Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results

School of Industrial and Management Engineering, Korea University

Jinyong Jeong

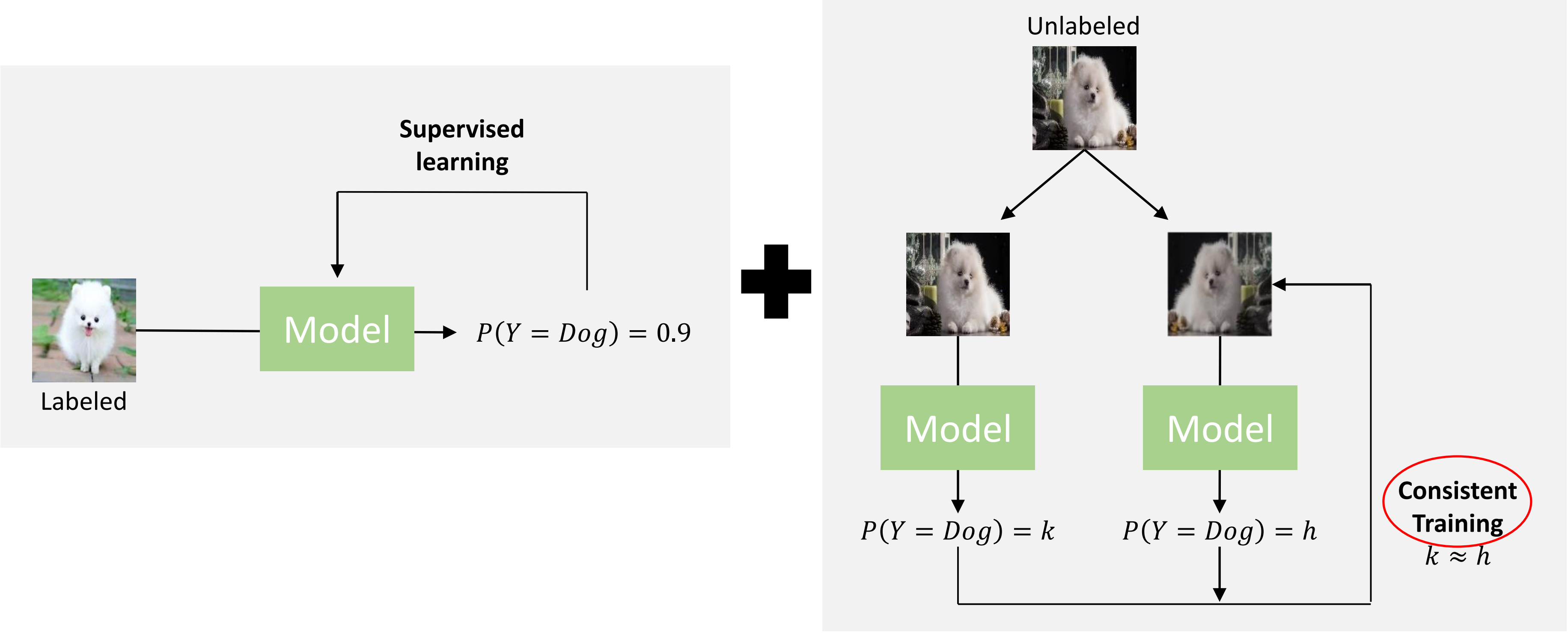
Contents

- ❖ Background
- ❖ Research Purpose
- ❖ Proposed Method
- ❖ Experiments and Results
- ❖ Conclusion

Background

❖ **Consistency regularization (Consistency training)**

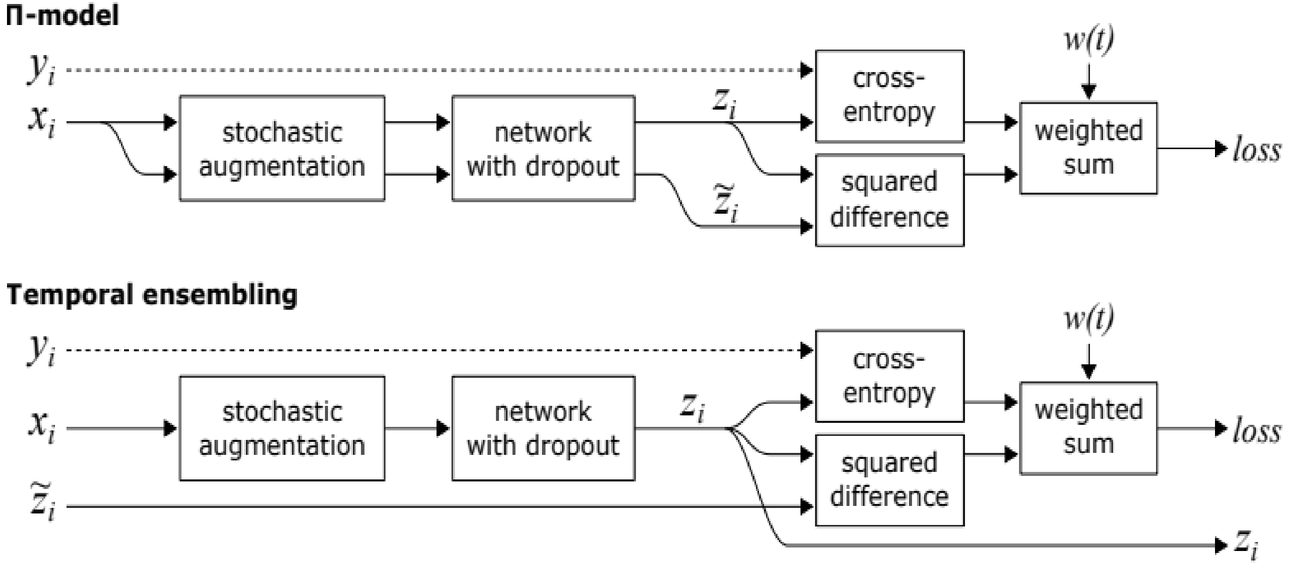
- Unlabeled data point에 작은 perturbation(e.g. augmentation, dropout)을 주어도 예측 결과는 일관성이 있을 것이라는 가정
- π-model, Temporal Ensembling, Mean Teacher



Background

❖ Consistency regularization (Consistency training)

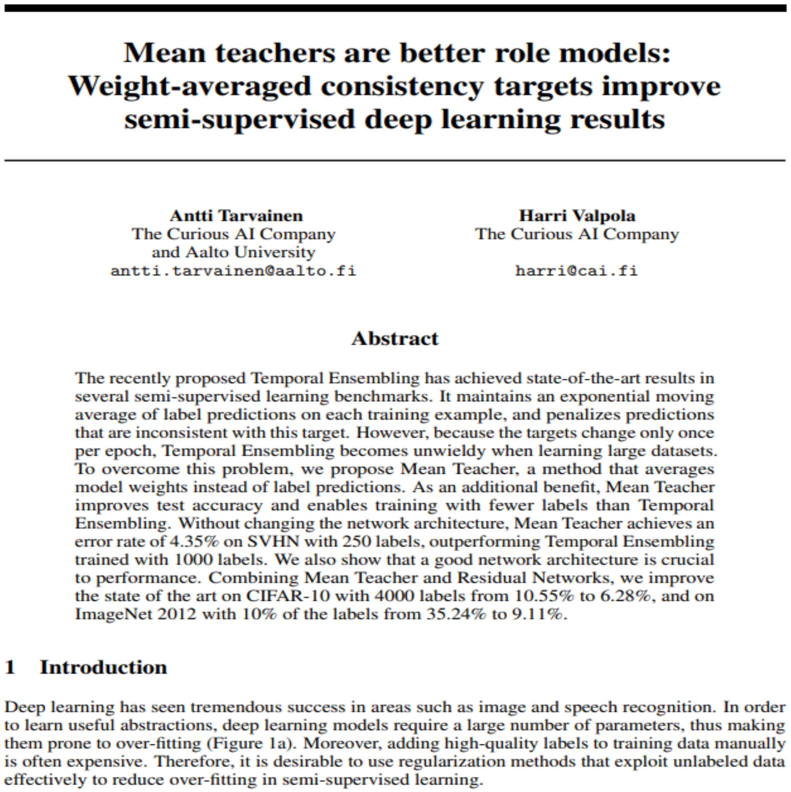
- 입력 데이터나 모델에 작은 perturbation(e.g. augmentation, dropout)을 주어도 예측 결과는 일관성이 있을 것이라는 가정
- π-model, Temporal Ensembling, Mean Teacher은 모두 Consistency training 기반 방법들임



Research Purpose

❖ **Mean teachers are better role models: weight-averaged consistency targets improve semi-supervised deep learning results** (NeurIPS, 2017)

- 2022년 08월 15일 기준 2322회 인용됨



Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results

Antti Tarvainen
The Curious AI Company
and Aalto University
antti.tarvainen@aalto.fi

Harri Valpola
The Curious AI Company
harri@cai.fi

Abstract

The recently proposed Temporal Ensembling has achieved state-of-the-art results in several semi-supervised learning benchmarks. It maintains an exponential moving average of label predictions on each training example, and penalizes predictions that are inconsistent with this target. However, because the targets change only once per epoch, Temporal Ensembling becomes unwieldy when learning large datasets. To overcome this problem, we propose Mean Teacher, a method that averages model weights instead of label predictions. As an additional benefit, Mean Teacher improves test accuracy and enables training with fewer labels than Temporal Ensembling. Without changing the network architecture, Mean Teacher achieves an error rate of 4.35% on SVHN with 250 labels, outperforming Temporal Ensembling trained with 1000 labels. We also show that a good network architecture is crucial to performance. Combining Mean Teacher and Residual Networks, we improve the state of the art on CIFAR-10 with 4000 labels from 10.55% to 6.28%, and on ImageNet 2012 with 10% of the labels from 35.24% to 9.11%.

1 Introduction

Deep learning has seen tremendous success in areas such as image and speech recognition. In order to learn useful abstractions, deep learning models require a large number of parameters, thus making them prone to over-fitting (Figure 1a). Moreover, adding high-quality labels to training data manually is often expensive. Therefore, it is desirable to use regularization methods that exploit unlabeled data effectively to reduce over-fitting in semi-supervised learning.

Research Purpose

❖ **Mean teachers are better role models: weight-averaged consistency targets improve semi-supervised deep learning results** (NeurIPS, 2017)

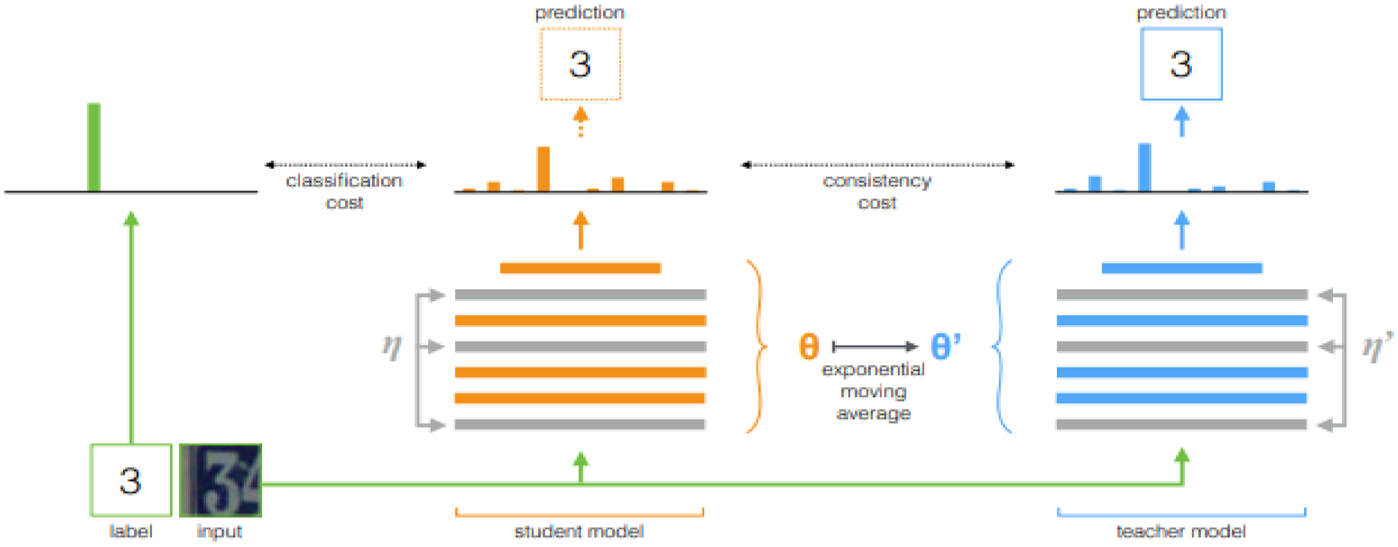
- 기존 SOTA 모델인 Temporal Ensembling 문제점
  - ✓ 학습 과정에서 얻은 앙상블 예측 값을 unlabeled data의 training target으로 사용하는 방식
  - ✓ 그러나 target이 epoch마다 한번 업데이트하여 학습 속도가 느리다는 단점이 존재
  - ✓ On-line learning에는 부적합함

- 본 논문에서는 Temporal Ensembling의 한계점을 보완하기 위해 Mean Teacher 방법 제안
  - ✓ Single network(parameter sharing)를 사용하는 Temporal Ensembling과는 다르게 teacher model과 student model을 사용함
  - ✓ Target을 step(iteration)마다 업데이트하여 학습 속도를 개선

Proposed Method

❖ Mean Teacher

- Labeled, unlabeled 데이터가 student model과 teacher model를 거침
- Student와 teacher model에 각각 noise(dropout) η과 η'를 적용
- Classification cost와 consistency cost 두가지 loss 사용

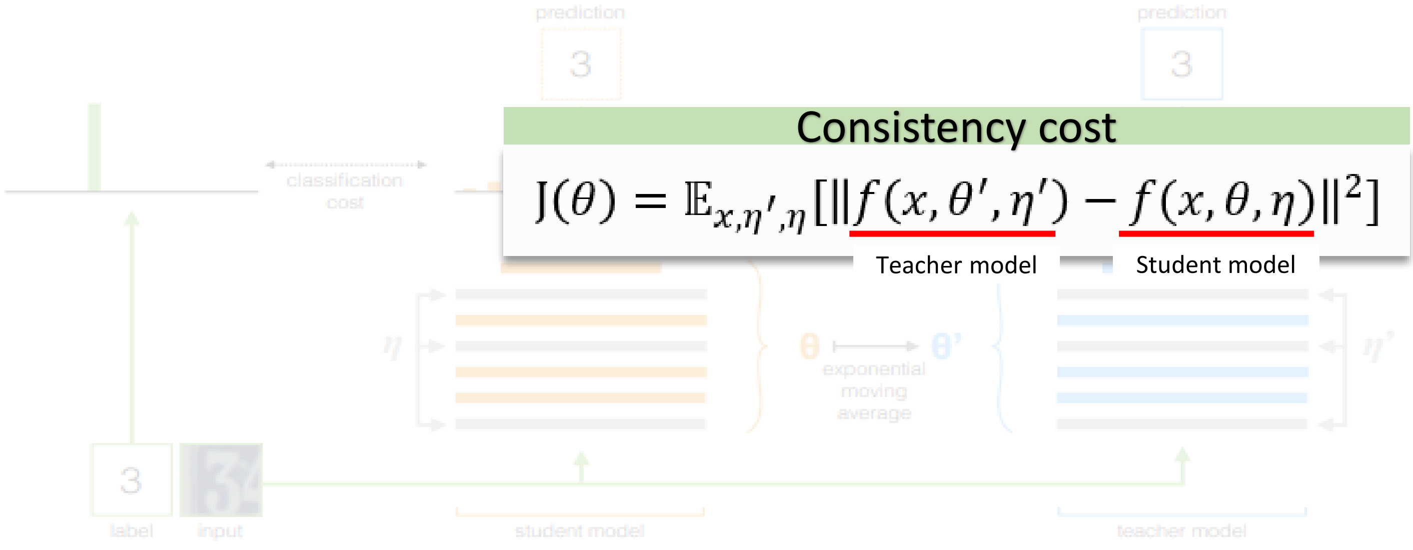


[Mean Teacher]

Proposed Method

❖ Mean Teacher – Loss

- Classification cost(Cross-entropy loss)는 labeled data에 대해서만 계산
- Consistency cost(MSE loss)는 labeled, unlabeled data에 대해서 계산
- 두 가지 Loss는 weighted sum
  - ✓ Consistency cost의 가중치에 0부터 ramp-up 방식 적용



Consistency cost

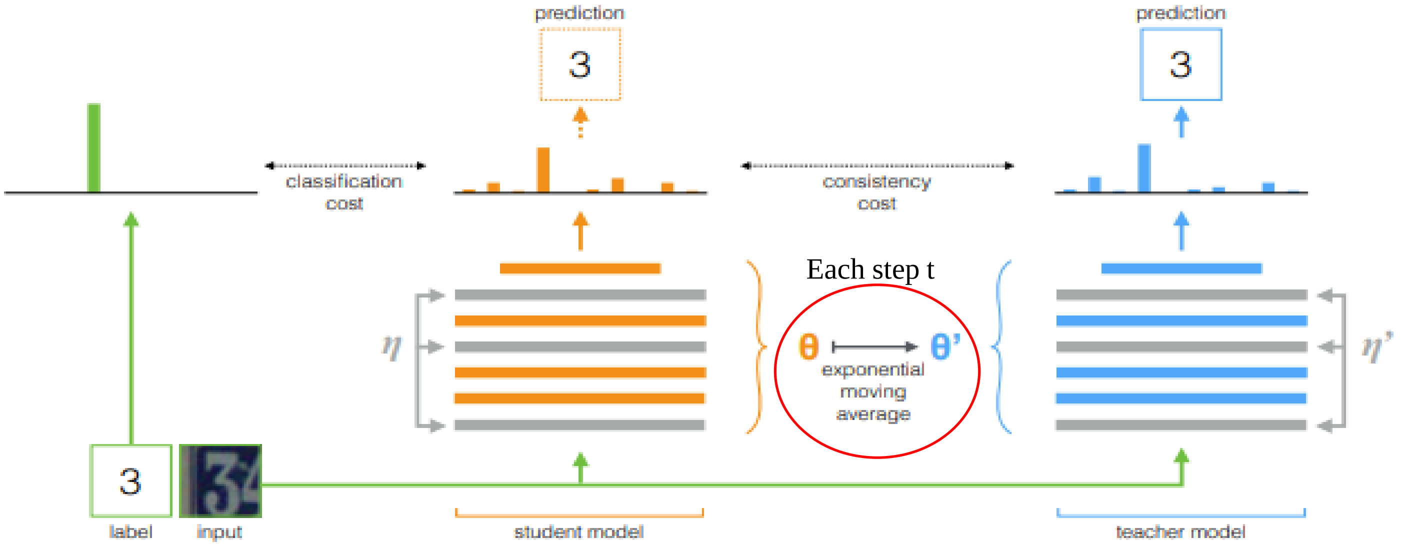
$$J(\theta) = \mathbb{E}_{x,\eta',\eta}[\|\underbrace{f(x,\theta',\eta')}_{\text{Teacher model}} - \underbrace{f(x,\theta,\eta)}_{\text{Student model}}\|^2]$$

[Mean Teacher]

Proposed Method

❖ Mean Teacher – Weight update process

- Student model 가중치는 gradient descent로 업데이트 함
- student model 가중치의 exponential moving average를 사용하여 Teacher model 가중치 업데이트
  - ✓ $\theta'_t = \alpha\theta'_{t-1} + (1-\alpha)\theta_t$
- Training step t마다 업데이트 시켜주기 때문에 teacher/student 사이에 빠른 피드백을 줌
- Mean teacher방법은 Temporal Ensembling과 다르게 large dataset, on-line learning에 적합함



[Mean Teacher]

Experiments and Results

❖ SVHN dataset

- Architecture : 13-layer ConvNet
- Regularizations : random translations, gaussian noise, dropout
- Ramp-up : from 0 to its final value during the first 80 epochs
- Consistency cost : MSE loss

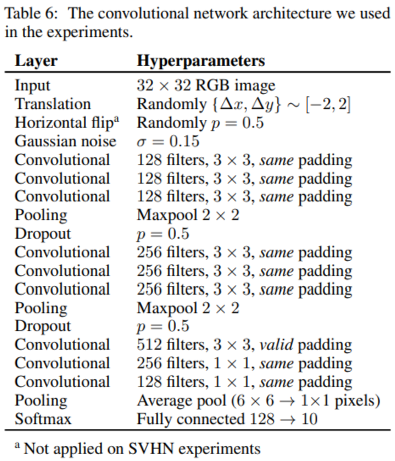


Table 6: The convolutional network architecture we used in the experiments.

Layer	Hyperparameters
Input	32×32 RGB image
Translation	Randomly $\{\Delta x, \Delta y\} \sim [-2, 2]$
Horizontal flip[a]	Randomly $p = 0.5$
Gaussian noise	$\sigma = 0.15$
Convolutional	128 filters, 3×3, *same* padding
Convolutional	128 filters, 3×3, *same* padding
Convolutional	128 filters, 3×3, *same* padding
Pooling	Maxpool 2×2
Dropout	$p = 0.5$
Convolutional	256 filters, 3×3, *same* padding
Convolutional	256 filters, 3×3, *same* padding
Convolutional	256 filters, 3×3, *same* padding
Pooling	Maxpool 2×2
Dropout	$p = 0.5$
Convolutional	512 filters, 3×3, *valid* padding
Convolutional	256 filters, 1×1, *same* padding
Convolutional	128 filters, 1×1, *same* padding
Pooling	Average pool ($6 \times 6 \to 1 \times 1$ pixels)
Softmax	Fully connected $128 \to 10$

[a] Not applied on SVHN experiments

[13-layer ConvNet]

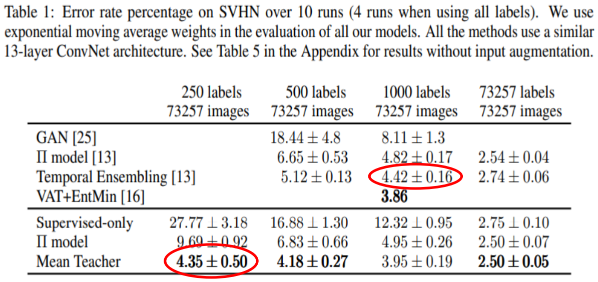


Table 1: Error rate percentage on SVHN over 10 runs (4 runs when using all labels). We use exponential moving average weights in the evaluation of all our models. All the methods use a similar 13-layer ConvNet architecture. See Table 5 in the Appendix for results without input augmentation.

	250 labels 73257 images	500 labels 73257 images	1000 labels 73257 images	73257 labels 73257 images
GAN [25]		18.44 ± 4.8	8.11 ± 1.3	
Π model [13]		6.65 ± 0.53	4.82 ± 0.17	2.54 ± 0.04
Temporal Ensembling [13]		5.12 ± 0.13	4.42 ± 0.16	2.74 ± 0.06
VAT+EntMin [16]			**3.86**	
Supervised-only	27.77 ⊥ 3.18	16.88 ⊥ 1.30	12.32 ⊥ 0.95	2.75 ⊥ 0.10
Π model	9.69 ± 0.92	6.83 ± 0.66	4.95 ± 0.26	2.50 ± 0.07
Mean Teacher	**4.35 ± 0.50**	**4.18 ± 0.27**	3.95 ± 0.19	**2.50 ± 0.05**

[W/ input augmentation]

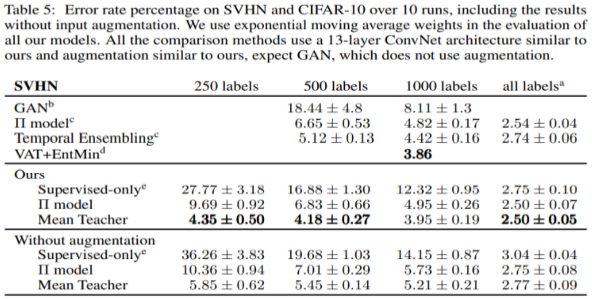


Table 5: Error rate percentage on SVHN and CIFAR-10 over 10 runs, including the results without input augmentation. We use exponential moving average weights in the evaluation of all our models. All the comparison methods use a 13-layer ConvNet architecture similar to ours and augmentation similar to ours, expect GAN, which does not use augmentation.

SVHN	250 labels	500 labels	1000 labels	all labels[a]
GAN[b]		18.44 ± 4.8	8.11 ± 1.3	
Π model[c]		6.65 ± 0.53	4.82 ± 0.17	2.54 ± 0.04
Temporal Ensembling[c]		5.12 ± 0.13	4.42 ± 0.16	2.74 ± 0.06
VAT+EntMin[d]			**3.86**	
Ours				
Supervised-only[e]	27.77 ± 3.18	16.88 ± 1.30	12.32 ± 0.95	2.75 ± 0.10
Π model	9.69 ± 0.92	6.83 ± 0.66	4.95 ± 0.26	2.50 ± 0.07
Mean Teacher	**4.35 ± 0.50**	**4.18 ± 0.27**	3.95 ± 0.19	**2.50 ± 0.05**
Without augmentation				
Supervised-only[e]	36.26 ± 3.83	19.68 ± 1.03	14.15 ± 0.87	3.04 ± 0.04
Π model	10.36 ± 0.94	7.01 ± 0.29	5.73 ± 0.16	2.75 ± 0.08
Mean Teacher	5.85 ± 0.62	5.45 ± 0.14	5.21 ± 0.21	2.77 ± 0.09

[W/O input augmentation]

✓ Mean Teacher의 경우 적은 labeled data(250 labels)를 사용하여도 기존 SOTA 모델인 Temporal ensembling(1000 labels) 보다 좋은 성능을 보임

Experiments and Results

❖ SVHN dataset with extra unlabeled training data

- Architecture : 13-layer ConvNet
- Regularizations : random translations, gaussian noise, dropout
- Ramp-up : from 0 to its final value during the first 80 epochs
- Consistency cost : MSE loss
- Unlabeled data를 각각 10만장, 20만장 추가 사용하여 실험 진행

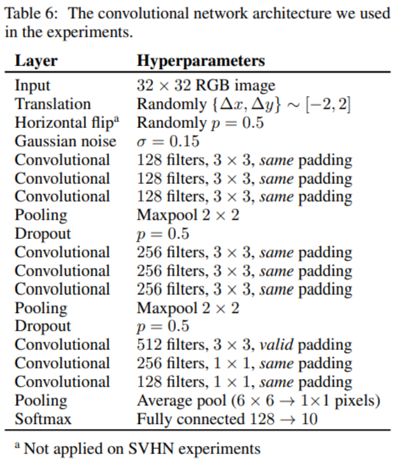


Table 6: The convolutional network architecture we used in the experiments.

Layer	**Hyperparameters**
Input	32 × 32 RGB image
Translation	Randomly $\{\Delta x, \Delta y\} \sim [-2, 2]$
Horizontal flip[a]	Randomly $p = 0.5$
Gaussian noise	$\sigma = 0.15$
Convolutional	128 filters, 3 × 3, *same* padding
Convolutional	128 filters, 3 × 3, *same* padding
Convolutional	128 filters, 3 × 3, *same* padding
Pooling	Maxpool 2 × 2
Dropout	$p = 0.5$
Convolutional	256 filters, 3 × 3, *same* padding
Convolutional	256 filters, 3 × 3, *same* padding
Convolutional	256 filters, 3 × 3, *same* padding
Pooling	Maxpool 2 × 2
Dropout	$p = 0.5$
Convolutional	512 filters, 3 × 3, *valid* padding
Convolutional	256 filters, 1 × 1, *same* padding
Convolutional	128 filters, 1 × 1, *same* padding
Pooling	Average pool (6 × 6 → 1×1 pixels)
Softmax	Fully connected 128 → 10

[a] Not applied on SVHN experiments

[13-layer ConvNet]

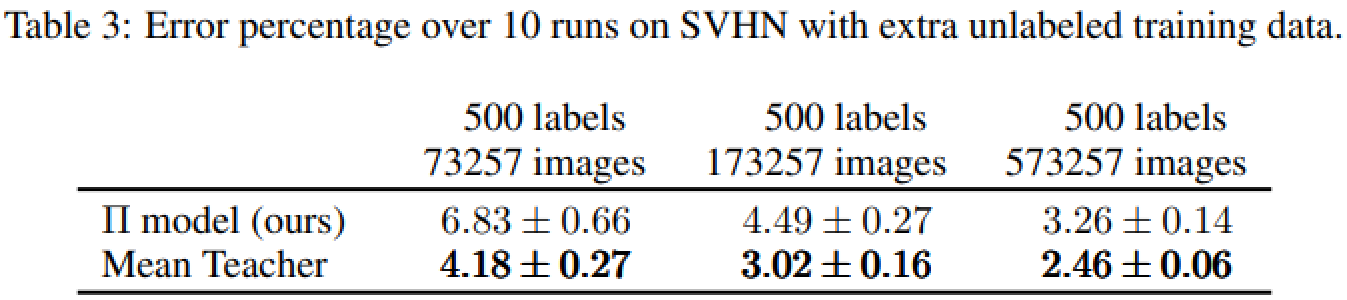


Table 3: Error percentage over 10 runs on SVHN with extra unlabeled training data.

	500 labels 73257 images	500 labels 173257 images	500 labels 573257 images
Π model (ours)	6.83 ± 0.66	4.49 ± 0.27	3.26 ± 0.14
Mean Teacher	$\mathbf{4.18 \pm 0.27}$	$\mathbf{3.02 \pm 0.16}$	$\mathbf{2.46 \pm 0.06}$

✓ Unlabeled data를 추가로 사용할수록 Mean Teacher 모델 성능이 향상되는 것을 확인

Experiments and Results

❖ CIFAR-10 dataset

- Architecture : 13-layer ConvNet, 12-block(26-layer) ResNet with Shake-Shake regularization
- Regularizations : random translations, gaussian noise, dropout, horizontal flip
- Ramp-up : from 0 to its final value during the first 80 epochs
- Consistency cost : MSE loss

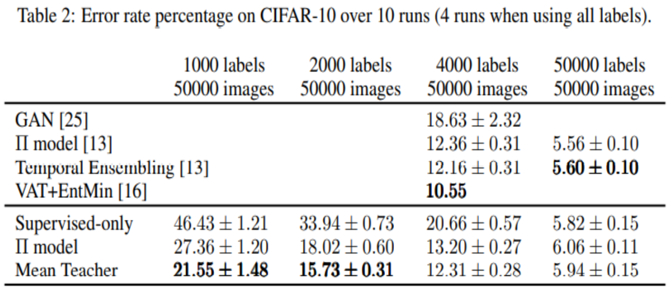
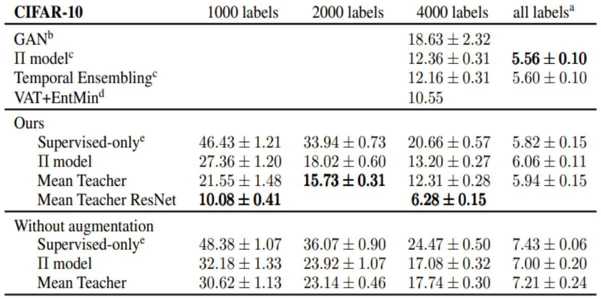


Table 2: Error rate percentage on CIFAR-10 over 10 runs (4 runs when using all labels).

	1000 labels 50000 images	2000 labels 50000 images	4000 labels 50000 images	50000 labels 50000 images
GAN [25]			18.63 ± 2.32	
Π model [13]			12.36 ± 0.31	5.56 ± 0.10
Temporal Ensembling [13]			12.16 ± 0.31	**5.60 ± 0.10**
VAT+EntMin [16]			**10.55**	
Supervised-only	46.43 ± 1.21	33.94 ± 0.73	20.66 ± 0.57	5.82 ± 0.15
Π model	27.36 ± 1.20	18.02 ± 0.60	13.20 ± 0.27	6.06 ± 0.11
Mean Teacher	**21.55 ± 1.48**	**15.73 ± 0.31**	12.31 ± 0.28	5.94 ± 0.15



CIFAR-10	1000 labels	2000 labels	4000 labels	all labels[a]
GAN[b]			18.63 ± 2.32	
Π model[c]			12.36 ± 0.31	**5.56 ± 0.10**
Temporal Ensembling[c]			12.16 ± 0.31	5.60 ± 0.10
VAT+EntMin[d]			10.55	
Ours				
Supervised-only[e]	46.43 ± 1.21	33.94 ± 0.73	20.66 ± 0.57	5.82 ± 0.15
Π model	27.36 ± 1.20	18.02 ± 0.60	13.20 ± 0.27	6.06 ± 0.11
Mean Teacher	21.55 ± 1.48	**15.73 ± 0.31**	12.31 ± 0.28	5.94 ± 0.15
Mean Teacher ResNet	**10.08 ± 0.41**		**6.28 ± 0.15**	
Without augmentation				
Supervised-only[e]	48.38 ± 1.07	36.07 ± 0.90	24.47 ± 0.50	7.43 ± 0.06
Π model	32.18 ± 1.33	23.92 ± 1.07	17.08 ± 0.32	7.00 ± 0.20
Mean Teacher	30.62 ± 1.13	23.14 ± 0.46	17.74 ± 0.30	7.21 ± 0.24

[W/ input augmentation] **[W/O input augmentation]**

✓ Mean Teacher의 architecture를 ResNet으로 설정했을 때, 더 좋은 성능을 보임

Experiments and Results

❖ CIFAR-10 dataset & ImageNet 2012 (10% of the labels)

- CIFAR-10 Architecture : 12-block(26-layer) ResNet with Shake-Shake regularization
- ImageNet Architecture : 50-block(152-layer) ResNeXt
- Regularizations : random translations, gaussian noise, dropout, horizontal flip
- Ramp-up : from 0 to its final value during the first 80 epochs
- Consistency cost : MSE loss

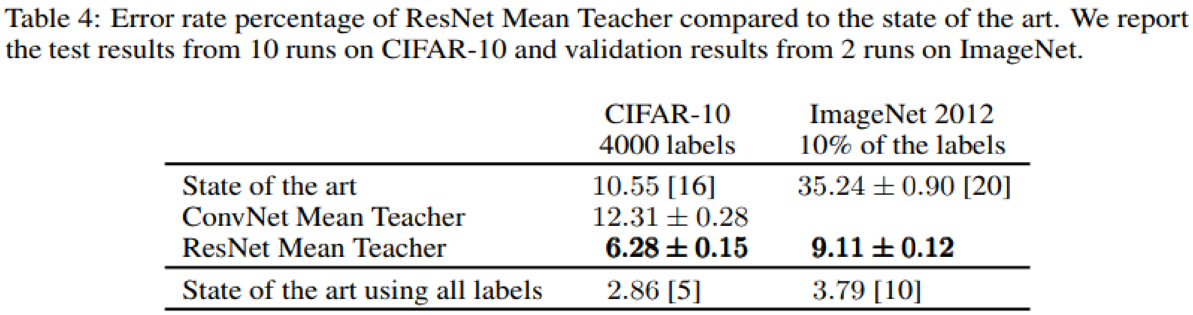


Table 4: Error rate percentage of ResNet Mean Teacher compared to the state of the art. We report the test results from 10 runs on CIFAR-10 and validation results from 2 runs on ImageNet.

	CIFAR-10 4000 labels	ImageNet 2012 10% of the labels
State of the art	10.55 [16]	35.24 ± 0.90 [20]
ConvNet Mean Teacher	12.31 ± 0.28	
ResNet Mean Teacher	**6.28 ± 0.15**	**9.11 ± 0.12**
State of the art using all labels	2.86 [5]	3.79 [10]

✓ Mean Teacher의 architecture를 ResNet 기반으로 설정했을 때, 기존 SOTA 성능보다 좋은 성능을 보임

Conclusion

❖ Mean Teacher

- 본 논문에서는 Temporal Ensembling의 단점을 개선시킨 Mean Teacher를 제안함
  - ✓ Student model의 가중치에 exponential moving average를 적용하여 업데이트
  - ✓ Epoch마다 update → Step마다 update
  - ✓ Model간 빠른 피드백을 통해 학습 속도를 개선함
- Large dataset, on-line learning에 적합한 방법임

Thank You