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 $1 - \sigma(z) = \sigma(-z)$.

$$\begin{aligned}
\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}, y) &= \frac{-y}{\sigma(z)}\frac{\partial\sigma(z)}{\partial z}\nabla_{\boldsymbol{x}} z - \frac{1-y}{\sigma(-z)}\frac{\partial\sigma(-z)}{\partial z}\nabla_{\boldsymbol{x}} z \\
&= \frac{-y}{\sigma(z)}\sigma(z)\sigma(-z)\boldsymbol{w} + \frac{1-y}{\sigma(-z)}\sigma(-z)\sigma(z)\boldsymbol{w} \\
&= -y\sigma(-z)\boldsymbol{w} + (1-y)\sigma(z)\boldsymbol{w}
\end{aligned}$$

The gradient is orthogonal to the decision boundary and points in the direction of the wrong class, depending on y.

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.

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

Since the loss is convex w.r.t. the data, taking a gradient step of magnitude ϵ towards the wrong class will result in the maximum increase in loss:

$$\begin{aligned}
\tilde{\boldsymbol{x}}^* &= \boldsymbol{x} - \epsilon\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2} \text{ if } y = 1 \\
\tilde{\boldsymbol{x}}^* &= \boldsymbol{x} + \epsilon\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2} \text{ if } y = 0
\end{aligned}$$

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

For the sample to change classification we need to have $\sigma(z) = 0.5 \Leftrightarrow \boldsymbol{w}^T\tilde{\boldsymbol{x}} + b = 0$. Plugging in the perturbation we get for $y = 1$:

$$\begin{aligned} \boldsymbol{w}^T\boldsymbol{x} - \boldsymbol{w}^T\epsilon\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2} + b = 0 \\ \boldsymbol{w}^T\boldsymbol{x} - \epsilon\|\boldsymbol{w}\|_2 + b = 0 \\ \frac{1}{\|\boldsymbol{w}\|_2}(\boldsymbol{w}^T\boldsymbol{x} + b) = \epsilon \end{aligned}$$

Thus, for a misclassification we need $\epsilon > \frac{1}{\|\boldsymbol{w}\|_2}(\boldsymbol{w}^T\boldsymbol{x} + b)$.

Analogously for $y = 0$ we obtain $\epsilon > \frac{1}{\|\boldsymbol{w}\|_2}(-\boldsymbol{w}^T\boldsymbol{x} - b)$

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

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

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

$$\begin{aligned} \hat{\mathcal{L}}(\boldsymbol{x}, y) &= \max_{\|\tilde{\boldsymbol{x}}-\boldsymbol{x}\|_2 \le \epsilon} \mathcal{L}(\tilde{\boldsymbol{x}}, y) \\ &= \mathcal{L}(\tilde{\boldsymbol{x}}^*, y) \\ &= -y\log(\sigma(\boldsymbol{w}^T\boldsymbol{x} - \epsilon\frac{\boldsymbol{w}^T\boldsymbol{w}}{\|\boldsymbol{w}\|_2} + b)) - (1-y)\log(\sigma(-\boldsymbol{w}^T\boldsymbol{x} - \epsilon\frac{\boldsymbol{w}^T\boldsymbol{w}}{\|\boldsymbol{w}\|_2} - b)) \\ &= -y\log(\sigma(\boldsymbol{w}^T\boldsymbol{x} - \epsilon\|\boldsymbol{w}\|_2 + b)) - (1-y)\log(\sigma(-\boldsymbol{w}^T\boldsymbol{x} - \epsilon\|\boldsymbol{w}\|_2 - b)) \end{aligned}$$

Consider the case $y = 1$ ($y = 0$ follows symmetrically). The input to the sigmoid function is shifted to the left (i.e. negative direction) by $\epsilon\|\boldsymbol{w}\|_2$, reducing the predicted probability of the sample $\boldsymbol{x}$ belonging to class 1. Thus, only if $\boldsymbol{w}^Tx + b \ge \epsilon\|\boldsymbol{w}\|_2$ the sample will be classified as belonging to class 1. We can interpret this as trying to enforce that each sample has at least a distance of $\epsilon\|\boldsymbol{w}\|_2$ to the decision boundary. Moreover, this margin is proportional to the norm of the weight vector, so simply increasing the norm of $\boldsymbol{w}$ does not lead to the desired outcome, since we can move $\epsilon\|\boldsymbol{w}\|_2$ units towards the decision boundary for a unit norm change on the sample $\boldsymbol{x}$. Note that, in contrast to support vector machines (SVMs), even when the samples have a margin of at least $\epsilon\|\boldsymbol{w}\|_2$ to the decision boundary, we have non-zero loss and continue training.

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

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

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.

We first define $u_{max} := \max_k \boldsymbol{u}_k$, i.e. the largest upper bound.

Now we introduce the following constraints:

$$\begin{aligned}
& y \leq \boldsymbol{x}_k + (1 - b_k)(u_{max} - \boldsymbol{l}_k) && \forall 1 \leq k \leq K-1 && (1)\\
& y \geq \boldsymbol{x}_i && \forall 1 \leq k \leq K-1 && (2)\\
& \boldsymbol{b}_k \in \{0, 1\} && \forall 1 \leq k \leq K-1 && (3)\\
& \sum_{k=1}^{K-1} \boldsymbol{b}_k = 1 && && (4)
\end{aligned}$$

The last constraint (4) simply ensures that only one element in $\boldsymbol{b}$ is 1 and all others are zero.

The only valid assignment of $\boldsymbol{b}$ is to have $\boldsymbol{b}_k = 1$ for the (unique) maximum value $\boldsymbol{x}_k = \max(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{K-1})$. To see this, consider the case that $\boldsymbol{b}_k = 1$ but $\boldsymbol{x}_k$ is not the maximum value. Then, (1) resolves to $y \leq \boldsymbol{x}_k$. However, for the maximum value $\boldsymbol{x}_{max} > \boldsymbol{x}_k$ we have from (2) $y \geq \boldsymbol{x}_{max}$, leads to a contradiction.

Consider the case $\boldsymbol{b}_k = 1$ and the corresponding value $\boldsymbol{x}_k$ is indeed the (unique) maximum. (1) and (2) imply that $y = \boldsymbol{x}_k$. The remaining values $\boldsymbol{b}_i$ are zero, and in this case we need to show that (1) and (2) are never binding, regardless of the values $\boldsymbol{x}_i$. (2) is not binding since $\boldsymbol{x}_i$ is not the maximum value. (1) is not binding because we have that $\boldsymbol{x}_i + u_{max} - \boldsymbol{l}_i \geq u_{max} \geq y$.