**초록**

# **제목**

|  |  |  |
| --- | --- | --- |
| 이연우 | 정민기 | 한대일 |
| 인공지능응용학과 | 인공지능응용학과 | 인공지능응용학과 |
| 21102376 | 21102385 | 21102398 |

사람의 육안은 쉽게 구별할 수 있지만, 딥러닝 모델에게는 구별이 쉽지 않은 데이터를 생성하는 기술을 adversarial attack이라고 한다. 이 프로젝트에선 adversarial attack을 구현하여, 이미지 분류 모델에 대한 adversarial attack의 성능을 측정한다. 그리고 adversarial attack에 대하여 data augmentation 및 모델의 층 수 증대 등의 효과적인 방어책을 모색하여 결과를 도출한다. 실험 결과, data augmentation 중에서 가장 효율적인 방어책은 Random Rotation으로 나타났다. 또한 모델의 층이 증가할수록 adversarial attack에 대한 방어 능력이 높아진다는 것이 확인되었다.

1. **서론**

본 프로젝트는 머신러닝2 강의 4주차에서 다루었던 adversarial example에 대한 의구심을 토대로 한다. 이미지 분류 문제에서 adversarial example란, adversarial attack으로 생성한 이미지 데이터를 의미한다. adversarial attack이란 전통적인 가중치 업데이트 방식과 반대되는 방향으로 가중치를 업데이트하여 DNN 이미지 분류 모델이 다루는 데이터에 대하여 최악에 가까운 노이즈 이미지를 생성하는 이론이다.

본 프로젝트는 상기의 adversarial attack에 대하여 효과적인 방어 방법이 무엇인지 찾아내는 것을 그 골자로 한다. 구체적으로 몇 가지 data augmentation 이론과 모델의 레이어 깊이에 대한 변화를 통해서 효과적인 방어 방법에 대하여 고찰해 볼 것이다.

데이터셋으로는 MNIST 데이터셋과 CIFAR10 데이터셋을 사용한다. 두 데이터셋에 대하여 같은 실험을 반복하여, 실험의 일관성을 확인할 것이다. DNN 모델은 LeNet-5모델을 수정하여 사용할 것이다.

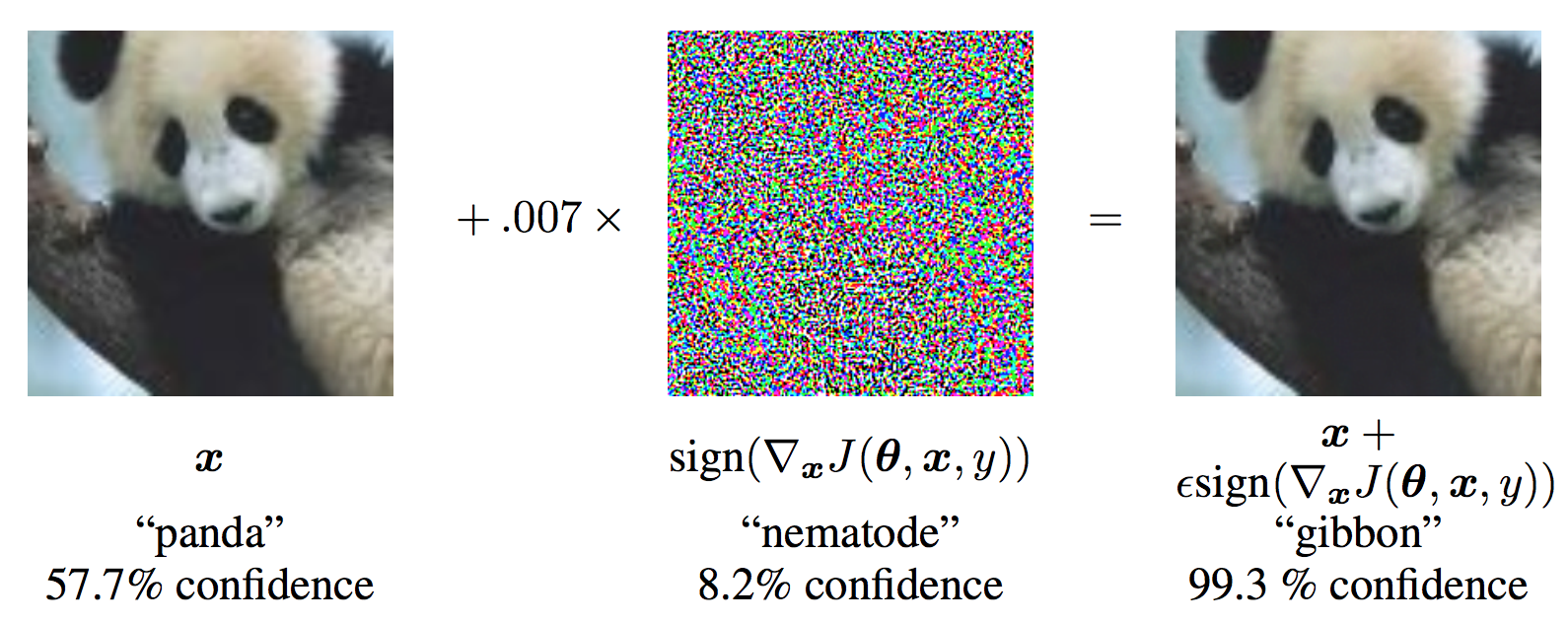
adversarial attack 모델로는 Fast Gradient Sign Method(FGSM)을 사용할 것이다. FGSM은 신경망 모델이 적대적 변동 (adversarial perturbation)을 견디기에는 지나치게 선형적이라는 가정하에, 이미지 분류의 결정 경계를 넘을 수 있는 최소한의 노이즈 분포를 찾는 방법론이다.

Data augmentation 방법으로는 Random Horizontal Flip, Random Rotation, Random Affine, Color Jitter, Gaussian Blur를 각각 사용하여 모델을 학습시킨 뒤, 각각의 모델에 대하여 adversarial attack을 감행하여, epsilon에 따른 accuracy 감소량을 확인할 것이다.

모델의 레이어 깊이는 각각 5층, 7층, 8층 구조로 깊이를 늘려가며 adversarial attack에 대한 방어 능력을 측정할 것이다.

1. **본론**

앞서 언급하였듯이 본 프로젝트의 핵심이 되는 adversarial attack의 방법으로 FSGM을 사용한다. FSGM은 Ian Goodfellow가 고안한 adversarial attack 방법론으로 가장 널리 쓰이는 adversarial attack 방법론 중의 하나이다. 이 방법론은 놀라울 정도로 강력하면서도 직관적이다. 모델이 그래디언트를 업데이트하는 방향을 고려하여 노이즈를 생성하도록 고안되었다. 즉, 그래디언트를 역전파하는 과정에서 손실을 최소화하는 방향으로 진행하기보다는, 오히려 이를 최대화하는 방향으로 입력 데이터를 업데이트하도록 고안되었다. 입력 데이터 그래디언트의 손실값을 활용하여 입력 데이터의 손실을 최대화하는 것이다.



예를 들어, 위의 그림을 살펴보면, 57.7%의 신뢰도로 “panda”라고 모델이 분류한 이미지 에 대하여, 우리가 바라는 라벨 , 그리고 모델 파라미터 를 바탕으로 모델을 업데이터하기 위한 손실 을 계산한다. 그리고 입력 데이터의 그래디언트 에 대한 방향을 sign 함수로 나타낸다. 이를 adversarial attack의 강도라고 볼 수 있는 가중치 을 곱한 뒤 입력 데이터에 반영하면 우리가 원하는 adversarial example를 생성할 수 있다. 이러한 adversarial example를 모델에 입력할 경우, 모델은 adversarial example를 “panda”가 아닌 “gibbon”으로 분류하게 되는 것이다. 즉, 육안으로는 명백하게 “panda”로 보이는 이미지를 DNN 모델은 전혀 다른 이미지로 분류한다는 것이 adversarial attack의 의의라고 할 수 있다.

FSGM에 대한 방어책으로 data augmentation과 네트워크 깊이 증대를 고안하였다. data augmentation 방법론으로는 Random Horizontal Flip, Random Rotation, Random Affine, Color Jitter, Gaussian Blur을 선택하였다.

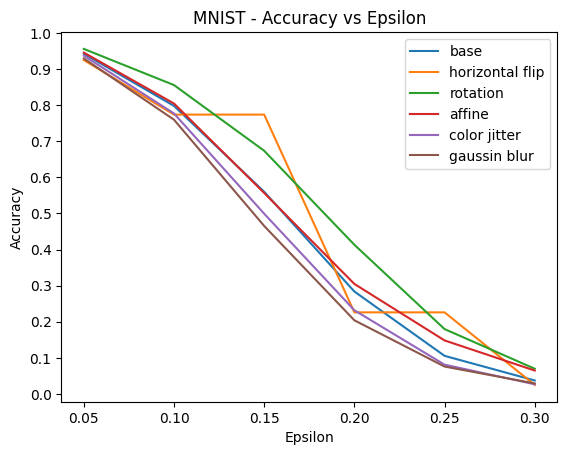
Random Horizontal Flip은 주어지는 이미지를 임의의 확률로 수평 방향으로 뒤집는 변형 방법이다. Random Rotation이란 주어진 이미지를 임의의 각도로 회전하는 변형 방법이다. Random Affine은 이미지를 랜덤하게 회전 및 이동하는 변형을 의미한다. Color Jitter는 이미지의 명도, 채도, 대비를 랜덤하게 변화시키는 변형을 의미한다. Gaussian Blur는 이미지를 가우시안 분포에 따라 흐릿하게 처리하는 변형 방법을 의미한다.

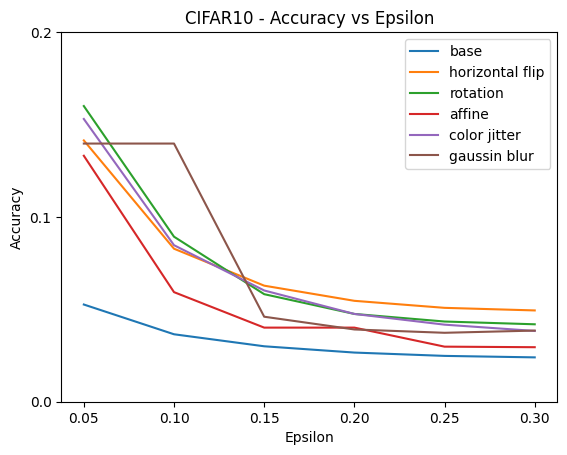
기본적인 학습 모델에 위 data augmentation 방법 5가지를 각각 적용한 5개의 모델에 대하여 가중치 데이터를 저장할 것이다. 그리고 이렇게 저장된 가중치 데이터에 대하여 각각 adversarial attack을 감행하여 어떠한 data augmentation 방법론이 adversarial attack에 대하여 가장 강건한지를 확인할 것이다.

모델의 레이어 깊이의 경우, 각각 5층, 7층, 8층 구조로 설계하였으며, 각 모델에 대한 파라미터 수는 MNIST 데이터셋을 기준으로 다음과 같이 설계하였다.

|  |  |  |  |
| --- | --- | --- | --- |
| **# Param** | **Net-5** | **Net-7** | **Net-8** |
| **MNIST** | 131,140 | 501,940 | 687,890 |
| **CIFAR10** | 62,006 | 153,662 | 216,406 |

1. **실험 결과**
   1. **Data Augmentation**





MNIST 데이터셋과 CIFAR10 데이터셋에 대하여 각각 adversarial attack을 시행하였다.으로 각각 실험하였을 때, FSGM이 잘 동작하는 것을 확인할 수 있다.

\*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **MNIST** | | **CIFAR10** | |
| **Acc** | **vanilla** | **to A.A.** | **vanilla** | **to A.A.** |
| **BM** | 0.9895 | 0.7976 | 0.7214 | 0.0365 |
| **RHF** | 0.9834 | 0.7736 | 0.7421 | 0.0828 |
| **RR** | 0.9900 | 0.8553 | 0.7251 | 0.0893 |
| **RA** | 0.9907 | 0.8042 | 0.7188 | 0.0593 |
| **CJ** | 0.9904 | 0.7767 | 0.7315 | 0.0847 |
| **GB** | 0.9895 | 0.7584 | 0.7251 | 0.0686 |

*(BM: Base Model, RHF: Random Horizontal Flip, RR: Random Rotation, RA: Random Affine, CJ: Color Jitter, GB: Gaussian Blur, A.A.: Adversarial Attack)*

Base Model은 data augmentation을 적용하지 않은 모델에 대한 실험 결과를 의미한다. Vanilla는 adversarial attack을 적용하지 않은 경우에 대한 정확도를 나타내는 행이다. To A.A.은 adversarial attack을 적용한 경우 정확도를 나타내는 행을 의미한다.

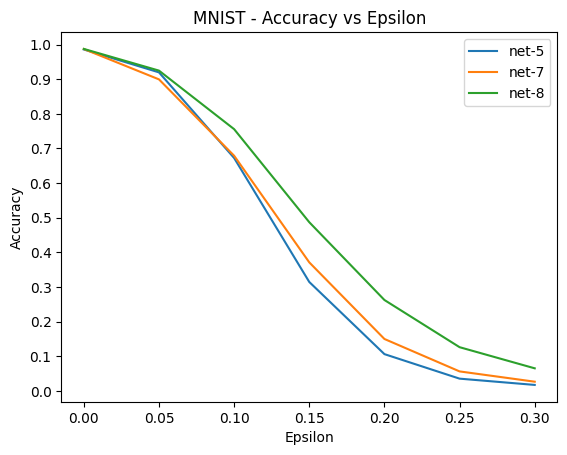
Data augmentation을 적용하지 않은 경우와 data augmentation을 적용한 경우를 비교하였을 때, MNIST 데이터셋의 경우 그다지 유의미한 차이를 보이지 않는다. 반면, CIFAR10 데이터셋의 경우 data augmentation을 적용하였을 때 adversarial attack에 대한 방어 능력이라고 할 수 있는 Acc to A.A.의 수치가 확연이 높음을 확인할 수 있다.

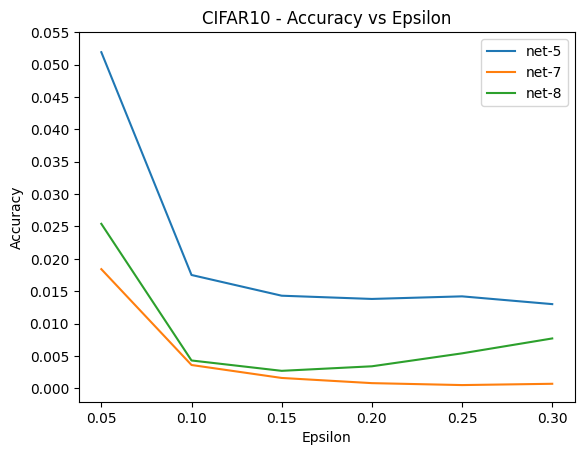
각 data augmentation 방법에 대하여 방어 능력을 비교해 보면, Random Rotation이 가장 강건함을 확인할 수 있다. 근본적인 모델의 정확도가 그다지 높지 않음에도 불구하고, Acc to A.A.이 가장 높은 수치를 보이기 때문이다.

* 1. **The Number of Parameters**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **MNIST** | | | **CIFAR10** | | |
| **Net-5** | **Net-6** | **Net-7** | **Net-5** | **Net-6** | **Net-7** |
| **0** | 0.9857 | 0.9870 | 0.9870 | 0.5514 | 0.5599 | 0.5546 |
| **0.05** | 0.9196 | 0.8994 | 0.9247 | 0.0519 | 0.0184 | 0.0254 |
| **0.1** | 0.6725 | 0.6790 | 0.7557 | 0.0175 | 0.0036 | 0.0043 |
| **0.15** | 0.3146 | 0.3712 | 0.4871 | 0.0143 | 0.0016 | 0.0027 |
| **0.2** | 0.1064 | 0.1501 | 0.2626 | 0.0138 | 0.0008 | 0.0034 |
| **0.25** | 0.0356 | 0.0565 | 0.1264 | 0.0142 | 0.0000 | 0.0054 |
| **0.3** | 0.0175 | 0.0268 | 0.0654 | 0.0013 | 0.0000 | 0.0077 |

레이어 깊이에 따른 모델의 방어 능력을 고려할 때, MNIST 데이터셋 모델의 경우 파라미터 수에 따른 정확도 증가가 뚜렷하게 비례함을 확인할 수 있다. 반면 CIFAR10 데이터셋 모델의 경우, 정확도의 증가가 그다지 뚜렷하지 않음을 확인할 수 있는데 이는 모델이 과소적합 되었기 때문이라고 의심된다.





1. **결론**

본 프로젝트에서는 대표적인 adversarial attack

방법인 FSGM에 대하여 data augmentation과 레이어 깊이에 따른 방어 능력을 실험하였다. Data augmentation의 경우 adversarial attack에 대해 유의미한 방어 능력이 확인되었으며, Random Rotation 알고리즘이 가장 효율적인 방법인 것으로 측정되었다. 각각의 데이터셋에 대한 적절한 파라미터를 계산하는 것은 추가적인 실험이 필요한 것으로 보이나 파라미터의 개수를 증가시키는 것 역시 효과적인 방어 방법으로 확인된다.

**참고 문헌**

[1] Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572 (2014).

[2] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.

[3] “Adversarial Example Generation - PyTorch Tutorials 1.13.0+cu117 documentation” https://pytorch.org/tutorials/beginner/fgsm\_tutorial.html?highlight=adversarial+attack