A survey on deep semi-supervised learning algorithms

Ani Vanyan, Hrant Khachatrian

September 13, 2020

Overview

Semi-supervised learning

- **Semi-supervised learning** is about making use of unlabeled data
  - It is useful when the available labeled data is small
- Unlabeled data is cheap
  - Although not every type of unlabeled data is useful

Consistency regularization

The idea is to make sure the neural network produces similar results for the augmented versions of the same unlabeled image.

- Definitions
  - X is the labeled dataset with samples $(x, p) \in X$
  - U is the dataset without labels
  - $f_\theta(x)$ is a function (neural network): outputs probabilities of the labels
  - $Augment(x)$ is a *stochastic* function that augments the sample x
  - $H(\cdot, \cdot)$ is the cross entropy loss
- Consistency regularization:
  - $L_U = \frac{1}{|U|} \sum_{x \in U} \| f_\theta(Augment(x)) - f_\theta(Augment(x)) \|_2^2$

Π-model[1]

- Π-model
  - $L = L_X + \lambda(t) L_U$
  - $L_X = \frac{1}{|X|} \sum_{(x,p) \in X} H(p, f_\theta(x))$
  - $L_U = \frac{1}{|U|} \sum_{x \in U} \| f_\theta(Augment(x)) - f_\theta(Augment(x)) \|_2^2$
  - $\lambda(t)$ slowly grows from 0 to its final value λ
    - which is a hyperparameter
- $Augment(x)$
  - Translation by $a \sim Uniform(-2, 2)$ pixels
  - Horizontal flip (for CIFAR-10/100 only)

[1] Samuli Laine and Timo Aila. "Temporal ensembling for semi-supervised learning". In: *ICLR* (2017).

The problem of the unstable target

The problem with consistency loss is that it is not stable. In Π-model the partial solution was given. The trick is called temporal ensembling.

$$f_{temp.ens}(x) = \alpha f_{temp.ens}(Augment(x)) + (1 - \alpha) f_\theta(Augment(x))$$

$$L_U = \frac{1}{|U|} \sum_{x \in U} \| f_\theta(Augment(x)) - f_{temp.ens}(x) \|_2^2$$

The first formula is computed once per epoch.

Mean Teacher[2]

- Two separate models: a Student network with θ parameters and a Teacher with θ' parameters.
- $L_X = \frac{1}{|X|} \sum_{(x,p) \in X} H(p, f_\theta^{student}(x))$
- Teacher's parameters are updated at each iteration
- $\theta'_t = \alpha \theta'_{t-1} + (1 - \alpha)\theta_t$
- $L_U = \frac{1}{|U|} \sum_{x \in U} \left\| f_\theta^{student}(Augment(x)) - f_{\theta'}^{teacher}(Augment(x)) \right\|_2^2$
- Teacher is *not* trained via backpropagation.

[2] Antti Tarvainen and Harri Valpola. "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results". In: *Advances in neural information processing systems*. 2017, pp. 1195–1204.

Virtual Adversarial Training[3]

- Regular adversarial training
  - For each $(x, p) \in X$
  - $L_{adv} = H\left(p, f_\theta(x + r_{adv})\right)$
  - $r_{adv} = \underset{r:\|r\|<\epsilon}{\arg\max}\, H(p, f_\theta(x + r))$
- Virtual adversarial training
  - For each $x \in U$
  - $L_{adv} = H\left(f_\theta(x), f_\theta(x + r_{adv})\right)$
  - $r_{adv} = \underset{r:\|r\|<\epsilon}{\arg\max}\, H(f_\theta(x), f_\theta(x + r))$
  - Fast approximation method for r_{adv}
- Entropy minimization
  - $L_{ent} = \frac{1}{|X|+|U|} \sum_{x \in X \cup U} H(f_\theta(x))$

[3] Takeru Miyato et al. "Virtual adversarial training: a regularization method for supervised and semi-supervised learning". In: *IEEE transactions on pattern analysis and machine intelligence* 41.8 (2018), pp. 1979–1993.

Realistic Evaluation[4]

- Reproducing results
  - Reimplemented many known methods in a single code repository
  - WideResNet-28-2, batch normalization, leaky ReLU
  - Adam optimizer with fixed β_1 and β_2
  - Fixed data augmentation and preprocessing
    - Although, different for CIFAR-10 and SVHN!
  - Equal hyperparameter tuning budget!
    - 1000 trials of Gaussian-Process-based black box optim. in Google Cloud
    - Model selection on the full validation set
    - Different initial learning rates for different methods!
    - VAT's ϵ was different on CIFAR-10 and SVHN

[4] Avital Oliver et al. "Realistic evaluation of deep semi-supervised learning algorithms". In: *Advances in Neural Information Processing Systems*. 2018, pp. 3235–3246.

Realistic Evaluation[5]

- Transfer learning
  - Transfer from (resized) ImageNet is a strong baseline, ignored in most papers
- Class distribution mismatch
  - Adding unlabeled data from a mismatched set of classes can actually **hurt** the performance compared to not using unlabeled data at all

[5] Avital Oliver et al. "Realistic evaluation of deep semi-supervised learning algorithms". In: *Advances in Neural Information Processing Systems*. 2018, pp. 3235–3246.

MixMatch[6]

- Similar underlying model, but with different X' and U'
  - $X', U' = MixMatch(X, U, T, K, \alpha)$
  - $L_X = \frac{1}{|X'|} \sum_{(x', p') \in X'} H(p', f_\theta(x'))$
  - $L_U = \frac{1}{|U'|} \sum_{(x', q') \in U'} \|q' - f_\theta(x')\|_2^2$
  - $L = L_X + \lambda(t) L_U$

[6] David Berthelot et al. "Mixmatch: A holistic approach to semi-supervised learning". In: *accepted at NeurIPS'19* (2019).

MixUp Regularization[7]

MixMatch uses trick for obtaining new unlabeled examples called MixUp.

- Constructs new pairs by convex combination of existing pairs
- For each pair (x_1, p_1) and (x_2, p_2):
  - Sample $\lambda \sim Beta(\alpha, \alpha)$
  - $\lambda' = max(\lambda, 1 - \lambda)$ to make sure it's close to 0
    - This step did not exist in the original MixUp paper
  - $x' = \lambda' x_1 + (1 - \lambda') x_2$
  - $p' = \lambda' p_1 + (1 - \lambda') p_2$
  - Return (x', p')
- $MixUp(A, B)$ is a set of samples "closer" to A

[7] Hongyi Zhang et al. "mixup: Beyond empirical risk minimization". In: *arXiv preprint arXiv:1710.09412* (2017).

ReMixMatch[8]

It is a more complicated version of MixMatch.
The two main additions are:

- Distribution alignment. The predicted probabilities on a batch of unlabeled examples are scaled to match the distribution of the labels present in the labeled subset.
- Anchored augmentation. The target label for unlabeled examples is determined using weakly augmented versions of the images, while the prediction for the same images is computed by using strongly augmented versions.

[8] David Berthelot et al. "ReMixMatch: Semi-Supervised Learning with Distribution Alignment and Augmentation Anchoring". In: *arXiv preprint arXiv:1911.09785 / accepted at ICLR'2020* (2019).

Unsupervised Data Augmentation[10]

UDA is quite similar to Virtual Adversarial Training, but replaces virtual adversarial example generation with a very strong augmenation called RandAugment[9].

UDA uses a training technique called Training Signal Annealing to reduce overfitting when there is a huge gap between the amount of unlabeled data and labeled data.

[9] Ekin D Cubuk et al. "RandAugment: Practical data augmentation with no separate search". In: *arXiv preprint arXiv:1909.13719* (2019).

[10] Qizhe Xie et al. "Unsupervised data augmentation". In: *arXiv preprint arXiv:1904.12848* (2019).

FixMatch[13]

FixMatch uses CutOut[11] along with RandAugment or CTAugment[12] as a strong augmentation procedure known from previous papers (UDA and ReMixMatch, respectively).

In contrast to UDA and other methods, FixMatch performs arg max on the guessed label, so it essentially becomes equivalent to pseudo-labeling. Additionally, FixMatch ignores the guessed labels if the confidence is lower than $\tau = 0.95$ threshold.

---

[11] Terrance DeVries and Graham W Taylor. "Improved regularization of convolutional neural networks with cutout". In: *arXiv preprint arXiv:1708.04552* (2017).

[12] David Berthelot et al. "ReMixMatch: Semi-Supervised Learning with Distribution Alignment and Augmentation Anchoring". In: *arXiv preprint arXiv:1911.09785 / accepted at ICLR'2020* (2019).

[13] Kihyuk Sohn et al. "Fixmatch: Simplifying semi-supervised learning with consistency and confidence". In: *arXiv preprint arXiv:2001.07685* (2020).

Results

Method	**Model**	**Parameters**	**Error** (%)	
			CIFAR-10(4k)	SVHN(1k)
Π-Model (2016)	Conv-Large	3.1M	12.36 ± 0.31	4.82 ± 0.17
Mean Teacher (2017)	Conv-Large	3.1M	12.31 ± 0.28	3.95 ± 0.19
VAT + EntMin (2017)	Conv-Large	3.1M	10.55 ± 0.05	3.86 ± 0.11
Mean Teacher (2017)	Shake-Shake	26M	6.28 ± 0.15	-
UDA(RandAugment) (2019)	Shake-Shake	26M	3.7	-
MixMatch (2019)	WRN	26M	4.95 ± 0.08	-
UDA(RandAugment) (2019)	WRN-28-2	1.5M	5.29 ± 0.25	**2.55 ± 0.09**
ReMixMatch (2019)	WRN-28-2	1.5M	5.14 ± 0.04	2.83 ± 0.30
FixMatch(RA) (2020)	WRN-28-2	1.5M	**4.26 ± 0.05**	**2.28 ± 0.11**
FixMatch(CTA) (2020)	WRN-28-2	1.5M	**4.31 ± 0.15**	**2.36 ± 0.19**
UDA(RandAugment) (2019)	PyramidNet	26M	**2.7**	-

Thanks.