Semi-Supervised Learning with Meta-Gradient

Xin-Yu Zhang, Taihong Xiao, Haolin Jia, Ming-Ming Cheng, Ming-Hsuan Yang

xinyuzhang@mail.nankai.edu.cn

Semi-Supervised Learning

Semi-supervised learning (SSL): labeled data + unlabeled data $\Longrightarrow$ better generalization ability.

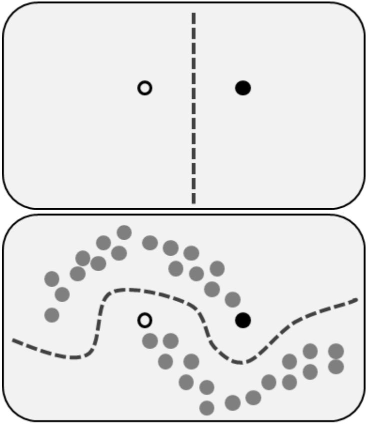

Figure: Illustration of the role of unlabeled data[1].

[1] https://en.wikipedia.org/wiki/Semi-supervised_learning

Consistency-Based SSL

Basic assumption: prediction consistency against perturbations of the input signals or model weights.

[2] Miyato *et al.* Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. In *TPAMI*, 2018

[3] Tarvainen & Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017

Consistency-Based SSL

Basic assumption: prediction consistency against perturbations of the input signals or model weights.

Two research directions for consistency-based SSL: **perturbing in the adversarial directions**[2] and **finding better "role models"**[3].

[2] Miyato *et al.* Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. In *TPAMI*, 2018

[3] Tarvainen & Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017

Consistency-Based SSL

Basic assumption: prediction consistency against perturbations of the input signals or model weights.

Two research directions for consistency-based SSL: **perturbing in the adversarial directions**[2] and **finding better "role models"**[3].

Problems: the consistency loss is rather generic and does not fully exploit the label information (not designed for the specific task).

[2] Miyato *et al.* Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. In *TPAMI*, 2018

[3] Tarvainen & Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017

Consistency-Based SSL

Basic assumption: prediction consistency against perturbations of the input signals or model weights.

Two research directions for consistency-based SSL: **perturbing in the adversarial directions**[2] and **finding better "role models"**[3].

Problems: the consistency loss is rather generic and does not fully exploit the label information (not designed for the specific task).

Our work is to bridge the gap between the consistency loss and the label information.

[2] Miyato *et al.* Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. In *TPAMI*, 2018

[3] Tarvainen & Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017

Overview

Challenge. The consistency loss, which is calculated from the unlabeled data, seems to have no relationship with the labeled data.

Challenge. The consistency loss, which is calculated from the unlabeled data, seems to have no relationship with the labeled data.

Solution. Magic of *meta-learning*: unfolding and differentiating through one SGD step.

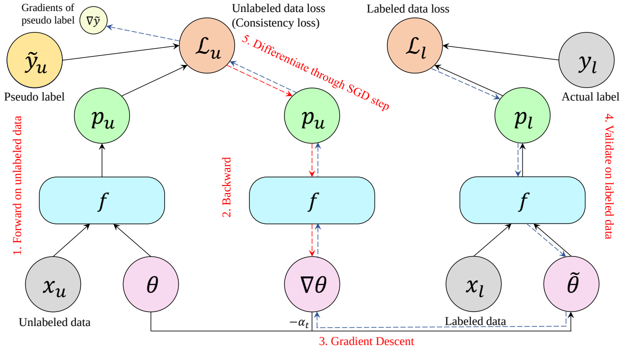

Figure: Illustration of the meta-learning philosophy.

Derivation

Formulation:

$$
\begin{aligned}
&\min_{\mathcal{Y}} \sum_{k=1}^{N^l} \mathcal{L}(\boldsymbol{x}_k^l, \boldsymbol{y}_k; \boldsymbol{\theta}^*(\mathcal{Y})) \\
&\text{s.t. } \boldsymbol{\theta}^*(\mathcal{Y}) = \arg\min_{\theta} \sum_{i=1}^{N^u} \mathcal{L}(\boldsymbol{x}_i^u, \widehat{\boldsymbol{y}}_i; \boldsymbol{\theta}).
\end{aligned}
\tag{1}
$$

Derivation

Formulation:

$$\begin{aligned} &\min_{\mathcal{Y}} \sum_{k=1}^{N^l} \mathcal{L}(\boldsymbol{x}_k^l, \boldsymbol{y}_k; \boldsymbol{\theta}^*(\mathcal{Y})) \\ &\text{s.t. } \boldsymbol{\theta}^*(\mathcal{Y}) = \arg\min_{\theta} \sum_{i=1}^{N^u} \mathcal{L}(\boldsymbol{x}_i^u, \widehat{\boldsymbol{y}}_i; \boldsymbol{\theta}). \end{aligned} \tag{1}$$

Solving Eq. (1) exactly is impossible, we adopt online approximation on the batch level.

Derivation

Initialize pseudo-labels:

$$\widetilde{\boldsymbol{y}}_i = f(\boldsymbol{x}_i^u; \boldsymbol{\theta}_t). \tag{2}$$

Derivation

Initialize pseudo-labels:

$$\widetilde{\boldsymbol{y}}_i = f(\boldsymbol{x}_i^u; \boldsymbol{\theta}_t). \tag{2}$$

Compute the unlabeled data loss and back-propagate the gradients:

$$\begin{aligned} \mathcal{L}(\boldsymbol{x}_i^u, \widetilde{\boldsymbol{y}}_i; \boldsymbol{\theta}_t) &= \Phi(f(\boldsymbol{x}_i^u; \boldsymbol{\theta}_t), \widetilde{\boldsymbol{y}}_i), \\ \nabla\boldsymbol{\theta}_t &= \frac{1}{B^u}\sum_{i=1}^{B^u} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}_i^u, \widetilde{\boldsymbol{y}}_i; \boldsymbol{\theta}_t). \end{aligned} \tag{3}$$

Derivation

Initialize pseudo-labels:

$$\widetilde{\boldsymbol{y}}_i = f(\boldsymbol{x}_i^u; \boldsymbol{\theta}_t). \tag{2}$$

Compute the unlabeled data loss and back-propagate the gradients:

$$\begin{aligned}\mathcal{L}(\boldsymbol{x}_i^u, \widetilde{\boldsymbol{y}}_i; \boldsymbol{\theta}_t) &= \Phi(f(\boldsymbol{x}_i^u; \boldsymbol{\theta}_t), \widetilde{\boldsymbol{y}}_i), \\ \nabla\boldsymbol{\theta}_t &= \frac{1}{B^u}\sum_{i=1}^{B^u} \nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{x}_i^u, \widetilde{\boldsymbol{y}}_i; \boldsymbol{\theta}_t).\end{aligned} \tag{3}$$

Apply one SGD step on the model parameters:

$$\widetilde{\boldsymbol{\theta}}_{t+1} = \boldsymbol{\theta}_t - \alpha_t \nabla\boldsymbol{\theta}_t, \tag{4}$$

where α_t is the learning rate of the inner loop.

Derivation

Evaluate on the labeled data and differentiate the labeled data loss:

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{x}_k^l, \boldsymbol{y}_k; \widetilde{\boldsymbol{\theta}}_{t+1}) &= \Phi(f(\boldsymbol{x}_k^l; \widetilde{\boldsymbol{\theta}}_{t+1}), \boldsymbol{y}_k), \\
\nabla \widetilde{\boldsymbol{y}}_i &= \frac{1}{B^l} \sum_{k=1}^{B^l} \nabla_{\widetilde{\boldsymbol{y}}_i} \mathcal{L}(\boldsymbol{x}_k^l, \boldsymbol{y}_k; \widetilde{\boldsymbol{\theta}}_{t+1}).
\end{aligned} \tag{5}
$$

Derivation

Evaluate on the labeled data and differentiate the labeled data loss:

$$\mathcal{L}(\boldsymbol{x}_k^l, \boldsymbol{y}_k; \widetilde{\boldsymbol{\theta}}_{t+1}) = \Phi(f(\boldsymbol{x}_k^l; \widetilde{\boldsymbol{\theta}}_{t+1}), \boldsymbol{y}_k),$$
$$\nabla \widetilde{\boldsymbol{y}}_i = \frac{1}{B^l} \sum_{k=1}^{B^l} \nabla_{\widetilde{\boldsymbol{y}}_i} \mathcal{L}(\boldsymbol{x}_k^l, \boldsymbol{y}_k; \widetilde{\boldsymbol{\theta}}_{t+1}). \tag{5}$$

Perform one SGD step on the pseudo-labels:

$$\widehat{\boldsymbol{y}}_i = \widetilde{\boldsymbol{y}}_i - \beta_t \nabla \widetilde{\boldsymbol{y}}_i, \tag{6}$$

where β_t is the meta learning rate,

Derivation

Evaluate on the labeled data and differentiate the labeled data loss:

$$\begin{aligned} \mathcal{L}(\boldsymbol{x}_k^l, \boldsymbol{y}_k; \widetilde{\boldsymbol{\theta}}_{t+1}) &= \Phi(f(\boldsymbol{x}_k^l; \widetilde{\boldsymbol{\theta}}_{t+1}), \boldsymbol{y}_k), \\ \nabla \widetilde{\boldsymbol{y}}_i &= \frac{1}{B^l} \sum_{k=1}^{B^l} \nabla_{\widetilde{\boldsymbol{y}}_i} \mathcal{L}(\boldsymbol{x}_k^l, \boldsymbol{y}_k; \widetilde{\boldsymbol{\theta}}_{t+1}). \end{aligned} \tag{5}$$

Perform one SGD step on the pseudo-labels:

$$\widehat{\boldsymbol{y}}_i = \widetilde{\boldsymbol{y}}_i - \beta_t \nabla \widetilde{\boldsymbol{y}}_i, \tag{6}$$

where β_t is the meta learning rate,

Compute the consistency loss from the unlabeled data and the updated pseudo-labels.

Algorithm 1 Meta-Learning Algorithm.

Input: regular learning rates $\{\alpha_t\}$,
meta learning rates $\{\beta_t\}$

for $t := 1$ *to #iters* **do**

$$\{(\boldsymbol{x}_k^l, \boldsymbol{y}_k)\}_{k=1}^{B^l} \leftarrow \text{BatchSampler}(\mathcal{D}^l)$$
$$\{\boldsymbol{x}_i^u\}_{i=1}^{B^u} \leftarrow \text{BatchSampler}(\mathcal{D}^u)$$
$$\widetilde{\boldsymbol{y}}_i = f(\boldsymbol{x}_i^u; \boldsymbol{\theta}_t)$$
$$\mathcal{L}(\boldsymbol{x}_i^u, \widetilde{\boldsymbol{y}}_i; \boldsymbol{\theta}_t) = \Phi(f(\boldsymbol{x}_i^u; \boldsymbol{\theta}_t), \widetilde{\boldsymbol{y}}_i)$$
$$\nabla\boldsymbol{\theta}_t = \frac{1}{B^u}\sum_{i=1}^{B^u} \nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{x}_i^u, \widetilde{\boldsymbol{y}}_i; \boldsymbol{\theta}_t)$$
$$\widetilde{\boldsymbol{\theta}}_{t+1} = \boldsymbol{\theta}_t - \alpha_t\nabla\boldsymbol{\theta}_t$$
$$\mathcal{L}(\boldsymbol{x}_k^l, \boldsymbol{y}_k; \widetilde{\boldsymbol{\theta}}_{t+1}) = \Phi(f(\boldsymbol{x}_k^l; \widetilde{\boldsymbol{\theta}}_{t+1}), \boldsymbol{y}_k)$$
$$\nabla\widetilde{\boldsymbol{y}}_i = \frac{1}{B^l}\sum_{k=1}^{B^l} \nabla_{\widetilde{\boldsymbol{y}}_i}\mathcal{L}(\boldsymbol{x}_k^l, \boldsymbol{y}_k; \widetilde{\boldsymbol{\theta}}_{t+1})$$
$$\widehat{\boldsymbol{y}}_i = \widetilde{\boldsymbol{y}}_i - \beta_t\nabla\widetilde{\boldsymbol{y}}_i$$
$$\mathcal{L}(\boldsymbol{x}_i^u, \widehat{\boldsymbol{y}}_i; \boldsymbol{\theta}_t) = \Phi(f(\boldsymbol{x}_i^u; \boldsymbol{\theta}_t), \widehat{\boldsymbol{y}}_i)$$
$$\nabla\widehat{\boldsymbol{\theta}}_t = \frac{1}{B^u}\sum_{i=1}^{B^u} \nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{x}_i^u, \widehat{\boldsymbol{y}}_i; \boldsymbol{\theta}_t)$$
$$\boldsymbol{\theta}_{t+1} = \text{Optimizer}(\boldsymbol{\theta}_t, \nabla\widehat{\boldsymbol{\theta}}_t, \alpha_t)$$

end

Experiments on **SVHN**, **CIFAR-10**, and **CIFAR-100**.

Method	SVHN	CIFAR-10	CIFAR-100
Π-Model [7]	4.82%	12.36%	39.19%
TE [7]	4.42%	12.16%	38.65%
MT [9]	3.95%	12.31%	-
MT+SNTG [26]	3.86%	10.93%	-
VAT [8]	5.42%	11.36%	-
VAT+Ent [8]	3.86%	10.55%	-
VAT+Ent+SNTG [26]	3.83%	9.89%	-
VAT+VAdD [27]	3.55%	9.22%	-
MA-DNN [28]	4.21%	11.91%	34.51%
Co-training [29]	3.29%	8.35%	34.63%
MT+fastSWA [10]	-	9.05%	33.62%
TNAR-VAE [11]	3.74%	8.85%	-
ADA-Net [24]	4.62%	10.30%	-
ADA-Net+fastSWA [24]	-	8.72%	-
DualStudent [30]	-	8.89%	32.77%
Ours	**3.15%**	**7.78%**	**30.74%**
Fully-Supervised	2.67%	4.88%	22.10%

Figure: Semi-supervised classification results.

Experiments on **ImageNet** dataset.

Method	**Top-1**	**Top-5**
Labeled-Only	53.65%	31.01%
MT [9]	49.07%	23.59%
Co-training [29]	46.50%	22.73%
ADA-Net [24]	44.91%	21.18%
Ours	**44.87%**	**18.88%**
Fully-Supervised	29.15%	10.12%

Figure: Semi-supervised classification results on ImageNet.

Experiments on ImageNet

Experiments on **ImageNet** dataset.

Method	Top-1	Top-5
Labeled-Only	53.65%	31.01%
MT [9]	49.07%	23.59%
Co-training [29]	46.50%	22.73%
ADA-Net [24]	44.91%	21.18%
Ours	**44.87%**	**18.88%**
Fully-Supervised	29.15%	10.12%

Figure: Semi-supervised classification results on ImageNet.

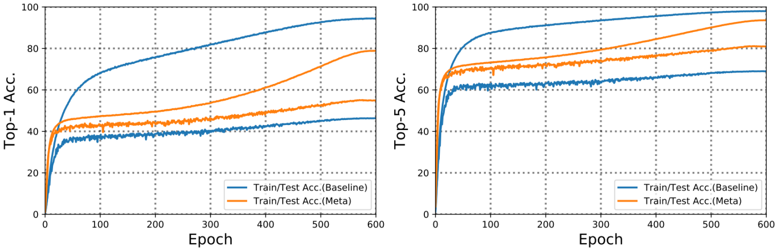

Figure: Accuracy curves of the baseline method and the meta-learning algorithm.

Further information

Refer to our paper[4] for further details:

(i) First-order approximation of the second-order derivative;
(ii) Incorporation of Mix-up augmentation in the algorithm;
(iii) Convergence analysis of the meta-learning algorithm;
(iv) Detailed ablation study and feature visualization.

The source codes are available:
`https://github.com/Sakura03/SemiMeta`.

[4] `https://arxiv.org/abs/2007.03966`

Thanks!