## Machine Learning for Graphs and Sequential Data Exercise Sheet 04

# Robustness of Machine Learning Models I

---

**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 $\epsilon$ 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.

Express the equality constraint

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

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.