Notes

Yu Liu

June 2020

1 reasons for adversarial examples

1. [3]: deep neural networks have big Lipschtiz constant and it's the necessary condition.
2. [1]: adversarial examples come from the linear nature of the models, and non-linear models like RBF can resist adversarial examples. It's a trade-off between easy-to-train and resist-adversarial-examples. And also this kind of adversarial examples occur in a continuous regions of the 1-D subspace defined by some key direction(FGSM).
3. [2]: adversarial vulnerability is a result of the models' sensitivity to well-generalizing features in the data, especially the non-robust features. And this article tries to show that robust and non-robust features exist, by showing that standard training on robust data lead to a robust classifier, and standard training on non-robust data lead to a standard classifier. Given a trained classifier, we want to get robust and non-robust feature(I'm not sure because I haven't fully understand this paper).

2 how to deal with it

2.1 how to generate the 'attack'

1. [3]:box-constrained optimization problem:

$$min \;\; ||r||_2$$

$$s.t. \;\; f(x+r) = l, x+r \in [0,1]^m$$

and its approximation problem L-BFGS:

$$min(c|r| + loss_f(x+r, l)), s.t. x+r \in [0,1]^m$$

for some input x and wrong label l

2. [1]: fast gradient sign method(FGSM):the perturbation is $\eta = \epsilon sign(\nabla_x J(\theta, x, y))$, and this comes from the linear approximation with $||\eta||_\infty \leq \epsilon$. We want to get perturbation with the biggest impact to the loss function J around a small area around x, so we can first use linear approximation around x, and compute η that will make the biggest change to the result. It also shows that only the direction matters.

   This article shows us a specific example using logistic regression.

2.2 how to 'defend'

1. [3]:use the hard-negative mining principle.
2. [1]: use:

$$\hat{J}(\theta, x, y) = \alpha J(\theta, x, y) + (1 - \alpha) J(\theta, x + \epsilon sign(\nabla_x J(\theta, x, y)), y)$$

   the latter part is just a regularization term, and we can understand it as a min-max error, that is, we want to minimize the maximal error we shall get, and around point x, we use linear approximation to get it's maximum error, η.

   This article shows us a specific example using logistic regression.

2.3 how to verification

1. [4]: only this article talks about the verification problem, which means to determine whether a model is robust around point x. Verification is hard, but this article propose co-design to make verification easy. That is to take it into account when designing model.

   This article focus on k-layers, fully-connected, feed-forward DNN classifiers with ReLU.

   The article shows that weight-sparsity and ReLU-stability can make verification faster, due to that we can transform verification problem to a MILP, in which fewer variables and less ReLU branches helps.

   The specific verification problem is :

$$min(f(x', W)_y - f(x', W)_{y^*})$$
$$s.t. \;\; x' \in Adv(x)$$

   where $Adv(x)$ means the area around x.

   In weight-sparsity part, we can use l_1-regularization and small weight pruning.Notice that l_1-regularization is inherent in adversarial training, cause $f(x') = Wx + b + W(x' - x) \leq f(x) + \epsilon||W||_1$.

   In ReLU-stability part, we use approximation of the upper and lower bound to determine whether this ReLU layer has 2 brunches. If we call the upper and lower bound u, l, than the determine function is:

$$F = sign(u)sign(l)$$

and we can use a differentiable function:

$$F' = -tanh(1 + ul)$$

to approximate it.

The advantage of this co-design method is that it can be added to any model just with a small weight to the loss function to get a big success in verification.

3 My questions

First, there is a phenomenon that a set adversarial examples for one model are usually adversarial examples again for other different models. In Section 1, I cannot understand how reason 2 explain it. Reason 3 can interpret it because it's the features matter. But in [1] it says RBF can resist this kind of adversarial examples, but in my point of view, RBF is also catching features, so there is a contradiction, maybe I make mistake about RBF?

Second, in [3], the perturbation is with l_2-constraint, and in [1] and [4] it is with l_∞-constraint, it seems that in [2] it's with l_2-constraint (I'm not sure because I haven't fully understand this paper). I think it is easier to analysis under $l-\infty$-constraint. So can we do some pre-processing like image restoration to make sure it is approximately under $l - \infty$-constraint?

References

[1] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[2] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pages 125–136, 2019.

[3] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[4] Kai Y Xiao, Vincent Tjeng, Nur Muhammad Shafiullah, and Aleksander Madry. Training for faster adversarial robustness verification via inducing relu stability. *arXiv preprint arXiv:1809.03008*, 2018.