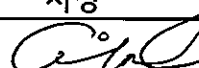


## 학부 연구생(랩 인턴) 수행 계획서

성명	정호용	학번	2017320243
소속대학	정보대학	소속학과	컴퓨터학과
학년	3	연락처	010-6490-4192
소속 연구실	정보시스템보안연구실	지도교수	허준범
수행기간	2019년 12월 01일 ~ 2019년 12월 31일	비고사항	
수행 계획 요약	<p>프로젝트명: Disclosing NN Structure for Better Adversarial Attacks (기존 프로젝트명 수정)</p> <p>1. timing side channel을 통해 target neural network 구조 파악 (grey-box 상황 구축)</p> <p>2. grey-box와 black-box attack의 성능 비교 (필요 query 수, adversarial attack 성공률)</p> <p>3. 기타 side-channel 이용 가능 여부 파악 (memory 등)</p>		

기업체 멘토 확인	회사명	성명	서명
	(주)아이오디큐브	이영준	

## ( 12 )월 랩인턴 업무 요약 보고서

2019.12.01. ~ 2019.12.31.

<b>랩인턴 업무 제목</b>	Disclosing NN Structure for Better Adversarial Attacks
------------------	--

### 1. 주요 수행 내용

1. timing side channel을 통해 target neural network 구조 파악
  - V Duddu, et al. *Stealing Neural Networks via Timing Side Channels* 참조
  - 간단한 실험을 위해 VGG11, VGG13, VGG16, VGG19 4가지로 구성
  - 본 논문에서는 강화학습+RNN을 사용해 target network 구조를 유추했지만 시간상의 문제로 각 target network의 inference time만 측정 및 비교
  - layer 수를 제외한 하이퍼파라미터는 공격자가 알고 있다고 가정

### 2. grey-box와 black-box attack의 성능 비교

- grey-box 상황에서 adversarial attack을 진행한 경우와 black-box 상황에서 adversarial attack을 진행한 경우 비교 (attack success rate와 필요 query 수 측면에서)
- attack success rate 상승함을 확인

### 3. 기타 side-channel 이용 가능 여부 파악

- target network와 공격자가 동일 machine 상에 위치할 경우, timing 외에 여러 side-channel 사용 가능해짐
- 공격자가 edge device 통제권을 갖고있는 시나리오 보다는 클라우드 상의 MLaaS (Machine Learning as a Service)에서 사용할 수 있는 side-channel에 집중

### 2. 기타 내용(현장업무에 대한 소견, 향후 활용, 개선 계획 등)

#### - 소견


기말고사 기간과 겹쳐 12월은 프로젝트 진행이 어려웠음

grey-box 상황의 attack success rate 상승은 확인했지만, 필요 query 수는 정량적으로 수치화하지 못함

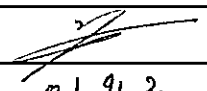
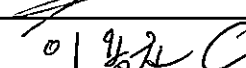
#### - 향후 활동

각 attack 시나리오 별 필요 query 수 수치화

공격자가 target network의 머신에 접근할 권한이 있는 경우, timing 이외의 다른 side-channel 경로 사용 시나리오 구성

기업체 멘토 확인	회사명	성명	서명
	(주)아이오씨유브	이 동 호	

## 랩인턴 최종 결과 보고서

Disclosing NN Structure for Better Adversarial Attacks				
수행기간	2019.09.16. ~ 2019.12.31.			
이 름	정호용	학 번	2017320243	
소속 연구실	정보시스템보안연구실	지도교수(서명)		
기업체 멘토	소속	(주)아이오티큐브	이름(서명)	

### 1. 연구의 개요

adversarial attack의 목표는 classifier로 하여금 잘못된 결과를 이끌어내도록 input을 조작하는 것이다. 이때, 원본 input에 가해지는 perturbation은 육안으로 식별 불가능하도록 해야 한다. 지금까지 다양한 adversarial attack 알고리즘들과, 이에 대한 defence strategy 또한 소개되었다.

White-box adversarial attack이란, 공격자가 target classifier의 내부 정보를 모두 알고 있는 공격 시나리오이다. learning rate, layer 수 등의 하이퍼파라미터 뿐만 아니라 인공신경망 내부 모든 노드의 가중치까지 target classifier의 내부 정보에 해당한다.

이에 반해 black-box adversarial attack이란, 공격자가 target classifier의 내부 정보를 전혀 모르는 상황에서의 공격 시나리오이다. 공격자는 target classifier에게 query를 던져 특정 input에 대한 classifier output을 알아낼 수 있는 점을 제외하곤 아무런 정보도 알지 못한다. 따라서 white-box 시나리오보다 적용범위가 더 넓다. 하지만 black-box adversarial attack의 경우, query를 통해 알아낸 input-output pair로 substitute network를 구축하고, 이에 맞춰 adversarial attack을 진행해야 하기에 피치못할 성능 감소가 존재한다. 뿐만 아니라, input-output pair를 알아내기 위한 query의 수도 무시할 수 없다.

본 프로젝트에서는 side-channel attack을 통해 target classifier의 정보를 수집하고, 이를 기반으로 보다 효율적이고 강력한 black-box adversarial attack 시나리오를 구축한다.

### 2. 연구수행 방법

#### 1) 연구 목표

side-channel을 통해 target neural network 구조를 파악하고, 이를 기반으로 보다 효과적인 black-box adversarial attack 시나리오 구축함을 목표로 한다. 이 때 adversarial attack의 성능이 어느정도 상승하는지 알아본다.

#### 2) 연구 수행 내용 및 방법

우선, 성능 평가의 기준이 될 white-box attack과 black-box attack을 설정하기 위해 다

양한 white-box attack과 black-box attack 구현 및 비교한다.

adversarial attack의 target network는 VGG16 네트워크를 CIFAR10 dataset에 맞게끔 변형하여 사용한다. (마지막 3개의 fully-connected layer 대신에 하나의 512-to-10 fully-connected layer을 사용한다.)

본 프로젝트에서의 모든 adversarial attack은 non-targeted로 설정한다.

white-box adversarial attack은 크게 Gradient-based approach와 Network-based approach로 나눠서 분석한다.

Gradient-based approach 중 대표적인 두가지 알고리즘을 분석한다.

- Fast Gradient Sign Method (FGSM)
- Projected Gradient Descent (PGD)

Network-based approach 중 대표적인 세가지 알고리즘을 분석한다.

- Perturbation - Adversarial Transformation Network (P-ATN)
- Adversarial Auto-Encoder (AAE)
- Adversarial Generative Adversarial Network (AdvGAN)

black-box adversarial attack은 Substitute neural network를 사용하여 진행한다.

white-box approach 중 성능이 가장 좋은 알고리즘을 선택하고, 이를 기준으로 실험을 진행한다.

side-channel을 통한 black-box adversarial attack의 성능이 전통적인 black-box attack에서 얼마나 개선되었고, white-box attack의 성능에 얼마나 근접한지 분석한다.

### 3. 연구 결과

#### 1) 연구 수행 결과

Gradient-based approach 구현 결과

- Fast Gradient Sign Method (FGSM)

Epsilon: 0	Test Accuracy = 918 / 1000 = 0.918
Epsilon: 0.03	Test Accuracy = 569 / 1000 = 0.569
Epsilon: 0.06	Test Accuracy = 481 / 1000 = 0.481
Epsilon: 0.09	Test Accuracy = 437 / 1000 = 0.437
Epsilon: 0.12	Test Accuracy = 400 / 1000 = 0.4
Epsilon: 0.15	Test Accuracy = 376 / 1000 = 0.376
Epsilon: 0.18	Test Accuracy = 351 / 1000 = 0.351

attack success rate를 높이기 위해서는 epsilon을 크게 설정해야 하는데, epsilon이 커질수록 perturbation이 육안으로 식별 가능해진다. 따라서 본 연구에 적합한 adversarial attack이 아니다.

- Projected Gradient Descent (PGD)

FGSM의 개선된 버전인 만큼, perturbation range를 제한한 상태에서도 거의 100%의 attack success rate를 보인다. (본 프로젝트에서는 편의상 perturbation range 단위를 L-infinity distance로 설정했다.)

Network-based approach 구현 결과

- Perturbation - Adversarial Transformation Network (P-ATN)

```

self.conv = nn.Sequential(
    nn.Conv2d(3, 512, 3, stride=2, padding=1),
    nn.ReLU(),
    nn.Conv2d(512, 256, 3, stride=2, padding=1),
    nn.ReLU(),
    nn.Conv2d(256, 128, 3, stride=2, padding=1),
    nn.ReLU(),
)

self.fc = nn.Sequential(
    nn.Linear(2048, 512),
    nn.ReLU(),
    nn.Linear(512, 3072),
    nn.Tanh(),
)

```

correct/adv\_correct: 9084/3699  
 avg.L\_2: 6.7575  
 avg.L\_inf: 0.1255  
 (50 epochs)

준수한 공격 성능을 보이지만, PGD에는 미치지 못한다. output image에서 perturbation이 어느정도 식별 가능하다.

#### - Adversarial Auto-encoder (AAE)

```

self.conv = nn.Sequential(
    nn.Conv2d(3, 12, 3),
    nn.ReLU(),
    nn.Conv2d(12, 24, 3),
    nn.ReLU(),
    nn.Conv2d(24, 48, 3),
    nn.ReLU(),
)

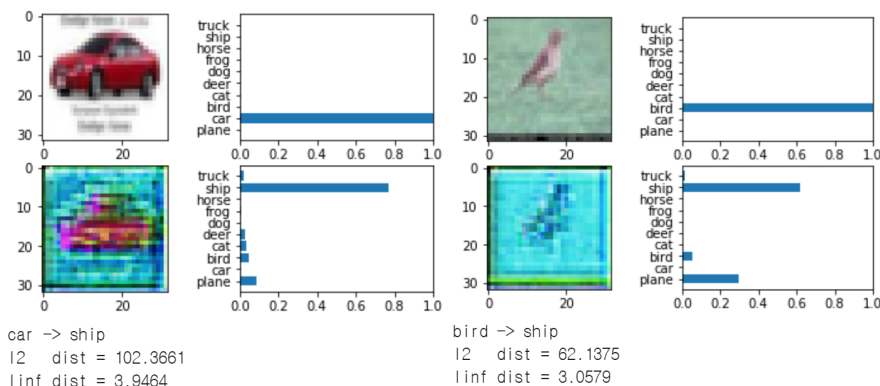
self.deconv = nn.Sequential(
    nn.ConvTranspose2d(48, 24, 3),
    nn.ReLU(),
    nn.ConvTranspose2d(24, 12, 3),
    nn.ReLU(),
    nn.ConvTranspose2d(12, 3, 3),
    nn.Tanh(),
)

```

correct/adv\_correct: 9084/3378  
 avg.L\_2: 14.9406  
 avg.L\_inf: 1.3152  
 (100 epochs)

perturbation mask를 만들어내는 P-ATN이나 AdvGAN과 달리, perturbed image 자체를 만들어내기 때문에 perturbation range를 결정적으로 제한할 수 없다. 따라서 결과물의 perturbation이 육안으로 식별 가능하기에 본 프로젝트와는 적합하지 않다.

#### - Adversarial Generative Adversarial Network (AdvGAN)



AdvGAN은 디버깅 문제(perturbation range 제한 실패)로 기간 내 구현에 실패했다.

위 구현 결과에 따라, black-box adversarial attack 중 가장 좋은 성능을 보인 PGD를 선택했다.

PGD의 경우, 각 sample에 대해 target network에 10번의 query와, 각 query에 대한 gradient 정보가 필요하다. 이에 반해 substitute neural network를 통한 black-box attack은 substitute network가 구성되면, 더 이상의 오버헤드 query가 필요하지 않다. 하지만, substitute network를 구성하기까지 많은 query를 필요로 한다.

공격자는 substitute network에 대한 아무런 정보가 없으므로 모든 하이퍼파라미터를 추정하여 학습을 진행해야 한다. 이 부분에서 substitute network 구조를 알아내기 위해 timing side-channel을 이용한다.

구체적인 구현은 Stealing Neural Networks via Timing Side Channels (V Duddu et. al., 2019)에 소개된 방법을 따른다. (논문에서는 강화학습과 RNN을 사용해 target network 구조를 유추했지만, 시간상의 문제로 target network의 inference time만을 측정해 비교했다.)

모델 구조를 잘못 추측해서 adversarial attack을 진행하였을 때와, timing side-channel을 통해 정확히 모델 구조를 예측한 후 adversarial attack을 진행하였을 때의 성능을 비교한다.

Target Network: VGG16

	Network	Attack success rate
Naive suspicion	VGG11	54.24%
Accurate suspicion (via timing side-channels)	VGG16	76.19%

## 2) 연구 전후 차이점

timing side-channel만으로는 target network에 대해 알아낼 수 있는 정보에 한계가 있다. (본 프로젝트에서는 target network 종류를 몇가지 이내로 한정되었기 때문에 timing side-channel만으로도 정확한 모델 구조 파악이 가능했다. 하지만 이는 현실성이 없는 설정이다. 또한 본 프로젝트에서는 timing 측정 시 같은 로컬 머신에서 실행된 inference time을 측정했기에, network overhead가 추가된 현실적인 MLaaS 환경에서는 성능이 하락할 것으로 예상된다.) 한정된 정보만으로도 개선된 adversarial attack이 가능하지만, 보다 나은 공격 성능을 위해 다른 side-channel 경로도 사용해야 한다. 하지만 대부분의 side-channel exploit 시나리오는 공격자가 edge device의 통제권을 갖고있는 hardware side-channel을 기반으로 하기 때문에 본 프로젝트의 시나리오와는 괴리가 있다. 따라서 MLaaS (Machine Learning as a Service) 상에서도 사용할 수 있는 side-channel attack 시나리오가 필요하다.

## 4. 향후 계획

timing side-channel을 통해 모델 구조를 알아낸 상황에서의 attack success rate 상승은 확인했지만, 필요 query 수는 정량적으로 수치화하지 못했다. 따라서 각 attack 시나리오 별 필요 query 수 수치화가 필요하다.

정보량이 제한적인 timing side-channel만 사용하여 공격을 진행하는 것은 현실성이 부족한 시나리오이다. 기타 side-channel을 통한 공격 강화가 필요하다. (e.g. 공격자가 target network의 머신에 접근할 권한이 있는 경우, memory side-channel이 사용 가능하다.)