

Hafta 13 - Adversarial ML
BGM 565 - Siber Güvenlik için Makine Öğrenme Yöntemleri
Bilgi Güvenliği Mühendisliği
Yüksek Lisans Programı

Dr. Ferhat Özgür Çatak
ozgur.catak@tubitak.gov.tr

İstanbul Şehir Üniversitesi
2018 - Bahar

İçindekiler

1 Adversarial Examples

- Giriş
- Hedef Algoritmalar
- Adversarial Examples in the Human Brain
- Lets fool a binary linear classifier

2 Adversarial Machine Learning

- Giriş
- Tehdit Modelleri

• Saldırı Sınıflandırması

- CleverHans
- IBM -Adversarial-Robustness- Toolbox

3 Saldırılar

- Fast gradient sign method (FGSM)
- Jacobian-based Saliency Map Attack
- DeepFool
- Virtual Adversarial Training

İçindekiler

1

Adversarial Examples

- Giriş
- Hedef Algoritmalar
- Adversarial Examples in the Human Brain
- Lets fool a binary linear classifier

2

Adversarial Machine Learning

- Giriş
- Tehdit Modelleri

- Saldırı Sınıflandırması

- CleverHans

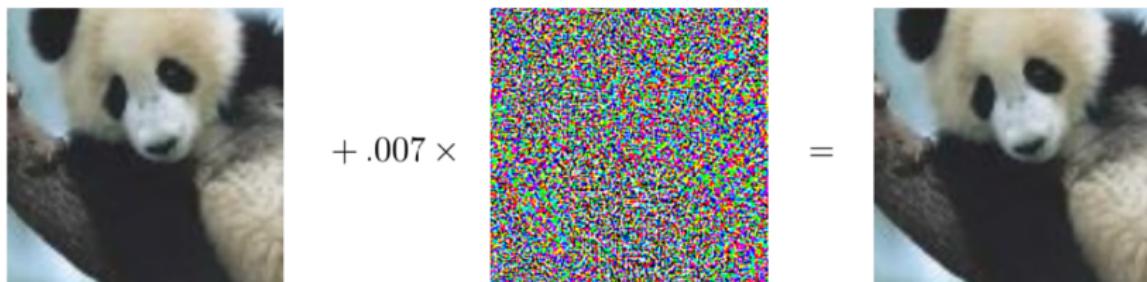
- IBM -Adversarial-Robustness-Toolbox

3

Saldırılar

- Fast gradient sign method (FGSM)
- Jacobian-based Saliency Map Attack
- DeepFool
- Virtual Adversarial Training

Adversarial Examples



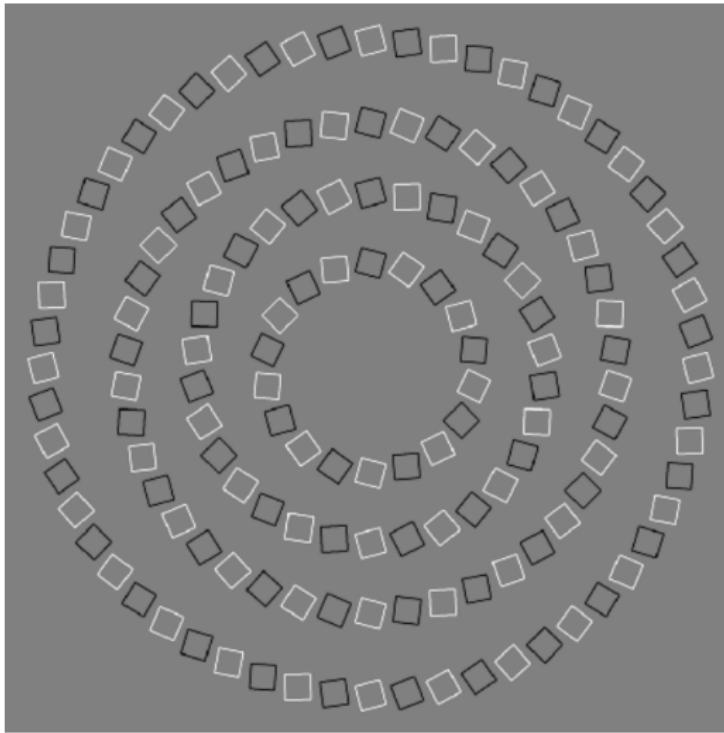
Adversarial Examples

- ▶ “Adversarial Classification” Dalvi et al 2004: **fool spam filter**
 - ▶ “Evasion Attacks Against Machine Learning at Test Time” Biggio 2013: **fool neural nets**
 - ▶ Szegedy et al 2013: fool ImageNet classifiers imperceptibly
Goodfellow et al 2014: cheap, closed form attack

Hedef Algoritmalar

- ▶ Sadece Neural Network Algoritmaları hedef değildir.
 - ▶ Doğrusal Yöntemler
 - ▶ Logistic regression
 - ▶ SVMs
 - ▶ Decision trees
 - ▶ Nearest neighbors

Adversarial Examples in the Human Brain



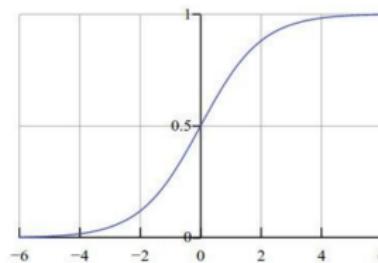
Eşmerkezli
daireler, iç içe
spiraller değil.

Şekil: (Pinna and Gregory, 2002)

Lets fool a binary linear classifier I

Lets fool a binary linear classifier: (logistic regression)

$$P(y=1 \mid x; w, b) = \frac{1}{1+e^{-(w^T x + b)}} = \sigma(w^T x + b)$$



Since the probabilities of class 1 and 0 sum to one, the probability for class 0 is $P(y=0 | x; w, b) = 1 - P(y=1 | x; w, b)$. Hence, an example is classified as a positive example ($y = 1$) if $\sigma(w^T x + b) > 0.5$, or equivalently if the score $w^T x + b > 0$.

Şekil: http://cs231n.stanford.edu/slides/2016/winter1516_lecture9.pdf

Lets fool a binary linear classifier II

Lets fool a binary linear classifier:

X	2	-1	3	-2	2	2	1	-4	5	1	← input example
W	-1	-1	1	-1	1	-1	1	1	-1	1	← weights

class 1 score = dot product:

$$= -2 + 1 + 3 + 2 + 2 - 2 + 1 - 4 - 5 + 1 = -3$$

=> probability of class 1 is $1/(1+e^{(-(-3))}) = 0.0474$

i.e. the classifier is **95%** certain that this is class 0 example.

$$P(y=1 \mid x; w, b) = \frac{1}{1 + e^{-(w^T x + b)}} = \sigma(w^T x + b)$$

Lets fool a binary linear classifier III

Lets fool a binary linear classifier:

X	2	-1	3	-2	2	2	1	-4	5	1	← input example
W	-1	-1	1	-1	1	-1	1	1	-1	1	← weights
adversarial x	1.5	-1.5	3.5	-2.5	2.5	1.5	1.5	-3.5	4.5	1.5	

class 1 score before:

$$-2 + 1 + 3 + 2 + 2 - 2 + 1 - 4 - 5 + 1 = -3$$

=> probability of class 1 is $1/(1+e^{(-(-3))}) = 0.0474$

$$-1.5 + 1.5 + 3.5 + 2.5 - 1.5 + 1.5 - 3.5 - 4.5 + 1.5 = 2$$

=> probability of class 1 is now $1/(1+e^{-(2)}) = 0.88$

i.e. we improved the class 1 probability from 5% to 88%

This was only with 10 input dimensions. A 224x224 input image has 150,528.

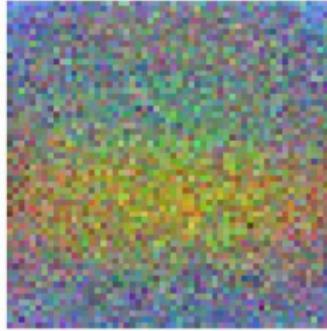
(It's significantly easier with more numbers, need smaller nudge for each)

Lets fool a binary linear classifier IV

1.0% kit fox



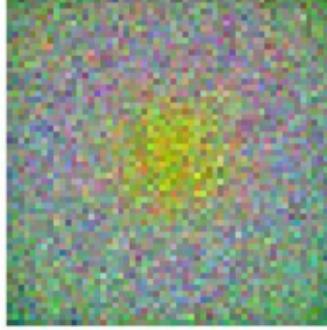
3.9% school bus



8.3% goldfish



12.5% daisy



İçindekiler

1 Adversarial Examples

- Giriş
- Hedef Algoritmalar
- Adversarial Examples in the Human Brain
- Lets fool a binary linear classifier

2 Adversarial Machine Learning

- Giriş
- Tehdit Modelleri

● Saldırı Sınıflandırması

- CleverHans
- IBM -Adversarial-Robustness-Toolbox

3 Saldırılar

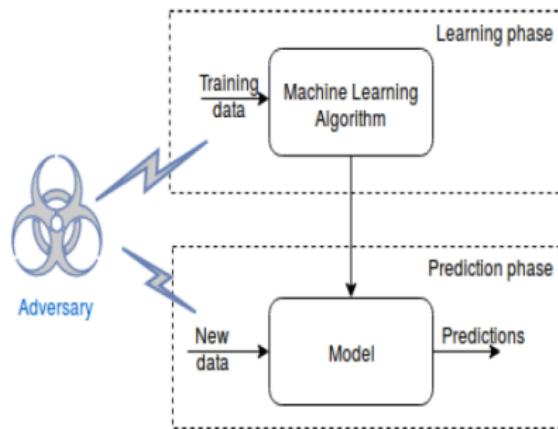
- Fast gradient sign method (FGSM)
- Jacobian-based Saliency Map Attack
- DeepFool
- Virtual Adversarial Training

Saldırgan Makine Öğrenmesi I

Adversarial Machine Learning

Saldırgan Makine Öğrenmesi

- ▶ ML sistemine karşı çalışan bir saldırganına, modelin etkinliğini azaltmak için çalışmasıdır.



Şekil: Saldırgan Makine Öğrenmesi

Tehdit Modelleri

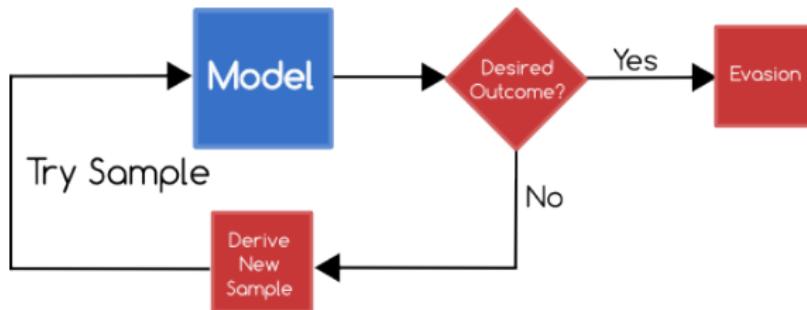
Tehdit Modelleri

- ▶ **Etki**
 - ▶ **Causative**: Öğrenme prosesini etkilemektedir. (**Poisoning**)
 - ▶ **Exploratory**: Eğitilmiş modele karşı yapılan saldırılar. (**Evasion**)
- ▶ **Güvenlik ihlalleri**: Yanlış sınıflandırmalar, sistemin kullanılamaz hale gelmesi (DoS).

Evasion Attacks

Evasion Attacks

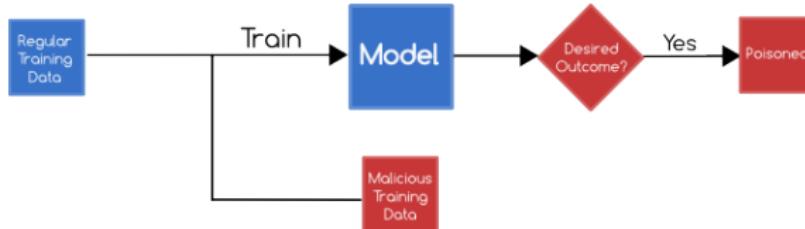
- ▶ En basit model saldırıları, öğrenme sonucunu atlatmaya çalışan saldırılardır.
- ▶ Saldırgan sadece kısmi sonuçları gözlemlleyebilir.
- ▶ Örnek: spam e-postaları göndermek isteyen bir saldırgan, ilk olarak, spam e-postalarını zararsız olarak sınıflandırmak için bir yol bulmak üzere modele karşı bir dizi farklı e-posta içeriğini deneyebilir.



Poisoning Attacks

Poisoning Attacks

- ▶ Bir saldırgan, öğrenim sonucunu etkilemek amacıyla eğitim verilerinizi etkilemeye odaklanabilir.
- ▶ **Örnek:** Anormal trafiği algılayan bir sınıflandırıcıyı eğitmek için şu anda ağ trafiğinin toplandığını bilen bir saldırgan, bu ağa trafik oluşturabilir, böylece model oluşturulduğunda, saldırı bağlantılarını anomali olarak tespit edemez.



Saldırı Sınıflandırması

	Ettirgen (Causative)	Keşif (Explatory)
Hedefli	Sınıflandırıcı, belirli pozitif örneklerde yanlış eğitilmiştir.	Pozitif örneklerin belirli bir alt kümelerinin yanlış sınıflandırılması
Ayrım Yap-mayan (Indis-crminate)	Sınıflandırıcı genellikle pozitif örneklerde yanlış eğitilir.	Yanlış sınıflandırılan pozitif örnekler
	Eğitim Aşamasında	Test Aşamasında

CleverHans

Cleverhans

- ▶ Açık-kaynak kütüphane
- ▶ <https://github.com/openai/cleverhans>
- ▶ Tensorflow üzerinde geliştirilmiş bir kütüphane
- ▶ Saldırılar için standart uygulama
 - ▶ Standard implementation of attacks
- ▶ Keras desteği bulunmaktadır.



IBM -Adversarial-Robustness-Toolbox

IBM-ART

IBM-ART

- ▶ Makine öğrenimi modelleri için saldırılardan ve savunma yöntemlerinin hızlı analizine izin vermektedir.
- ▶ <https://github.com/IBM/adversarial-robustness-toolbox>
- ▶ Kullanım alanları
 - ▶ Measuring model robustness
 - ▶ Model hardening
 - ▶ Runtime detection

[IBM / adversarial-robustness-toolbox](#)

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights

This is a library dedicated to adversarial machine learning. Its purpose is to allow rapid crafting and analysis of attacks and defense methods for machine learning models. The Adversarial Robustness Toolbox provides an implementation for many state-of-the-art methods for attacking and defending classifiers. <https://developer.ibm.com/code/open/p...>

adversarial-examples python machine-learning deep-learning deep-neural-networks defense-methods attack

465 commits 1 branch 1 release 7 contributors MIT

İçindekiler

1 Adversarial Examples

- Giriş
- Hedef Algoritmalar
- Adversarial Examples in the Human Brain
- Lets fool a binary linear classifier

2 Adversarial Machine Learning

- Giriş
- Tehdit Modelleri

- Saldırı Sınıflandırması
- CleverHans
- IBM -Adversarial-Robustness-Toolbox

3 Saldırılar

- Fast gradient sign method (FGSM)
- Jacobian-based Saliency Map Attack
- DeepFool
- Virtual Adversarial Training

Fast gradient sign method (FGSM) I

Fast gradient sign method (FGSM)

- ▶ Bu yöntem, gradyan yönünde piksel genişliğinde pertürbasyon ekleyerek rakip bir görüntüyü hesaplamaktadır.

$$\mathbf{X}^{adv} = \mathbf{X} + \epsilon \cdot sign(\nabla_{\mathbf{X}} \ell(\mathbf{X}, y_{true})) \quad (1)$$

- ▶ \mathbf{X} : Clean image (input)
- ▶ \mathbf{X}^{adv} : perturbated adversarial image
- ▶ ℓ : loss function
- ▶ y_{true} : true label for input \mathbf{X}

Fast gradient sign method (FGSM) II

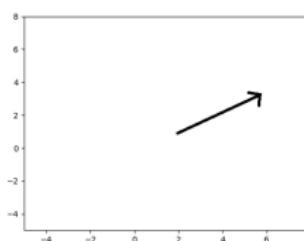
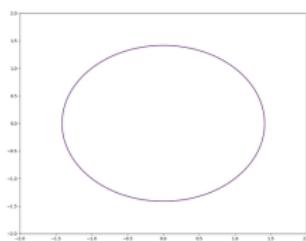
Gradient

- ▶ Bir fonksiyonun bir noktadaki gradyan değeri bu noktada yer alan eğimi ifade eder.
- ▶ Bir fonksiyonun $f(x_1, x_2, \dots, x_n)$, bir noktadaki (x_1, x_2, \dots, x_n) gradyanı bulunması için türevi ∇f alınarak bu nokta yerine konulur.

Fast gradient sign method (FGSM) III

Örnek: $f(x, y) = x^2 + y^2 - 1.5$

$$\nabla f(x, y) \Rightarrow \begin{aligned} \frac{\partial f}{\partial x} &= 2x \\ \frac{\partial f}{\partial y} &= 2y \end{aligned} \Rightarrow \begin{aligned} (2, 1) \text{ noktasında gradyan} \\ \nabla f = (2 \times 2, 2 \times 1) = (4, 1) \end{aligned}$$



Numerical Gradient

`numpy.gradient`

```
>>> f = np.array([1, 2, 4, 7, 11, 16], dtype=float)
>>> np.gradient(f)
array([ 1. ,  1.5,  2.5,  3.5,  4.5,  5. ])
```

Fast gradient sign method (FGSM) IV

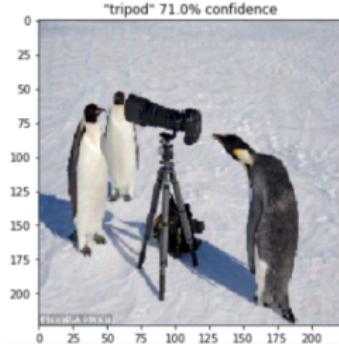
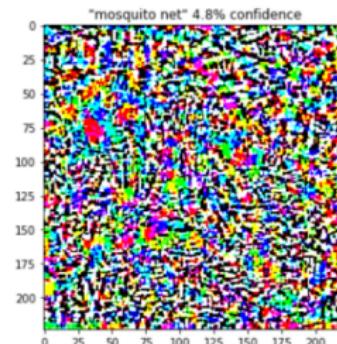
$$\mathbf{X}^{adv} = \mathbf{X} + \epsilon \cdot sign(\nabla_{\mathbf{X}} \ell(\mathbf{X}, y_{true}))$$

- ▶ Her bir pixel için kayıp fonksiyonunun gradyanı hesaplanır.
 - ▶ $\nabla_{\mathbf{X}} \ell(\mathbf{X}, y_{true}) \in \mathbb{R}^{m \times n \times k}$ - (width, height, channels)
- ▶ $\eta = \epsilon \cdot sign(\nabla_{\mathbf{X}} \ell(\mathbf{X}, y_{true}))$ sadece eğimin işaretini kullanılarak pixel değerlerinin azalmasına veya artmasına bakılır.
- ▶ düşük bir ϵ değeri ile çarpılarak *perturbation matrix* oluşturulur.
- ▶ $\mathbf{X}^{adv} = \mathbf{X} + \eta$

Fast gradient sign method (FGSM) V

 x

$+ \varepsilon \operatorname{sign}(\nabla_x L(\theta, x, y)) =$

 x_{adv} 

Şekil: white-box attacks.

Fast gradient sign method (FGSM) VI

White-Box Problemi

- ▶ **Çözüm:** Aynı sınıflandırma probleminin çözümü için h modeli oluşturur.
- ▶ h modeline FGSM kullanarak adversarial örnekler oluşturarak gönder.
- ▶ Sınıflandırma sonucuna göre parametreleri değiştir.

Fast gradient sign method (FGSM) VII

Targeted fast gradient sign method (T-FGSM)

- ▶ Gradient ile ters yönde perturbation işlemi gerçekleştirilmesi
- ▶ Hedef bir etiket olması

$$\mathbf{X}^{adv} = \mathbf{X} - \epsilon \cdot sign(\nabla_{\mathbf{X}} \ell(\mathbf{X}, y_{target}))$$

- ▶ y_{target} : the target label for the adversarial attack.

Fast gradient sign method (FGSM) VIII

Iterative fast gradient sign method (I-FGSM)

- ▶ T gradyan adımı

$$\mathbf{X}_0^{adv} = \mathbf{X}$$

$$\mathbf{X}_{t+1}^{adv} = \mathbf{X}_t^{adv} + \alpha \cdot \text{sign}(\nabla_{\mathbf{X}} \ell(\mathbf{X}, y_{target}))$$

- ▶ $\alpha = \frac{\epsilon}{T}$

Lab

Lab - 1

Jacobian-based Saliency Map Attack (JSMA) I

JSMA

- ▶ Bir ağ modelinin sınıf etiketleri olasılığa göre hesaplanır.
 $\text{label}(\mathbf{x}) = \arg \max_j F_j(\mathbf{x})$
- ▶ Bir saldırgan, \mathbf{x} örneğinin etiketini t olmasını istiyorsa, $F_t(\mathbf{x})$ değerini artırmalı, diğer sınıflar için $F_j(\mathbf{x})$ olasılığı azaltmalıdır.
- ▶ Bu amaçla bazı nitelikleri artırarak yapabilir. Bu amaçla **saliency map** (çıkıntı/göze çarpma) oluşturur.

$$S(\mathbf{X}, t)[i] = \begin{cases} 0 & \text{if } \frac{\partial F_t(\mathbf{X})}{\partial \mathbf{x}_i} < 0 \text{ or } \sum_{j \neq t} \frac{\partial F_j(\mathbf{X})}{\partial \mathbf{x}_i} > 0 \\ \left(\frac{\partial F_t(\mathbf{x})}{\partial \mathbf{x}_i} \right) \left\| \sum_{j \neq t} \frac{\partial F_j(\mathbf{X})}{\partial \mathbf{x}_i} \right\| & \text{otherwise} \end{cases}$$

- ▶ 1. satırda eğer hedef etiket t için azalma varsa veya diğer sınıfların olasılığının toplamında artış varsa 0 olarak kabul edilir.
- ▶ \mathbf{x}_i ifadesi artırılıyorsa $\frac{\partial F_t(\mathbf{x})}{\partial \mathbf{x}_i}$ ifadesi pozitif olmalıdır.
- ▶ \mathbf{x}_i ifadesi azaltılıyorsa $\frac{\partial F_t(\mathbf{x})}{\partial \mathbf{x}_i}$ ifadesi negatif olmalıdır.

Jacobian-based Saliency Map Attack (JSMA) II

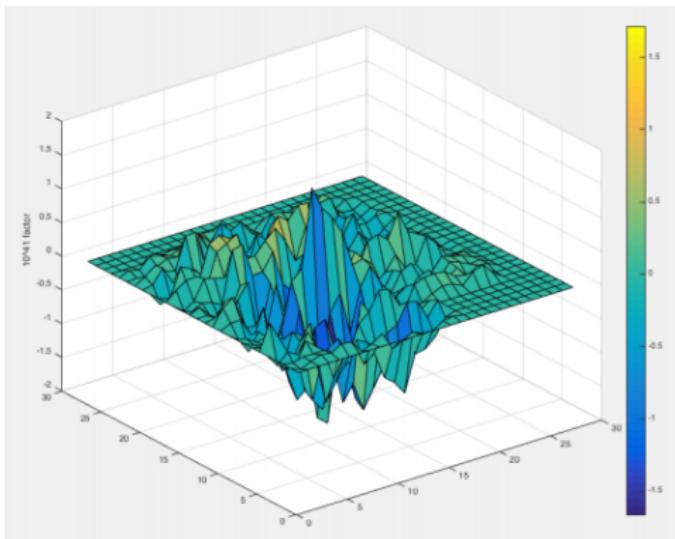


Fig. 7: Saliency map of a 784-dimensional input to the LeNet architecture (cf. validation section). The 784 input dimensions are arranged to correspond to the 28x28 image pixel alignment. Large absolute values correspond to features with a significant impact on the output when perturbed.

Jacobian-based Saliency Map Attack (JSMA) III

Algorithm 1 Crafting adversarial samples

\mathbf{X} is the benign sample, \mathbf{Y}^* is the target network output, \mathbf{F} is the function learned by the network during training, Υ is the maximum distortion, and θ is the change made to features. This algorithm is applied to a specific DNN in Algorithm 2.

Input: \mathbf{X} , \mathbf{Y}^* , \mathbf{F} , Υ , θ

- 1: $\mathbf{X}^* \leftarrow \mathbf{X}$
 - 2: $\Gamma = \{1 \dots |\mathbf{X}|\}$
 - 3: **while** $\mathbf{F}(\mathbf{X}^*) \neq \mathbf{Y}^*$ and $\|\delta_{\mathbf{X}}\| < \Upsilon$ **do**
 - 4: Compute forward derivative $\nabla \mathbf{F}(\mathbf{X}^*)$
 - 5: $S = \text{saliency_map}(\nabla \mathbf{F}(\mathbf{X}^*), \Gamma, \mathbf{Y}^*)$
 - 6: Modify $\mathbf{X}_{i_{max}}^*$ by θ s.t. $i_{max} = \arg \max_i S(\mathbf{X}, \mathbf{Y}^*)[i]$
 - 7: $\delta_{\mathbf{X}} \leftarrow \mathbf{X}^* - \mathbf{X}$
 - 8: **end while**
 - 9: **return** \mathbf{X}^*
-

Lab

Lab - 2

DeepFool I

DeepFool

- Minimal Perturbation \mathbf{r} , Sufficient to change the estimated label $\hat{k}(\mathbf{x})$:

$$\Delta(\mathbf{x}; \hat{k}) = \min_{\mathbf{r}} \|\mathbf{r}\|_2 \text{ s.t. } \hat{k}(\mathbf{x} + \mathbf{r}) \neq \hat{k}(\mathbf{x})$$

- $\Delta(\mathbf{x}; \hat{k})$: the robustness of \hat{k} at point \mathbf{x}

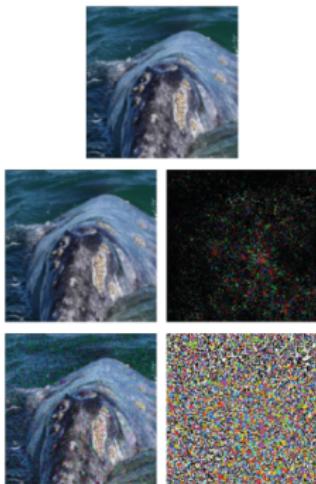


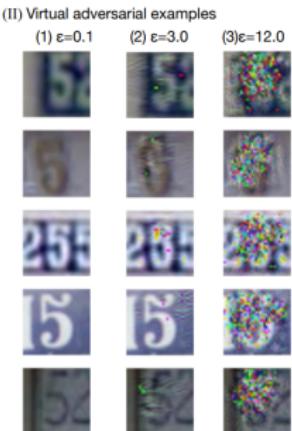
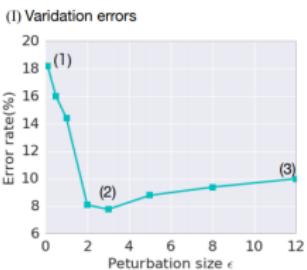
Figure 1: An example of adversarial perturbations. First row: the original image \mathbf{x} that is classified as $\hat{k}(\mathbf{x})$ =“whale”. Second row: the image $\mathbf{x} + \mathbf{r}$ classified as $\hat{k}(\mathbf{x} + \mathbf{r})$ =“turtle” and the corresponding perturbation \mathbf{r} computed by DeepFool. Third row: the image classified as “turtle” and the corresponding perturbation computed by the fast gradient sign method [4]. DeepFool leads to a smaller perturbation.

Virtual Adversarial Training I

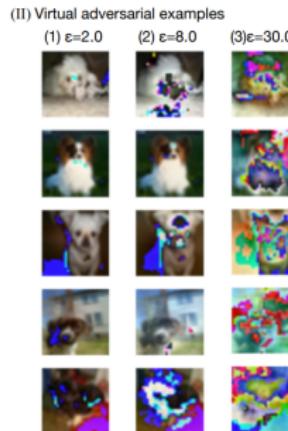
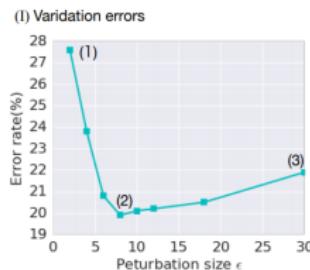
Virtual Adversarial Training

- ▶ Yeni bir regularization yöntemi
- ▶ GAN benzeri bir yöntem. Bir model ortaya çıkarılmaktadır.
- ▶ Oluşan model bir **sınıflandırıcıdır**. Üretici bir model değildir.

Virtual Adversarial Training II



(a) SVHN



(b) CIFAR-10

Lab

Lab - 3