Machine Learning for Graphs and Sequential Data

Robustness of Machine Learning Models

Lecturer: Prof. Dr. Stephan Günnemann

cs.cit.tum.de/daml

Summer Term 2023

Data Analytics and
Machine Learning

Roadmap

1. Introduction
2. Construction of adversarial examples
3. Improving robustness
4. Certifiable robustness
   – Exact certification
   – **Convex relaxations**
   – Lipschitz-continuity
   – Randomized smoothing

Certification via Convex Relaxation

Recall our **goal**: develop an algorithm that answers the question:

"Is the classifier f_θ around the sample $\mathbf{x}$ adversarial-free (within an ϵ-ball measured by some norm)?"

Exact certification returns **YES** if and only if there is no adversarial example within an ϵ ball around the input sample (**NO** otherwise)

Now we allow the following answers:

- **YES**: We must have that for all $\tilde{\mathbf{x}} \in \mathcal{P}(\mathbf{x})$: $\arg\max F(\tilde{\mathbf{x}}) = \arg\max F(\mathbf{x})$
- **POTENTIALLY NOT / MAYBE:** In this case we have no guarantees.
- [**NO:** There must exist a $\tilde{\mathbf{x}} \in \mathcal{P}(\mathbf{x})$ such that $\arg\max F(\tilde{\mathbf{x}}) \neq \arg\max F(\mathbf{x})$]

$F(\tilde{\mathbf{x}}) = \hat{\mathbf{x}}^{(L)}$

Recall: Exact Certification

- We call $\mathrm{m_t} = F(\mathbf{x})_{c^*} - F(\mathbf{x})_t$ the classification margin of classes c^* and t.
- Worst-case margin (given $\mathcal{P}(\mathbf{x})$):

positive <=> negative

$$m_t^* = \min_{\tilde{\mathbf{x}}} \; F(\tilde{\mathbf{x}})_{c^*} - F(\tilde{\mathbf{x}})_t$$

$$\begin{aligned} \textit{subject to} \quad & \|\tilde{\mathbf{x}} - \mathbf{x}\|_p \le \epsilon \\ & \boldsymbol{y}^{(0)} = \tilde{\mathbf{x}} \\ & \hat{\mathbf{x}}^{(l)} = \boldsymbol{W}_l \boldsymbol{y}^{(l-1)} + \boldsymbol{b}_l \qquad \forall l = 1 \dots L \\ & \boldsymbol{y}^{(l)} = \mathrm{ReLU}\big(\hat{\mathbf{x}}^{(l)}\big) \qquad \forall l = 1 \dots L-1 \end{aligned}$$

- $m_t^* > 0$: the classifier's prediction **cannot** be changed from class c^* to t

- As seen previously, solving for m_t^* is NP-hard.
- The (only) problem was the ReLU constraint

Idea: Relaxed Classification Margin

- Instead of solving the exact optimization problem

$$m_t^* = \min_{\tilde{\mathbf{x}}, \boldsymbol{y}^{(l)}, \hat{\boldsymbol{x}}^{(l)}} F(\tilde{\mathbf{x}})_{c^*} - F(\tilde{\mathbf{x}})_t$$

$$\begin{aligned} \textit{subject to} \quad & \|\tilde{\mathbf{x}} - \mathbf{x}\|_p \leq \epsilon \\ & \boldsymbol{y}^{(0)} = \tilde{\mathbf{x}} \\ & \hat{\mathbf{x}}^{(l)} = \boldsymbol{W}_l \boldsymbol{y}^{(l-1)} + \boldsymbol{b}_l \qquad \forall l = 1 \dots L \\ & \boldsymbol{y}^{(l)} = \mathrm{ReLU}\left(\hat{\mathbf{x}}^{(l)}\right) \qquad \forall l = 1 \dots L-1 \end{aligned}$$

Note: We treat $\boldsymbol{y}^{(l)}, \hat{\boldsymbol{x}}^{(l)}$ here as variables; though, due to the equality constraints their values are completely determined based on $\tilde{\boldsymbol{x}}$

we solve a **relaxed** optimization problem

- E.g. with some constraints relaxed or removed

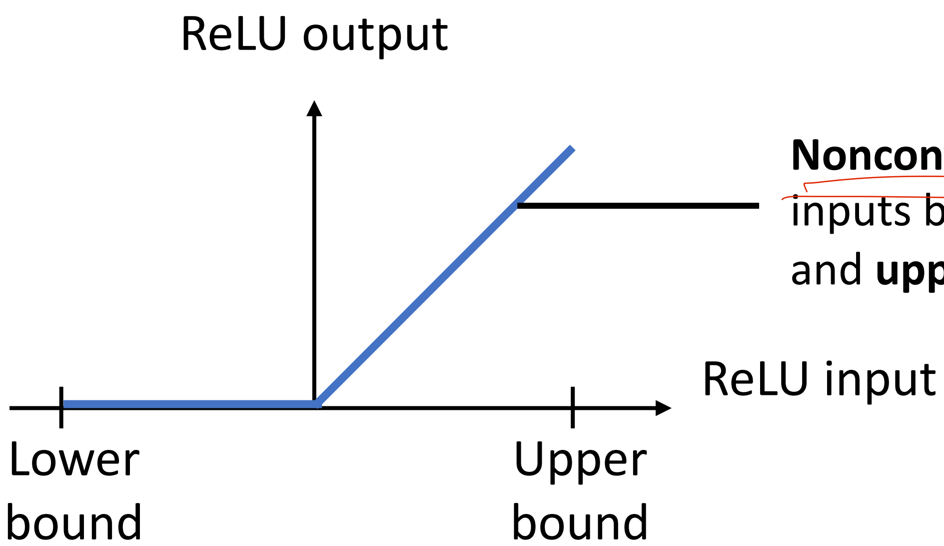
➢ Results in a **lower bound** $\underline{m}_t^*$ on the true minimum m_t^*.

- If $\underline{m}_t^* > 0$: the classifier's prediction **cannot** be changed from class c^* to t
- Can make the optimization possible in **polynomial time**.
- The **price** we pay is that we cannot make a **YES** or **NO** statement in some cases.

ReLU: Nonconvex Output Set

Original constraint

$$\boldsymbol{y}^{(l)} = \mathrm{ReLU}\left(\hat{\mathbf{x}}^{(l)}\right)$$

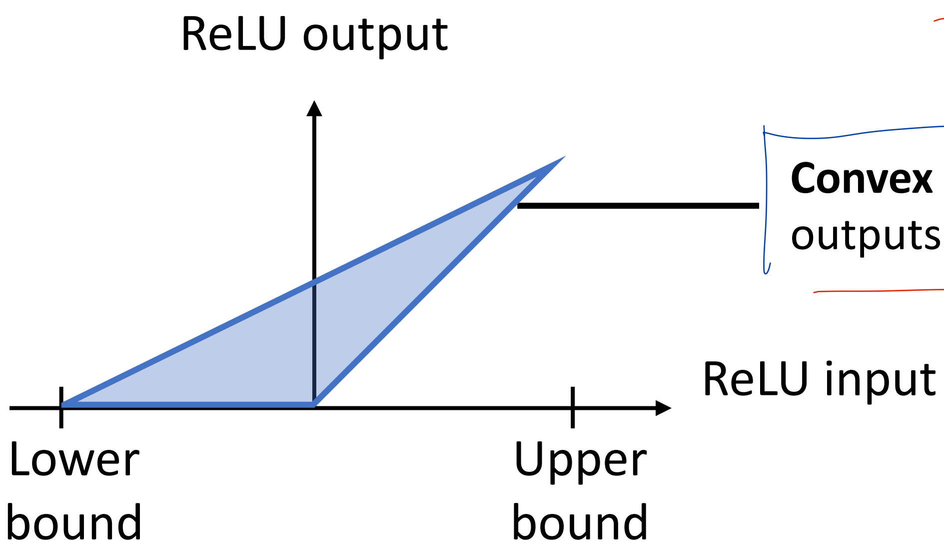


Nonconvex output set for inputs between the **lower** and **upper bound**

Convex ReLU Relaxation: Illustration

Relaxed constraint

$$\boldsymbol{y}^{(l)} \in \mathrm{Triangle}\left(\hat{\mathbf{x}}^{(l)}\right)$$



Convex set of possible outputs.

Note: The output of the ReLU activation is **no longer deterministic** but a **variable** to optimize over (like the input)!

Convex ReLU Relaxation: Formal Definition

We replace the $\boldsymbol{y}^{(l)} = \mathrm{ReLU}\left(\hat{\mathbf{x}}^{(l)}\right)$ constraint by (see [Wong and Kolter, 2018]):

1) For unstable units ($\boldsymbol{l}_i^{(l)} < \mathbf{0} \wedge \boldsymbol{u}_i^{(l)} > \mathbf{0}$):
   - $\boldsymbol{y}_i^{(l)} \geq 0$
   - $\boldsymbol{y}_i^{(l)} \geq \hat{\mathbf{x}}_i^{(l)}$
   - $\left(\boldsymbol{u}_i^{(l)} - \boldsymbol{l}_i^{(l)}\right) \boldsymbol{y}_i^{(l)} - \boldsymbol{u}_i^{(l)} \hat{\mathbf{x}}_i^{(l)} \leq -\boldsymbol{u}_i^{(l)} \boldsymbol{l}_i^{(l)}$

no a_i

Note: This relaxation is equivalent to relaxing the integer constraint $\boldsymbol{a}_i \in \{0, 1\}$ we have seen in the MILP to $\boldsymbol{a}_i \in [0,1]$.

$y \leq \frac{x}{u-l} - \frac{l}{u-l}$ ⇐ line

2) For stably active units ($\boldsymbol{l}_i^{(l)} \geq \mathbf{0}$):
   - $\boldsymbol{y}_i^{(l)} = \hat{\mathbf{x}}_i^{(l)}$
3) For stably inactive units ($\boldsymbol{u}_i^{(l)} \leq \mathbf{0}$):
   - $\boldsymbol{y}_i^{(l)} = \mathbf{0}$

Where $[\boldsymbol{l}^{(l)}, \boldsymbol{u}^{(l)}]$ denote the **element-wise lower** and **upper bounds** on the ReLU input at layer l, which we have encountered in the exact certification session.

These are **linear** constraints only!

Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. ICML 2018.

Overall LP

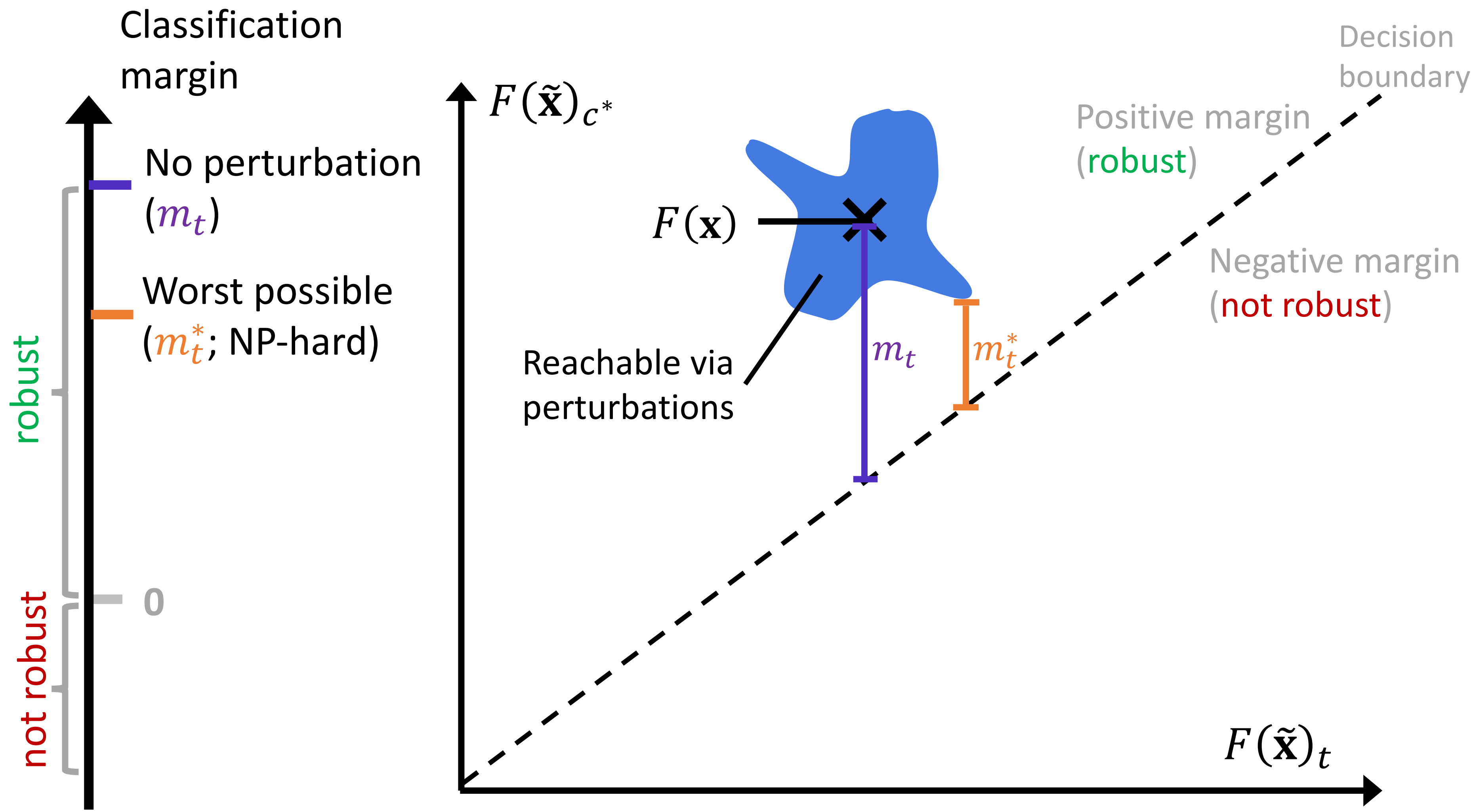
- Since now all constraints are linear, we obtain a linear program (LP):

$$\underline{m}_t^* = \min_{\tilde{\mathbf{x}}, \boldsymbol{y}^{(l)}, \hat{\mathbf{x}}^{(l)}} [\hat{\mathbf{x}}^{(L)}]_{c^*} - [\hat{\mathbf{x}}^{(L)}]_t$$

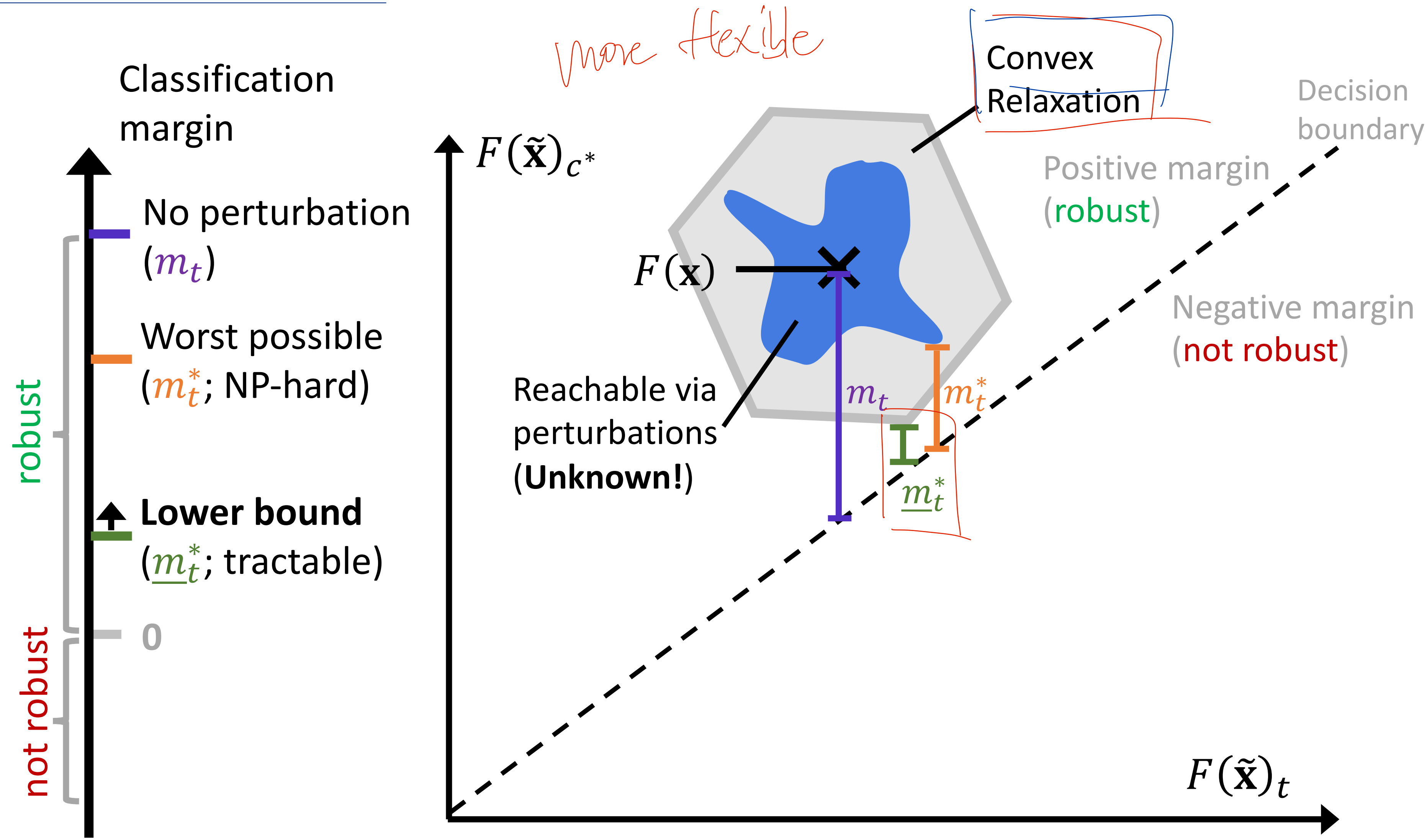
$$\begin{aligned}
\textit{subject to}\quad & \mathbf{x}_i - \tilde{\mathbf{x}}_i \leq \epsilon \quad \forall i \\
& \tilde{\mathbf{x}}_i - \mathbf{x}_i \leq \epsilon \quad \forall i \\
& \boldsymbol{y}^{(0)} = \tilde{\mathbf{x}} \\
& \hat{\mathbf{x}}^{(l)} = \boldsymbol{W}_l \boldsymbol{y}^{(l-1)} + \boldsymbol{b}_l \qquad \forall l = 1 \ldots L \\
& \left.\begin{aligned}
& \boldsymbol{y}_i^{(l)} \geq \hat{\mathbf{x}}_i^{(l)} \\
& \boldsymbol{y}_i^{(l)} \geq 0 \\
& \left(\boldsymbol{u}_i^{(l)} - \boldsymbol{l}_i^{(l)}\right) \boldsymbol{y}_i^{(l)} - \boldsymbol{u}_i^{(l)} \hat{\mathbf{x}}_i^{(l)} \leq -\boldsymbol{u}_i^{(l)} \boldsymbol{l}_i^{(l)}
\end{aligned}\right\} \quad \forall l = 1 \ldots L-1, \forall i : \boldsymbol{l}_i^{(l)} < \mathbf{0} \wedge \boldsymbol{u}_i^{(l)} > \mathbf{0} \\
& \boldsymbol{y}_i^{(l)} = \hat{\mathbf{x}}_i^{(l)} \qquad \forall l = 1 \ldots L-1, \forall i : \boldsymbol{l}_i^{(l)} \geq \mathbf{0} \\
& \boldsymbol{y}_i^{(l)} = \mathbf{0} \qquad \forall l = 1 \ldots L-1, \forall i : \boldsymbol{u}_i^{(l)} \leq \mathbf{0}
\end{aligned}$$

Where $[\boldsymbol{l}^{(l)}, \boldsymbol{u}^{(l)}]$ denote the **element-wise lower** and **upper bounds** on the ReLU input at layer l.

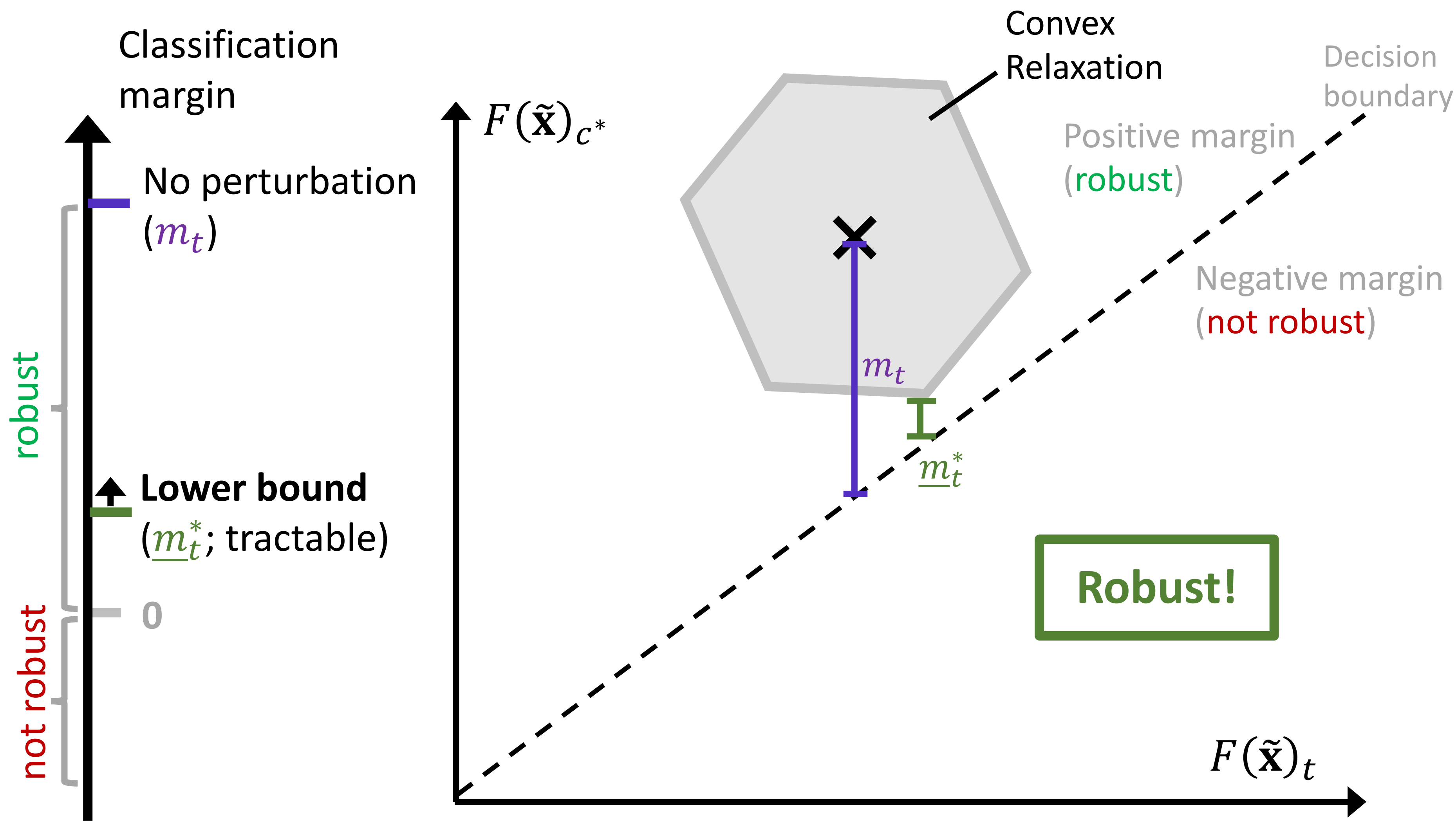
Recap: Exact Robustness Certification Illustration



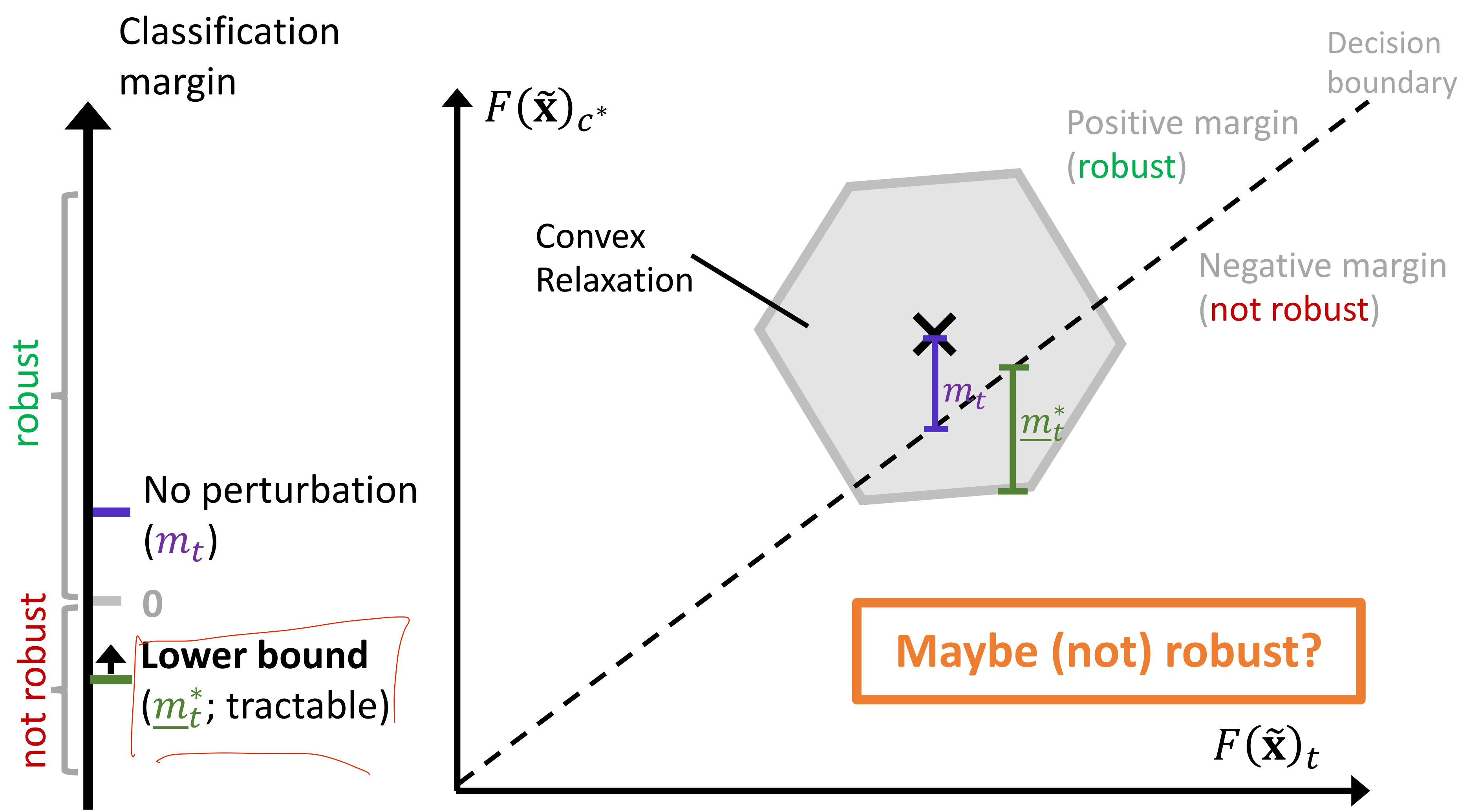
Robustness Certification via Convex Relaxation



Robustness Certification via Convex Relaxation



Robustness Certification via Convex Relaxation



Maybe (not) robust?

通过放宽ReLU激活函数，我们将稳健性认证的问题变成了一个线性程序à可计算的问题

Intermediate Summary

我们付出的代价是，有些实例我们无法决定。

- By relaxing the ReLU activation function we have turned the problem of robustness certification into a **linear program** → tractable to compute
- The price we paid is that there are instances for which we **cannot decide**.

max的优化是一个扰动的实例 **x** $ 。
- 我们可以把它送入原来的神经网络，观察分类是否有变化。
- 如果是：我们有一个对抗性的例子，因此证明了非稳健性à我们的算法可以报告 "NO"，即不是无对抗的。

- The $\arg\min$ of the optimization is a **perturbed instance** $\tilde{\mathbf{x}}$.
  - We can feed it into the original neural network and observe whether the classification changes.
  - If yes: we have an **adversarial example** and therefore proven non-robustness → our algorithm can report "**NO**", i.e. not adversarial-free

与精确认证相比，下限和上限的严格程度影响着松弛的质量，即我们返回MAYBE的频率。

- In contrast to exact certification, the **tightness of the lower and upper bounds** influence the quality of the relaxation, i.e. how often we return **MAYBE**.

more maybe

Certification of GNNs via Convex Relaxation

How does certification via convex relaxation work for GNNs?

Rephrase the original **goal**: develop an algorithm that answers the question:

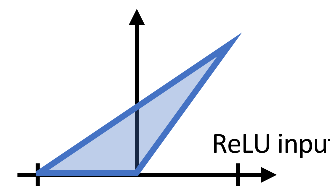
"Is the GNN f_θ around the features $\mathbf{X}$ and adjacency matrix $\mathbf{A}$ adversarial-free (within an ϵ**-**ball(s) measured by some norm)?"

Allowed answers in the relaxed setting:

- **YES**: If for all $\tilde{\mathbf{x}} \in \mathcal{P}_X(\mathbf{x}), \widetilde{\mathbf{A}} \in \mathcal{P}_A(\mathbf{A})$: $\arg\max F\left(\tilde{\mathbf{x}}, \widetilde{\mathbf{A}}\right) = \arg\max F(\mathbf{x}, \mathbf{A})$
- **POTENTIALLY NOT / MAYBE:** In this case we have no guarantees.
- **[NO:** If any $\tilde{\mathbf{x}} \in \mathcal{P}_X(\mathbf{x}), \widetilde{\mathbf{A}} \in \mathcal{P}_A(\mathbf{A})$: $\arg\max F\left(\tilde{\mathbf{x}}, \widetilde{\mathbf{A}}\right) \neq \arg\max F(\mathbf{x}, \mathbf{A})$]

1. Graph and Attributes may change simultaneously
2. The nodes of a graph are non i.i.d.
3. L_0-ball perturbations is natural for discrete data

Exact / Relaxed Certification for GNNs



Already challenging if we are only allowed to perturb $\mathbf{X}$

Proposed approaches so far are focusing on specific architectures and/or only attribute or structure perturbations:

- One can generalize the relaxed certification setting via linear programs to **attribute perturbations** on a **GCN** (Zügner and Günnemann, 2019).
- Certifying a **GCN** against **structure perturbations** can be formulized via a Jointly Constraint Bilinear Program (Zügner and Günnemann, 2020).
- To certify a **PPNP** model w.r.t. **structure perturbations**, we may solve a Quadratically Constrained Linear Program (Bojchevski and Günnemann, 2019).
    - under specific perturbation models ("local budget"; max x perturbations per node) one can perform certification exactly in polynomial time; for a global budget (max x perturbations overall), the problem becomes NP-hard and, thus, requires relaxation for efficiency

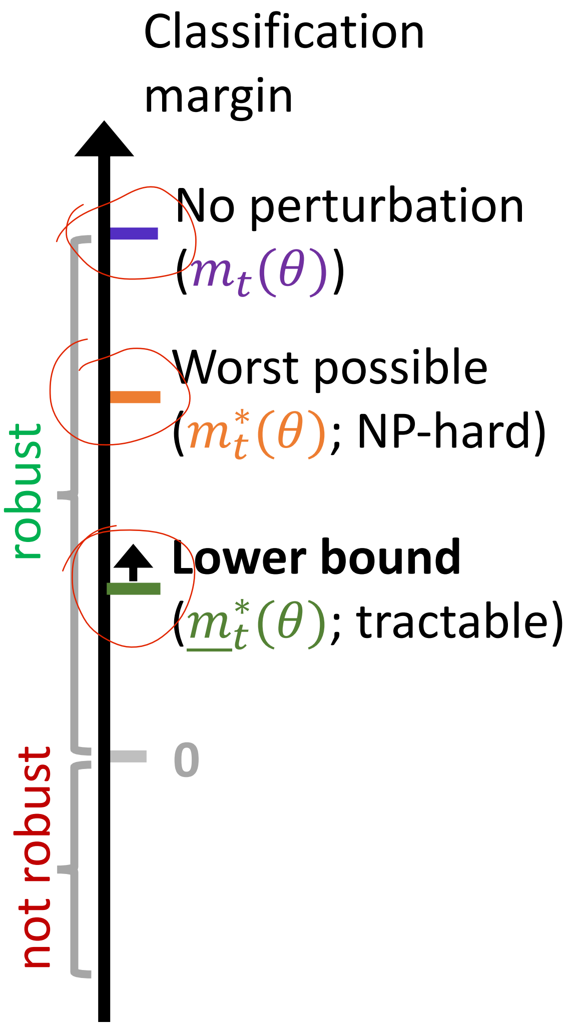
Daniel Zügner and Stephan Günnemann. Certifiable robustness and robust training for graph convolutional networks. SIGKDD 2019.
Daniel Zügner and Stephan Günnemann. Certifiable Robustness of Graph Convolutional Networks under Structure Perturbations. SIGKDD 2020.
Aleksandar Bojchevski and Stephan Günnemann. Certifiable Robustness to Graph Perturbations. NeurIPS, 2019.

Improving the Robustness via Certificates

Can we use the derived (lower-bound) margins

to improve the model robustness?

Notation: Margins w.r.t. θ



- Previously we assumed a fixed neural network with given weights/biases per layer
  - let's indicate all these parameters by θ
- Important: The optimization problems (objective + constraints) depend on θ; thus, also the optimal solutions (i.e. the margins!) of these problems depend on θ

➢ different weights θ lead to different (worst-case) margins

- During (robust) training we aim to find a good θ, e.g., via gradient descent
- To make this dependency explicit we write $m_t^*(\theta)$

Recall: Robust Training

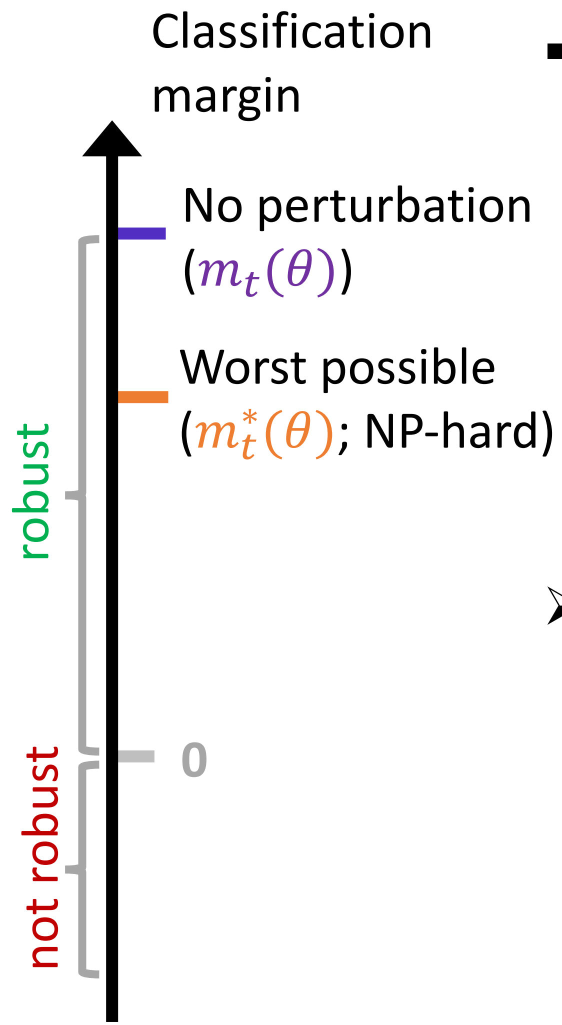
- In robust training we aim to optimize the robust loss w.r.t. θ:

$$\min_{\theta} \mathbb{E}_{(\mathbf{x},y)\in\mathbb{P}_{\text{data}}} \left[\sup_{\tilde{\mathbf{x}}\in\mathcal{P}(\mathbf{x})} \ell\left(f_\theta(\tilde{\mathbf{x}}), y\right) \right]$$

  - The challenge is how to compute $\nabla_\theta \left(\sup_{\tilde{\mathbf{x}}\in\mathcal{P}(\mathbf{x})} \ell\left(f_\theta(\tilde{\mathbf{x}}), y\right) \right)$

- The ReLU relaxation via an LP lends itself nicely to efficiently **optimize a robust loss** based on the certification
  - Note: here we focus on the general idea only

Robust Training



- To keep the discussion simple, let's assume the loss is the following margin loss:
  - if instance is correctly classified → loss = 0
  - if misclassified → loss = margin to the decision boundary (in logit space)

margin

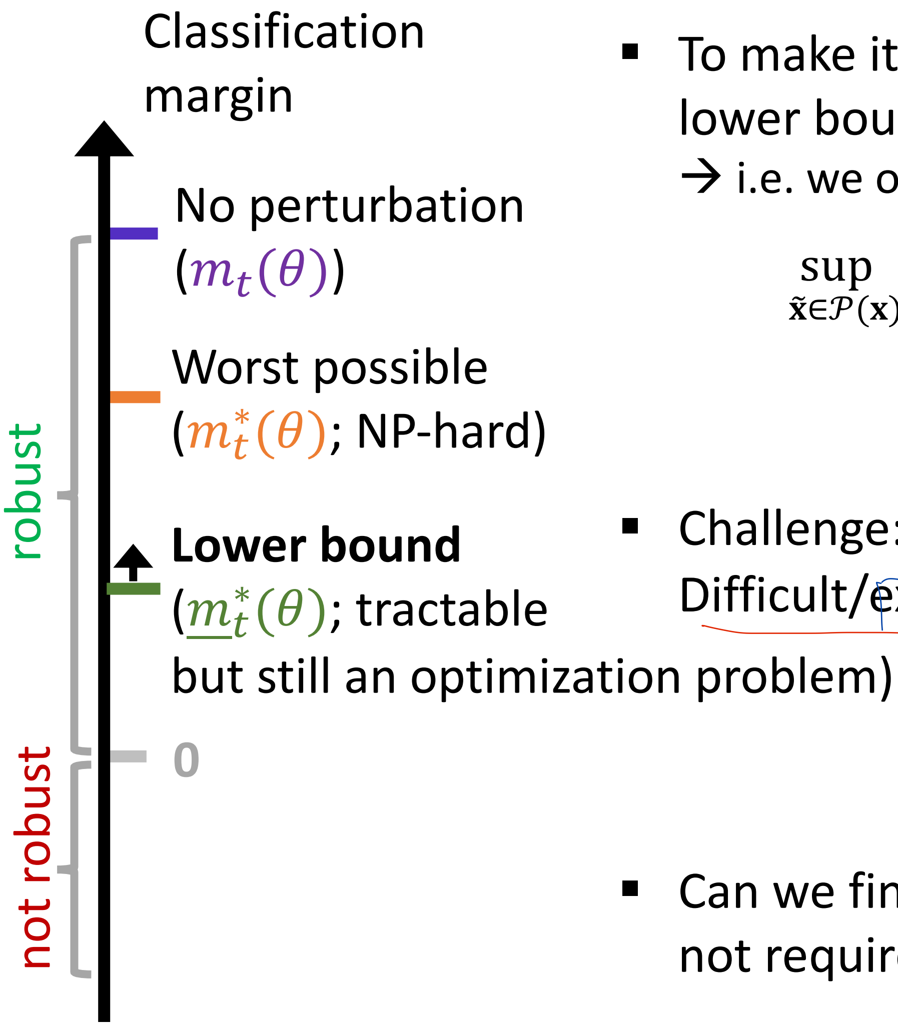
$$\ell\left(f_\theta(\tilde{\mathbf{x}}), y\right) = \max_{t \neq y} \max\left(F_\theta(\tilde{\mathbf{x}})_t - F_\theta(\tilde{\mathbf{x}})_y, 0\right)$$

correct classify

➢ Thus, the supremum evaluates to

$$\boxed{\sup_{\tilde{\mathbf{x}} \in \mathcal{P}(\mathbf{x})} \ell\left(f_\theta(\tilde{\mathbf{x}}), y\right) = \max_{t \neq y} \max\left(-m_t^*(\theta), 0\right)}$$

Robust Training with Lower Bounds



- To make it tractable, we instead optimize via the lower bound
  → i.e. we optimize a more "pessimistic" loss

$$\sup_{\tilde{\mathbf{x}} \in \mathcal{P}(\mathbf{x})} \ell\left(f_\theta(\tilde{\mathbf{x}}), y\right) = \max_{t \neq y} \max(-m_t^*(\theta), 0)$$

$$\boxed{\leq \max_{t \neq y} \max(-\underline{m}_t^*(\theta), 0)}$$

- Challenge: $\underline{m}_t^*(\theta)$ is obtained via an LP.
  Difficult/expensive to get the gradient $\nabla_\theta \underline{m}_t^*(\theta)$.

- Can we find another lower bound which does not require to solve an optimization problem?

Recap: Strong Duality

primal

$$\min_{\boldsymbol{x}} h_0(\boldsymbol{x})$$
$$\text{s.t. } h_i(\boldsymbol{x}) \leq 0 \quad i = 1 \dots M$$

dual

$$\max_{\boldsymbol{\alpha}} g(\boldsymbol{\alpha})$$
$$\text{s.t. } \boldsymbol{\alpha}_i \geq 0 \quad i = 1 \dots M$$

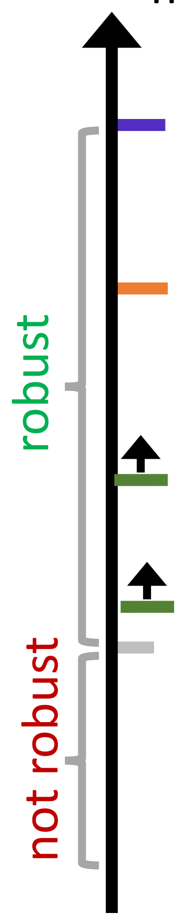
In our case, the primal is the LP from the slide „Overall LP"

In our case, both primal and dual depend additionally on θ. Thus, it would be more accurate to write $g_\theta(\boldsymbol{\alpha})$

$$h_0(\boldsymbol{x}') \geq \boxed{h(\boldsymbol{x}^*)} = \boxed{g(\boldsymbol{\alpha}^*)} \geq g(\boldsymbol{\alpha}')$$

any feasible $\boldsymbol{x}'$ optimal $\boldsymbol{x}^*$ optimal $\boldsymbol{\alpha}^*$ any feasible $\boldsymbol{\alpha}'$

Robust Training via Duality



Classification margin

No perturbation ($m_t(\theta)$)

Worst possible ($m_t^*(\theta)$; NP-hard)

Lower bound = minimum of primal LP = maximum of dual LP
($m_t^*(\theta)$; tractable but still an optimization problem)

2. Lower bound = value obtained for a feasible point of the dual LP
($m_t^{dual}(\theta)$; tractable since just some analytical function)

$g(\alpha^*) \geq g(\alpha)$

- We do not need to perform optimization to get a lower bound
- Just plug in some feasible point $\boldsymbol{\alpha}'$ into the objective function of the dual LP
- $m_t^{dual}(\theta) = g_\theta(\boldsymbol{\alpha}')$

Robust Training via Duality

Classification margin

No perturbation ($m_t(\theta)$)

Worst possible ($m_t^*(\theta)$; NP-hard)

robust

not robust

0

$$\sup_{\tilde{\mathbf{x}}\in\mathcal{P}(\mathbf{x})} \ell\left(f_\theta(\tilde{\mathbf{x}}), y\right) = \max_{t\neq y} \max(-\underbrace{m_t^*(\theta)}_{\text{NP}}, 0)$$

$$\leq \max_{t\neq y} \max(-\underbrace{\underline{m}_t^*(\theta)}_{\text{LP}}, 0)$$

$$\leq \max_{t\neq y} \max(-\underbrace{\underline{m}_t^{dual}(\theta)}_{\text{No optimation}}, 0)$$

$$= \max_{t\neq y} \max(-g_\theta(\boldsymbol{\alpha}'), 0)$$

Lower bound = minimum of primal LP = maximum of dual LP
($\underline{m}_t^*(\theta)$; tractable but still an optimization problem)

2. Lower bound = value obtained for a feasible point of the dual LP
($\underline{m}_t^{dual}(\theta)$; tractable since just some analytical function)

We reached our goal: $\nabla_\theta g_\theta(\boldsymbol{\alpha}')$ can be easily computed!

Summary

- In robust training we aim to optimize the robust loss:

$$\min_{\theta} \mathbb{E}_{(\mathbf{x},y)\in\mathbb{P}_{\text{data}}} \left[\sup_{\tilde{\mathbf{x}}\in\mathcal{P}(\mathbf{x})} \ell\left(f_\theta(\tilde{\mathbf{x}}), y\right) \right]$$

replace

- 1. We replaced the supremum by an even larger value (i.e. by a more tractable bound)
- 2. Instead of deriving the bound via an optimization problem, we used the concept of duality (any feasible point of the dual leads to a valid bound)

- Comparison to adversarial training: The supremum was replaced by a simple surrogate (loss evaluated at an adversarial point)

- In both cases
  - Computing the gradient ∇_θ of the bound/surrogate is (relatively) easy
  - You obtain ML models which are more robust

Questions – Robustness (II)

1. When the optimal value from our **convex relaxation**, $\underline{m}_t^*$, is negative, this means that …
   a) The classifier is not robust (w.r.t. the current sample $\boldsymbol{x}$)
   b) The classifier is robust (w.r.t. the current sample $\boldsymbol{x}$)
   c) We cannot make a statement

2. Same question but now the **exact certification**, m_t^*, is negative

3. Can you think about scenarios where $m_t^* = \underline{m}_t^*$?

Recommended Reading

- Lecture 13: Certified Defenses II: Convex Relaxations of Jerry Li‘s course on Robustness in Machine Learning (CSE 599-M), https://jerryzli.github.io/robust-ml-fall19.html