UNSUPERVISED DATA AUGMENTATION FOR CONSISTENCY TRAINING

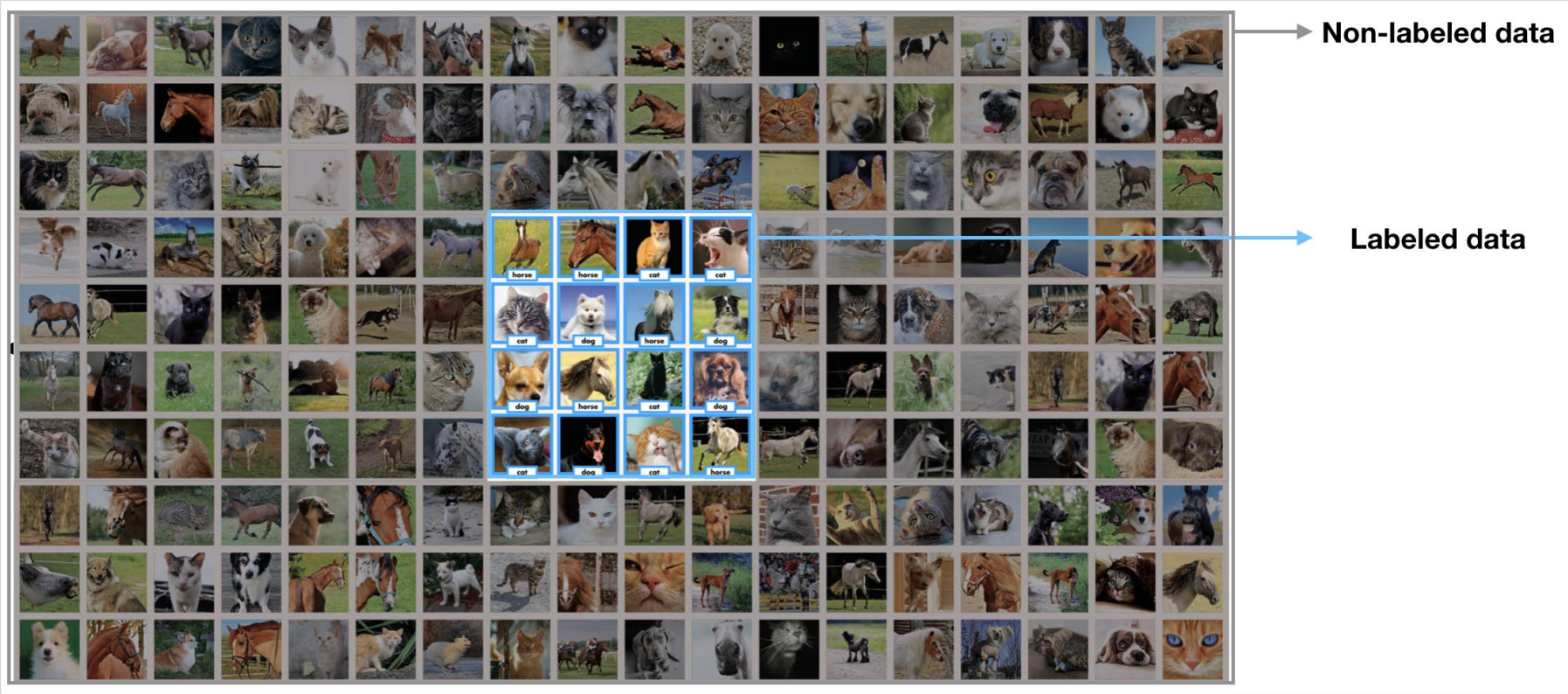
Qizhe Xie[1,2], **Zihang Dai**[1,2], **Eduard Hovy**[2], **Minh-Thang Luong**[1], **Quoc V. Le**[1]
[1] Google Research, Brain Team, [2] Carnegie Mellon University
{qizhex, dzihang, hovy}@cs.cmu.edu, {thangluong, qvl}@google.com

김 성 철

Contents

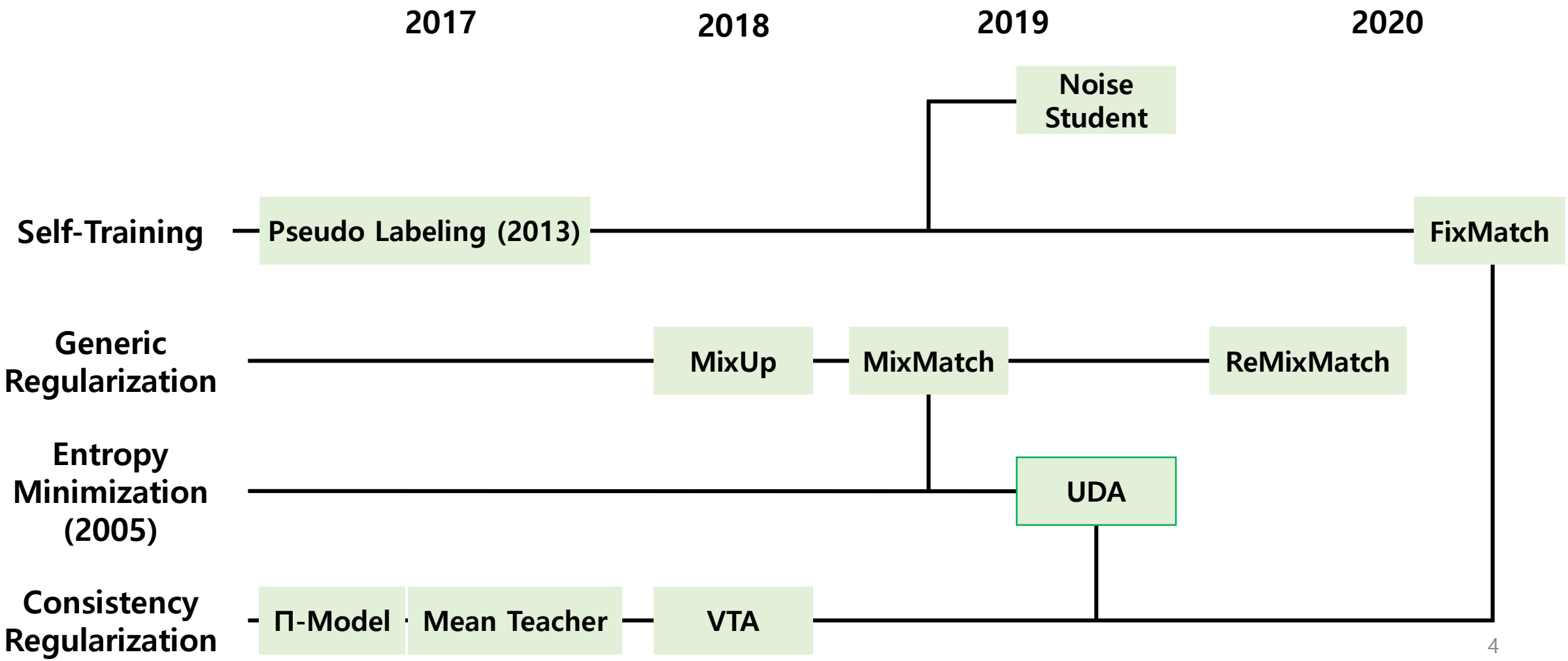
Semi-supervised learning

- **목표**
  - Labeled data가 많지 않은 상황에서 unlabeled data의 도움을 받아 성능을 높이자!



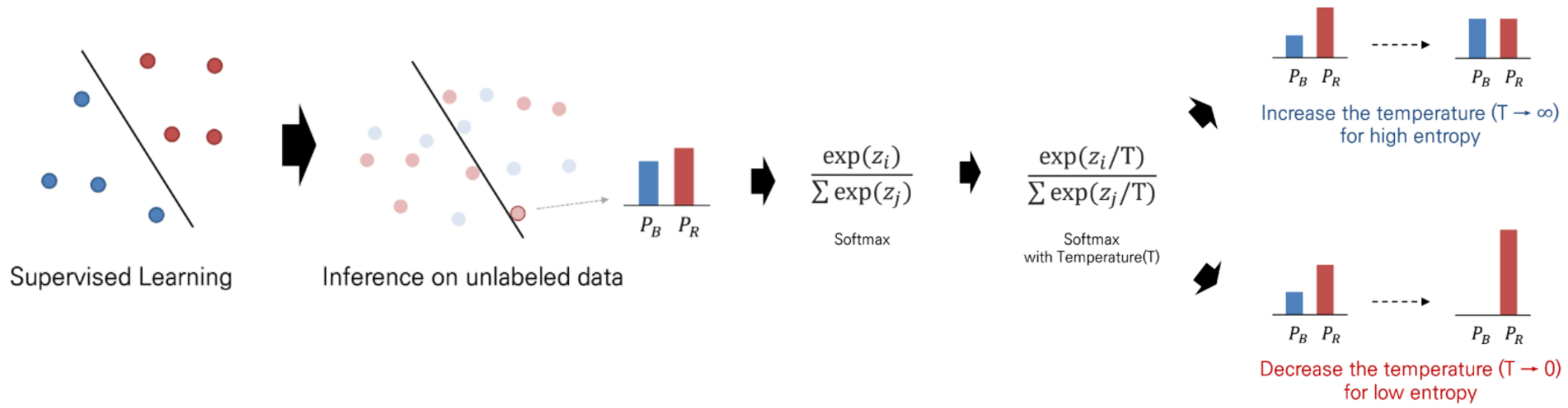
https://steemit.com/kr/@jiwoopapa/realistic-evaluation-of-semi-supervised-learning-algorithms

Semi-supervised learning



Semi-supervised learning

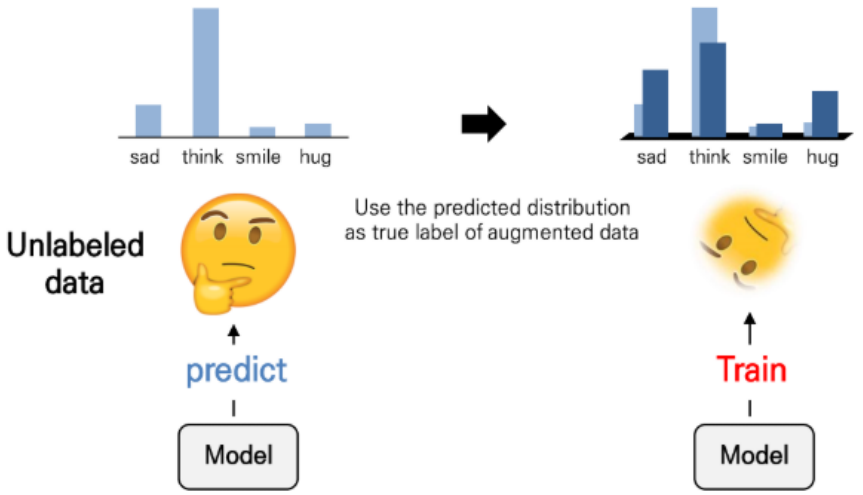
- **Entropy Minimization**
  - 예측값(softmax)의 confidenc를 높이기 위해 사용
  - 주로 softmax temperature를 이용
  - Temperature를 1보다 적게 설정할수록 entropy가 작아짐



https://papers.nips.cc/paper/2740-semi-supervised-learning-by-entropy-minimization.pdf
http://dsba.korea.ac.kr/seminar/?mod=document&uid=68
http://dsba.korea.ac.kr/seminar/?mod=document&uid=248

Semi-supervised learning

- **Consistency Regularization**
  1. 모델을 이용해 unlabeled data의 분포 예측
  2. Unlabeled data에 noise 추가 (data augmentation)
  3. 예측한 분포를 augmented data의 정답 label로 사용해 모델로 학습



http://dsba.korea.ac.kr/seminar/?mod=document&uid=248

Semi-supervised learning

- Consistency Regularization
  - Π-Model (ICLR 2017)

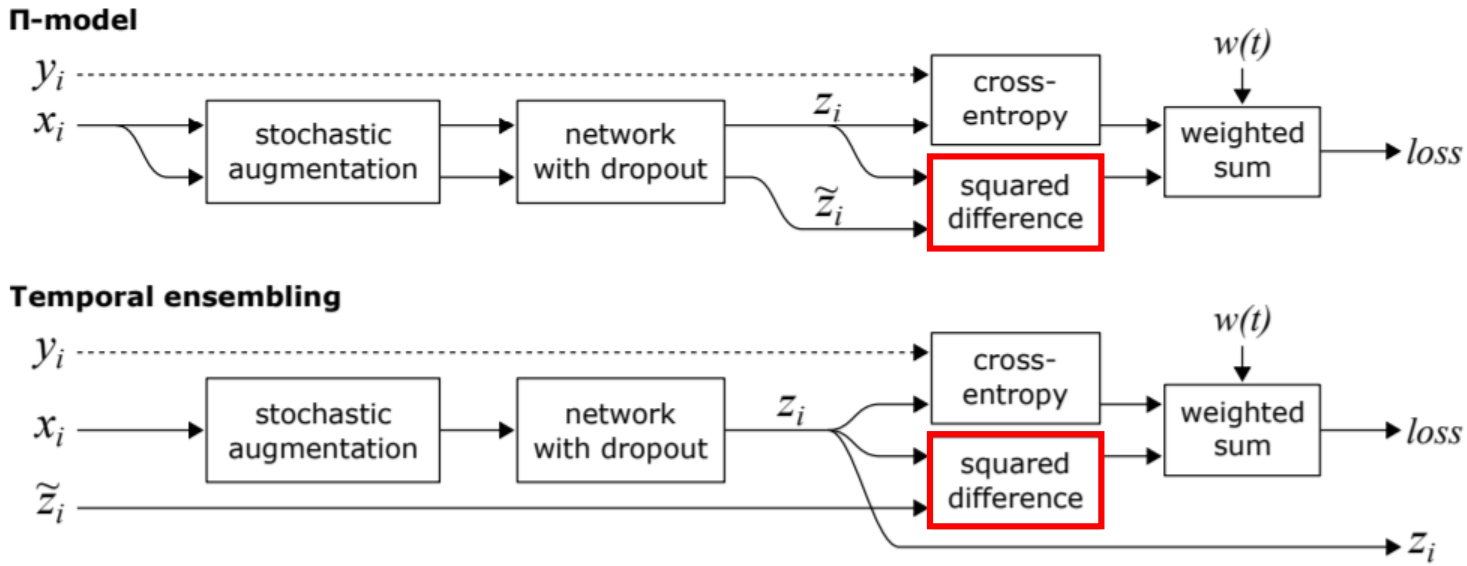
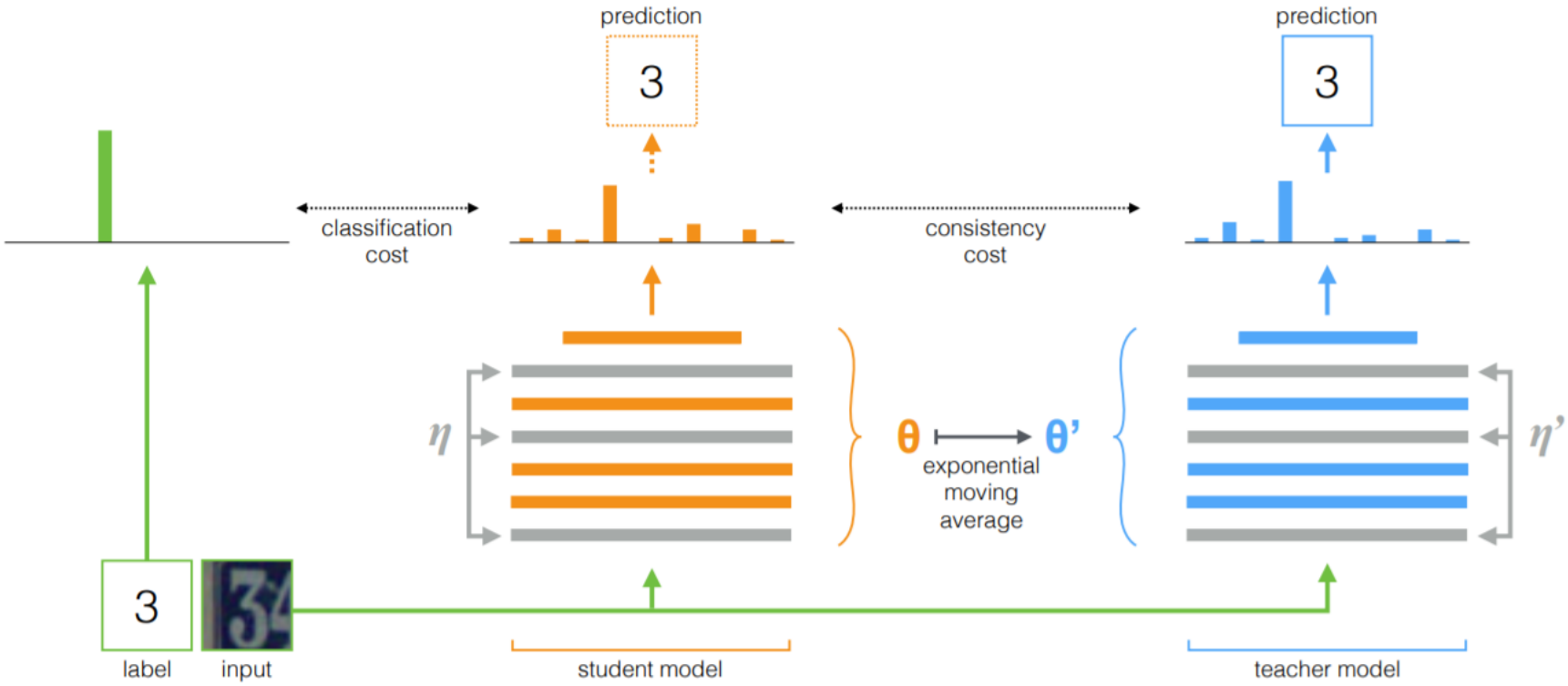


Figure 1: Structure of the training pass in our methods. Top: Π-model. Bottom: temporal ensembling. Labels y_i are available only for the labeled inputs, and the associated cross-entropy loss component is evaluated only for those.

https://arxiv.org/pdf/1610.02242.pdf

Semi-supervised learning

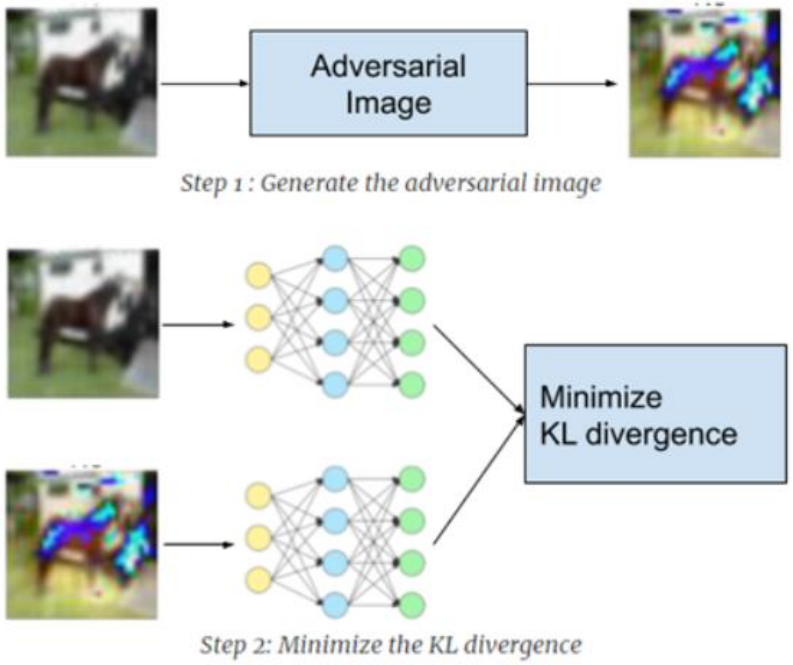
- **Consistency Regularization**
  - Mean Teacher (NIPS 2017)



https://arxiv.org/pdf/1703.01780.pdf

Semi-supervised learning

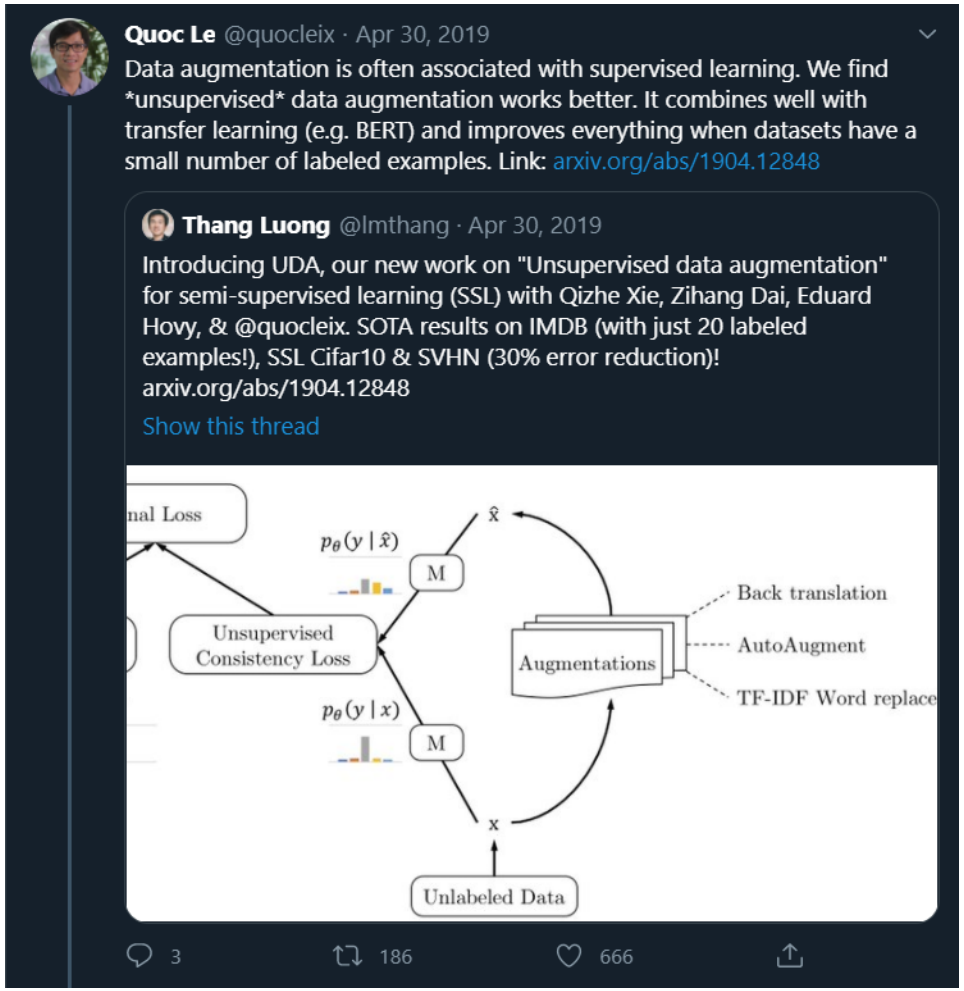
- **Consistency Regularization**
  - Virtual Adversarial Training (TPAMI 2018)



Virtual Adversarial Training (Takeru Miyato et al., 2017)

https://arxiv.org/pdf/1704.03976.pdf
https://medium.com/@kabbi159/semi-supervised-learning-%EC%A0%95%EB%A6%AC-a7ed58a8f023

Unsupervised Data Augmentation (UDA)



https://twitter.com/quocleix/status/1123103668318769152

Unsupervised Data Augmentation (UDA)

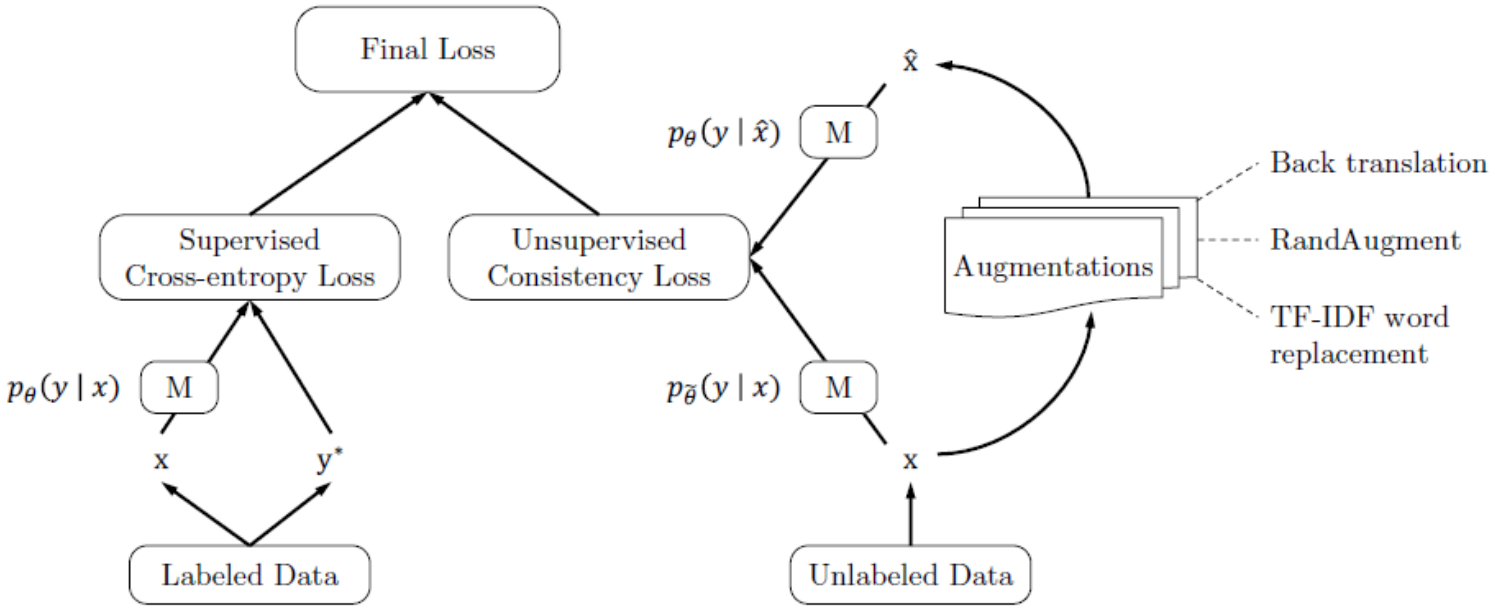


Figure 1: Training objective for UDA, where M is a model that predicts a distribution of y given x.

Unsupervised Data Augmentation (UDA)

- UDA
  - Given an input x, compute the output distribution $p_\theta(y|x)$ and a noised version $p_\theta(y|x,\epsilon)$ by injecting a small noise ϵ. The noise can be applied to x or hidden states.
  - Minimize a divergence metric between the two distributions $\mathcal{D}\big(p_\theta(y|x)||p_\theta(y|x,\epsilon)\big)$.
  - 모델을 ϵ에 대해 덜 민감하게 만들고, input (or hidden) space의 변화에 대해 smoother하게 만듬
    - Consistency loss를 최소화하는 것은 label information을 labeled examples에서 unlabeled ones로 점차 진행

$$\min_\theta \mathcal{J}(\theta) = \underbrace{\mathbb{E}_{x,y^* \in L}[-\log p_\theta(y^*|x)]}_{\text{supervised cross entropy}} + \lambda \underbrace{\mathbb{E}_{x \in U} \mathbb{E}_{\hat{x} \sim q(\hat{x}|x)} \left[\mathcal{D}_{\mathrm{KL}}\left(p_{\tilde{\theta}}(y|x)||p_\theta(y|\hat{x})\right)\right]}_{\text{unsupervised consistency training loss}}$$

- $\lambda\ (=1)$: **a weighting factor to balance** the supervised cross entropy and the unsupervised consistency training loss
- $q(\hat{x}|x)$: a data augmentation transformation
- $\tilde{\theta}$: a *fixed* copy of the current parameters θ indicating that the gradient is not propagated through $\tilde{\theta}$

Unsupervised Data Augmentation (UDA)

- UDA

$$\min_\theta \mathcal{J}(\theta) = \mathbb{E}_{x,y^* \in L}[-\log p_\theta(y^*|x)] + \lambda \mathbb{E}_{x \in U} \mathbb{E}_{\hat{x} \sim q(\hat{x}|x)} \left[\mathcal{D}_{\mathrm{KL}} \left(p_{\tilde{\theta}}(y|x) || p_\theta(y|\hat{x}) \right) \right]$$

supervised cross entropy — unsupervised consistency training loss

- $\lambda\ (= 1)$: **a weighting factor to balance the supervised cross entropy and the unsupervised consistency training loss**
- $q(\hat{x}|x)$: a data augmentation transformation
- $\tilde{\theta}$: a *fixed* copy of the current parameters θ indicating that the gradient is not propagated through $\tilde{\theta}$
- Different batch size for the supervised data and the unsupervised data
- Supervised training $(p_\theta(y|x))$과 prediction on unlabeled examples $(p_{\tilde{\theta}}(y|x))$의 discrepancy를 줄이기 위해, unlabeled examples에도 같은 augmentation 적용

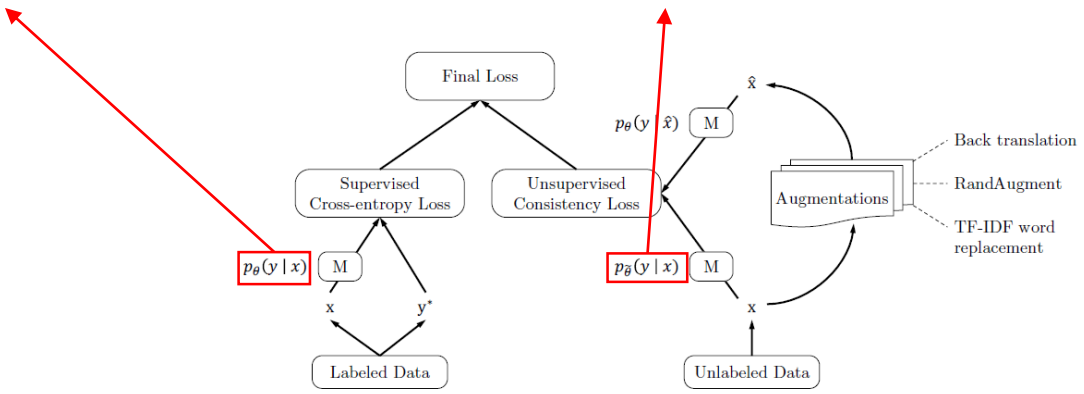


Figure 1: Training objective for UDA, where M is a model that predicts a distribution of y given x.

Unsupervised Data Augmentation (UDA)

- **Augmentation Strategies for Different Tasks**
  - RandAugment for Image Classification
    - AutoAugment와 달리 search가 필요 없음
  - Back-translation for Text Classification
  - Word replacing with TF-IDF for Text Classification

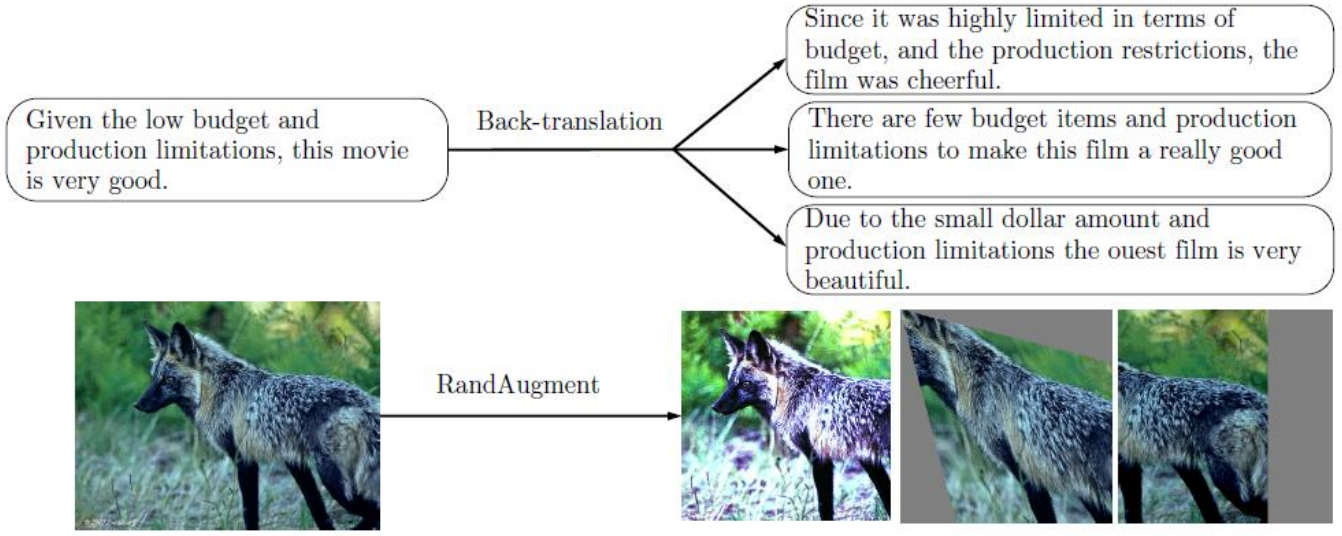


Figure 2: Augmented examples using back-translation and RandAugment.

Unsupervised Data Augmentation (UDA)

- **Training Signal Annealing for Low-data Regime**
  - Semi-supervised learning에서 unlabeled data와 labeled data의 양은 큰 차이가 있음
    - 모델의 크기가 클수록 소량의 labeled data에 overfitting되고 unlabeled data에는 underfitting되기 쉬움
  - Training Signal Annealing (TSA)
    - 학습 초반에는 정답 레이블에 대한 confidence가 정해진 threshold보다 높은 labeled data를 사용 X

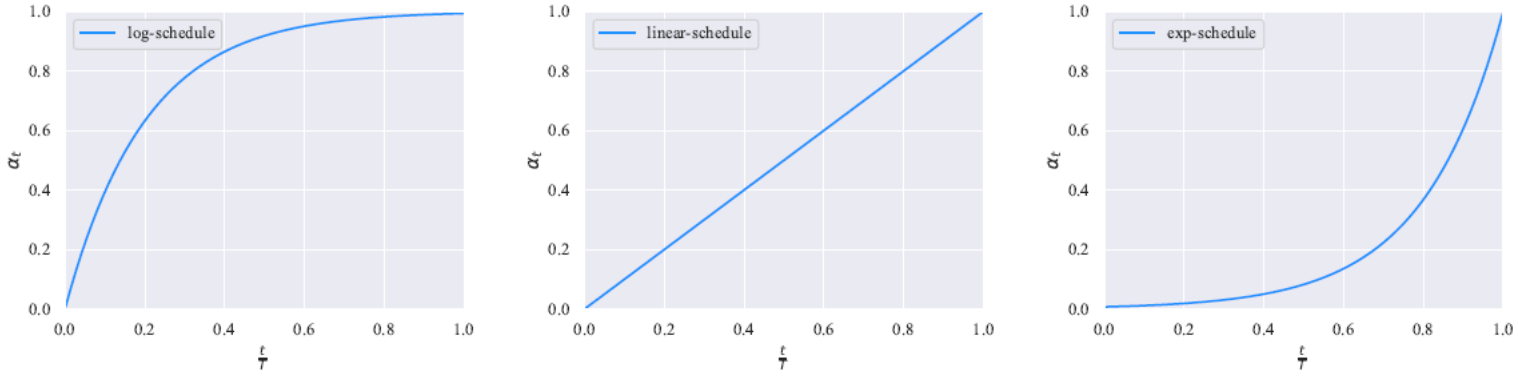


Figure 3: Three schedules of TSA. We set $\eta_t = \alpha_t * (1 - \frac{1}{K}) + \frac{1}{K}$. α_t is set to $1 - \exp(-\frac{t}{T} * 5)$, $\frac{t}{T}$ and $\exp((\frac{t}{T} - 1) * 5)$ for the log, linear and exp schedules.

Unsupervised Data Augmentation (UDA)

- Training Signal Annealing for Low-data Regime

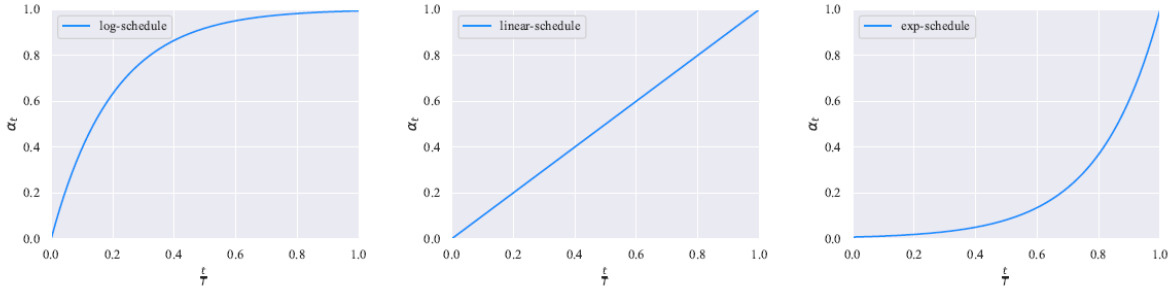
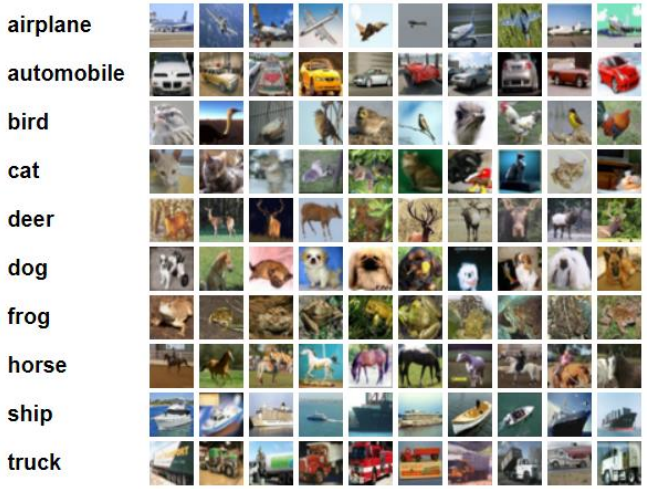


Figure 3: Three schedules of TSA. We set $\eta_t = \alpha_t * (1 - \frac{1}{K}) + \frac{1}{K}$. α_t is set to $1 - \exp(-\frac{t}{T} * 5)$, $\frac{t}{T}$ and $\exp((\frac{t}{T} - 1) * 5)$ for the log, linear and exp schedules.

- training step t에서 맞춘 category에 대한 모델의 predicted probability $p_\theta(y^*|x)$가 threshold η_t보다 높으면 loss function에서 제거
  - K : the number of categories, $(\eta_t = \alpha_t * \left(1 - \frac{1}{k}\right) + \frac{1}{K})$
  - Log-schedule $(\alpha_t = 1 - \exp\left(-\frac{t}{T} * 5\right))$: 모델이 과적합될 것 같지 않을 때 (labeled example의 수가 많거나 모델이 effective regularization 사용)
  - Linear-schedule $(\alpha_t = \frac{t}{T})$
  - Exp-schedule $(\alpha_t = \exp\left(\left(\frac{t}{T} - 1\right) * 5\right))$: 문제가 쉽거나 labeled example의 수가 제한적일 때

Experiments

- **Dataset**
  - Language
    - IMDb, Yelp-2, Yelp-5, Amazon-2, Amazon-5
    - Language dataset을 활용한 실험은 발표자료에서 제외
  - Vision
    - CIFAR-10, SVHN



CIFAR-10

SVHN (Street View House Number)

https://www.cs.toronto.edu/~kriz/cifar.html
http://ufldl.stanford.edu/housenumbers/

Experiments

- **Correlation between Supervised and Semi-supervised Performances**
  - CIFAR-10으로 augmentation별 + supervised/semi-supervised 성능 비교
    - Augmentation : Crop & Flip / Cutout / RandAugment
    - Stronger data augmentation일수록 좋음
    - Supervised에서 좋은 성능을 내면 semi-supervised에서도 좋은 성능을 보임 (Table 1)
    - Transformation의 수가 늘어날수록 대체로 좋은 성능을 보임 (Figure 6)

Augmentation (# Sup examples)	Sup (50k)	Semi-Sup (4k)
Crop & flip	5.36	16.17
Cutout	4.42	6.42
RandAugment	**4.23**	**5.29**

Table 1: Error rates on CIFAR-10.

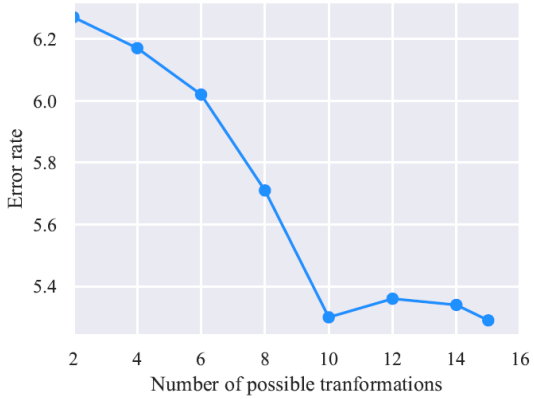
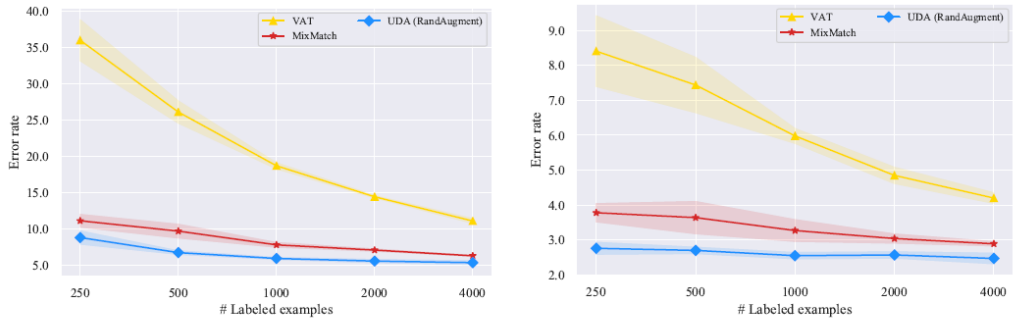


Figure 6: Error rate of UDA on CIFAR-10 with different numbers of possible transformations in RandAugment. UDA achieves lower error rate when we increase the number of possible transformations, which demonstrates the importance of a rich set of augmentation transformations.

Experiments

- **Algorithm Comparison on Vision Semi-supervised Learning Benchmarks**
  - 현재 semi-supervised learning algorithm과 비교 (CIFAR-10, SVHN)
    - **Vary the size of labeled data**
      - Wide-ResNet-28-2 + varied supervised data sizes
      - Virtual Adversarial Training (VAT)와 MixMatch와 비교
      - 모든 측면에서 UDA가 다른 알고리즘들보다 성능이 좋음
      - Noise based augmentation한 VAT에서는 real image에서는 볼 수 없는 high-frequency artifact가 포함되므로 성능 저하



(a) CIFAR-10 (b) SVHN

Figure 4: Comparison with two semi-supervised learning methods on CIFAR-10 and SVHN with varied number of labeled examples.

Experiments

- **Algorithm Comparison on Vision Semi-supervised Learning Benchmarks**
  - 현재 semi-supervised learning algorithm과 비교 (CIFAR-10, SVHN)
    - **Vary the size of labeled data**

Methods / # Sup	250	500	1,000	2,000	4,000
Pseudo-Label	49.98 ± 1.17	40.55 ± 1.70	30.91 ± 1.73	21.96 ± 0.42	16.21 ± 0.11
Π-Model	53.02 ± 2.05	41.82 ± 1.52	31.53 ± 0.98	23.07 ± 0.66	17.41 ± 0.37
Mean Teacher	47.32 ± 4.71	42.01 ± 5.86	17.32 ± 4.00	12.17 ± 0.22	10.36 ± 0.25
VAT	36.03 ± 2.82	26.11 ± 1.52	18.68 ± 0.40	14.40 ± 0.15	11.05 ± 0.31
MixMatch	11.08 ± 0.87	9.65 ± 0.94	7.75 ± 0.32	7.03 ± 0.15	6.24 ± 0.06
UDA (RandAugment)	**8.76 ± 0.90**	**6.68 ± 0.24**	**5.87 ± 0.13**	**5.51 ± 0.21**	**5.29 ± 0.25**

Table 7: Error rate (%) for CIFAR-10.

Methods / # Sup	250	500	1,000	2,000	4,000
Pseudo-Label	21.16 ± 0.88	14.35 ± 0.37	10.19 ± 0.41	7.54 ± 0.27	5.71 ± 0.07
Π-Model	17.65 ± 0.27	11.44 ± 0.39	8.60 ± 0.18	6.94 ± 0.27	5.57 ± 0.14
Mean Teacher	6.45 ± 2.43	3.82 ± 0.17	3.75 ± 0.10	3.51 ± 0.09	3.39 ± 0.11
VAT	8.41 ± 1.01	7.44 ± 0.79	5.98 ± 0.21	4.85 ± 0.23	4.20 ± 0.15
MixMatch	3.78 ± 0.26	3.64 ± 0.46	3.27 ± 0.31	3.04 ± 0.13	2.89 ± 0.06
UDA (RandAugment)	**2.76 ± 0.17**	**2.70 ± 0.09**	**2.55 ± 0.09**	**2.57 ± 0.09**	**2.47 ± 0.15**

Table 8: Error rate (%) for SVHN.

Experiments

- **Algorithm Comparison on Vision Semi-supervised Learning Benchmarks**
  - 현재 semi-supervised learning algorithm과 비교 (CIFAR-10, SVHN)
    - **Comparison with published results**

Method	Model	# Param	CIFAR-10 (4k)	SVHN (1k)
Π-Model (Laine & Aila, 2016)	Conv-Large	3.1M	12.36 ± 0.31	4.82 ± 0.17
Mean Teacher (Tarvainen & Valpola, 2017)	Conv-Large	3.1M	12.31 ± 0.28	3.95 ± 0.19
VAT + EntMin (Miyato et al., 2018)	Conv-Large	3.1M	10.55 ± 0.05	3.86 ± 0.11
SNTG (Luo et al., 2018)	Conv-Large	3.1M	10.93 ± 0.14	3.86 ± 0.27
VAdD (Park et al., 2018)	Conv-Large	3.1M	11.32 ± 0.11	4.16 ± 0.08
Fast-SWA (Athiwaratkun et al., 2018)	Conv-Large	3.1M	9.05	-
ICT (Verma et al., 2019)	Conv-Large	3.1M	7.29 ± 0.02	3.89 ± 0.04
Pseudo-Label (Lee, 2013)	WRN-28-2	1.5M	16.21 ± 0.11	7.62 ± 0.29
LGA + VAT (Jackson & Schulman, 2019)	WRN-28-2	1.5M	12.06 ± 0.19	6.58 ± 0.36
mixmixup (Hataya & Nakayama, 2019)	WRN-28-2	1.5M	10	-
ICT (Verma et al., 2019)	WRN-28-2	1.5M	7.66 ± 0.17	3.53 ± 0.07
MixMatch (Berthelot et al., 2019)	WRN-28-2	1.5M	6.24 ± 0.06	2.89 ± 0.06
Mean Teacher (Tarvainen & Valpola, 2017)	Shake-Shake	26M	6.28 ± 0.15	-
Fast-SWA (Athiwaratkun et al., 2018)	Shake-Shake	26M	5.0	-
MixMatch (Berthelot et al., 2019)	WRN	26M	4.95 ± 0.08	-
UDA (RandAugment)	WRN-28-2	1.5M	5.29 ± 0.25	**2.55 ± 0.09**
UDA (RandAugment)	Shake-Shake	26M	3.7	-
UDA (RandAugment)	PyramidNet	26M	**2.7**	-

Table 3: Comparison between methods using different models where PyramidNet is used with ShakeDrop regularization. On CIFAR-10, with only 4,000 labeled examples, UDA matches the performance of fully supervised Wide-ResNet-28-2 and PyramidNet+ShakeDrop, where they have an error rate of 5.4 and 2.7 respectively when trained on 50,000 examples without RandAugment. On SVHN, UDA also matches the performance of our fully supervised model trained on 73,257 examples without RandAugment, which has an error rate of 2.84.

Experiments

- **Scalability Test on the ImageNet Dataset**
  - ImageNet + ResNet-50
    1. **10% of the supervised data of ImageNet** while using all other data as unlabeled data
    2. All images in ImageNet as supervised data + filtering to 1.3M images from JFT → unlabeled data
       - Unlabeled data in out-of-domain은 모으기 쉽지만 in-domain data의 분포와 너무 다르면 성능을 저하시킴
         → 먼저 labeled data로 학습하고 out-of-domain data 중 높은 confidence를 갖는 데이터를 골라 unlabeled data로 사용 (정답유무는 필요 없음)
    - Supervised baseline과 비교했을 대 모든 측면에서 좋은 성능을 보임
    - Labeled data의 scale을 조절할 수 있을 뿐만 아니라, out-of-domain unlabeled data도 사용할 수 있음 (S4L, CPC와 유사한 결과)

Methods	SSL	10%	100%
ResNet-50	✗	55.09 / 77.26	77.28 / 93.73
w. RandAugment		58.84 / 80.56	78.43 / 94.37
UDA (RandAugment)	✓	**68.78 / 88.80**	**79.05 / 94.49**

Table 5: Top-1 / top-5 accuracy on ImageNet with 10% and 100% of the labeled set. We use image size 224 and 331 for the 10% and 100% experiments respectively.

Experiments

- **Ablation Studies for TSA**
  - CIFAR-10 : 4k labeled examples and 50k unlabeled examples
    - CIFAR-10에 대해 linear-schedule가 가장 좋은 성능을 보임

TSA schedule	Yelp-5	CIFAR-10
✗	50.81	5.67
log-schedule	49.06	5.67
linear-schedule	45.41	**5.29**
exp-schedule	**41.35**	7.81

Table 6: Ablation study for Training Signal Annealing (TSA) on Yelp-5 and CIFAR-10. The shown numbers are error rates.

Appendix

- **Additional Training Techniques**
  - Sharpening Predictions
    - Entropy minimization
      - Augmented data의 예측 값이 낮은 entropy를 가지도록 (=예측이 더 sharp하도록) entropy objective term을 전체 objective에 추가
    - Confidence-based masking
      - 예측의 confidence가 낮은 unlabeled data를 학습에 이용 X
    - Softmax temperature controlling
      - Unlabeled data의 예측 값을 계산할 때 1 미만의 softmax temperature를 적용, augmented 데이터의 타겟이 더 sharp하도록 함
  - Labeled 데이터가 매우 적은 경우 confidence-based masking, softmax temperature controlling이 유용, labeled 데이터가 많은 경우 entropy minimization이 효과가 있었음

https://medium.com/platfarm/unsupervised-data-augmentation-for-consistency-training-5bcd52d3f01b

감 사 합 니 다