Machine Learning for Graphs and Sequential Data Exercise Sheet 07

Robustness of Machine Learning Models

Exercises marked with a (*) will be discussed in the in-person exercise session.

Problem 1: (*) Suppose we have a trained binary logistic regression classifier with weight vector $\boldsymbol{w} \in \mathbb{R}^d$ and bias $b \in \mathbb{R}$. Given a sample $\boldsymbol{x} \in \mathbb{R}^d$ we want to construct an adversarial example via gradient descent on the binary cross entropy loss:

$$\mathcal{L}(\boldsymbol{x}, y) = -y \log(\sigma(z)) - (1-y)\log(1-\sigma(z)),$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$ is the logistic sigmoid function, $z = \boldsymbol{w}^T\boldsymbol{x} + b$, and $y \in \{0, 1\}$ is the class label of the sample at hand.

a) Derive the gradient $\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}, y)$. How do you interpret the result?

   Hint: You may use the relation $1 - \sigma(z) = \sigma(-z)$.

b) Provide a closed-form expression for the worst-case perturbed instance $\tilde{\boldsymbol{x}}^*$ (measured by the loss $\mathcal{L}$) for the perturbation set $\mathcal{P}(x) = \{\tilde{\boldsymbol{x}} : \|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_2 \leq \epsilon\}$, i.e.

$$\tilde{\boldsymbol{x}}^* = \underset{\|\tilde{\boldsymbol{x}}-\boldsymbol{x}\|_2 \leq \epsilon}{\arg\max}\ \mathcal{L}(\tilde{\boldsymbol{x}}, y)$$

c) What is the smallest value of ϵ for which the sample $\boldsymbol{x}$ is misclassified (assuming it was correctly classified before)?

d) We would now like to perform adversarial training. Provide a closed-form expression of the worst-case loss

$$\hat{\mathcal{L}}(\boldsymbol{x}, y) = \max_{\|\tilde{\boldsymbol{x}}-\boldsymbol{x}\|_2 \leq \epsilon} \mathcal{L}(\tilde{\boldsymbol{x}}, y)$$

   as a function of $\boldsymbol{x}$ and $\boldsymbol{w}$. How do you interpret the results?

Problem 2: (*) In the lecture on exact certification of neural network robustness we have considered $K-1$ optimization problems (one for each incorrect class) of the form (c.f. slide 42):

$$m_t^* = \min_{\tilde{\boldsymbol{x}}, \boldsymbol{y}^{(l)}, \hat{\boldsymbol{x}}^{(l)}, \boldsymbol{a}^{(l)}} [\hat{\boldsymbol{x}}^{(L)}]_{c^*} - [\hat{\boldsymbol{x}}^{(L)}]_t \quad \text{subject to MILP constraints.}$$

That is, for each class $t \neq c^*$, we optimize for the **worst-case margin** m_t^*, and conclude that the classifier is robust if and only if

$$\min_{t \neq c^*} m_t^* \geq 0.$$

However, we can equivalently solve the following single optimization problem:

$$m^* = \min_{\tilde{\boldsymbol{x}}, \boldsymbol{y}^{(l)}, \hat{\boldsymbol{x}}^{(l)}, \boldsymbol{a}^{(l)}} \Big([\hat{\boldsymbol{x}}^{(L)}]_{c^*} - y \Big) \quad \text{subject to } y = \max_{t \neq c^*} [\hat{\boldsymbol{x}}^{(L)}]_t \ \wedge \text{MILP constraints,}$$

where we have introduced a new variable y into the objective function.

Problem 1: (*) Suppose we have a trained binary logistic regression classifier with weight vector $\boldsymbol{w} \in \mathbb{R}^d$ and bias $b \in \mathbb{R}$. Given a sample $\boldsymbol{x} \in \mathbb{R}^d$ we want to construct an adversarial example via gradient descent on the binary cross entropy loss:

$$\mathcal{L}(\boldsymbol{x}, y) = -y \log(\sigma(z)) - (1-y)\log(1-\sigma(z)),$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$ is the logistic sigmoid function, $z = \boldsymbol{w}^T\boldsymbol{x} + b$, and $y \in \{0, 1\}$ is the class label of the sample at hand.

a) Derive the gradient $\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}, y)$. How do you interpret the result?

   Hint: You may use the relation $1 - \sigma(z) = \sigma(-z)$.

b) Provide a closed-form expression for the worst-case perturbed instance $\tilde{\boldsymbol{x}}^*$ (measured by the loss $\mathcal{L}$) for the perturbation set $\mathcal{P}(x) = \{\tilde{\boldsymbol{x}} : \|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_2 \leq \epsilon\}$, i.e.

$$\tilde{\boldsymbol{x}}^* = \underset{\|\tilde{\boldsymbol{x}}-\boldsymbol{x}\|_2 \leq \epsilon}{\arg\max}\ \mathcal{L}(\tilde{\boldsymbol{x}}, y)$$

c) What is the smallest value of ϵ for which the sample $\boldsymbol{x}$ is misclassified (assuming it was correctly classified before)?

d) We would now like to perform adversarial training. Provide a closed-form expression of the worst-case loss

$$\hat{\mathcal{L}}(\boldsymbol{x}, y) = \max_{\|\tilde{\boldsymbol{x}}-\boldsymbol{x}\|_2 \leq \epsilon} \mathcal{L}(\tilde{\boldsymbol{x}}, y)$$

as a function of $\boldsymbol{x}$ and $\boldsymbol{w}$. How do you interpret the results?

P1 a) $\sigma(w^T x + b)$

$$\frac{\partial w^T x}{x} = w$$

$$\frac{\partial \sigma(z)}{z} = \sigma(z)\,\sigma(-z)$$

$$\nabla_x L = \frac{-y}{\cancel{\sigma(z)}} \cdot \cancel{\sigma(z)}\,\sigma(-z) \cdot w - \frac{(1-y)}{\underbrace{1-\sigma(z)}_{=\cancel{\sigma(-z)}}}\,\cancel{\sigma(-z)}\,\sigma(z) \cdot w$$

b) CE $\to$ convex

$y=1$ $\quad \boxed{x^* = x - \epsilon \frac{w}{\|w\|_2}}$ / $y=0$ $\quad x^* = x + \epsilon \frac{w}{\|w\|_2}$

$\left\| \epsilon \frac{w}{\|w\|_2} \right\|_2 = \epsilon \Leftarrow$ valid perturbatio

c) misclassified $\quad \sigma(z) = 0.5 \Leftrightarrow w^T x + b = 0$

$$w^T x - w^T \epsilon \frac{w}{\|w\|_2} + b = 0$$

$$\frac{1}{\|w\|_2}(w^T x + b) = \epsilon$$

For misclasification $\quad \epsilon > \frac{1}{\|w\|_2}(w^T x + b)$

d) $L(\tilde{x}, y) = -y \log\left(w^T\left(x - \epsilon \frac{w}{\|w\|_2}\right) + b\right) - (1-y)\log\left(-w^T\left(x - \epsilon \frac{w}{\|w\|_2}\right) + b\right)$

Problem 2: (*) In the lecture on exact certification of neural network robustness we have considered $K-1$ optimization problems (one for each incorrect class) of the form (c.f. slide 42):

$$m_t^* = \min_{\hat{\boldsymbol{x}}, \boldsymbol{y}^{(l)}, \hat{\boldsymbol{x}}^{(l)}, \boldsymbol{a}^{(l)}} [\hat{\boldsymbol{x}}^{(L)}]_{c^*} - [\hat{\boldsymbol{x}}^{(L)}]_t \quad \text{subject to MILP constraints.}$$

That is, for each class $t \neq c^*$, we optimize for the **worst-case margin** m_t^*, and conclude that the classifier is robust if and only if

$$\min_{t \neq c^*} m_t^* \geq 0.$$

However, we can equivalently solve the following single optimization problem:

$$m^* = \min_{\hat{\boldsymbol{x}}, \boldsymbol{y}^{(l)}, \hat{\boldsymbol{x}}^{(l)}, \boldsymbol{a}^{(l)}} \left([\hat{\boldsymbol{x}}^{(L)}]_{c^*} - y \right) \quad \text{subject to } y = \max_{t \neq c^*} [\hat{\boldsymbol{x}}^{(L)}]_t \wedge \text{MILP constraints,}$$

where we have introduced a new variable y into the objective function.

Express the equality constraint

$l_k \leq x_k \leq u_k$

$$y = \max(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{K-1})$$

using only linear and integer constraints. To simplify notation, here $\boldsymbol{x}_k \in \mathbb{R}$ denotes the logit corresponding to the k-th incorrect class, and $\boldsymbol{l}_k$ and $\boldsymbol{u}_k$ its corresponding lower and upper bound.

Hint: You might want to introduce binary variables to indicate which logit is the maximum.

P2 $y \leq x_u + (1-b_k)(u_{max} - l_k)$

$y \geq x_y$

$b_k \in \{0,1\}$ ⇐ indicate which x_k is maximum

$\sum_{i=1}^{k} b_k = 1$ ⇐ one-hot

Problem 3: On slide 15 of the robustness chapter, we have defined an optimization problem for untargeted attacks, i.e. we aim to have the sample $\hat{\boldsymbol{x}}$ classified as **any** class other than the correct one:

$$\min_{\hat{\boldsymbol{x}}} \mathcal{D}(\boldsymbol{x}, \hat{\boldsymbol{x}}) + \lambda \cdot L(\hat{\boldsymbol{x}}, y)$$

The loss function is defined as:

$$L(\hat{\boldsymbol{x}}, y) = \left[Z(\hat{\boldsymbol{x}})_y - \max_{i \neq y} Z(\hat{\boldsymbol{x}})_i \right]_+ ,$$

where $[\boldsymbol{x}]_+$ is shorthand for $\max(\boldsymbol{x}, 0)$ and $Z(\boldsymbol{x})_i = \log f(\boldsymbol{x})_i$ (i.e. log probability of class i. Here, $L(\hat{\boldsymbol{x}}, y)$ is positive if $\hat{\boldsymbol{x}}$ is classified correctly and 0 otherwise.

Provide an alternative loss function to turn this attack into a targeted attack, i.e. we aim to have the sample $\boldsymbol{x}$ classified as a *specific* target class t.

$$L(\hat{x}, y) = \left[\max Z(\hat{x})_i - Z(\hat{x})_t \right]_+$$

Express the equality constraint

$$y = \max(\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_{K-1})$$

using only linear and integer constraints. To simplify notation, here $\boldsymbol{x}_k \in \mathbb{R}$ denotes the logit corresponding to the k-th incorrect class, and $\boldsymbol{l}_k$ and $\boldsymbol{u}_k$ its corresponding lower and upper bound.

Hint: You might want to introduce binary variables to indicate which logit is the maximum.

Problem 3: On slide 15 of the robustness chapter, we have defined an optimization problem for untargeted attacks, i.e. we aim to have the sample $\hat{\boldsymbol{x}}$ classified as **any** class other than the correct one:

$$\min_{\hat{\boldsymbol{x}}} \mathcal{D}(\boldsymbol{x}, \hat{\boldsymbol{x}}) + \lambda \cdot L(\hat{\boldsymbol{x}}, y)$$

The loss function is defined as:

$$L(\hat{\boldsymbol{x}}, y) = \left[Z(\hat{\boldsymbol{x}})_y - \max_{i \neq y} Z(\hat{\boldsymbol{x}})_i \right]_+ ,$$

where $[\boldsymbol{x}]_+$ is shorthand for $\max(\boldsymbol{x}, 0)$ and $Z(\boldsymbol{x})_i = \log f(\boldsymbol{x})_i$ (i.e. log probability of class i. Here, $L(\hat{\boldsymbol{x}}, y)$ is positive if $\hat{\boldsymbol{x}}$ is classified correctly and 0 otherwise.

Provide an alternative loss function to turn this attack into a targeted attack, i.e. we aim to have the sample $\boldsymbol{x}$ classified as a *specific* target class t.

Problem 4: Recall from slide 41 the MILP constraints expressing the ReLU activation function:

$$\begin{aligned} &y_i \leq x_i - l_i(1 - a_i), \\ &y_i <= a_i \cdot u_i, \\ &y_i \geq x_i, \\ &y_i \geq 0, \\ &a_i \in \{0, 1\}, \end{aligned}$$

where $u_i, l_i \in \mathbb{R}$ are upper and lower bounds on the value of the ReLU input x_i.

Show that – for an unstable unit (i.e. $u_i > 0 \wedge l_i < 0$) – a continuous relaxation on a leads to the convex relaxation constraints on slide 54. That is, replacing the constraint $a_i \in \{0, 1\}$ with $a_i \in [0, 1]$ yields

$$(u_i - l_i)y_i - u_i x_i \leq -u_i l_i.$$

Problem 5: Convex relaxations of non-linearities are not limited to ReLU. For this exercise, we consider the ReLU6 non-linearity

$$\mathrm{ReLU6}(x) = \min(\max(0, x), 6),$$

which is used in MobileNet models performing low-precision computations on mobile devices.

Given input bounds l and u with $l \leq x \leq u$, provide a set of linear constraints corresponding to the convex hull of $\left\{ \begin{pmatrix} x & \mathrm{ReLU6}(x) \end{pmatrix}^T \mid l \leq x \leq u \right\}$.

Hint: You have to make a case distinction over different ranges of l and u.

Problem 4: Recall from slide 41 the MILP constraints expressing the ReLU activation function:

$$\begin{aligned}
&y_i \leq x_i - l_i(1 - a_i),\\
&y_i <= a_i \cdot u_i,\\
&y_i \geq x_i,\\
&y_i \geq 0,\\
&a_i \in \{0, 1\},
\end{aligned}$$

where $u_i, l_i \in \mathbb{R}$ are upper and lower bounds on the value of the ReLU input x_i.

Show that – for an unstable unit (i.e. $u_i > 0 \wedge l_i < 0$) – a continuous relaxation on a leads to the convex relaxation constraints on slide 54. That is, replacing the constraint $a_i \in \{0, 1\}$ with $a_i \in [0, 1]$ yields

$$(u_i - l_i)y_i - u_i x_i \leq -u_i l_i.$$

$$y \leq \min\left(x_i - l_i(1-a_i),\ a_i \cdot u_i\right) \iff a_i \in [0,1]$$

choose ~~loosest~~ loosest possible

$$y \leq \max_{a_i} \min\left(x_i - l_i(1-a_i),\ a_i \cdot u_i\right)$$

maximal at intersection of two linear function

$$x_i - l_i(1-a_i) = a_i \cdot u_i$$

$$a_i = \frac{x_i - l_i}{u_i - l_i} \iff y_i \leq \frac{x_i - l_i}{u_i - l_i} u_i$$

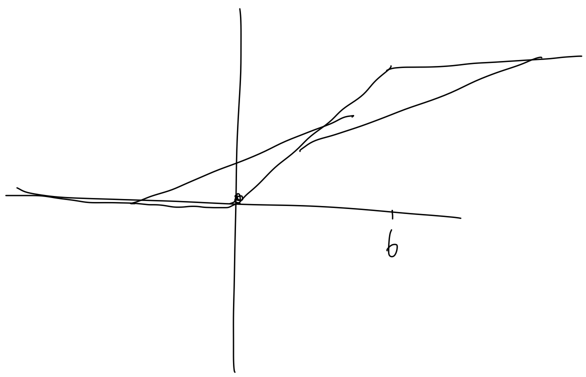
Problem 5: Convex relaxations of non-linearities are not limited to ReLU. For this exercise, we consider the ReLU6 non-linearity

$$\mathrm{ReLU6}(x) = \min(\max(0, x), 6),$$

which is used in MobileNet models performing low-precision computations on mobile devices.

Given input bounds l and u with $l \leq x \leq u$, provide a set of linear constraints corresponding to the convex hull of $\left\{ \begin{pmatrix} x & \mathrm{ReLU6}(x) \end{pmatrix}^T \mid l \leq x \leq u \right\}$.

Hint: You have to make a case distinction over different ranges of l and u.



$y = 0$

$y = x$

$y = 6$

$L < 0 < u < 6$

$(L, 0) \quad (u, u)$

$0 = La + b$

$u = ua + b$

$a = \frac{u}{u - L} \qquad b = -\frac{Lu}{u - L}$

$y \leq \frac{u}{u-L} x - \frac{Lu}{u-L} \qquad y \geq 0, \ y \geq x$

$0 < L < 6, \ u > 6$

$(L, L), \ (u, 6)$

$6 = ua + b$

$L = La + b$

$a = \frac{L - 6}{L - u} \qquad b = \frac{L(L-u) - u(L-6)}{L - u}$

Randomized smoothing

Problem 6: (*) In the previous exercise we investigated the adversarial robustness of linear classifiers

$$f(x) = \mathrm{I}[\boldsymbol{w}^T\boldsymbol{x} + b > 0]$$

with weight vector $\boldsymbol{w} \in \mathbb{R}^d$ and bias $b \in \mathbb{R}$, mapping samples from $\mathbb{R}^d$ to binary labels $\{0, 1\}$.

Given such a linear classifier f, we can define the randomly smoothed classifier $g : \mathbb{R}^d \mapsto \{0, 1\}$ with

$$g(\boldsymbol{x}) = \mathrm{argmax}_{c\in\{0,1\}} g_c(\boldsymbol{x})$$

and

$$g_c(\boldsymbol{x}) = \Pr_{\boldsymbol{\epsilon}}\left(f(\boldsymbol{x} + \boldsymbol{\epsilon}) = c\right) = \begin{cases} \Pr_{\boldsymbol{\epsilon}}\left(\boldsymbol{w}^T(\boldsymbol{x} + \boldsymbol{\epsilon}) + b \leq 0\right) & \text{if c} = 0 \\ \Pr_{\boldsymbol{\epsilon}}\left(\boldsymbol{w}^T(\boldsymbol{x} + \boldsymbol{\epsilon}) + b > 0\right) & \text{else} \end{cases},$$

where $\boldsymbol{\epsilon} \in \mathbb{R}^d$ is a random variable.

For this exercise, we assume that $\boldsymbol{\epsilon}$ follows an isotropic normal distribution, i.e. $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ with elementwise standard deviation $\sigma \in \mathbb{R}_+$.

As discussed in the lecture, evaluating randomly smoothed classifier is typically not tractable and requires sampling. This is however not the case for our simple linear classifier.

Given input $\boldsymbol{x} \in \mathbb{R}^d$, weights $\boldsymbol{w} \in \mathbb{R}^d$ and bias $b \in \mathbb{R}$, show that $g_0(\boldsymbol{x}) = \Phi_{0,1}\left(-\frac{\boldsymbol{w}^T\boldsymbol{x}}{\sigma||\boldsymbol{w}||_2} - \frac{b}{\sigma||\boldsymbol{w}||_2}\right)$, where $\Phi_{0,1} : \mathbb{R} \mapsto [0, 1]$ is the cumulative distribution of the standard normal distribution $\mathcal{N}(0, 1)$.

Hint: $\Pr_{\boldsymbol{\epsilon}}\left(\boldsymbol{w}^T(\boldsymbol{x} + \boldsymbol{\epsilon}) + b \leq 0\right)$ can alternatively be written as:

$$\int_{\mathbb{R}^d} \mathrm{I}\left[\boldsymbol{w}^T(\boldsymbol{x} + \boldsymbol{\epsilon}) + b \leq 0\right] \mathcal{N}(\boldsymbol{\epsilon} \mid \mathbf{0}, \sigma^2\mathbf{I})\, d\boldsymbol{\epsilon}.$$

Randomized smoothing for discrete data

For the sake of simplicity, we consider a slightly different setup than in the lecture. In this exercise, we assume no knowledge about $f_\theta(\mathbf{x})$ respectively $g(\mathbf{x})_c$ (usually we would estimate a lower bound of $g(\mathbf{x})_c$ via Monte Carlo sampling, but here we do not).

We use the same sparsity-aware randomization scheme $\phi(\mathbf{x})$ as in the lecture:

$$g(\mathbf{x})_c = \mathcal{P}(f(\phi(\mathbf{x})) = c) = \sum_{\tilde{\mathbf{x}} \text{ s.t. } f(\tilde{\mathbf{x}})=c} \prod_{i=1}^{n^2} \mathcal{P}(\tilde{\mathbf{x}}_i|\mathbf{x}_i) \tag{1}$$

with

$$\mathcal{P}(\tilde{\mathbf{x}}_i|\mathbf{x}_i) = \begin{cases} p_d^{\mathbf{x}_i} p_a^{1-\mathbf{x}_i} & \tilde{\mathbf{x}}_i = 1 - \mathbf{x}_i \\ (1 - p_d)^{\mathbf{x}_i}(1 - p_a)^{1-\mathbf{x}_i} & \tilde{\mathbf{x}}_i = \mathbf{x}_i \end{cases} \tag{2}$$

and the number of nodes n. For an illustration we refer to Slide 15 “Smoothed Classifier for Discrete Data”

Problem 6: (*) In the previous exercise we investigated the adversarial robustness of linear classifiers

$$f(x) = \mathrm{I}[\boldsymbol{w}^T\boldsymbol{x} + b > 0]$$

with weight vector $\boldsymbol{w} \in \mathbb{R}^d$ and bias $b \in \mathbb{R}$, mapping samples from $\mathbb{R}^d$ to binary labels $\{0, 1\}$.

Given such a linear classifier f, we can define the randomly smoothed classifier $g : \mathbb{R}^d \mapsto \{0, 1\}$ with

$$g(\boldsymbol{x}) = \mathrm{argmax}_{c \in \{0,1\}} g_c(\boldsymbol{x})$$

and

$$g_c(\boldsymbol{x}) = \Pr_{\boldsymbol{\epsilon}}\left(f(\boldsymbol{x} + \boldsymbol{\epsilon}) = c\right) = \begin{cases} \Pr_{\boldsymbol{\epsilon}}\left(\boldsymbol{w}^T(\boldsymbol{x} + \boldsymbol{\epsilon}) + b \leq 0\right) & \text{if } \ c = 0 \\ \Pr_{\boldsymbol{\epsilon}}\left(\boldsymbol{w}^T(\boldsymbol{x} + \boldsymbol{\epsilon}) + b > 0\right) & \text{else} \end{cases},$$

where $\boldsymbol{\epsilon} \in \mathbb{R}^d$ is a random variable.

For this exercise, we assume that $\boldsymbol{\epsilon}$ follows an isotropic normal distribution, i.e. $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ with elementwise standard deviation $\sigma \in \mathbb{R}_+$.

As discussed in the lecture, evaluating randomly smoothed classifier is typically not tractable and requires sampling. This is however not the case for our simple linear classifier.

Given input $\boldsymbol{x} \in \mathbb{R}^d$, weights $\boldsymbol{w} \in \mathbb{R}^d$ and bias $b \in \mathbb{R}$, show that $g_0(\boldsymbol{x}) = \Phi_{0,1}\left(-\frac{\boldsymbol{w}^T\boldsymbol{x}}{\sigma\|\boldsymbol{w}\|_2} - \frac{b}{\sigma\|\boldsymbol{w}\|_2}\right)$, where $\Phi_{0,1} : \mathbb{R} \mapsto [0, 1]$ is the cumulative distribution of the standard normal distribution $\mathcal{N}(0, 1)$.

Hint: $\Pr_{\boldsymbol{\epsilon}}\left(\boldsymbol{w}^T(\boldsymbol{x} + \boldsymbol{\epsilon}) + b \leq 0\right)$ can alternatively be written as:

$$\int_{\mathbb{R}^d} \mathrm{I}\left[\boldsymbol{w}^T(\boldsymbol{x} + \boldsymbol{\epsilon}) + b \leq 0\right] \mathcal{N}(\boldsymbol{\epsilon} \mid \mathbf{0}, \sigma^2\mathbf{I}) \; d\boldsymbol{\epsilon}.$$

$z = w^T x + b \quad \Rightarrow Ay + c \qquad y \sim N(\mu, \Sigma)$

$N(A\mu + c, A\Sigma A^T)$

$z \sim N \sim (\mu, \alpha)$

$\mu = w^T 0 + w^T x + b = w^T x + b$

$\sigma = \sqrt{w^T \sigma^2 I w} = \sigma\sqrt{w^T w} = \sigma \|w\|_2$

Problem 7: (*) Given an arbitrary graph $\mathbf{x}$, and a perturbed one $\mathbf{x}'$ where $\mathbf{x}'$ differs from $\mathbf{x}$ in exactly one edge. What is the worst-case base classifier $h^*(\mathbf{x})$? In this context, we refer to the worst-case base classifier $h^*(\mathbf{x})$ as the classifier that has the largest drop in classification confidence between $g(\mathbf{x})_c$ and $g(\mathbf{x}')_c$. Or in other words, $h^*(\mathbf{x})$ results in the most instable smooth classifier if we switch a single edge. This motivates the importance of analyzing robustness for graph neural networks (or other models with discrete input data).

50% c 2^{n^2-1}

Problem 8: (*) How many of the possible graphs $\tilde{\mathbf{x}}$ does the worst-case base classifier assign the label c (see Problem 1)? To be more specific, we are looking for a term reflecting the absolute number and not a ratio?

Problem 9: What is $g(\mathbf{x}')_c$, $g(\mathbf{x})_c$, and $g(\mathbf{x}')_c - g(\mathbf{x})_c$ for the worst-case base classifier $h^*(\mathbf{x})$ (see Problem 1)? Please derive the equations (given $p_a + p_d < 1$). Subsequently, we would like to know the precise values for $p_a = 0.001$ and $p_d = 0.1$.

n^2 eade 2^{n^2} possibility