Wasserstein adversarial examples via projected Sinkhorn iterations

Eric Wong1  Frank R. Schmidt2  J. Zico Kolter13

Code: https://github.com/locuslab/projected_sinkhorn/

1School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, 15213, USA

2Bosch Center for Artificial Intelligence
Pittsburgh, PA 15222, USA

3Bosch Center for Artificial Intelligence
Renningen, Germany
Typical threat model: norm-bounded perturbation

\[ \ell_\infty + \text{image} = \text{image} \]

\[ \ell_2 + \text{image} = \text{image} \]

\[ \ell_0 + \text{image} = \text{image} \]
$\ell_p$ norms don't capture typical image transforms
These transforms move pixel mass short distances…
These transforms move pixel mass short distances…

and the Wasserstein metric measures “moving mass”
We propose Wasserstein balls as a threat model
The strongest known method for generating adversarial examples is projected gradient descent.
How to project onto the Wasserstein ball?
How to project onto the Wasserstein ball?

\[
\begin{align*}
\text{minimize} \quad & \|w - z\|_2^2 \\
\text{subject to} \quad & \Pi 1 = x \\
& \Pi^T 1 = z \\
& \langle \Pi, C \rangle \leq \epsilon
\end{align*}
\]
How to project onto the Wasserstein ball?

\[
\begin{align*}
\text{minimize} & \quad \|w - z\|_2^2 \\
\text{subject to} & \quad \Pi 1 = x \\
& \quad \Pi^T 1 = z \\
& \quad \langle \Pi, C \rangle \leq \epsilon
\end{align*}
\]

Quadratic program, quadratic number of variables → Costly!
Projected Sinkhorn Iteration: a fast (approximate) projection algorithm onto the Wasserstein ball

\[
\text{minimize } \sum_{\substack{z \in \mathbb{R}^n, \Pi \in \mathbb{R}^{n \times n}}} ||w - z||_2^2 + \frac{1}{\lambda} \sum_{ij} \Pi_{ij} \log(\Pi_{ij})
\]

subject to \n
\begin{align*}
\Pi 1 &= x \\
\Pi^T 1 &= z \\
\langle \Pi, C \rangle &\leq \epsilon
\end{align*}

- Entropy regularization term
- Local transport plans
- Block coordinate descent on the dual problem
CIFAR10 Wasserstein adversarial examples
We can also adversarially train robust networks

Adversarial training

PGD adversary

Projection algorithm onto Wasserstein balls
MNIST and CIFAR10 robustness curves

**MNIST**

- Standard
- $l_\infty$ robust
- Adv. training
- Binarize

**CIFAR10**

- Standard
- $l_\infty$ robust
- Adv. training

Adversarial accuracy vs $\epsilon$ radius for Wasserstein ball.
Wasserstein adversarial examples via projected Sinkhorn iterations

Poster #67 in the Pacific Ballroom

Code: https://github.com/locuslab/projected_sinkhorn