# Temperature Encoding: One Hot Way to Resist Adversarial Examples

J. Buckman, A. Roy, C. Raffel, I. Goodfellow

Google Brain

http://colinraffel.com/publications/iclr2018thermometer.pdf

Reviewed by : Bill Zhang

University of Virginia

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

# Outline

# Introduction

## Basic Premise and Motivation

- Machine learning models tend to create linear decision boundaries, which is one reason why adversarial examples are so effective
- Adding non-linear function (i.e. quadratic) makes model much harder to train accurately/efficiently
- Previous research has used one-hot representations of pixels effectively
- Inspired by quantization of input as method of introducing non-linearity
- Instead of replacing number with lower-bit representation, replace with binary vector

# Discretization

## Introduction

- Two types of discretization: one-hot and temperature
- Each pixel value in each color channel is mapped to a binary vector
- Discretization can keep all information or discard information while making significant change to model

# Discretization

## Why it Works

- Goodfellow et al. (2014) provides evidence that many network architectures are vulnerable to adversarial inputs because, empirically, the loss function of the networks tend to be highly linear w.r.t. inputs (where $x \in \mathbb{R}^n$); large $n$ means even small $\epsilon$ will affect prediction

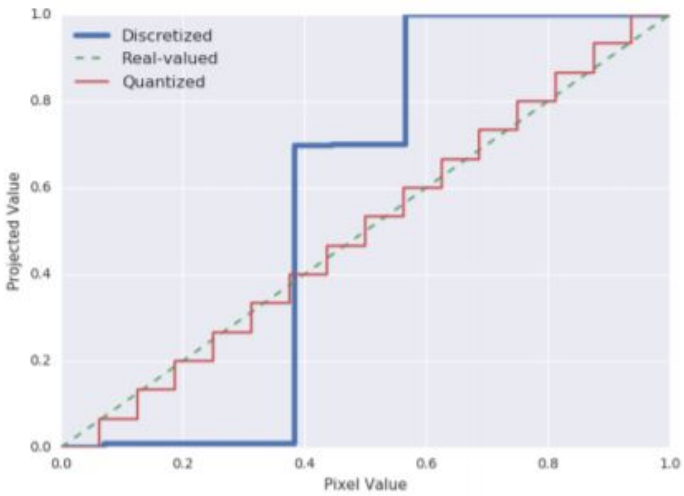
$$\mathbb{L}(\tilde{x}) = \sigma(w^T \tilde{x}) = \sigma(w^T(x + \eta)) = \sigma(x^T x + w^T \eta)$$

- Neural networks have the capacity to represent non-linear function, but those trained via SGD tend to converge to mostly-linear solutions (possibly because non-linearities introduced are either piecewise linear or mostly-linear in region of training)

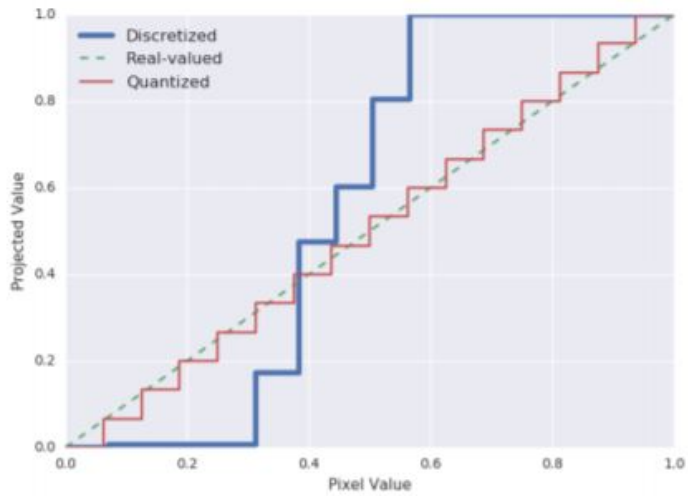
# Discretization

## Why it Works

- Quantization, although non-linear, is approximately linear when $\epsilon$ is larger than bucket size
- Discretizing input and using learned weights to project back onto scalar yields highly non-linear, non-differentiable function




(a) One hot encoding




(b) Thermometer encoding

# Discretization

## Types of Discretization

- First, describe quantization function $b$, where $0 < b_1 < b_2 < ... < b_k = 1$; for this paper, choose $b_i = i/k$. For a real number $\theta \in [0, 1]$, define $b(\theta)$ to be largest index $\alpha \in \{1, 2, ..., k\}$ s.t. $\theta \leq b_\alpha$
- One-hot encodings: simple to compute, does not preserve ordering or give information on how close two inputs are to each other

$$\chi(j)_l = \{1 \text{ if } l = j, 0 \text{ otherwise}\}$$

$$f_{onehot}(x_i) = \chi(b(x_i))$$

- Thermometer encodings: preserve ordering

$$\tau(j)_l = \{1 \text{ if } l \geq j, 0 \text{ otherwise}\}$$

$$f_{therm}(x_i) = \tau(b(x_i))$$

# Discretization

## Possible White-box Attacks

- Discretizing input makes traditional white-box attacks difficult to execute since backpropagation is impossible
- Discrete Gradient Ascent: first, initialize by placing each pixel within $\epsilon$ of true value; at each step, select bucket likely to do most "harm"

$$z_i^0 = f_{therm}(x_i + U(-\varepsilon, \varepsilon))$$

$$\text{harm}(z_i^t)_l = \begin{cases} (z_i^t - \tau(l))^\top \cdot \frac{\partial \mathbb{L}(z^t)}{\partial z_i^t} & \text{if } \exists(-\varepsilon \leq \eta \leq \varepsilon) \quad \text{s.t.} \quad b(x_i + \eta) = l \\ 0 & \text{otherwise.} \end{cases}$$

$$z_i^{t+1} = \tau\left(\arg\max\left(\text{harm}\left(z_i^t\right)\right)\right)$$

# Discretization

## Possible White-box Attacks

- Logit-Space Projected Gradient Ascent

$$z_i^t = \mathbb{C}\left(\sigma\left(\frac{u_i^t}{T^t}\right)\right)$$
$$z_i^{final} = \tau\left(\arg\max\left(u_i^{final}\right)\right)$$
$$T^t = T^{t-1} \cdot \delta$$

$$u_i^0 = \begin{cases} \mathcal{N}(\mathbf{0};\mathbf{1}) & \text{if } \exists(-\varepsilon \le \eta \le \varepsilon) \quad \text{s.t.} \quad b(x_i+\eta) = l \\ -\infty & \text{otherwise.} \end{cases}$$

$$\left(u_i^{t+1}\right)_l = \begin{cases} (u_i^t)_l + \xi \cdot \operatorname{sign}\left(\frac{\partial \mathbb{L}(z^t)}{\partial u_i^t}\right)_l & \text{if } \exists(-\varepsilon \le \eta \le \varepsilon) \quad \text{s.t.} \quad b(x_i+\eta) = l \\ -\infty & \text{otherwise.} \end{cases}$$

- Soften discrete encodings into continuous relaxations, represent dist. of embeddings as softmax of logits $u$, scaled by temperature $T$, anneal $T$ with $\alpha$ each step
- Init. logits from normal dist. if bucket within $\epsilon$ of original value, since continouous just perform standard PGA

# Experiments

## Procedure

- Compare models trained on discretized inputs with state-of-the-art adversarial defenses
- For MNIST, use CNN; for CIFAR-10, CIFAR-100, SVHN use Wide ResNet
- All quantized/discretized models use 16 levels
- Found that LS-PGA strictly better than DGA, so only used LS-PGA in results

# Results

- Comparison of adversarial robustness on MNIST (white and black-box) and CIFAR-10 (white and black-box)

| | Model | Clean | FGSM | PGD/LS-PGA |
|---|---|---|---|---|
| *Clean* | Vanilla (Madry) | 99.20 | 6.40 | - |
| | Vanilla | **99.30** | 0.19 | 0 |
| | Quantized | 99.19 | 1.10 | 0 |
| | One-hot | 99.13 | 0 | 0 |
| | Thermometer | 99.20 | 0 | 0 |
| *Adv. train* | Vanilla (Madry) | 98.80 | 95.60 | 93.20 |
| | Vanilla | 98.67 | 96.17 | 93.30 |
| | Quantized | 98.75 | **96.29** | 94.23 |
| | One-hot | 98.61 | 96.22 | **94.30** |
| | Thermometer | 99.03 | 95.84 | 94.02 |

Table 2: Comparison of adversarial robustness to *white-box attacks* on MNIST .

| Target \ Source | | *Clean* | | | *Adv. train* | | |
|---|---|---|---|---|---|---|---|
| | | **Vanilla** | **One-hot** | **Thermometer** | **Vanilla** | **One-hot** | **Thermometer** |
| *Clean* | Vanilla | 2.04 | 36.02 | 24.58 | 3.48 | 80.44 | 57.69 |
| | Quantized | 39.22 | 32.39 | 25.63 | 75.02 | 75.92 | 52.32 |
| | One-hot | 14.57 | 6.91 | 8.11 | 39.02 | 39.60 | 18.02 |
| | Thermometer | 41.12 | 14.30 | 10.98 | 61.84 | 59.16 | 32.93 |
| *Adv. train* | Vanilla (Madry) | - | - | - | 96.0 | - | - |
| | Quantized | 97.65 | 98.16 | 97.14 | 95.27 | 95.31 | 96.53 |
| | Vanilla | 97.62 | 98.05 | 97.06 | 95.43 | 95.38 | 96.23 |
| | One-hot | 97.78 | 98.48 | 97.87 | 96.87 | 96.60 | 96.87 |
| | Thermometer | **98.07** | **98.75** | **98.02** | **97.05** | **96.88** | **97.13** |

Table 3: Comparison of adversarial robustness to *black-box attacks* on MNIST .

# Results

| | Model | Clean | FGSM | PGD/LS-PGA |
|---|---|---|---|---|
| *Clean* | Vanilla (Madry) | **95.20** | 25.10 | 4.10 |
| | Vanilla | 94.29 | 46.15 | 1.66 |
| | Quantized | 93.49 | 43.89 | 3.57 |
| | One-hot | 93.26 | 52.07 | 53.11 |
| | Thermometer | 94.22 | 48.50 | 50.50 |
| *Adv. train* | Vanilla (Madry) | 87.3 | 60.3 | 50.0 |
| | Vanilla | 87.67 | 59.7 | 41.78 |
| | Quantized | 85.75 | 53.53 | 42.09 |
| | One-hot | 88.67 | 68.76 | 67.83 |
| | Thermometer | 89.88 | **80.96** | **79.16** |

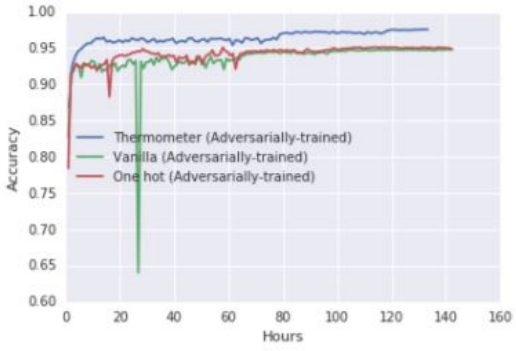
Table 4: Comparison of adversarial robustness to *white-box attacks* on CIFAR-10 .

| | Source / Target | *Clean* | | | *Adv. train* | | |
|---|---|---|---|---|---|---|---|
| | | **Vanilla** | **One-hot** | **Thermometer** | **Vanilla** | **One-hot** | **Thermometer** |
| *Clean* | Vanilla (Madry) | 0.0 | - | - | 79.7 | - | - |
| | Vanilla | 3.38 | 60.10 | 52.60 | 45.48 | 37.21 | 49.91 |
| | Quantized | 70.54 | 62.46 | 55.38 | 51.74 | 45.37 | 55.64 |
| | One-hot | 83.00 | 56.25 | 63.94 | 54.59 | 49.21 | 57.28 |
| | Thermometer | 80.33 | 66.22 | 53.45 | 57.04 | 51.03 | 60.90 |
| *Adv. train* | Vanilla (Madry) | 85.60 | - | - | 67.0 | - | - |
| | Vanilla | 85.60 | 74.99 | 73.78 | 67.0 | 50.09 | 71.03 |
| | Quantized | 84.56 | **82.43** | **82.22** | **72.52** | **72.29** | **79.43** |
| | One-hot | 86.01 | 77.19 | 77.70 | 61.92 | 60.02 | 72.89 |
| | Thermometer | **88.25** | 81.59 | 80.80 | 67.96 | 67.43 | 77.68 |

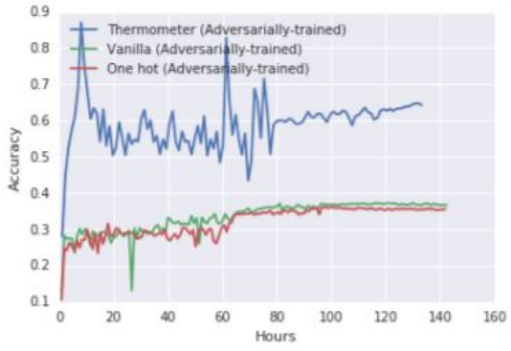
Table 5: Comparison of adversarial robustness to *black-box attacks* on CIFAR-10 .

# Results

- Plot convergence of accuracy over time of adversarially trained models and loss over steps for each attack type

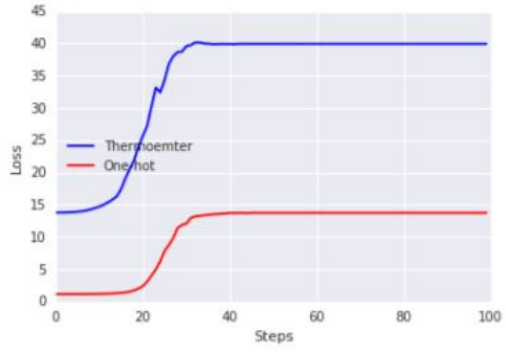



(a) Clean examples

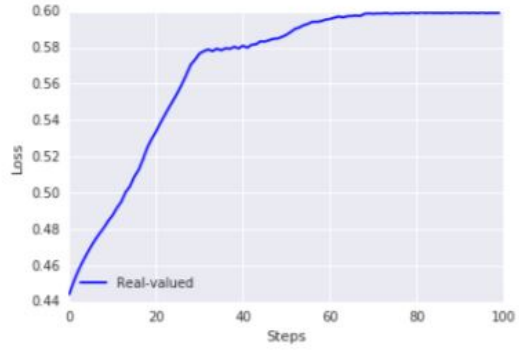



(b) Adversarial examples (iterative white-box)

Figure 2: Comparison of the convergence rate of various *adversarially trained* models on the SVHN dataset.




(a) Loss over steps of LS-PGA




(b) Loss over steps of PGD

Figure 3: Loss for iterated white-box attacks on various models on a randomly chosen data point from MNIST. By step 40, which is where we evaluate, the loss of the point found by iterative attacks has converged.

# Discussion and Conclusion

- Discretizing introduces extra parameters, but the added amount is negligible so not the reason why accuracy is better
- Thermometer encodings, in combination with adversarial training, can resulting in less vulnerable networks which are also significantly less linear w.r.t. inputs
- Supports Goodfellow's hypothesis that linearity of models is the cause of many vulnerabilities to adversarial examples

# References


- http://colinraffel.com/publications/iclr2018thermometer.pdf