Foundations of Deep Learning

Lecture 11

Adversarial Examples

Aurelien Lucchi

Fall 2024

Section 1

Adversarial Examples

Adversarial examples [SZS+13]

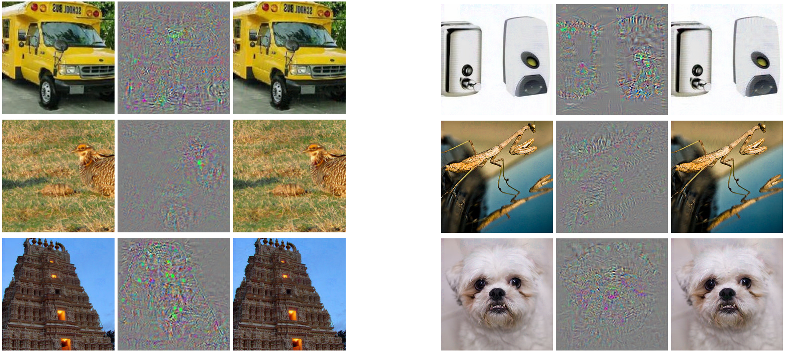
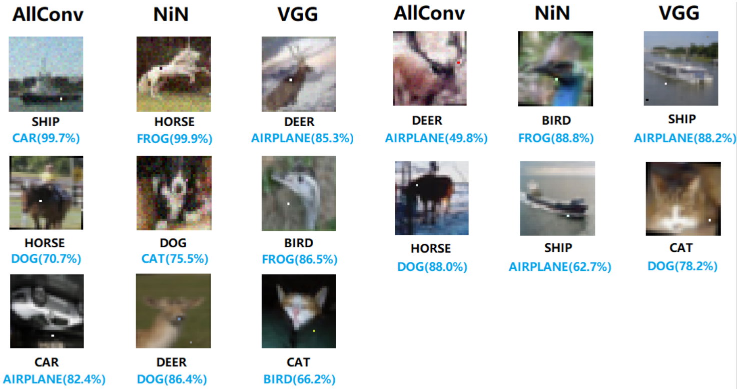


Figure: Adversarial examples generated for AlexNet.(Left) is a correctly predicted sample, (center) difference between correct image, and image predicted incorrectly magnified by 10x (values shifted by 128 and clamped), (right) adversarial example. All images in the right column are predicted to be an ostrich.

Adversarial examples: one-pixel attack [SVS19]



Classification setting

Data Given a training set $\mathcal{S} = (\mathbf{x}_i, y_i)_{i=1}^n$ consisting of pairs of feature vectors $\mathbf{x}_i \in \mathbb{R}^d$ and corresponding labels $y_i \in \{1, \dots k\}$ where k is the number of classes.

Model Define model $f_{\mathbf{w}}(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^k$ that is parametrized by $\mathbf{w} \in \mathbb{R}^d$.

Loss The parameter vector $\mathbf{w}$ is trained by optimizing a loss function $\ell : \mathbb{R}^k \times \mathbb{Z}_+ \to \mathbb{R}_+$, e.g. the cross-entropy:

$$\ell(f_{\mathbf{w}}(\mathbf{x}), y) = \log\left(\sum_{j=1}^{k} \exp(f_{\mathbf{w}}(\mathbf{x}))_j\right) - f_{\mathbf{w}}(\mathbf{x})_y,$$

where $f_{\mathbf{w}}(\mathbf{x})_j$ denotes the j-th entry of the vector $f_{\mathbf{w}}(\mathbf{x})$.

Definition of an adversarial example

Informal An adversarial example is an input to a machine learning model that was specifically designed to **cause the model to predict the wrong class.**

Formally Perturbation $\bar{\delta}$ such that $\bar{\mathbf{x}} = \mathbf{x} + \bar{\delta}$, where

$$\bar{\delta} = \underset{\delta \in \Delta}{\operatorname{argmax}}\, \ell(f_{\mathbf{w}}(\mathbf{x} + \delta), y).$$

We need a notion of distance for $\Delta \rightsquigarrow$ common choice is to use the L_∞ norm $\|\delta\|_\infty = \sup_i |\delta_i|$, i.e.

$$\Delta = \{\delta : \|\delta\|_\infty \leq \epsilon\},$$

for a small $\epsilon > 0$.

Finding adversarial examples

SGD

$$\delta_{k+1} = \delta_k + \eta \nabla_\delta \ell(f_{\mathbf{w}}(\mathbf{x} + \delta), y),$$

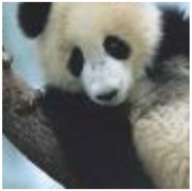
where $\eta > 0$ is a step-size parameter and $\delta_0 \in \Delta$.
After each iteration, project δ s.t. $\|\delta\|_\infty \leq \epsilon$.

Fast Gradient Sign Method Projecting onto this norm ball involves clipping values of δ to lie within the range $[-\epsilon, \epsilon]$.
A surrogate update involves taking the sign of the gradient

- If η large enough, the relative size of the entries does not matter
- Thus we can simply consider the sign of the entries.

$$\delta_{k+1} = \delta_k + \eta \, \mathrm{sign}(\nabla_\delta \ell(f_{\mathbf{w}}(\mathbf{x} + \delta)), y).$$

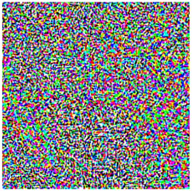
FGSM: adversarial example



$\mathbf{x}$

“panda”
57.7% confidence

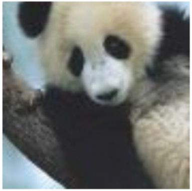
$+ .007 \times$



$\mathrm{sign}(\nabla_\delta \ell(f_{\mathbf{w}}(\mathbf{x} + \delta)), y)$

“nematode”
8.2% confidence

$=$



$\mathbf{x} + \eta\,\mathrm{sign}(\nabla_\delta \ell(f_{\mathbf{w}}(\mathbf{x} + \delta)), y)$

“gibbon”
99.3 % confidence

Section 2

Existence of Robust Networks

Setting: Two-layer network

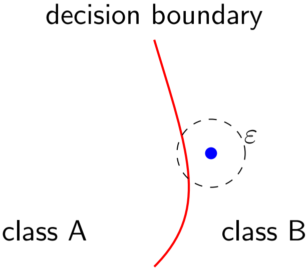
Network Consider a two-layer neural network of width k defined by

$$f(\mathbf{x}) = \sum_{j=1}^{k} a_j \sigma(\mathbf{w}_j^\top \mathbf{x} + b_j),$$

where $\mathbf{x} \in \mathbb{R}^d$ is the input, $\mathbf{w}_j \in \mathbb{R}^d$ and b_j are the weight vectors and offset of the first layer, while $a_j \in \mathbb{R}$ are the weights of the second layer.

Assumptions Assume $\mathbf{x}_i \in \sqrt{d}\mathbb{S}^{d-1}$
$\rightsquigarrow$ distance between every two inputs is at most $\mathcal{O}(\sqrt{d})$
$\rightsquigarrow$ perturbation of size $\mathcal{O}(\sqrt{d})$ is sufficient to flip the sign of the input (assuming $\exists i, j$ s.t. $f(\mathbf{x}_i) > 0$ and $f(\mathbf{x}_j) < 0$).

Illustration: robustness



Goal: want the size of the ϵ-ball to be large, i.e. need a large perturbation to change the class.

Existence of robust networks

The following theorem explicitly constructs an example of a neural network that is $\sqrt{d}$-robust.

Theorem 1 ([VYS22])

Let $\{(\mathbf{x}_i, y_i)\}_{i=1}^{m} \subseteq (\sqrt{d} \cdot \mathbb{S}^{d-1}) \times \{-1, 1\}$ *be a dataset. Let* $0 < c < 1$ *be a constant such that* $|\langle \mathbf{x}_i, \mathbf{x}_j \rangle| \leq c \cdot d$ *for every* $i \neq j$.

Then, there exists a depth-2 *ReLU network* f *of width* m *such that* $y_i f(\mathbf{x}_i) \geq 1$ *for every* $i \in [m]$*, and for every* $\mathbf{x}_i$ *flipping the sign of the output requires a perturbation of size larger than* $\frac{1-c}{4} \cdot \sqrt{d}$*. Thus,* f *is* $\sqrt{d}$*-robust w.r.t.* $\mathbf{x}_1, \ldots, \mathbf{x}_m$*.*

Proof

Section 3

Adversarial Examples at Initialization

Setting

Result A single step of gradient descent at initialization suffices to find an adversarial example.

Model

$$f(\mathbf{x}) = \frac{1}{\sqrt{k}} \sum_{j=1}^{k} a_j \sigma(\mathbf{w}_j^\top \mathbf{x}).$$

where $\mathbf{w}_j \overset{i.i.d}{\sim} \mathcal{N}\left(0, \frac{1}{d}\mathbf{I}_d\right)$ and $a_j \overset{i.i.d}{\sim}$ Unif$(\{-1, +1\})$.

CLT For $\mathbf{x} \in \sqrt{d} \cdot \mathbb{S}^{d-1}$ (so that $\mathbf{w}_j^\top \mathbf{x} \sim \mathcal{N}(0,1)$) and large width k, the distribution of $f(\mathbf{x})$ is a centered Gaussian with variance equal to

$$\mathbb{E}_{X\sim\mathcal{N}(0,1)}[\sigma(X)^2] = \mathcal{O}(1).$$

Preliminaries

Theorem 2 (Bernstein's inequality)

Let (X_j) be i.i.d. centered random variables such that there exists $\sigma, c > 0$ such that for all integers $q \geq 2$,

$$\mathbb{E}[|X_j|^q] \leq \frac{q!}{2}\sigma^2 c^{q-2}\,.$$

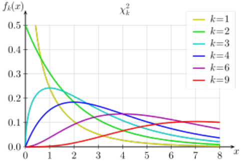
Then with probability at least $1-\gamma$ one has:

$$\sum_{j=1}^{k} X_j \leq \sqrt{2\sigma^2 k \log(1/\gamma)} + c\log(1/\gamma)\,.$$

Preliminaries

Sum of square normal random variables
If $X_1, X_2, \dots, X_k$ are independent standard normal random variables (i.e., $X_i \sim \mathcal{N}(0,1)$), then $Z = \sum_{i=1}^{k} X_i^2$ follows a chi-square distribution with k degrees of freedom. This is denoted as $Z \sim \chi_k^2$.

Figure: χ^2 distribution



Concentration of χ^2 random variables Let $X_1, \dots, X_k$ be i.i.d. standard Gaussians, then with probability at least $1-\gamma$, one has:

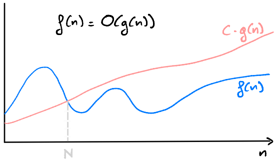
$$\left| \sum_{j=1}^{k} X_j^2 - k \right| \le 4\sqrt{k \log(2/\gamma)}\,.$$

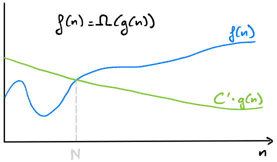
Preliminaries

We will also use various mathematical notations, such as $\mathcal{O}, \Omega$ that allow us to relate the growth of various functions.

- $f(n) = \mathcal{O}(g(n))$ if there exists constants N and C such that $f(n) < C|g(n)|$ for all $n > N$.

- $f(n) = \Omega(g(n))$ if there exists constants N and C' such that $f(n) > C'|g(n)|$ for all $n > N$.





Main result

Assumption 1

Let σ be differentiable almost everywhere, and assume that there exists $\sigma', c' > 0$ such that for all integers $q \geq 2$,

$$\mathbb{E}_{X \sim \mathcal{N}(0,1)}[|\sigma'(X)|^{2q}] \leq \frac{q!}{2}\sigma'^2 c'^{q-2}\,.$$

Main result: Theorem

Theorem 3

Let $\gamma \in (0,1)$ *and* σ *be non-constant, Lipschitz and with Lipschtiz derivative. Assume* $k \geq \mathcal{O}(\log^3(1/\gamma))$ *and* $d \geq \mathcal{O}(\log(k/\gamma)\log(1/\gamma))$*. Then, there exists* $\eta > 0$ *such that with probability at least* $1-\gamma$ *one has:*

$$\operatorname{sign}(f(\mathbf{x})) \neq \operatorname{sign}(f(\mathbf{x}+\eta\nabla f(\mathbf{x})))\,.$$

Proof

We will give a proof sketch of the theorem that will rely on the following proposition.

Proposition 4

Assume σ satisfies Assumption 1. Then with high probability,

$$\|\nabla f(\mathbf{x})\| = \Omega(1).$$

Proof

Section 4

Adversarial Training

Risk

True risk Define the true risk as the expected loss under the true distribution of the samples denoted by $\mathcal{D}$, i.e.

$$R(f_{\mathbf{w}}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\ell(f_{\mathbf{w}}(\mathbf{x}), y)].$$

Empirical risk Typically only have access to a finite set of samples $\mathcal{S} = (\mathbf{x}_i, y_i)_{i=1}^n$. We thus consider the empirical risk defined as

$$\hat{R}(f_{\mathbf{w}}) = \frac{1}{|\mathcal{S}|} \sum_{(\mathbf{x},y)\in\mathcal{S}} \ell(f_{\mathbf{w}}(\mathbf{x}), y).$$

Adversarial risk

We modify the loss to consider the whole $\Delta(\mathbf{x})$ region around each sample as follows:

$$R_{\mathsf{adv}}(f_{\mathbf{w}}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\max_{\delta\in\Delta(\mathbf{x})} \ell(f_{\mathbf{w}}(\mathbf{x}+\delta), y)\right].$$

The empirical counterpart can simply be defined as

$$\hat{R}_{\mathsf{adv}}(f_{\mathbf{w}}) = \frac{1}{|\mathcal{S}|}\sum_{(\mathbf{x},y)\sim\mathcal{S}} \max_{\delta\in\Delta(\mathbf{x})} \ell(f_{\mathbf{w}}(\mathbf{x}+\delta), y).$$

Stochastic Gradient Descent (SGD)

SGD update

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \frac{\eta}{|\mathcal{B}|} \sum_{(\mathbf{x},y)\sim\mathcal{B}} \nabla_{\mathbf{w}} \max_{\delta\in\Delta(\mathbf{x})} \ell(f_{\mathbf{w}}(\mathbf{x}+\delta), y)$$

where $\eta > 0$ and $\mathcal{B} \subseteq \mathcal{S}$ is a mini-batch is a chosen step-size.

Gradient computation By Danskin's theorem:

$$\nabla_{\mathbf{w}} \max_{\delta\in\Delta(\mathbf{x})} \ell(f_{\mathbf{w}}(\mathbf{x}+\delta), y) = \nabla_{\mathbf{w}} \ell(f_{\mathbf{w}}(\mathbf{x}+\delta^*), y).$$

Active area of research

- Adversarial attacks and defenses are active area of research
- No silver bullet, most models are vulnerable to attacks
- Many technical difficulties to implement them (e.g. typically don't have access to the weights of the trained model)

Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai.
One pixel attack for fooling deep neural networks.
IEEE Transactions on Evolutionary Computation, 23(5):828–841, 2019.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus.
Intriguing properties of neural networks.
arXiv preprint arXiv:1312.6199, 2013.

Gal Vardi, Gilad Yehudai, and Ohad Shamir.
Gradient methods provably converge to non-robust networks.
arXiv preprint arXiv:2202.04347, 2022.