Efficient Adversarial Input Generation via Neural Net Patching

Tooba Khan, Kumar Madhukar & Subodh Sharma
{tooba.khan.jcs21, madhukar, svs}@iid.ac.in
Indian Institute of Technology Delhi

Introduction

- Presence of adversarial examples for deep neural networks(DNNs) limit the areas where they can be used.
- To use DNNs in safety critical domains, they have to be made robust against adversarial inputs.
- Scalability makes the adversarial input generation problem practically challenging.
- The generated adversarial inputs generally lack important characteristics like naturalness and output-impartiality.

Architecture

- Given a first layer modification, we only need to solve linear equations to find an adversarial input.
- AIGENT finds modification in the first weight layer using an iterative approach.
- It begins by finding modifications in the middle weight layer of the network and divides the network into two halves.
- Then, the later half of the network is discarded and a modification is found in the middle layer of the extracted network.
- Once we have a modification in the first weight layer, it is used to find an adversarial input.

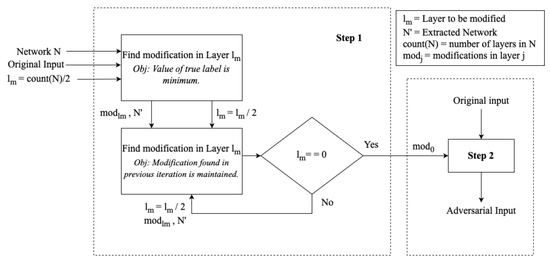


Figure 1: Architecture of AIGENT

Example

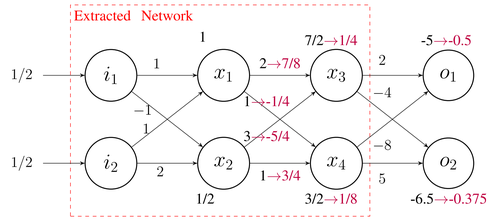


Figure 2: Middle-layer modification and sub-net extraction

- The above network has 3 weight layers. The first step finds a modification in layer 2.
- Fig.2 shows the modification found in the middle layer of the network.
- A subnetwork is extracted from inputs to layer 2.
- Modification is found in middle layer of extracted network i.e 1^{st} weight layer.
- As shown in Fig.3, the generated adversarial input changes the output of the original network.
- For the given network and input, originally the output is $o_1 > o_2$ and we want a modification such that $o_1 < o_2$.
- Since we want to increase o_2 we mark it as increment neuron and similarly o_1 as decrement neuron.
- This marking can be propagated to all network layers and speeds the constraint solving[Elboher et al., 2020].
- Edge weights only increase if connected to increment neurons and decrease for decrement neurons.

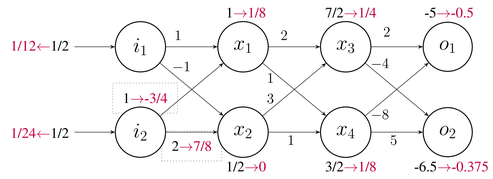


Figure 3: Adversarial inputs from first-layer modification. The adversarial input and the corresponding values of each neuron are written in red. The modification required in first layer weights are shown in black boxes.

Metrics of Evaluation

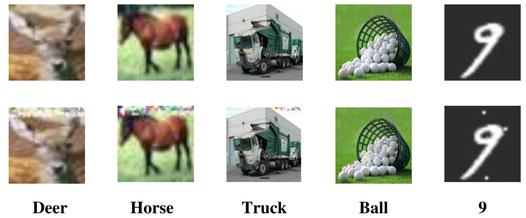
- **FID [Harel-Canada et al., 2020]:**
  - Measures naturalness i.e indicates whether set of two images are visibly different or not.
  - FID closer to 0 indicates that the adversarial images are natural, and are therefore desirable.
- **Defect Detection:**
  - The attack success rate, or the number of benchmarks for which our tool could successfully produce an adversarial image.
- **Pielou Score [Harel-Canada et al., 2020]:**
  - It is the measure of output impartiality. Reflect whether the adversarial image generation is biased towards any one of the output classes or not.

Results

- AIGENT performs better than all the other approaches in terms of FID.
- The adversarial images generated by AIGENT are natural and visibly quite similar to the corresponding original images.
- AIGENT modifies far fewer pixels as compared to the other approaches.
- AIGENT was able to achieve a good Pielou score on all the benchmark datasets.
- Although black box methods achieve higher defect detection, they modify 100% pixels which leads to visibly distinguishable images.
- AIGENT performs well in terms of defect detection, while keeping the modification quite small.

S. No.	Technique	FID	Pielou score	L-2	L-∞	Time (seconds)	Pixels modified	Pixels modified (%)	Defect detection
Benchmark dataset: MNIST									
1	AIGENT	**0.001**	0.725	**1.82**	0.66	1.726	**24**	**3.06%**	72.00%
2	AIGENT (high defect)	0.03	0.74	4.1	0.80	1.799	**24**	**3.06%**	91.40%
3	FGSM	1.73	**0.95**	2.8	**0.1**	0.069	784	100.00%	**99.00%**
4	Black Box	1.98	0.14	6.58	0.23	**0.065**	784	100.00%	88.40%
5	DeepXplore	0.02	0.47	5.16	1	11.74	60	7.65%	45.66%
6	DLFuzz	0.17	0.88	2.29	0.39	30	586	74.74%	92.36%
Benchmark dataset: CIFAR-10									
1	AIGENT	**0.00009**	**0.927**	**1.6**	0.5	12.01	**12**	**0.39%**	**100.0%**
2	FGSM	0.071	0.92	5.5	**0.1**	**0.079**	3072	100.00%	**100.0%**
3	Black Box	0.44	0.703	13.04	0.23	0.082	3072	100.00%	76.20%
Benchmark dataset: ImageNet									
1	AIGENT	**0.00011**	0.75	**6.81**	**0.73**	35	**300**	**0.61%**	**98.60%**
2	FGSM	0.43	**0.87**	22	0.1	0.4	16384	100.00%	97.00%
3	Black Box	0.05	0.8	52	0.4	**0.3**	16384	100.00%	90.00%
4	DeepXplore	0.032	N.A	58.04	0.51	84	15658	95.57%	59.13%
5	DLFuzz	0.11	N.A	61.1	0.6	57	16102	98.28%	92.00%

Table 1: Comparison of AIGENT with other state-of-the-art techniques on MNIST, CIFAR-10 and ImageNet datasets. Bold values indicate the best figure for each metric.



Deer predicted as frog for CIFAR-10 | **Horse predicted as dog for CIFAR-10** | **Truck predicted as instrument on Imagenet** | **Ball predicted as skydiving on Imagenet** | **9 predicted as 5 on MNIST**

Figure 4: Adversarial images produced by AIGENT (bottom row), and the corresponding original images (top row)

Conclusion

- Adversarial inputs are useful for adversarial training of DNNs, which can make the network robust.
- We have proposed a technique to generate adversarial inputs via patching of neural networks.
- AIGENT does significantly better than the state-of-the-art i.e it produces natural images, with a tiny fraction of pixels changed.

Future Work

- Better algorithms for DNN patching would make our technique more efficient.
- It would also be useful to find a minimal patch in order to get the closest adversarial example.

References

[Elboher et al., 2020] Elboher, Y. Y., Gottschlich, J., and Katz, G. (2020). An abstraction-based framework for neural network verification. In *Computer Aided Verification: 32nd International Conference, CAV 2020, Los Angeles, CA, USA, July 21–24, 2020, Proceedings, Part I*, page 43–65, Berlin, Heidelberg. Springer-Verlag.

[Harel-Canada et al., 2020] Harel-Canada, F., Wang, L., Gulzar, M. A., Gu, Q., and Kim, M. (2020). *Is Neuron Coverage a Meaningful Measure for Testing Deep Neural Networks?*, page 851–862. Association for Computing Machinery, New York, NY, USA.