Labs
Machine Learning Course
Fall 2025

EPFL
School of Computer and Communication Sciences
Bob West
epfml.github.io/cs433-2025/

Problem Set 9, Nov 13, 2025 (Adversarial Robustness)

Security and robustness of machine learning models remain critical concerns in modern AI systems. Despite significant advances in model performance, neural networks are still vulnerable to adversarial attacks—carefully crafted perturbations that can cause models to misclassify inputs (e.g., images) while remaining imperceptible to humans. Understanding and defending against such attacks is essential both for deploying ML systems in real-world applications and for scientific understanding.

Goals. The goals of this exercise are to:

- Understand how to generate adversarial examples in practice.
- Use adversarial examples in adversarial training to obtain more robust models.
- Understand what adversarial examples correspond to in the simple case of linear models.

Problem 1 (Adversarial training for linear models):

It can be often very insightful to analyze what a method corresponds to in a simple setting of linear models.

Assume we have input points $\boldsymbol{x}_i \in \mathbb{R}^d$ and binary labels $y_i \in \{-1, 1\}$. Let ℓ be a monotonically decreasing margin-based loss function, for example the hinge loss $\ell(z) = \max\{0, 1 - z\}$ or logistic loss $\ell(z) = \log(1 + \exp(-z))$ that you have seen before.

Consider the adversarial training objective for a linear model $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}$ with respect to ℓ_2 adversarial perturbations:

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \max_{\|\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i\|_2 \leq \varepsilon} \ell\left(y_i \cdot \boldsymbol{w}^\top \hat{\boldsymbol{x}}_i\right).$$

- Find a closed-form solution of the inner maximization problem $\max_{\|\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i\|_2 \leq \varepsilon} \ell\left(y_i \cdot \boldsymbol{w}^\top \hat{\boldsymbol{x}}_i\right)$ and the minimizer $\hat{\boldsymbol{x}}_i^\star$.

  Solution:

$$\begin{aligned}\max_{\|\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i\|_2 \leq \varepsilon} \ell\left(y_i \cdot \boldsymbol{w}^\top \hat{\boldsymbol{x}}_i\right) &= \max_{\|\boldsymbol{\delta}\|_2 \leq \varepsilon} \ell\left(y_i \cdot \boldsymbol{w}^\top (\boldsymbol{x} + \boldsymbol{\delta})\right) = \ell\left(\min_{\|\boldsymbol{\delta}\|_2 \leq \varepsilon} y_i \cdot \boldsymbol{w}^\top (\boldsymbol{x} + \boldsymbol{\delta})\right) \\ &= \ell\left(y_i \cdot \boldsymbol{w}^\top \boldsymbol{x} + \min_{\|\boldsymbol{\delta}\|_2 \leq \varepsilon} y_i \boldsymbol{w}^\top \boldsymbol{\delta}\right) = \ell\left(y_i \cdot \boldsymbol{w}^\top \boldsymbol{x} - \varepsilon \|\boldsymbol{w}\|_2\right)\end{aligned}$$

  where we used in the second equality the fact that the loss is monotonically decreasing in its margin, and thus flips the max to the min. Moreover, in the last equality we used the Cauchy-Schwartz inequality to solve the inner minimization problem of a linear function over the ℓ_2-ball:

$$y_i \boldsymbol{w}^\top \boldsymbol{\delta} \geq -\|y_i \boldsymbol{w}\|_2 \|\boldsymbol{\delta}\|_2 = -\|\boldsymbol{w}\|_2 \|\boldsymbol{\delta}\|_2 \geq -\|\boldsymbol{w}\|_2 \varepsilon.$$

  And then the equalities are attained for $\boldsymbol{\delta}^\star = -y_i \varepsilon \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}$, i.e. $\delta^\star$ is colinear with $\boldsymbol{w}$ and scaled by an appropriate constant. Thus, $\delta^\star$ is a minimizer of $\min_{\|\boldsymbol{\delta}\|_2 \leq \varepsilon} y_i \boldsymbol{w}^\top \boldsymbol{\delta}$. And equivalently we have that $\hat{\boldsymbol{x}}_i^\star = \boldsymbol{x}_i - y_i \varepsilon \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}$.

  This makes the overall adversarial training have the following form:

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell\left(y_i \cdot \boldsymbol{w}^\top \boldsymbol{x} - \varepsilon \|\boldsymbol{w}\|_2\right).$$

Note that adversarial training for linear models boils down to a convex optimization problem with respect to the weights $\boldsymbol{w}$, so it can be efficiently solved. This is unlike for neural networks where both inner maximization and outer minimization problems are computationally hard (but can be solved approximately, often with a lot of empirical success).

- In case of the hinge loss, $\ell(z) = \max\{0, 1 - z\}$, what is the connection between ℓ_2 adversarial training and the primal formulation of the soft-margin SVM?

  Solution: For the hinge loss, we have the following adversarial training objective:

  $$\min_{\boldsymbol{w} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \max\left\{0, 1 - y_i \cdot \boldsymbol{w}^\top \boldsymbol{x} + \varepsilon \left\|\boldsymbol{w}\right\|_2\right\}.$$

  And for the soft-margin SVM we have a quite similar objective:

  $$\min_{\boldsymbol{w} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \max\left\{0, 1 - y_i \cdot \boldsymbol{w}^\top \boldsymbol{x}\right\} + \varepsilon \left\|\boldsymbol{w}\right\|_2^2,$$

  with the only difference that the ℓ_2-regularization term is contained inside of the loss function and it is not squared (i.e. $\|\boldsymbol{w}\|_2$ instead of $\|\boldsymbol{w}\|_2^2$). Moreover, note that we have the following upper bound:

  $$\frac{1}{n} \sum_{i=1}^{n} \max\left\{0, 1 - y_i \cdot \boldsymbol{w}^\top \boldsymbol{x} + \varepsilon \left\|\boldsymbol{w}\right\|_2\right\} \leq \frac{1}{n} \sum_{i=1}^{n} \max\left\{0, 1 - y_i \cdot \boldsymbol{w}^\top \boldsymbol{x}\right\} + \varepsilon \left\|\boldsymbol{w}\right\|_2$$

  Thus, we can see that performing ℓ_2-regularized standard training leads to a similar effect as the ℓ_2 adversarial training. In particular, a small norm of the weights $\boldsymbol{w}$ is sufficient to have small adversarial loss.

- What if instead of ℓ_2 adversarial training, we performed ℓ_∞ adversarial training, how would the solution of the inner maximization problem change? Does the maximizer for ℓ_∞-perturbations resemble the Fast Gradient Sign Method (FGSM)?

  Solution: The only difference would be only in the step where we used the Cauchy-Schwartz inequality. Instead, we can use its generalization known as the *Hölder's inequality* which states that for ℓ_p-norms with $p \geq 1$ such that $\frac{1}{p} + \frac{1}{q} = 1$ we have $|\boldsymbol{w}^\top \boldsymbol{\delta}| \leq \left\|\boldsymbol{w}\right\|_p \left\|\boldsymbol{\delta}\right\|_q$ and the equality can be always attained for some choice of $\boldsymbol{\delta}$. Using this fact for $p = 1$ and $q = \infty$:

  $$y_i \boldsymbol{w}^\top \boldsymbol{\delta} \geq -\left\|y_i \boldsymbol{w}\right\|_1 \left\|\boldsymbol{\delta}\right\|_\infty = -\left\|\boldsymbol{w}\right\|_1 \left\|\boldsymbol{\delta}\right\|_\infty \geq -\left\|\boldsymbol{w}\right\|_1 \varepsilon \quad \implies \quad \min_{\left\|\boldsymbol{\delta}\right\|_\infty \leq \varepsilon} y_i \boldsymbol{w}^\top \boldsymbol{\delta} = -\varepsilon \left\|\boldsymbol{w}\right\|_1.$$

  Moreover, we can recover the minimizer $\boldsymbol{\delta}^\star$ by solving $y_i \boldsymbol{w}^\top \boldsymbol{\delta} = -\varepsilon \left\|\boldsymbol{w}\right\|_1$. We get that $\sum_{j=1}^{d} y_i w_j \delta_j = \sum_{j=1}^{d} -\varepsilon |w_j|$ for which we can find a solution by equating each term in the sum. This gives us $y_i w_j \delta_j = -\varepsilon |w_j|$ which implies that $\delta_j = -y_i \varepsilon \operatorname{sign}(w_j)$ or in a vector form $\boldsymbol{\delta}^\star = -y_i \varepsilon \operatorname{sign}(\boldsymbol{w})$. This solution is optimal since it satisfies $y_i \boldsymbol{w}^\top \boldsymbol{\delta} = -\varepsilon \left\|\boldsymbol{w}\right\|_1$ and is feasible $\left\|\boldsymbol{\delta}\right\|_\infty$.

  We also note that we can alternatively solve the optimization problem $\min_{\left\|\boldsymbol{\delta}\right\|_\infty \leq \varepsilon} y_i \boldsymbol{w}^\top \boldsymbol{\delta}$ directly, without relying on Hölder's inequality:

  $$\begin{aligned}
  \min_{\left\|\boldsymbol{\delta}\right\|_\infty \leq \varepsilon} y_i \boldsymbol{w}^\top \boldsymbol{\delta} &= \min_{\forall j \ |\delta_j| \leq \varepsilon} \sum_{j=1}^{d} y_i w_j \delta_j \overset{(a)}{=} \sum_{j=1}^{d} \min_{|\delta_j| \leq \varepsilon} y_i w_j \delta_j \overset{(b)}{=} \sum_{j=1}^{d} \min_{|\delta_j| = \varepsilon} y_i w_j \delta_j \\
  &= \sum_{j=1}^{d} \min\{-w_j \varepsilon, w_j \varepsilon\} = \sum_{j=1}^{d} -\varepsilon |w_j| = -\varepsilon \left\|\boldsymbol{w}\right\|_1,
  \end{aligned}$$

  where in (a) we used the separability of the objective and constraints over dimensions and in (b) the fact that a minimum of a linear function is attained at a boundary of the feasible set.

  Additionally, we note that for an arbitrary constrained convex optimization problem (that satisfies some mild technical conditions), we could solve the Karush-Kuhn-Tucker conditions to find the solution. In the case of $\min_{\left\|\boldsymbol{\delta}\right\|_\infty \leq \varepsilon} y_i \boldsymbol{w}^\top \boldsymbol{\delta}$, it would lead to the same closed-form solution as shown above.

  Regarding the interpretation of $\boldsymbol{\delta}^\star$, the choice $\boldsymbol{\delta}^\star = -y_i \varepsilon \operatorname{sign}(\boldsymbol{w})$ is very similar to the Fast Gradient Sign Method (FGSM), where instead of $\boldsymbol{w}$ we have the negative gradient of the loss function $-\nabla_x \ell(\boldsymbol{x}, y)$. This

is motivated by the fact that we want to maximize the loss (i.e. to make the prediction on $\boldsymbol{x}$ as dissimilar to the true label y as possible), and for this we try to maximize the *linear approximation* of the loss:

$$\max_{\|\boldsymbol{\delta}\|_\infty \leq \varepsilon} \ell(\boldsymbol{x} + \boldsymbol{\delta}, y) \approx \max_{\|\boldsymbol{\delta}\|_\infty \leq \varepsilon} \left[\ell(\boldsymbol{x}, y) + \nabla_x \ell(\boldsymbol{x}, y)^\top \boldsymbol{\delta}\right] = \max_{\|\boldsymbol{\delta}\|_\infty \leq \varepsilon} \nabla_x \ell(\boldsymbol{x}, y)^\top \boldsymbol{\delta},$$

where the maximizer is similarly given as $\boldsymbol{\delta}^\star = \varepsilon \cdot \mathsf{sign}(\nabla_x \ell(\boldsymbol{x}, y))$. In case of binary labels, monotonically decreasing ℓ, and a linear classifier, we get that $\boldsymbol{\delta}^\star = \varepsilon\, \mathsf{sign}(y\ell'(y\boldsymbol{w}^\top \boldsymbol{x})\boldsymbol{w}) = \varepsilon\, \mathsf{sign}(-y\boldsymbol{w}) = -y\varepsilon\, \mathsf{sign}(\boldsymbol{w})$, i.e. we recover the expression above.