Advanced Computer Vision

Exercise Sheet 10

Winter Term 2023
Prof. Dr. Niels Landwehr
Dr. Ujjwal

Available: 23.01.2024
Hand in until: 30.01.2024 23:59
Exercise session: 02.02.2024

Task 1 – Carlini-Wagner Attack [30 points]

The notebook *Ex10.ipynb* contains code to train a simple CNN on the MNIST data set. The model reaches a validation accuracy of around 99%.

Implement the Carlini-Wagner attack discussed in the lecture against this model and use it to create adversarial examples for the model. Specifically, we will look at the first test example from the MNIST data set (denoted by the variable $\mathbf{x}_0$ in the notebook *Ex10.ipynb*). The trained model correctly classifies this test instance as the digit "7". Using the Carlini-Wagner attack, generate a small perturbation $\boldsymbol{\delta} \in \mathbb{R}^{28\times 28}$ such that $\mathbf{x}_0 + \boldsymbol{\delta}$ will be classified as the digit "3". Remember to use the exact Carlini-Wagner criterion as shown in the lecture, which has a squared Euclidian norm to penalize deviations from the original input. Also remember to clip your adversarial example to the range of valid inputs, which means normalized pixel values between zero and one in this case. Also remember to work with class scores, not probabilities, in the objective of the Carlini-Wagner attack. Can you generate a perturbation $\boldsymbol{\delta}$ with $||\boldsymbol{\delta}|| < 1.5$ that leads to the desired classification?

Also try to generate an adversarial perturbation that leads to a high-confidence misclassification, with $p(y = 3|\mathbf{x} + \boldsymbol{\delta}, \boldsymbol{\theta}) > 0.9999$ and $||\boldsymbol{\delta}|| < 2$.

Hints: One way to implement the attack is the following. Take the model to be attacked, and insert a layer after the input layer that adds a perturbation $\boldsymbol{\delta} \in \mathbb{R}^{28\times 28}$ to an input $\mathbf{x}$ before passing $\mathbf{x} + \boldsymbol{\delta}$ on to the rest of the model. The actual perturbation $\boldsymbol{\delta}$ is defined by $28 \cdot 28$ learnable model weights. If you freeze all the other weights in the model and then optimize the model according to the Carlini-Wagner criterion, you will obtain an adversarial perturbation in the $28 \cdot 28$ model weights. Split up the Carlini-Wagner criterion into a regularizer on your new perturbation layer and a custom loss function. Then you can train the model using e.g. the Adam optimizer (on the single input $\mathbf{x}_0$). Playing with the hyperparameters λ and κ will give you different trade-offs between a small perturbation norm and the confidence of the misclassification.

Task 2 – Sparse Perturbations [20 points]

In this task, your goal is to generate a sparse adversarial perturbation $\boldsymbol{\delta}$ for the same model and the same instance $\mathbf{x}_0$ discussed in Task 1. To encourage sparsity in the learned perturbation, implement the Hoyer-Square regularizer given in [1], Equation (3), for the network weights representing the perturbation as a custom regularizer (instead of the L2-norm used in Task 1). Can you generate an adversarial example $\mathbf{x}_0 + \boldsymbol{\delta}$ that differs in only 10 pixels from the original example $\mathbf{x}_0$ and is still misclassified as digit "3"?

Hint: you may need to add a small constant to the denominator in the Hoyer regularizer to avoid numerical issues when weights are close to zero.

[1] Yang, Huanrui, Wei Wen, and Hai Li. "DeepHoyer: Learning sparser neural network with differentiable scale-invariant sparsity measures." International Conference on Learning Representations, 2020. https://arxiv.org/pdf/1908.09979.pdf.