# ML2021Spring HW10 Report

Economics TSU-FU,Li

r09323036

| Public Score | Private Score |
|---|---|
| 0.000 | 0.000 |

The methods I used to pass the strong baselines include:

I apply Momentum iterative fast gradient sign method and Esemble attack.

**Algorithm 2** MI-FGSM for an ensemble of models

**Input:** The logits of $K$ classifiers $\boldsymbol{l}_1, \boldsymbol{l}_2, ..., \boldsymbol{l}_K$; ensemble weights $w_1, w_2, ..., w_K$; a real example $\boldsymbol{x}$ and ground-truth label $y$;
**Input:** The size of perturbation $\epsilon$; iterations $T$ and decay factor $\mu$.
**Output:** An adversarial example $\boldsymbol{x}^*$ with $\|\boldsymbol{x}^* - \boldsymbol{x}\|_\infty \leq \epsilon$.

$\alpha = \epsilon/T$;
$\boldsymbol{g}_0 = 0$; $\boldsymbol{x}_0^* = \boldsymbol{x}$;
**for** $t = 0$ to $T - 1$ **do**
  Input $\boldsymbol{x}_t^*$ and output $\boldsymbol{l}_k(\boldsymbol{x}_t^*)$ for $k = 1, 2, ..., K$;
  Fuse the logits as $\boldsymbol{l}(\boldsymbol{x}_t^*) = \sum_{k=1}^{K} w_k \boldsymbol{l}_k(\boldsymbol{x}_t^*)$;
  Get softmax cross-entropy loss $J(\boldsymbol{x}_t^*, y)$ based on $\boldsymbol{l}(\boldsymbol{x}_t^*)$ and Eq. (9);
  Obtain the gradient $\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_t^*, y)$;
  Update $\boldsymbol{g}_{t+1}$ by Eq. (6);
  Update $\boldsymbol{x}_{t+1}^*$ by Eq. (7);
**end for**
**return** $\boldsymbol{x}^* = \boldsymbol{x}_T^*$.

```
model = ptcv_get_model('resnet20_cifar10', pretrained=True).to(device).eval()
model1 = ptcv_get_model('resnet56_cifar10', pretrained=True).to(device).eval()
model2 = ptcv_get_model('resnet110_cifar10', pretrained=True).to(device).eval()
model3 = ptcv_get_model('resnet164bn_cifar10', pretrained=True).to(device).eval()
model4 = ptcv_get_model('resnet272bn_cifar10', pretrained=True).to(device).eval()
model5 = ptcv_get_model('resnet542bn_cifar10', pretrained=True).to(device).eval()
model6 = ptcv_get_model('resnet1001_cifar10', pretrained=True).to(device).eval()
model7 = ptcv_get_model('resnet1202_cifar10', pretrained=True).to(device).eval()
model8 = ptcv_get_model('preresnet20_cifar10', pretrained=True).to(device).eval()
model9 = ptcv_get_model('preresnet272bn_cifar10', pretrained=True).to(device).eval()
model10 = ptcv_get_model('preresnet1202_cifar10', pretrained=True).to(device).eval()
model11 = ptcv_get_model('preresnet542bn_cifar10', pretrained=True).to(device).eval()
```