文章工作：
将对抗训练和鲁棒性验证结合在一起进行实现，并得到了良好的效果。

动机：
现在有很多工作是对抗训练，增强网络的鲁棒性，但是对抗训练并不提供鲁棒性保证，有可能被更强力的攻击所击破；有人已经做了将对抗训练和鲁棒性验证结合在一起进行，但是这样训练出来的网络的标准测试精度太低，不是良好的网络。所以本文工作如下：
将对抗训练和鲁棒性验证结合在一起，得到一个精度高且可验证鲁棒性高的网络。

贡献：
1:提出了一种通过对每层都进行凸包含来进行对抗训练的方法，这种方法训练出来一个精度高且可验证鲁棒性高的网络；
2:将先前工作中所用的凸包含扩展到本分层对抗训练工作中
3:本工作可以提高最新在cifar-10对抗训练的模型的精度和可验证鲁棒性
4:将本工作实现成一个系统，但是这个github工程现在消失找不到了

扰动区域定义

$$\mathbb{S}_0(\boldsymbol{x}) = \{\boldsymbol{x}' \in \mathbb{R}^{d_0}, ||\boldsymbol{x} - \boldsymbol{x}'||_\infty < \epsilon\}$$

网络定义：

$$h_\theta = h_\theta^k \circ h_\theta^{k-1} \cdots \circ h_\theta^1 \text{ and } h_\theta^i : \mathbb{R}^{d_{i-1}} \to \mathbb{R}^{d_i}$$

从某一层到输出层的网络定义

$$h_\theta^{i:k} = h_\theta^k \circ h_\theta^{k-1} \cdots \circ h_\theta^i.$$

要验证的约束，h是网络输出层的输出向量：

$$\boldsymbol{c}^T h_\theta(\boldsymbol{x}') + d < 0, \forall \boldsymbol{x}' \in \mathbb{S}_0(\boldsymbol{x})$$

比如我们要验证第一类的输出一直大于第二类，那么c=[1,-1]，d=0，即可。

这是训练网络时所需要用的损失函数，是一个双层优化，最小化最大分类损失：

$$\min_{\theta} \mathbb{E}_{(\mathrm{x},\mathrm{y})\sim D} \max_{\boldsymbol{x}' \in \mathbb{S}_0(\boldsymbol{x})} \mathcal{L}(h_\theta(\boldsymbol{x}'), y)$$

由于内层max不好计算，所以现有两种技术将之进行逼近：
1:对抗训练，对抗训练每次都使用对抗样本，一般的对抗样本不是最大损失对抗样本，所以这是一个下界；
2:对抗验证，由于是验证，所以找的必须是不小于最大损失的点，这是上界

文章验证的大概过程与采用的字母：
首先对输入层进行凸包含，就等于本身；
然后对网络计算进行凸松弛，针对relu函数，原本网络一层输出等于h(x)，输出域是S，松弛之后的计算是g(x)，输出域的凸包含是C。
每层都这样计算，最后验证输出层的凸包含是否满足所要验证的性质。

Latent adversarial examples:指的是由于凸包含带来的假的对抗样本，文章的训练目标就是最小化这些潜在对抗样本的数量

做法：首先对输入层进行鲁棒训练，和我们一般的鲁棒训练一样，找到对抗样本使对抗样本正确分类，然后如图1所示，递进分层进行优化，已经优化过的层不再动，对隐藏层优化的时候的优化目标函数就是：

$$\min_{\theta^{l+1:k}} \mathbb{E}_{(\mathrm{x},\mathrm{y})\sim D} \max_{\boldsymbol{x}'_l \in \mathbb{C}_l(\boldsymbol{x})} \mathcal{L}(h_\theta^{l+1:k}(\boldsymbol{x}'_l), y, \theta)$$

就相当于把隐藏层作为输入层进行优化了。

算法：第L层优化的算法
3:从原始数据随机取点x
4:计算x到第L层所产生的凸包含域
5:从这些凸包含域中随机取点x’
7:以x’作为初始点，计算凸包含域中的对抗样本，更新x’
9:用x’更新参数,只更新到L层，之前的就不更新了

他们不使用普通的区间计算，使用zonotope，表示成

$$\mathbb{C}_l(\boldsymbol{x}) = \{\boldsymbol{a}_l + \boldsymbol{A}_l\boldsymbol{e} \mid \boldsymbol{e} \in [-1, 1]^{m_l}\}$$

做pgd中projection的时候的做法如图2:
首先将点换到zonotope域，然后将系数clip到[-1,1]，再转回凸包含域

使用zonotope到最后其系数矩阵A可能非常庞大，两种方法：
1是系数矩阵可能是稀疏的，只计算非0的位置
2是使用已有的方法计算上下界，然后确定系数矩阵

验证的技术：
首先做一个处理：再进行一次优化，优化包含relu区域的直线的斜率，使relu区域最小

验证使用MILP进行编码验证