
Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning

Yongwon Jo

School of Industrial and Management Engineering, Korea University



KOREA
UNIVERSITY



DMQA

Contents

❖ Research Purpose

❖ RotNet

❖ Experiments

❖ Conclusion

Research Purpose

- ❖ **Virtual Adversarial Training: A Regularization Method for Supervised and Semi-supervised Learning**
 - 2019년 IEEE Transaction on Pattern Analysis and Machine Intelligence에서 발표된 논문
 - 2022년 8월 28일 기준 1,778회 인용
 - 학습 데이터에 과적합 되는 것을 방지하기 위한 Regularization 방법을 제안하는 논문

Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning

Takeru Miyato^{ID}, Shin-Ichi Maeda, Masanori Koyama, and Shin Ishii

Research Purpose

- ❖ **Virtual Adversarial Training: A Regularization Method for Supervised and Semi-supervised Learning**
 - Unlabeled data에 대한 가상 레이블을 생성하고 가상 레이블과 모델 Output 사이 Gradient 계산
 - Gradient 역방향에 대해 Perturbation을 가해 이미지 왜곡 (Local distribution smoothness)
 - 왜곡된 이미지에 대한 Output probability가 가상 레이블과 동일하도록 학습

- ❖ **Virtual Adversarial Training 장점**
 - Semi-supervised Learning에 적용 가능
 - Gradient를 계산할 수 있는 Parametric 모델에 모두 적용 가능
 - Hyperparameter 수가 적음
 - Parameter를 구하는 방식과 관계없이 적용 가능

Adversarial Training

❖ Adversarial Training

- $Loss_{adv}$ 를 최소화하는 방향으로 Parametric 모델 학습
 - $D[x, y]$ 는 x, y 사이 거리를 측정할 수 있는 비선형함수 (For example, Cross entropy)
 - x_l 은 Labeled data를 의미하며, $q(y|x_l)$ 은 Labeled data의 실제 범주 분포(One-hot vector)
- γ_{adv} 는 계산이 어렵기 때문에 $L_2 norm$ 또는 $L_\infty norm$ 으로 근사해 사용

$$Loss_{adv} := D[q(y|x_l), p(y|x_l + \gamma_{adv}, \theta)]$$

$$\text{where, } \gamma_{adv} := \underset{r; \|r\| < \epsilon}{\operatorname{argmax}} D[q(y|x_l), p(y|x_l + r)]$$

$$\gamma_{adv} \approx \epsilon \frac{g}{\|g\|_2}, \text{ where } g = \nabla_{x_l} D[h(y; y_l), p(y|x_l, \theta)] \text{ (L_2 norm Approximation)}$$

$$\gamma_{adv} \approx \epsilon * sign(g), \text{ where } g = \nabla_{x_l} D[h(y; y_l), p(y|x_l, \theta)] \text{ (L_∞ norm Approximation)}$$

Virtual Adversarial Training

❖ Virtual Adversarial Training (VAT)

- Unlabeled data(x_{ul})에 대해서는 $q(y|x_{ul})$ 을 산출할 수 없는 상황
- $p(y|x_{ul}, \hat{\theta})$ 는 Unlabeled data에 대한 예측 확률이며 이를 $q(y|x_{ul})$ 라 가정(Pseudo-label)
- Local distribution smoothness(LDS)를 아래와 같이 정의
- x_* 의 * 는 Labeled, Unlabeled 가 들어가는 자리
- 2차 미분을 사용해 γ_{vadv} 을 근사 계산하는 식은 논문 3.3에 존재하며 관심있을 때 수식확인

$$LDS(x_*, \theta) := D[p(y|x_*, \hat{\theta}), p(y|x_* + \gamma_{vadv}, \theta)]$$

$$\text{where, } \gamma_{vadv} := \underset{r; \|r\| < \varepsilon}{\operatorname{argmax}} D[p(y|x_*, \hat{\theta}), p(y|x_* + r)]$$

Virtual Adversarial Training

❖ Virtual Adversarial Training (VAT) 전체 손실 함수

- $l(D_l, \theta)$ 는 Labeled data에 대한 손실 함수이며 Negative Loglikelihood function 사용
- R_{vadv} 는 VAT를 사용한 Regularization Term
- VAT 적용을 위한 Hyperparameter는 ϵ , α 총 2가지이며 논문 내 실험은 $\alpha=1$ 로 고정

$$LDS(x_*, \theta) := D[p(y|x_*, \hat{\theta}), p(y|x_* + \gamma_{vadv}, \theta)]$$

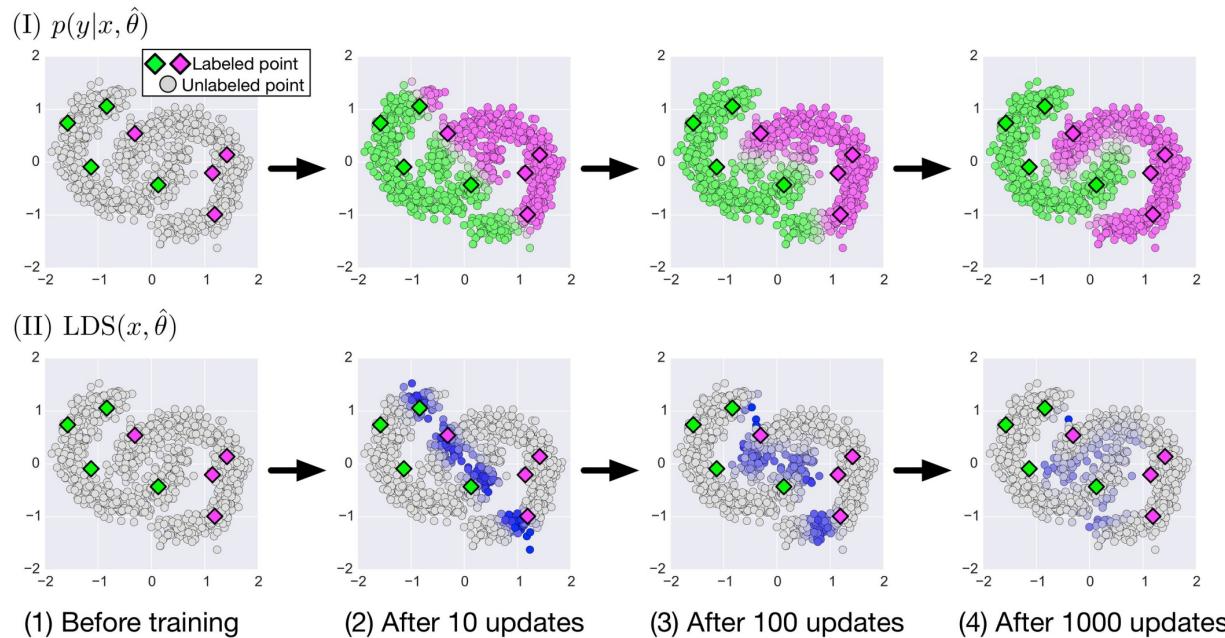
where, $\gamma_{vadv} := \underset{r; \|r\| < \varepsilon}{\operatorname{argmax}} D[p(y|x_*, \hat{\theta}), p(y|x_* + r)]$

$$\text{Loss} = l(D_l, \theta) + \alpha * \frac{1}{N_l + N_{ul}} \sum_{x_* \in D_l, D_{ul}} LDS(x_*, \theta)$$

Virtual Adversarial Training

❖ Virtual Adversarial Training (VAT) 적용에 따른 2D Feature space 시각화

- 첫번째 행은 Iteration 반복에 따른 예측 결과 시각화
- 두번째 행은 Unlabeled data에 대한 거리 측도에 대한 시각화
- Iteration이 진행될수록 정확하게 Unlabeled data에 대해 예측하는 것을 확인 가능
- Decision Boundary근처에서 거리 측도 값이 높아지는 것을 확인



Experiments

❖ Supervised Learning 성능 비교(MNIST, CIFAR-10)

- (MNIST) 해당 데이터를 위해 개발된 모델 Ladder Network 와 유사한 성능
- (CIFAR-10) 단순한 모델임에도 ResNet, DenseNet과 유사한 성능
- 하지만 모델 Smoothness (Regularization) 성능은 다른 모델 대비 잘 이루어 진다는 것을 시각적으로 증명 (Paper Figure 2)

TABLE 1
Test Performance of Supervised Learning Methods
on MNIST with 60,000 Labeled Examples in the
Permutation Invariant Setting

| Method | Test error rate(%) |
|--|---------------------|
| SVM (Gaussian kernel) | 1.40 |
| Dropout [39] | 1.05 |
| Adversarial, L_∞ norm constraint [14] | 0.78 |
| Ladder networks [33] | 0.57 (± 0.02) |
| Baseline (MLE) | 1.11 (± 0.06) |
| RPT | 0.84 (± 0.03) |
| Adversarial, L_∞ norm constraint | 0.79 (± 0.03) |
| Adversarial, L_2 norm constraint | 0.71 (± 0.03) |
| VAT | 0.64 (± 0.05) |

TABLE 2
Test Performance of Supervised Learning Methods
Implemented with CNN on CIFAR-10 with 50,000
Labeled Examples

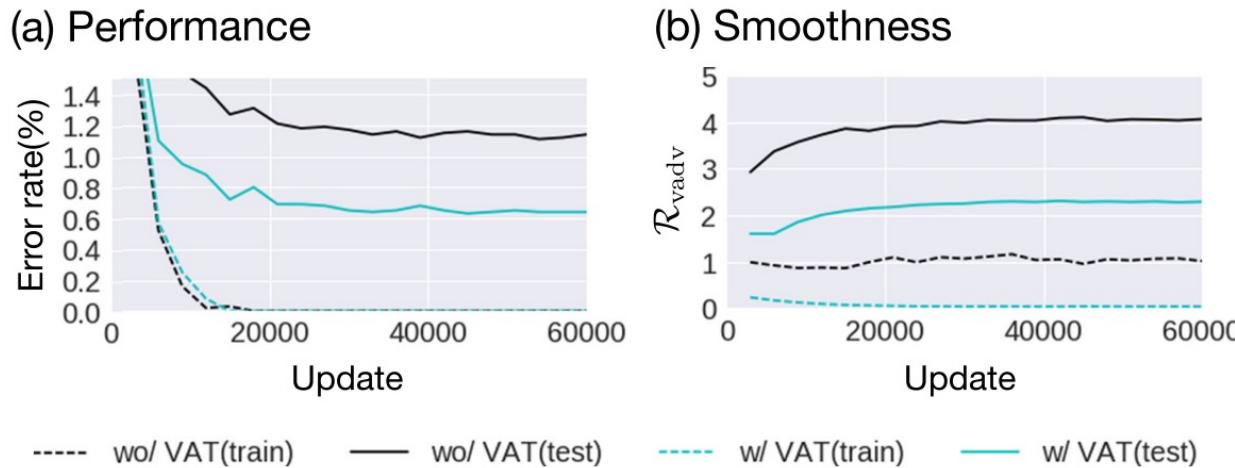
| Method | Test error rate(%) |
|------------------------------|---------------------|
| Network in Network [26] | 8.81 |
| All-CNN [38] | 7.25 |
| Deeply Supervised Net [25] | 7.97 |
| Highway Network [40] | 7.72 |
| ResNet (1001 layers) [17] | 4.62 (± 0.20) |
| DenseNet (190 layers) [18] | 3.46 |
| Baseline (only with dropout) | 6.67 (± 0.07) |
| RPT | 6.30 (± 0.04) |
| VAT | 5.81 (± 0.02) |

The top part cites the results provided by the original paper. The bottom part shows the performance achieved by our implementation.

Experiments

❖ Supervised Learning 성능 비교(MNIST, CIFAR-10)

- (MNIST) 해당 데이터를 위해 개발된 모델 Ladder Network 와 유사한 성능
- (CIFAR-10) 단순한 모델임에도 ResNet, DenseNet과 유사한 성능
- 하지만 모델 Smoothness (Regularization) 성능은 다른 모델 대비 잘 이루어 진다는 것을 시각적으로 증명 (Paper Figure 2)



Experiments

❖ Semi-Supervised Learning 성능 비교(MNIST, SVHN, and SVHN)

- (MNIST) Generative model 기반 Semi-supervised Learning 보다 성능이 뛰어나다고 서술
 - 하지만 Test error rate 기준 GAN with FM 보다는 성능이 떨어짐 (이상한 서술...)
- (SVHN, CIFAR-10) 다른 Semi-supervised Learning 방법론 대비 뛰어난 성능
 - EntMin: Regularization Term으로 conditional entropy 추가 사용
 - $R_{cent} = -\frac{1}{N_l} \sum_{x \in D_l, D_u} \sum_y p(y|x, \theta) * \log p(y|x, \theta)$

TABLE 3
Test Performance of Semi-Supervised Learning Methods
on MNIST with the Permutation Invariant Setting

| Method | Test error rate(%) | |
|----------------------|---------------------|---------------------|
| | $N_l = 100$ | $N_l = 1000$ |
| TSVM [8] | 16.81 | 5.38 |
| PEA [5] | 5.21 | 2.87 |
| DGM (M1+M2) [22] | 3.33 (± 0.14) | 2.40 (± 0.02) |
| CatGAN [37] | 1.91 (± 0.1) | 1.73 (± 0.18) |
| Skip DGM [27] | 1.32 (± 0.07) | |
| Ladder networks [33] | 1.06 (± 0.37) | 0.84 (± 0.08) |
| Auxiliary DGM [27] | 0.96 (± 0.02) | |
| GAN with FM [36] | 0.93 (± 0.07) | |
| RPT | 6.81 (± 1.30) | 1.58 (± 0.54) |
| VAT | 1.36 (± 0.03) | 1.27 (± 0.11) |

The value N_l stands for the number of labeled examples in the training set. The top part cites the results provided by the original paper. The bottom part shows the performance achieved by our implementation. (PEA = Pseudo Ensembles Agreement, DGM = Deep Generative Models, FM = feature matching).

TABLE 4
Test Performance of Semi-Supervised Learning Methods
on SVHN and CIFAR-10 without Image Data Augmentation

| Method | Test error rate(%) | |
|--------------------------------------|----------------------|--------------------------|
| | SVHN $N_l = 1000$ | CIFAR-10 $N_l = 4000$ |
| SWWAE [48] | 23.56 | |
| *Skip DGM [27] | 16.61 (± 0.24) | |
| *Auxiliary DGM [27] | 22.86 | |
| Ladder networks, Γ model [33] | | 20.40 (± 0.47) |
| CatGAN [37] | | 19.58 (± 0.58) |
| GAN with FM [36] | 8.11 (± 1.3) | 18.63 (± 2.32) |
| II model [24] | 5.43 (± 0.25) | 16.55 (± 0.29) |
| (on Conv-Small used in [36]) | | |
| RPT | 8.41 (± 0.24) | 18.56 (± 0.29) |
| VAT | 6.83 (± 0.24) | 14.87 (± 0.13) |
| (on Conv-Large used in [24]) | | |
| VAT | 5.77 (± 0.32) | 14.18 (± 0.38) |
| VAT+EntMin | 4.28 (± 0.10) | 13.15 (± 0.21) |

The value N_l stands for the number of labeled examples in the training set. The top part cites the results provided by the original paper. The middle and bottom parts show the performance achieved by our implementation. The asterisk(*) stands for the results on the permutation invariant setting. (DGM = Deep Generative Models, FM = feature matching).

Experiments

❖ Semi-Supervised Learning with Data Augmentation 성능 비교(MNIST, SVHN, and SVHN)

- (SVHN, CIFAR-10) 다른 Semi-supervised Learning 방법론 대비 뛰어난 성능
 - EntMin: Regularization Term으로 conditional entropy 추가 사용
 - $R_{cent} = -\frac{1}{N_l + N_{ul}} \sum_{x \in D_l \cup D_{ul}} \sum_y p(y|x, \theta) * \log p(y|x, \theta)$
- 데이터 증강 기법 적용으로 성능 향상 달성

TABLE 5
Test Performance of Semi-Supervised Learning Methods
on SVHN and CIFAR-10 *with* Image Data Augmentation

| Method | Test error rate(%) | |
|------------------------------|----------------------|--------------------------|
| | SVHN $N_l = 1000$ | CIFAR-10 $N_l = 4000$ |
| Π model [24]. | 4.82 (± 0.17) | 12.36 (± 0.31) |
| Temporal ensembling [24] | 4.42 (± 0.16) | 12.16 (± 0.24) |
| Sajjadi et al. [35] | | 11.29 (± 0.24) |
| (On Conv-Large used in [24]) | | |
| VAT | 5.42 (± 0.22) | 11.36 (± 0.34) |
| VAT+EntMin | 3.86 (± 0.11) | 10.55 (± 0.05) |

The value N_l stands for the number of labeled examples in the training set. The performance of all methods other than Sajjadi et al. [35] are based on experiments with the moderate data augmentation of translation and flipping (see Appendix D for more detail). Sajjadi et al. [35] used extensive image augmentation, which included rotations, stretching, and shearing operations. The top part cites the results provided by the original paper. The bottom part shows the performance achieved by our implementation.

Experiments

❖ Hyperparameter ϵ, α 변화에 따른 성능 비교(SVHN, and SVHN)

- Taylor expansion을 사용한 R_{vadv} Term 간소화

➤ $R_{vadv} \approx \max_r \left\{ \frac{1}{2} r^T H(x, \theta) r; \|r\|_2 < \epsilon \right\} = \frac{1}{2} \epsilon^2 \lambda_1(x, \theta)$

➤ λ_1 는 Hessian matrix 분해 시 가장 큰 Eigen value

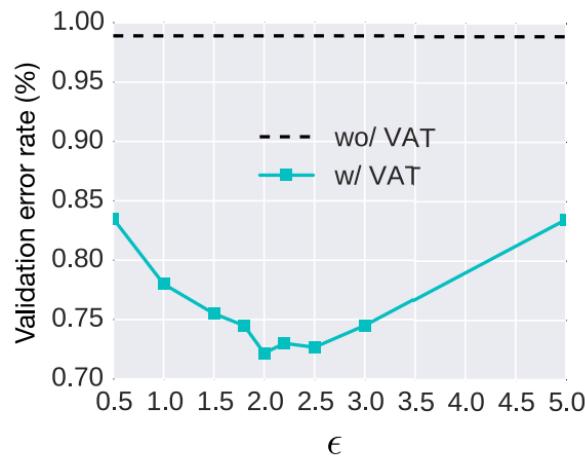
- 손실 함수는 아래와 같이 변경 가능
- ϵ, α 의 곱 형태로 표현 가능

$$Loss = l(D_l, \theta) + \alpha * \frac{1}{2} \epsilon^2 \lambda_1(x, \theta)$$

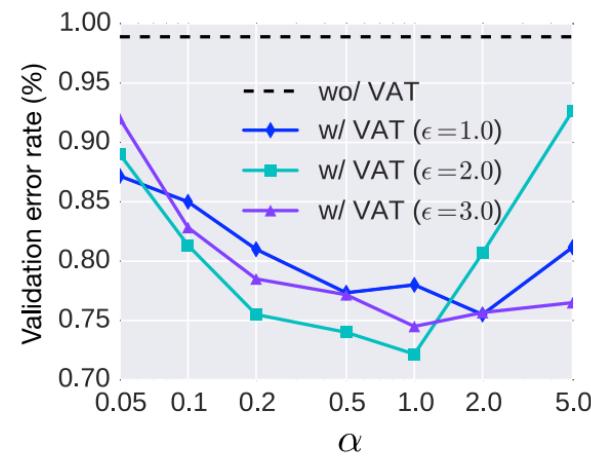
Experiments

❖ Hyperparameter ϵ, α 변화에 따른 성능 비교(SVHN, and SVHN)

- ($\alpha = 1$), ϵ 값에 따라 Validation error rate 감소했다가 증가하는 상황
- ϵ 값이 고정 후, α 값이 1일 때, 가장 Validation error rate가 작은 상황
- 두 파라미터 튜닝을 위해 많은 실험이 필요할 것으로 판단
- 대규모 Unlabeled dataset 사용 시 많은 시간이 필요할 것이라는 생각



(a) Effect of ϵ ($\alpha = 1$).



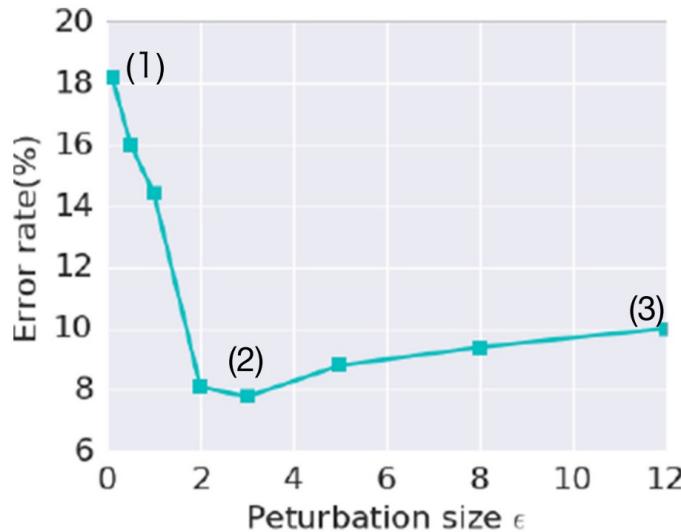
(b) Effect of α .

Experiments

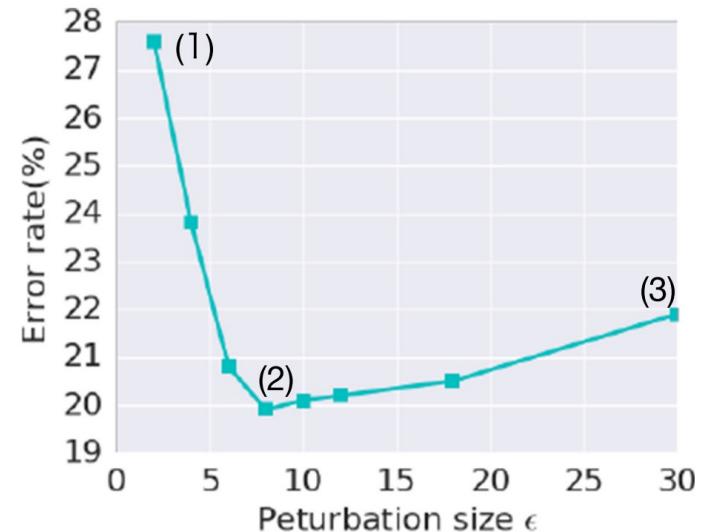
❖ Perturbation size 변화에 따른 성능 비교 (SVHN, and SVHN)

- Perturbation size에 따라 많은 노이즈가 가해지게 되며 입력 이미지에 많은 왜곡 발생
- 많은 왜곡이 발생함에 따라 Test error rate 증가
- 하지만 왜곡하지 않은 것 대비는 성능은 뛰어난 상황
- Noise에 강건해진 것으로 보이며 Label distribution이 정확하게 추론되었기 때문으로 판단

(I) Validation errors



(I) Validation errors



Experiments

❖ Perturbation size 변화에 따른 성능 비교 (SVHN, and SVHN)

- Perturbation size에 따라 많은 노이즈가 가해지게 되며 입력 이미지에 많은 왜곡 발생
- 많은 왜곡이 발생함에 따라 Test error rate 증가
- 하지만 왜곡하지 않은 것 대비는 성능은 뛰어난 상황
- Noise에 강건해진 것으로 보이며 Label distribution이 정확하게 추론되었기 때문으로 판단

(II) Virtual adversarial examples

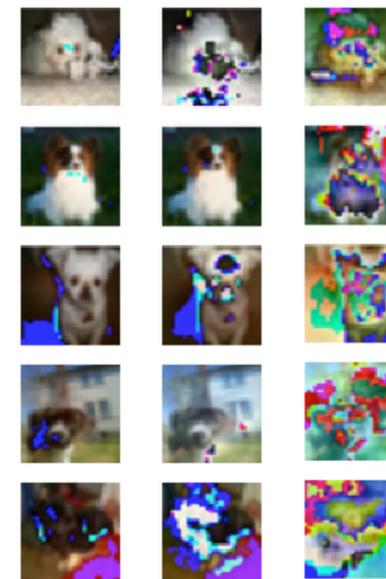
(1) $\epsilon=0.1$ (2) $\epsilon=3.0$ (3) $\epsilon=12.0$



(a) SVHN

(II) Virtual adversarial examples

(1) $\epsilon=2.0$ (2) $\epsilon=8.0$ (3) $\epsilon=30.0$



(b) CIFAR-10

Conclusion

❖ Conclusion

- Gradient를 사용해 Perturbation을 발생시켜 이미지 왜곡
- 왜곡된 이미지의 출력 값과 기존 이미지의 출력 값이 유사해지도록 학습(Regularization)
- SVHN, CIFAR-10에 대한 Semi-supervised Learning 방법론의 SOTA 달성

❖ 제안 방법론에 대한 나의 의견

- Hyperparameter tuning 시 많은 시간이 필요할 것으로 생각
- 기존 이미지에 대해 손실 함수 계산 및 Backpropagation 후 Gradient 산정이 필요하기에 시간이 오래 걸릴 것으로 판단
- 학습 시간을 줄이며 Adversarial example을 생성할 수 있을 지에 대한 의문점 발생

Thank you