# Defensive Distillation is Not Robust to Adversarial Examples

N. Carlini, D. Wagner

University of California, Berkeley

arXiv: 1607.04311

Reviewed by : Bill Zhang

University of Virginia

https://qdata.github.io/deep2Read/

# Outline

# Introduction

## Basic Premise and Motivation

- Defensive distillation was proposed as a defense against adversarial examples
- Using a slight modification to the standard adversarial attacks, distilled networks can be attacked
- Distillation only works on standard attack methods

# Background

## Adversarial Examples, Papernot's Attack

- Adversarial examples are instances $x'$ which are very close to a valid instance $x$ w.r.t. some distance metric, but with $C(\theta, x) \neq C(\theta, x')$; can be targeted or untargeted
- Papernot's attack uses a greedy approach where each iteration the attack picks and modifies pairs of pixels which make the target classification $t$ most likely; repeated until misclassification or 112 pixels modified
- Attack uses gradient to approximate each pixel's importance

# Background

## Papernot's Attack

- Define the following

$$\alpha_{pq} = \Sigma_{i \in \{p,q\}} \frac{\partial Z(x)_t}{\partial x_i}$$

$$\beta_{pq} = (\Sigma_{i \in \{p,q\}} \Sigma_{j=0}^{9} \frac{\partial Z(x)_j}{\partial x_i}) - \alpha_{pq}$$

  where $\alpha_{pq}$ represents how much changing $(p, q)$ will change the target classification and $\beta_{pq}$ represents how much it will change the other outputs

- Then, pick
  $(p*, q*) = argmax_{(p,q)}(-\alpha_{pq} * \beta_{pq}) * (\alpha_{pq} > 0) * (\beta_{pq} < 0)$
- Note that this attack uses the logits, not the softmax for gradients

# Background

## Defensive Distillation

- Three steps: train a teacher network using standard methods (and temperature 1), evaluate teacher set on each example in training set to obtain soft labels, train second network on soft labels as targets with some temperature $T$
- Then, to classify, run the distilled network using temperature $T = 1$
- By training using $T$, the logits (inputs to the softmax) become on average $T$ times larger to minimize cross-entropy loss; causes network to become much more confident when using $T = 1$ to predict
- $T = 100$ was found to be most difficult to successfully attack

# Breaking Distillation

## Why Distillation Works

- Papernot's attack uses the logit, not softmax layer
- $\alpha$ and $\beta$ (as defined above) represent changes to the inputs of the softmax layer
- ALL changes are equal importance even though relative differences between terms should be taken into account
- It is surprising that Papernot's attack even works on non-distilled networks and it could theoretically fail to find adversarial examples due to this flaw

# Breaking Distillation

## Why Distillation Works

- Since the distilled network is trained on $Z(\theta, x)/T$, the network essentially learns to multiply all logits by $T$; this was experimentally verified
- This only magnifies the flaw in Papernot's attack, causing it to fail spectacularly on distilled networks

# Breaking Distillation

## Modifying the Attack

- First, instead of taking gradient w.r.t. logits, take it w.r.t. output of network
- To deal with vanishing gradients (due to large absolute values of logits), artificially divide logits by T before passing them through: $\hat{F}(\theta, x) = softmax(Z(\theta, x)/T)$
- Second, use $\alpha - \beta$ instead of the product as used previously to slightly improve accuracy
- Using these modifications, can search individual pixels instead of pairs of pixels:

$$p* = argmax_p(2\frac{\partial \hat{F}(x)_t}{\partial x_p} - \Sigma_{j=0}^{9}\frac{\partial \hat{F}(x)_j}{\partial x_p})$$

- For $T = 100$, attacks had 96.4% success rate while only changing on average 36.4 pixels; works on all $T$ from 1 to 100
- 86% success with average of 45 pixels changed on non-distilled networks

# Conclusion

- It is insufficient to create defenses just for current attacks; must take into account how it might be attacked in the future
- While it is impossible to test on all possible attacks, should look for arguments that existing attacks cannot be adapted
- Demonstrating that a defense works on sub-optimal attacks does not imply it will stop other attacks

# References


- https://arxiv.org/pdf/1607.04311.pdf