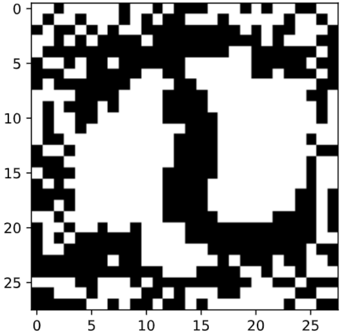
Deep Learning

Adversarial Training Basics



Learning goals

- Basics of adversarial training
- Adversarial training for linear models

ADVERSARIAL TRAINING

- To modify a trained model so that it is more resistant to such attacks, adversarial training can be performed.

- To do so, we minimize the **empirical adversarial risk** which measures the worst-case empirical loss of a model, if we are able to manipulate every input **x** in the training data set within the feasible set $\Delta(\mathbf{x})$:

$$\min_{\boldsymbol{\theta}} \mathcal{R}_{adv}(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{i=1}^{N} \max_{\boldsymbol{\delta} \in \Delta(\mathbf{x})} L(y^{(i)}, f(\mathbf{x}^{(i)} + \boldsymbol{\delta} | \boldsymbol{\theta}))$$

ADVERSARIAL TRAINING

- To solve the optimization problem, we use SGD over $\boldsymbol{\theta}$. In each SGD step $t \in \{1, 2, ...\}$ we repeatedly choose a minibatch of size m and repeat the following until a stopping criterion is met:
  1. For each $(\mathbf{x}^{(i)}, y^{(i)}), i = 1, ..., m$, we compute an adversarial example

$$\boldsymbol{\delta}^*(\mathbf{x}^{(i)}) = \underset{\boldsymbol{\delta} \in \Delta(\mathbf{x}^{(i)})}{\arg\max}\, L(y^{(i)}, f(\mathbf{x}^{(i)} + \boldsymbol{\delta} | \boldsymbol{\theta}^{[t]}))$$

  2. Then we compute the gradient of the empirical adversarial risk given $\boldsymbol{\delta}^* = (\boldsymbol{\delta}^*(\mathbf{x}^{(1)}), ..., \boldsymbol{\delta}^*(\mathbf{x}^{(m)}))$ and update $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}^{[t+1]} := \boldsymbol{\theta}^{[t]} - \alpha \frac{1}{m} \sum_{i=1}^{m} \nabla_{\boldsymbol{\theta}} L(y^{(i)}, f(\mathbf{x}^{(i)} + \boldsymbol{\delta}^*(\mathbf{x}^{(i)}) | \boldsymbol{\theta}^{[t]}))$$

- The first step is derived from Danskin's theorem, which states that the gradient of the inner function (maximization term) is simply given by the gradient of the function evaluated at its maximum.

Linear Models

LINEAR MODELS

- In case of linear models, the inner maximization problem can be solved exactly. We show this in the case of binary classification using linear models.
- Recall, the hypothesis space for logistic regression consists of models of the form:

$$\mathcal{H} = \left\{ f : \mathbb{R}^p \to [0,1] \;\middle|\; f(\mathbf{x}) = \tau\left(\sum_{j=1}^{p} \theta_j x_j + \theta_0\right), \boldsymbol{\theta} \in \mathbb{R}^p, \theta_0 \in \mathbb{R} \right\},$$

  where $\tau(z) = (1 + \exp(-z))^{-1}$ is the logistic sigmoid function.
- For class labels $y \in \{+1, -1\}$, the logistic loss is:

$$L\left(y, f(\mathbf{x} \mid \boldsymbol{\theta})\right) = \log(1 + \exp(-y(\underbrace{\sum_{j=1}^{p} \theta_j x_j}_{\boldsymbol{\theta}^T\mathbf{x}} + \theta_0))) \equiv \Psi(y(\boldsymbol{\theta}^T\mathbf{x} + \theta_0))$$

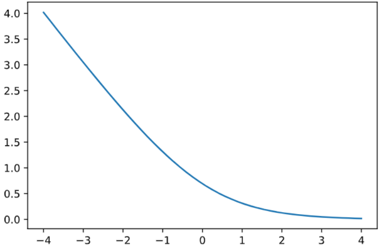
  where we define $\Psi(z) = \log(1 + \exp(-z))$.

LINEAR MODELS

- The inner maximization in the adversarial risk, which we saw earlier, can be written as:

$$\max_{\delta \in \Delta(\mathbf{x})} L(y, f(\mathbf{x} + \boldsymbol{\delta} | \boldsymbol{\theta})) = \max_{\delta \in \Delta(\mathbf{x})} \Psi(y(\boldsymbol{\theta}^T(\mathbf{x} + \boldsymbol{\delta}) + \theta_0))$$

- In this particular case, it is possible to solve the inner maximization exactly.
- First, note that Ψ is a monotonically decreasing function:



Kolter & Madry, 2019

LINEAR MODELS

- Maximizing such a monotonically decreasing function is equivalent to minimizing the argument.

- Therefore

$$\max_{\boldsymbol{\delta} \in \Delta(\mathbf{x})} \Psi(y(\boldsymbol{\theta}^T(\mathbf{x} + \boldsymbol{\delta}) + \theta_0)) = \Psi(\min_{\boldsymbol{\delta} \in \Delta(\mathbf{x})} y(\boldsymbol{\theta}^T(\mathbf{x} + \boldsymbol{\delta}) + \theta_0))$$
$$= \Psi(y(\boldsymbol{\theta}^T\mathbf{x} + \theta_0) + \min_{\boldsymbol{\delta} \in \Delta(\mathbf{x})} y(\boldsymbol{\theta}^T\boldsymbol{\delta}))$$

- We have to solve the problem

$$\min_{\boldsymbol{\delta} \in \Delta} y(\boldsymbol{\theta}^T\boldsymbol{\delta})$$

LINEAR MODELS

- To get a feel for the problem, let us consider the case where $y = +1$ and use $\Delta = \mathcal{B}_\epsilon^\infty$. The latter constrains each element of δ to lie between $-\epsilon$ and $+\epsilon$.

- The quantity $y(\boldsymbol{\theta}^T\boldsymbol{\delta})$ is then minimized when $\delta_j = -\epsilon$ for $\theta_j \geq 0$ and $\delta_j = \epsilon$ for $\theta_j < 0$.

- For $y = -1$, the signs would be flipped.

- The optimal solution then, is

$$\boldsymbol{\delta}^* = -y\epsilon \cdot \mathrm{sign}(\boldsymbol{\theta})$$

- Note that the optimal solution does not explicitly depend on **x**.

LINEAR MODELS

- The function value achieved by the solution is:

$$y \cdot \boldsymbol{\theta}^T \boldsymbol{\delta}^* = y \cdot \sum_j -y\epsilon \cdot \mathrm{sign}(\theta_j)\theta_j = -y^2\epsilon \sum_j |\theta_j| = -\epsilon \|\boldsymbol{\theta}\|_1$$

- Therefore, we have analytically computed the solution to the inner maximization problem! The solution is:

$$\max_{\boldsymbol{\delta} \in \Delta(\mathbf{x})} \Psi(y(\boldsymbol{\theta}^T(\mathbf{x}+\boldsymbol{\delta}) + \theta_0)) = \Psi(y(\boldsymbol{\theta}^T(\mathbf{x}+\boldsymbol{\delta})) - \epsilon\|\boldsymbol{\theta}\|_1)$$

- As a result, the adversarial risk, which was a min-max problem, has now been converted to a pure minimization problem:

$$\min_{\boldsymbol{\theta},\theta_0} \frac{1}{N} \sum_{i=1}^{N} \Psi\left(y^{(i)} \cdot \left(\boldsymbol{\theta}^T \mathbf{x}^{(i)} + \theta_0\right) - \epsilon\|\boldsymbol{\theta}\|_1\right)$$

- This problem is convex in $\{\boldsymbol{\theta}, \theta_0\}$ and can be solved exactly. An iterative optimizer such as SGD will also approach the global minimum.

MNIST EXAMPLE

- As an example, we look at the MNIST dataset, but this time we perform logistic regression and focus only on the classification of 0s vs. 1s.
- The logistic regression classifier was trained for 10 epochs with SGD on the training set.
- This model obtained a low misclassification rate of 0.0004 on the test set.
- To generate adversarial examples, Δ is defined as $\mathcal{B}_{0.2}^{\infty}$.
- As we saw earlier, the optimal perturbation $\boldsymbol{\delta}^*$ is $-y\epsilon \cdot \text{sign}(\boldsymbol{\theta})$.

MNIST EXAMPLE

- As δ^* does not directly depend on **x**, it is the "same" (ignoring the value of label y) across all examples. This is what it looks like:

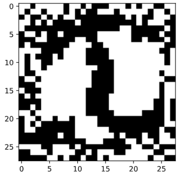


Figure: The optimal perturbation for images that contain 0. For images that contain 1, the signs would be flipped (Kolter & Madry, 2019). (The contrast between the black and white pixels is amplified for the sake of visualization.)

- The perturbation (*vaguely*) has a vertical line (like a 1) in black pixels, and a circle (like a 0) in white pixels. Intuition: When a given image is moved (translated) in the black direction, it is more likely to be classified as 1, whereas when moved in the white direction, it is more likely to be classified as 0.

MNIST EXAMPLE

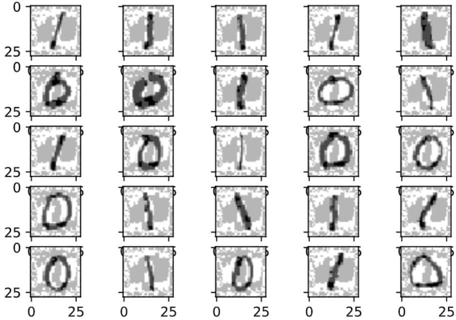


Figure: Perturbed images from the test set (Kolter & Madry, 2019).

- When all the images in test set are perturbed, the misclassification error of the model jumps from 0.0004 to 0.845!
- Interestingly, when the model is trained on similarly perturbed images from the training set (that is, the empirical adversarial risk is minimized), the misclassification error on the perturbed test set drops to 0.025.

REFERENCES

Zico Kolter and Aleksander Madry (2019)
Adversarial Robustness - Theory and Practice
https://adversarial-ml-tutorial.org/

Ian Goodfellow, Yoshua Bengio and Aaron Courville (2016)
Deep Learning
http://www.deeplearningbook.org/

Ian Goodfellow (2017)
Lecture 16 | Adversarial Examples and Adversarial Training
https://www.youtube.com/watch?v=CIfsB_EYsVI

Ian GoodfellowNicolas PapernotSandy HuangRocky DuanPieter AbbeelJack Clark (2017)
Attacking Machine Learning with Adversarial Examples
https://openai.com/blog/adversarial-example-research/

REFERENCES

Anh Nguyen, Jason Yosinski and Jeff Clune (2015)

Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images

https://arxiv.org/abs/1412.1897

Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinto (2012)

ImageNet Classification with Deep Convolutional Neural Networks. NIPS.

https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks

Maxence Prevost (2018)

Adversarial ResNet50

http://arxiv.org/abs/1207.0580

Mahmood Sharif, Sruti Bhagavatula and Lujo Bauer (2016)

Accessorize to a Crime: Real and Stealthy Attacks onState-of-the-Art Face Recognition. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security.

https://dl.acm.org/doi/10.1145/2976749.2978392

REFERENCES

- Goodfellow, Shlens (2014)
  Explaining and Harnessing Adversarial Examples
  https://github.com/maxpv/maxpv.github.io/blob/master/notebooks/Adversarial_ResNet50.ipynb

- Papernot , McDaniel, Goodfellow, Jha, Celik, Swamy (2016)
  Practical Black-Box Attacks against Machine Learning
  https://arxiv.org/abs/1602.02697

- Athalye , Engstrom, Ilyas, Kwok (2017)
  Synthesizing Robust Adversarial Examples
  https://arxiv.org/abs/1707.07397

- Tom B. Brown and Catherine Olsson, Research Engineers, Google Brain Team (2018)
  Introducing the Unrestricted Adversarial Examples Challenge
  https://ai.googleblog.com/2018/09/introducing-unrestricted-adversarial.html