CIS 7000-005: Debugging Data & Models Fall 2022

Robust Learning — October 20

Prof. Eric Wong

When you know what your model failures are, how do you fix them? One way is to retrain the model to be robust to these failures, an area known as robust learning.

- How to do adversarial training (robustness in the worst case)?
- How to do out of distribution robustness (robustness in the average case)?
- What are the empirical and guaranteed approaches, and what are the trade-offs between them?

1 Training for worst case robustness

1.1 Adversarial robustness

Robustness to adversarial examples has a checkered history. A lot of methods were proposed that simply didn't work. A brief history:

- In 2014 lots of attention brought to adversarial examples
- ICLR 2018 - 9 accepted papers on empirical adversarial robustness. 7 of the 9 were broken by Athalye et al 2018 before the conference even happened. There were also 2 certified defenses.
- January 2019 - white paper on adaptive attacks by Nicholas Carlini on how to really evaluate
- NeurIPS 2020 - 13 published papers on empirical adversarial robustness at ICLR, ICML, and NeurIPS are broken again by Tramer et al. 2020.
- Today - everyone uses some variant of adversarial training, the one defense not broken at ICLR 2018.

The concept behind adversarial training is simple. At a high level, we solve a minmax optimization problem:

$$\min_{\theta} \max_{\|\delta\| \leq \epsilon} \ell(f_\theta(x+\delta), y)$$

At each step, instead of training on the standard loss, we instead calculate an adversarial example for the current model and train on that loss instead. In practice we do this by running an adversarial attack, such as FGSM or its multi-step cousin PGD.

- The attack/defense landscape is assymetric. The attack that you use during training needs to be "strong enough" but does not have to be too strong. In contrast, at evaluation time you have to use a very strong attack.

- In fact it's possible to use FGSM adversarial training (one step), which is just 2x as long as standard training. The key is that a strong enough of an attack avoids a behavior known as *catastrophic overfitting*, but there are other ways to mitigate this as well.
- There is also *robust overfitt*, where adversarial robustness actually overfits if you train for too long even when standard deep learning does not.

1.2 Provable robustness

An alternative approach is to use provable guarantees to prove that no adversarial exists in training. At the core, this boils down to minimizing an upper bound on the adversarial loss:

$$\min_{\theta} \max_{\|\delta\| \leq \epsilon} \ell(f_\theta(x+\delta), y) \leq \min_{\theta} L(f_\theta, x, y, \epsilon)$$

These bounds can be gotten quickly with interval bound propagation or linear bounds based on linear programming or duality.

- Empirical defenses generally are faster and perform better.
- Provable defenses so far always take a hit to performance, and require training to get nonvacuous guarantees.

Randomized smoothing One other way to get provable robustness is to smooth a model over noise. The way this works is to

- Sample a lot of ablations with random noise added
- Take a majority vote
- Check that the winning margin is large enough to hold with high probability

Note that this replaces the original prediction of the model with a smoothed classifier instead, which is smoothed over ablations. In the case of Gaussian smoothing you can get ℓ_2 robustness guarantees. A brief overview of the proof is as follows. Let f be the base classifier, and let g be the smoothed classifier (so $g(x) = \arg\max_c \mathbb{P}(f(x+\epsilon) = c)$ where $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$. Let's also consider the simplified case of binary classification.

1. Suppose the top class has probability p_A, so f classifies $\mathcal{N}(x, \sigma^2 I)$ as A with probability $\geq p_A$. .
2. Consider a fixed perturbation δ. We want the probability that f classifies s $\mathcal{N}(x+\delta, \sigma^2 I)$ as A. If this probability is greater than $1/2$ then $g(x+\delta) = A$.
3. We want a statement for all possible f, so consider the worst case f which classifies s $\mathcal{N}(x, \sigma^2 I)$ with probability $\geq p_A$, but minimizes the probability that $\mathcal{N}(x+\delta, \sigma^2 I)$ is A.

4. By a similar argument to the Neyman Pearson lemma, this worst-case classifier is the linear classifier $f(x') = \begin{cases} A \text{ if } \delta^T(x' - x) \leq \sigma \|\delta\|_2 \Phi^{-1}(p_A) \\ B \text{ otherwise} \end{cases}$

5. For this worst case classifier, f classifies $\mathcal{N}(x+\delta, \sigma^2 I)$ as A with probability $\Phi\left(\Phi^{-1}(p_A) - \frac{\|\delta\|_2}{\sigma}\right)$. Solving this for $1/2$ we get the condition $\|\delta\|_2 < \sigma\Phi^{-1}(p_A)$.

Generalized smoothing results for other ℓ_p norms are also possible, see Yang et al 2020.

Derandomized smoothing We can use a similar smoothing and voting aggregation approach to get patch robustness. Specifically, we can do a majority vote over image ablations. If the winning class has a large enough of a margin, we can guarantee robustess to patches of a certain size. Interestingly, classifiers are still somewhat accurate even when using only very small image ablations.

2 Training for average case robustness

Instead of adversarial robustness, we can instead consider "natural" kinds of distribution shift. While this may seem like an easier problem, it is not quite as easy as often these kinds of distribution shifts are not mathematically easy to formulate.

ERM Empirical risk minimization is the standard ML setting, where we do standard training on the training data. There is no considerations for out of distribution shifts in this training approach.

$$\theta_{ERM} = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim p}[\ell(\theta; x, y)]$$

Group DRO Group Distributionally Robust Optimization is a form of ERM that increases the importance of domains that have larger errors.

$$\theta_{DRO} = \arg\min_{\theta} \max_{g\in\mathcal{G}} \mathbb{E}_{(x,y)\sim p_g}[\ell(\theta; x, y)]$$

CVaR Conditional Value at Risk is like group DRO but where the set of groups are all subpopulations of size α.

MixUp MixUp is a form of data augmentation that trains on "virtual examples" formed by combining training examples.

$$\theta_{ERM} = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim p,(x',y')\sim p}[\ell(\theta; \lambda x + (1-\lambda)x', \lambda y + (1-\lambda)y')]$$

JTT Just train twice — first train a low complexity ERM model to identify errors, then train a second model that upweights the error set. The hope is that identified errors correspond to more challenging groups.

Distribution matching These techniques try to ensure that encoded features have similar distributions across different domains via a penalty term, such as by matching the mean and covariance (CORAL, Sun & Saenko 2016) and MMD (Gretton et al. 2012).

$$\theta_{ERM} = \arg\min_{\theta} \ell_{ERM}(\theta) + \lambda \ell_{penalty}$$

IRM Invariant Risk Minimization aims to find a representation that is invariant to different distributions, or a representation that is simultaneously optimal across different environments. It does so by separating out invariant features from environmental features.

$$\theta_{IRM} = \arg\min_{(w,\Phi)} \sum_{i} \mathbb{E}_{(x,y)\sim p_i}[\ell(w \circ \Phi; x, y)]$$

subject to

$$w \in \arg\min_{w'} \mathbb{E}_{(x,y)\sim p_i}[\ell(w' \circ \Phi; x, y)]$$

for all environments p_i.

Meta learning Meta learning techniques can also be used to meta learn classifiers that generalize to new domains.

2.1 Current state of affairs

Unfortunately it seems that ERM, when compared fairly, can actually outperform all of these approaches. In fact, some negative theoretical results were published for IRM:

- For linear f, if the number of environments surpasses the number of environmental features, e.g. $E > d_e$, then w uses only invariant features with minimax optimal risk. If $E \leq d_e$, then w relies on invariant features.
- For linear f, there exists a linear (Φ, w) that uses only environmental features that achieves lower risk than the invariant features.
- For nonlinear f, there exists nonlinear (Φ, w) that is optimal under the IRM objective and nearly identical to the invariant predictor on training data, but is equivalent to ERM when the test and train environments have sufficiently different means, and in fact fails to use invariant features outside of close overlap with the training environment.

3 References

Cohen, Jeremy, Elan Rosenfeld, and Zico Kolter. "Certified adversarial robustness via randomized smoothing." International Conference on Machine Learning. PMLR, 2019.

Yang, Greg, et al. "Randomized smoothing of all shapes and sizes." International Conference on Machine Learning. PMLR, 2020.

Salman, Hadi, et al. "Certified patch robustness via smoothed vision transformers." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

Rosenfeld, Elan, Pradeep Ravikumar, and Andrej Risteski. "The risks of invariant risk minimization." arXiv preprint arXiv:2010.05761 (2020).