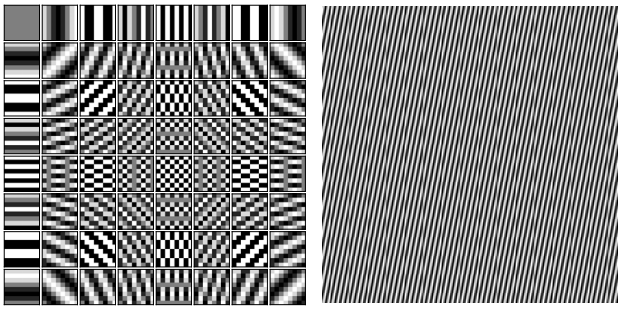

Figure 3. Left: Visualization of Fourier basis in 8×8 space. Row i and column j shows $(F_8)_i \otimes (F_8)_j$. Right: An example of perturbations created by Single Fourier attack in Alg. 1. This perturbation was used in later evaluation (Sec. 5.7).

One naive approach to approximately solve the problem is testing all frequencies with a batch of images and find a frequency with the highest fool ratio. The batchsize controls the variance of the evaluation of each frequency. Even if we do the brute-force search, we can create UAPs within a reasonable amount of time thanks to the simplicity of our formulation. As more query efficient methods, we can also use Bayesian optimization techniques [34, 3]. We show that the search of the frequency has a favorable property for such methods in Sec. 5.4. This suggests our algorithm is useful even when only small numbers of queries are allowed to create UAPs.

Our formulation and algorithm have the following two key features. First, we formulated the creation of UAPs as an optimization problem of two discrete variables. On the other hand, the original problem has the same number of parameters with the input size, which can be tens of thousands. This reduction of parameters to optimize is a significant simplification. Second, our algorithm requires neither model parameters nor output logits. Prior UAPs creation algorithms require access to models or substituted models created by attackers. These requirements have made the attacks less practical. In our algorithm, we only require the information on the predicted label by the target. Thus, the algorithm is available in broader settings.

5. Experiments

We presented a characterization of the universal adversarial directions through Fourier basis functions in Sec. 4. To show that the characterization well describes the nature of the universal adversarial directions, we conducted a series of experiments. Primarily, we answer the following questions.

1. Whether Fourier basis characterization is better than others such as characterization using the standard basis (Sec. 5.2).
2. Whether the sensitivity to the Fourier basis directions is unique to convolutional networks (Sec. 5.3).
3. Whether UAPs are related to Fourier basis directions (Sec. 5.5).
4. Whether current white-box attacks are also related to Fourier basis directions (Sec 5.6).
5. Whether manipulation on a single Fourier basis can image-agnostically change predictions of various convolutional neural networks and datasets (Sec. 5.7).

5.1. Evaluation setups

This section describes the evaluation setups. A more detailed explanation can be found in the supplementary material. We used MNIST [21], fashion-MNIST [42], SVHN [26], CIFAR10, CIFAR100 [18], and ILSVRC2015 [30] as datasets. We used a multi-layer perceptron (MLP) consisting of 1000–1000 hidden layer with ReLU activation, LeNet [20], WideResNet [43], DenseNet-BC [11], and VGG [32] with batch-normalization for evaluations on datasets except for ILSVRC2015. For ILSVRC2015, we used ResNet50 [9], DenseNet, VGG16, and GoogLeNet [35]. For VGG16 and GoogLeNet, we added a batch-normalization layer after each convolution for faster training. We used the fool ratio as a metric, which is the percentage of data that models changed its prediction, following Moosavi-Dezfooli *et al*. [22].

5.2. Fourier domain vs pixel domain

We analyzed the sensitivity of deep convolutional neural networks to the directions of Fourier basis functions in Sec. 4. To empirically support the analysis, we investigated the sensitivity on each Fourier basis. For comparison, we checked the sensitivity on the standard basis directions, which is the manipulation on each pixel. We also tested the sensitivity in random directions (see Sec. 5.7). We first describe the method we used to study the sensitivity. For Fourier basis, we applied a single Fourier attack (Algorithm. 1) and calculated its fool ratio on a single minibatch for each frequency. We bounded the size of perturbations by $30/255$ in ℓ_∞-norm for MNIST, FMNIST, and SVHN, $20/255$ for ILSVRC2015, and $10/255$ for CIFAR10 and CIFAR100. For a standard basis, we added $255/255$ to each pixel and then clipped to range from zero to one for

Algorithm 2: Creation of heatmap

foreach *(i,j) in frequencies*:
  $B :=$ Randomly select Minibatch;
  $y \leftarrow$ Forward(B);
  $y' \leftarrow$ Forward(B + noise);
  Heatmap$_{i,j} \leftarrow$ FoolRatio(y, y');

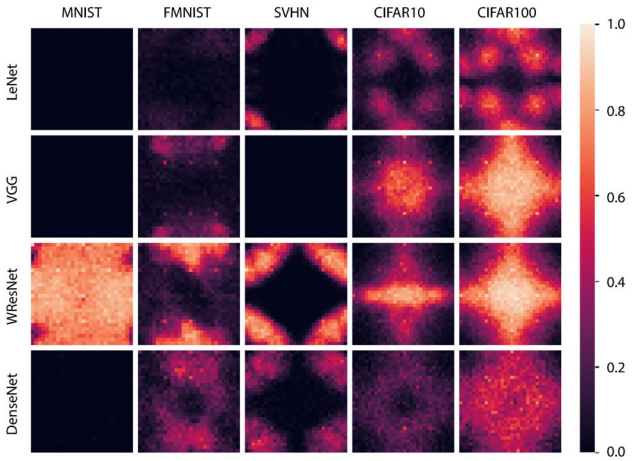

Figure 4. Visualization of sensitive spot of convolutional networks in Fourier domain. Coordinate (i, j) of each image represents fool ratio on a single minibatch when we used Algorithm 1 as a perturbation. White areas are spots with high fool ratio. The center of each image corresponds to a high-frequency area. The perturbation sizes were $30/255$ for MNIST, FMNIST, and SVHN, and $10/255$ for CIFAR10 and CIFAR100. The creation of this heatmap is described in Algorithm. 2.

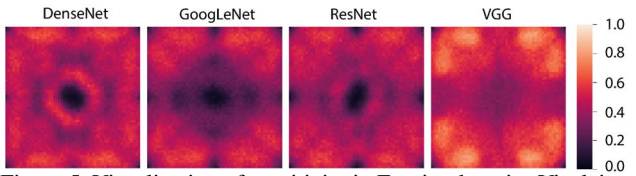

Figure 5. Visualization of sensitivity in Fourier domain. Visulaization procedure is the same with Figure 4. We can see that most sensitive frequency is neither hight nor low frequencies, and it lies in the middle. For reference, frequency distributions in natural images and random noise can be found in Figure 8.

attack creation, which is an analogy of Algorithm 1. Using heat maps, we visualized the results for Fourier basis on ILSVRC2015 in Figure 5 and the results on the other datasets in Figure 4. The algorithm to create the heat map is described in Algorithm 2.

We observed that in most cases except for MNIST, architectures tend to have some sensitive spots in the Fourier domain. Especially on CIFAR10 and CIFAR100, VGG and Wide-ResNet showed near 90% and 99% fooling ratio to some directions. The result means that the predictions be-

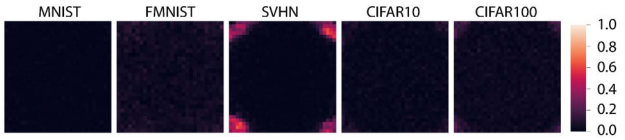

Figure 6. Visualization of the sensitivity of multilayer perceptrons (MLPs) in the Fourier domain. MLPs did not have sensitive spot as CNNs in most cases and they were more resistant to directions of Fourier basis.

came almost random guess. Since all Fourier basis directions are orthogonal, Figure 4 highlights that there are hundreds of directions that networks are weak independent of their inputs. While it has been known that there are tens of orthogonal directions for transferable or universal adversarial examples, to the best of our knowledge, this is the fastest method to find a large number of orthogonal directions for which networks are universally vulnerable. Contrastive to Fourier basis, experiments in standard basis achieved almost 0% fool ratio in all settings. In this experiment, we showed the existence of sensitive spots of convolutional networks in the Fourier domain and the effectiveness of the characterization by Fourier basis directions.

5.3. Convolutional networks vs. MLP

In Sec. 5.2 we observed that various convolutional neural networks are sensitive to some Fourier basis directions. To see whether the sensitivity to the Fourier basis functions is caused by network architectures as suggested in Sec. 4 or the nature of image processing, we compared the sensitivity of convolutional neural networks and MLP to Fourier basis functions. We used the same method as Sec. 5.2 for the comparison. Figure 6 shows the results for the MLP trained on various datasets. The MLP did not show vulnerability to some vectors in the Fourier basis. The contrastive activation pattern of convolutional networks and multilayer perceptrons supports our analysis of the sensitivity in Sec. 4. This result suggests the possibility that changing architectures is a useful measure to mitigate adversarial examples, especially UAPs. Since prior defense work has mostly focused on training methods [7, 19], this opens another research direction for defense methods. For example, we may use the information of the weak spots in the Fourier domain to choose which models to use for ensembles.

5.4. Co-occurrence of sensitivity

In the evaluation in Secs. 5.2 and 5.3, we observed that convolutional networks showed similar sensitivity to the Fourier basis directions with similar frequencies. Since Sec. 4 does not cover this phenomenon, we explain it here. In convolutional networks, the convolution kernel size is typically much smaller than the input size. The size of the kernel restricts the expressiveness of convolutional layers. This restriction makes convolutional layers respond simi-

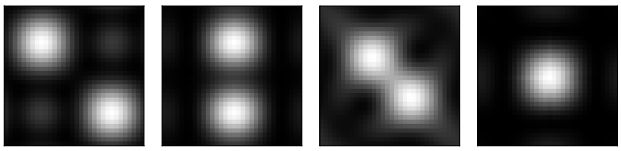

Figure 7. Coordinate (i, j) of each image shows the magnitude of outputs of a convolutional layer when input was $(F_{32})_i \otimes (F_{32})_j$. The kernel size of each convolutional layer is 3. They were trained to maximize the output against $(F_{32})_8 \otimes (F_{32})_8, (F_{32})_8 \otimes (F_{32})_{16}, (F_{32})_{12} \otimes (F_{32})_{12}, and (F_{32})_{16} \otimes (F_{32})_{16}$, respectively.

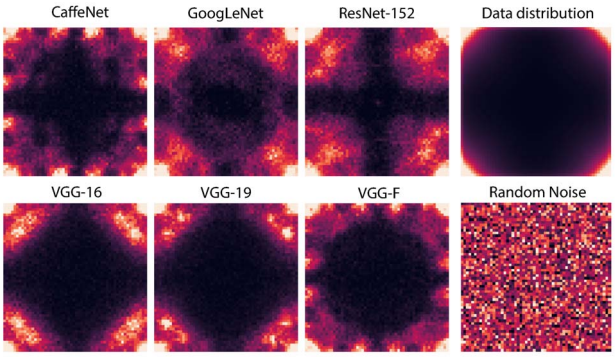

Figure 8. Visualization of UAPs calculated for various architectures on ILSVRC2012 by Moosavi-Dezfooli *et al.* [22] in the Fourier domain. Coordinate (i, j) corresponds to the fool ratio of $(F_{224})_i \otimes (F_{224})_j$. White spots had higher fool ratio.

larly to similar frequencies. To see the co-occurrence of the sensitivity, we trained convolutional layers with kernel size 3×3 and the input size 32×32 so that the ℓ_2-norm of their outputs are maximized when one specific Fourier basis is fed as its input. Then we tested the ℓ_2-norm of the layer's outputs when their inputs are other Fourier basis functions. Figure 7 shows the result. The result confirms the hypothesis that convolutional layers respond similarly to Fourier basis directions with similar frequency. In other words, the optimization problem of frequency of Algorithm 1 has a small Lipschitz constant. This property is known to be favorable for optimizations in many algorithms including Bayesian optimizations [34].

5.5. UAPs in Fourier domain

In this section, we investigate whether UAPs created by an existing method also have some specific patterns in the Fourier domain. For this analysis, we used precomputed UAPs for VGG16, VGG19, VGG-F, CaffeNet [13], ResNet152, and GoogLeNet by Moosavi-Dezfooli *et al.* [22]. Figure 8 shows the magnitude of each frequency of each UAP in log scale. For reference, Figure 8 also shows those of random noise and average magnitudes of each frequency of original training data in ILSVRC2015.

While architectures and training procedures differ, Figure 8 and Figure 5 share a similar tendency compared to

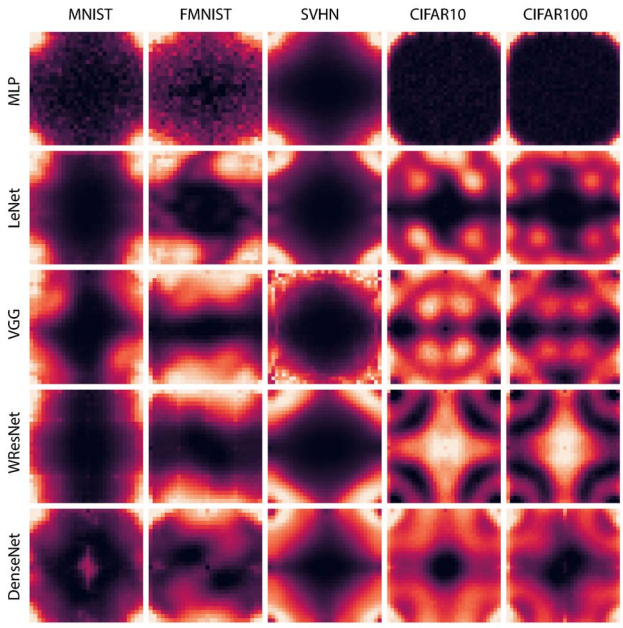

Figure 9. Visualization of FGSM attack in the Fourier domain in the same way as in Figure 8. FGSM had larger values in sensitive spots revealed in Figure 4. Center of each image is a high-frequency area.

the random noise and original images. For example, we can see from Figure 5 that the networks are relatively robust against high-frequency noises and sensitive to low and middle-frequency noises. From Figure 8, current UAPs appear to exploit the sensitivity. This suggests the effectiveness to consider Fourier domain to analyze existing UAPs.

5.6. Adversarial attacks in Fourier domain

In this section, we investigate whether current white box adversarial attacks also have some tendency in the Fourier domain. We studied FGSM [7], which is known to transfer better than naive iterative attacks [19]. Figure 9 shows the average magnitude of each vector in the Fourier basis of a perturbation created by FGSM on test data. Compared with Figure 4, which revealed sensitive spots in the Fourier domain, Figure 9 shows that the mass of FGSM concentrates almost in the sensitive spots. This experiment also shows that adversarial perturbations do not necessarily lie in a high-frequency area, which denies a common myth that adversarial perturbations tend to be high-frequency. Figure 9 also shows that the tendency of adversarial perturbations differs across datasets and architectures, which reminds us to test defense methods in various settings.

5.7. Effectivceness of Fourier attack

The analysis in Sec. 4 and experiments in Sec. 5.2 – Sec. 5.6 suggests the effectiveness of the Fourier basis functions as universal adversarial perturbations. We evaluated

Table 1. Fool ratio of random noise (upper rows) and SFA (Algorithm 1, lower rows) on various architectures and datasets. Despite the simpleness of our algorithm, some pairs dropped their accuracy to almost chance ratio. This results show that our characterization through Fourier basis functions effectively captures the sensitivity of networks.

	LeNet	WResNet	VGG	DenseNet
MNIST	0.1	55.8	0.0	0.1
Fashion MNIST	5.4	11.4	8.3	12.6
SVHN	3.1	5.2	0.0	4.6
CIFAR10	5.0	8.1	6.8	5.4
CIFAR100	13.0	26.4	25.9	22.5
MNIST	0.4	**90.2**	0.1	0.2
Fashion MNIST	12.5	**48.1**	**83.7**	**56.9**
SVHN	**64.9**	**90.8**	0.0	**50.5**
CIFAR10	**63.3**	**82.3**	**72.2**	**50.7**
CIFAR100	**83.4**	**93.7**	**95.8**	**72.3**

Table 2. Fool ratio of Fourier basis attack on various architectures on ILSVRC2015. Rand is random noise, SSFA is defined in Sec. 5.7. Attacks are bounded in $10/255$ and $20/255$ in ℓ_∞-norm. UAP denotes the best performing precomputed UAP in [22] per architecture. Since naively scaling them to $20/255$ can be unfairly advantageous to ours and we just omitted the evaluation. While comparable, our algorithm does not assume access to the same training data and also has no need to train models locally.

	GoogLeNet	ResNet	VGG	DenseNet
Rand(10)	8.2	8.5	11.5	9.5
Rand(20)	14.9	16.1	19.7	16.7
UAP(10) [22]	45.5	49.7	64.8	56.0
SFA(10)	34.6	38.7	49.7	36.8
SFA(20)	62.3	68.5	76.3	63.5
SSFA(10)	44.1	40.1	53.3	39.5
SSFA(20)	74.1	66.9	79.0	62.5

its ability to flip predictions on various datasets and architectures. We set the size of perturbations to $10/255$ in ℓ_∞ for CIFAR, and to $30/255$ for MNIST, FMNIST, and SVHN. We used frequencies with the highest fool ratio in Figure 4 as the perturbations. In the evaluation, we used Algorithm 1 with one fixed frequency per pair of dataset and architecture. For comparison, we calculated the fool ratio of random noise sampled from the ϵ-ball bounded in ℓ_∞-norm. Table 1 shows the result. Given the dataset and architecture-agnostic search space, the attack showed strong attack ability. Especially in CIFAR10 and CIFAR100 experiments, some architectures dropped prediction accuracy almost to that of random guessing. This effectiveness of Fourier basis attack highlights the sensitivity of current convolutional networks against Fourier features. In MNIST, however, the fool ratio was not as high as other datasets. Since MNIST is highly normalized dataset and easiest among them, we suspect that networks can better capture true signal from the inputs and are more robust to change of a single Fourier basis direction. From the viewpoints of architectures, LeNet and DenseNet were more robust than others. We explain this by their max-pooling layers. As max-pooling layers are not supported in Sec. 4, they add additional nonlinearities and mix Fourier basis.

We also tested Algorithm 1 on ILSVRC2015. For the evaluation, we fixed one frequency for all architectures and inputs[2]. In other words, we selected a single perturbation input and architecture agnostically. To choose the frequency, we took the average of Figure 5 and picked the frequency with the highest fool ratio. Figure 1 shows examples of created adversarial examples. We used $10/255$ and $20/255$ for the size of perturbations. Note that previous work used $10/255$ for the evaluation [22]. Examples of created UAPs are shown in Figure 1. We empirically found that taking the sign of Fourier basis can sometimes boost the performance of the attack. We named this attack Signed-SFA (SSFA), and we also tested the attack. In the evaluation, we also tested random perturbations and the best precomputed UAP from Moosavi-Dezfooli *et al*. [22] per architecture. The result is shown in Table 2. Compared to Moosavi-Dezfooli *et al*. [22], the fool ratio is comparable to their perturbations under this black-box setting. Note, since our algorithm does not need to train local model, our algorithm is more suitable in black-box settings.

6. Conclusion

From the analysis of linearized convolutional neural networks, we hypothesized that convolutional networks are sensitive to the directions of Fourier basis functions. Through empirical evaluations, we validated the sensitivity. The finding provides a better characterization of universal adversarial perturbations using Fourier basis functions. The characterization might be beneficial to the development of defense methods and the analysis of statistical generalization guarantees. As a by-product of our analysis, we proposed a black-box method to create universal adversarial perturbations. The algorithm does not require locally trained models for black-box attack and extends the potential use cases of universal adversarial perturbations.

Acknowledgement

YT was supported by Toyota/Dwango AI scholarship. IS was supported by KAKENHI 17H04693.

[2]The input sizes were the same (224×224) among the all architectures we tested.

References

[1] S. Arora, R. Ge, B. Neyshabur, and Y. Zhang. Stronger Generalization Bounds for Deep Nets via a Compression Approach. In *Proceedings of the 35th International Conference on Machine Learning*, pages 254–263, 2018.

[2] A. Athalye, N. Carlini, and D. Wagner. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In *Proceedings of the 35th International Conference on Machine Learning*, pages 274–283, 2018.

[3] A. D. Bull. Convergence Rates of Efficient Global Optimization Algorithms. *Journal of Machine Learnining Research*, pages 2879–2904, 2011.

[4] N. Carlini and D. A. Wagner. Towards Evaluating the Robustness of Neural Networks. In *Proceedings of the 2017 IEEE Symposium on Security and Privacy*, pages 39–57. IEEE Computer Society, 2017.

[5] J. W. Cooley and J. W. Tukey. An Algorithm for the Machine Calculation of Complex Fourier Series. *Mathematics of Computation*, pages 297–301, 1965.

[6] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[7] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and Harnessing Adversarial Examples. *International Conference on Learning Representations*, 2015.

[8] C. Guo, M. Rana, M. Cisse, and L. v. d. Maaten. Countering Adversarial Images using Input Transformations. *International Conference on Learning Representations*, 2018.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[10] G. E. Hinton and D. v. Camp. Keeping the Neural Networks Simple by Minimizing the Description Length of the Weights. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, pages 5–13, 1993.

[11] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely Connected Convolutional Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 2261–2269, 2017.

[12] A. K. Jain. *Fundamentals of digital image processing*. Prentice-Hall, Inc., 1989.

[13] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM International Conference on Multimedia*, page 675–678, 2014.

[14] J. Jo and Y. Bengio. Measuring the tendency of CNNs to Learn Surface Statistical Regularities. *CoRR*, abs/1711.11561, 2017.

[15] G. Karolina Dziugaite, Z. Ghahramani, and D. M. Roy. A study of the effect of JPG compression on adversarial images. *CoRR*, abs/1608.00853, 2016.

[16] V. Khrulkov and I. Oseledets. Art of singular vectors and universal adversarial perturbations. *CoRR*, 2017.

[17] J. Z. Kolter and E. Wong. Provable Defenses against Adversarial Examples via the Convex Outer Adversarial Polytope. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5286–5295, 2018.

[18] A. Krizhevsky. Learning Multiple Layers of Features from Tiny Images. *Computer Science Department, University of Toronto, Technical Report*, 2009.

[19] A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial Machine Learning at Scale. *International Conference on Learning Representations*, 2017.

[20] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based Learning Applied to Document Recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[21] Y. LeCun, C. Cortes, and C. J. C. Burges. The MNIST Database of Handwritten Digits. *http://yann.lecun.com/exdb/mnist/*, 1998.

[22] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1765–1773, 2017.

[23] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, and S. Soatto. Robustness of Classifiers to Universal Perturbations: A Geometric Perspective. *International Conference on Learning Representations*, 2018.

[24] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, 2016.

[25] K. R. Mopuri, U. Garg, and R. V. Babu. Fast Feature Fool: A data independent approach to universal adversarial perturbations. In *Proceedings of the British Machine Vision Conference*, 2017.

[26] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading Digits in Natural Images with Unsupervised Feature Learning. *Neural Information Processing Systems Workshop*, 2011.

[27] B. Neyshabur, S. Bhojanapalli, and N. Srebro. A PAC-Bayesian Approach to Spectrally-Normalized Margin Bounds for Neural Networks. In *International Conference on Learning Representations*, 2018.

[28] N. Papernot, P. McDaniel, and I. J. Goodfellow. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples. *CoRR*, abs/1605.07277, 2016.

[29] N. Papernot, P. D. McDaniel, I. J. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical Black-Box Attacks against Machine Learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519, 2017.

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, pages 211–252, 2015.

[31] H. Sedghi, V. Gupta, and P. M. Long. The Singular Values of Convolutional Layers. In *International Conference on Learning Representations*, 2019.

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[33] S. Song, Y. Chen, N.-M. Cheung, and C.-C. J. Kuo. Defense Against Adversarial Attacks with Saak Transform. *CoRR*, abs/1808.01785, 2018.

[34] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 1015–1022, 2010.

[35] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper with Convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[36] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing Properties of Neural Networks. *International Conference on Learning Representations*, 2014.

[37] F. Tramèr, A. Kurakin, N. Papernot, D. Boneh, and P. D. McDaniel. Ensemble Adversarial Training: Attacks and Defenses. *International Conference on Learning Representations*, 2018.

[38] F. Tramèr, N. Papernot, I. J. Goodfellow, D. Boneh, and P. D. McDaniel. The Space of Transferable Adversarial Examples. *CoRR*, abs/1704.03453, 2017.

[39] Y. Tsuzuku, I. Sato, and M. Sugiyama. Lipschitz-Margin Training: Scalable Certification of Perturbation Invariance for Deep Neural Networks. In *Advances in Neural Information Processing Systems 31*, pages 6542–6551. 2018.

[40] J. Uesato, B. O'Donoghue, P. Kohli, and A. Oord. Adversarial Risk and the Dangers of Evaluating Against Weak Attacks. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5025–5034, 2018.

[41] L. Weng, H. Zhang, H. Chen, Z. Song, C. Hsieh, L. Daniel, D. Boning, and I. Dhillon. Towards Fast Computation of Certified Robustness for ReLU Networks. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5276–5285, 2018.

[42] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *CoRR*, abs/1708.07747, 2017.

[43] S. Zagoruyko and N. Komodakis. Wide Residual Networks. In *Proceedings of the British Machine Vision Conference*, pages 87.1–87.12, 2016.