Boulder

Computer Vision; Advanced Topics; Semi-Supervised Learning

Maziar Raissi

Assistant Professor

Department of Applied Mathematics

University of Colorado Boulder

maziar.raissi@colorado.edu

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

Boulder

$\mathcal{D}_l = \{x_l^{(n)}, y_l^{(n)} | n = 1, \ldots, N_l\} \rightarrow$ labeled dataset

$\mathcal{D}_{ul} = \{x_{ul}^{(m)} | m = 1, \ldots, N_{ul}\} \rightarrow$ unlabeled dataset

Train $p(y|x, \theta)$ using $\mathcal{D}_l$ and $\mathcal{D}_{ul}$.

Adversarial Training

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

↳ divergence between two distributions

$r_{\text{adv}} := \arg\max_{r;\|r\|\leq\epsilon} D[q(y|x_l), p(y|x_l + r, \theta)]$

$D[p, p'] = -\sum_i p_i \log p'_i \rightarrow$ cross entropy

$q(y|x_l) \rightarrow$ true distribution

$q(y|x_l) \approx h(y; y_l) \rightarrow$ one-hot-vector

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

↳ for L_2 norm

$r_{\text{adv}} \approx \epsilon \text{sign}(g) \rightarrow$ for L_∞ norm

Virtual Adversarial Training

Let x_* represent either x_l or x_{ul}.

we have no direct information about $q(y|x_{ul})$

Replace $q(y|x)$ with its current estimate $p(y|x, \hat{\theta})$!

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

↳ Local Distributional Smoothness

$r_{\text{vadv}} := \arg\max_{r;\|r\|_2\leq\epsilon} D\big[p(y|x_*, \hat{\theta}), p(y|x_* + r)\big]$

↳ virtual adversarial perturbation

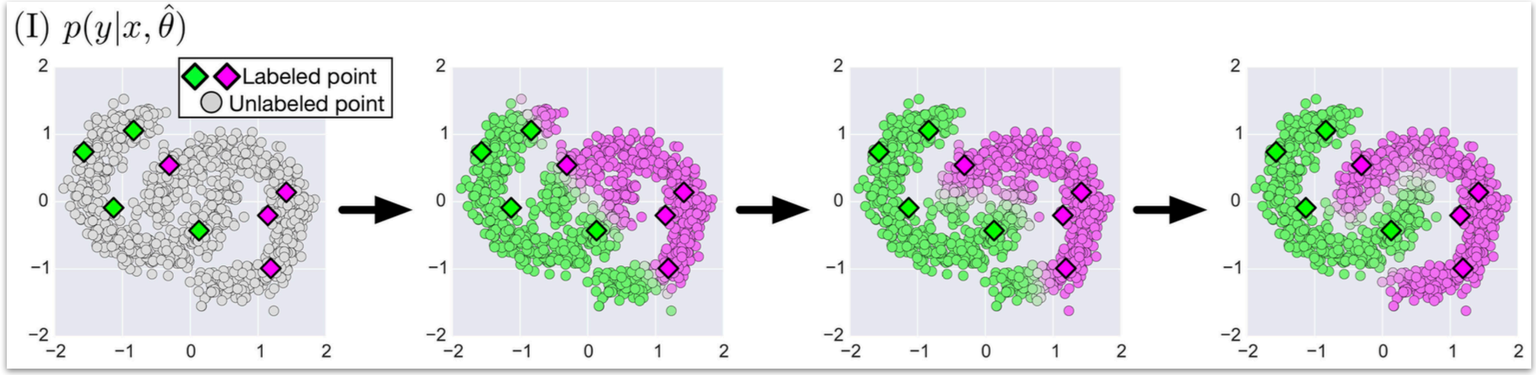
$\ell(\mathcal{D}_l, \theta) + \alpha \mathcal{R}_{\text{vadv}}(\mathcal{D}_l, \mathcal{D}_{ul}, \theta) \rightarrow$ full objective function

↳ negative log likelihood for labeled data

$$\mathcal{R}_{\text{vadv}}(\mathcal{D}_l, \mathcal{D}_{ul}, \theta) := \frac{1}{N_l + N_{ul}} \sum_{x_* \in \mathcal{D}_l, \mathcal{D}_{ul}} \text{LDS}(x_*, \theta)$$

Reduction LDS would make the model smooth at each data point.

VAT on semi-supervised learning can be given a similar interpretation as label propagation.



Power iteration method and the finite difference method

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

$D(r, x_*, \hat{\theta})$ takes the minimal value at $r = 0$ and $\nabla_r D(r, x_*, \hat{\theta})|_{r=0} = 0$.

$$D(r, x, \hat{\theta}) \approx \frac{1}{2} r^T H(x, \hat{\theta}) r$$

↳ Hessian matrix

$r_{\text{vadv}} \approx \arg\max_r \{r^T H(x, \hat{\theta}) r;\ \|r\|_2 \leq \epsilon\} = \epsilon \overline{u(x, \hat{\theta})} \rightarrow$ first dominant eigenvector of H with magnitude ϵ

$$r_{\text{vadv}} \approx \epsilon \frac{g}{\|g\|_2} \text{ where } g = \nabla_r D\big[p(y|x, \hat{\theta}), p(y|x + r, \hat{\theta})\big]\Big|_{r=\xi d}$$

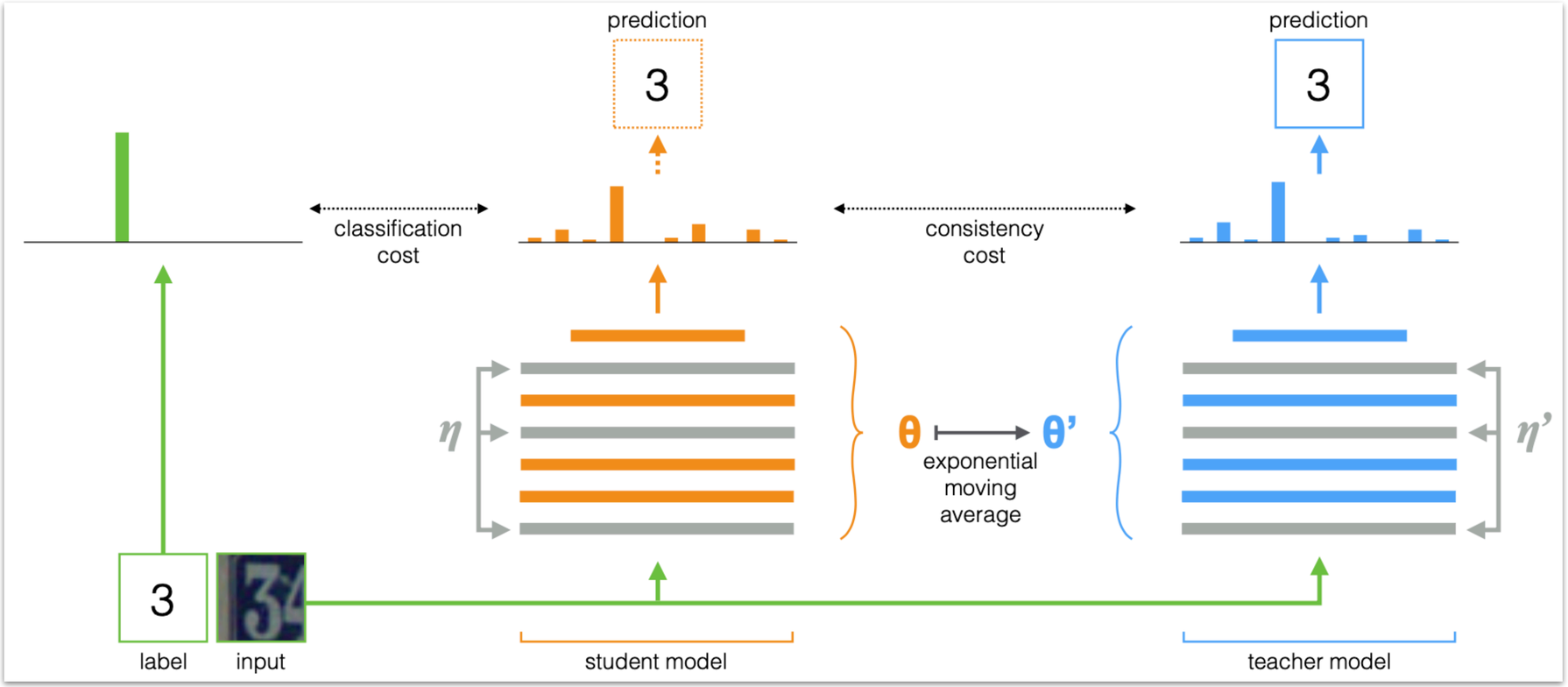
$\bar{v} \equiv \frac{v}{\|v\|_2}$

$\xi = 10^{-6}$, d is a randomly sampled unit vector

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

Mean teachers are better role models:

Weight-averaged consistency targets improve semi-supervised deep learning results



$J \rightarrow$ consistency loss

student model $\rightarrow$ weights θ and noise η

teacher model $\rightarrow$ weights θ' and noise η'

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

$\theta'_t = \alpha\theta'_{t-1} + (1-\alpha)\theta_t$

↳ Exponential Moving Average (EMA)

Three types of noise:

– random translations and horizontal flips of the input images

– Gaussian noise on the input layer

– dropout applied within the network

Ramp up the scale of consistency loss from zero to its final value

Temporal Ensembling: Maintains an exponential moving average of label predictions on each training example.

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	$\mathbf{5.60 \pm 0.10}$
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	$\mathbf{21.55 \pm 1.48}$	$\mathbf{15.73 \pm 0.31}$	12.31 ± 0.28	5.94 ± 0.15

Error rate percentage on SVHN over 10 runs (4 runs when using all labels)

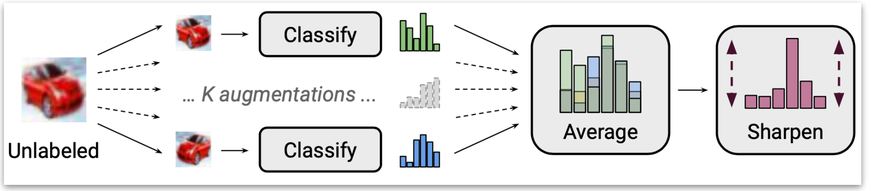
	250 labels 73257 images	500 labels 73257 images	1000 labels 73257 images	73257 labels 73257 images
GAN [25]		18.44 ± 4.8	8.11 ± 1.3	
Π model [13]		6.65 ± 0.53	4.82 ± 0.17	2.54 ± 0.04
Temporal Ensembling [13]		5.12 ± 0.13	4.42 ± 0.16	2.74 ± 0.06
VAT+EntMin [16]			**3.86**	
Supervised-only	27.77 ± 3.18	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	$\mathbf{4.35 \pm 0.50}$	$\mathbf{4.18 \pm 0.27}$	3.95 ± 0.19	$\mathbf{2.50 \pm 0.05}$

Tarvainen, Antti, and Harri Valpola. "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results." *arXiv preprint arXiv:1703.01780* (2017).

MixMatch: A Holistic Approach to Semi-Supervised Learning

Boulder

$p_{\text{model}}(y|x;\theta) \rightarrow$ a generic model which produces a distribution over class labels y for an input x with parameters θ



Consistency Regularization

encourage the model to produce the same output distribution when its inputs are perturbed

$\|p_{\text{model}}(y|\text{Augment}(x);\theta) - p_{\text{model}}(y|\text{Augment}(x);\theta)\|_2^2$

↳ a stochastic transformation (the two terms above are not identical)

"Mean Teacher" → replaces one of the terms above with the output of the model using an exponential moving average of model parameter values

"Virtual Adversarial Training" (VAT) → computing an additive perturbation to apply to the input which maximally changes the output class distribution

Entropy Minimization

encourage the model to output confident predictions on unlabeled data

minimize the entropy of $p_{\text{model}}(y|x;\theta)$ for unlabeled data

"Pseudo-Label" → constructing hard (1-hot) labels from high-confidence predictions on unlabeled data and using these as training targets in a standard cross-entropy loss

"Pseudo-Label" and "Sharpening" also achieve entropy minimization

Traditional Regularization

weight-decay and mixup

$(x_1, p_1), (x_2, p_2) \rightarrow$ pair of two examples with their corresponding labels

$\lambda \sim \text{Beta}(\alpha, \alpha)$

$\lambda' = \max(\lambda, 1-\lambda) \rightarrow$ vanilla mixup uses $\lambda' = \lambda$

$x' = \lambda' x_1 + (1-\lambda')x_2$

$p' = \lambda' p_1 + (1-\lambda')p_2$

x' closer to x_1 than x_2

MixMatch

$\mathcal{X}', \mathcal{U}' = \text{MixMatch}(\mathcal{X}, \mathcal{U}, T, K, \alpha)$

Algorithm 1 MixMatch takes a batch of labeled data $\mathcal{X}$ and a batch of unlabeled data $\mathcal{U}$ and produces a collection $\mathcal{X}'$ (resp. $\mathcal{U}'$) of processed labeled examples (resp. unlabeled with guessed labels).

Input: Batch of labeled examples and their one-hot labels $\mathcal{X} = \big((x_b, p_b); b \in (1, \dots, B)\big)$, batch of unlabeled examples $\mathcal{U} = \big(u_b; b \in (1, \dots, B)\big)$, sharpening temperature T, number of augmentations K, Beta distribution parameter α for MixUp.
for $b = 1$ **to** B **do**
  $\hat{x}_b = \text{Augment}(x_b)$ *// Apply data augmentation to x_b*
  for $k = 1$ **to** K **do**
    $\hat{u}_{b,k} = \text{Augment}(u_b)$ *// Apply k^{th} round of data augmentation to u_b*
  end for
  $\bar{q}_b = \frac{1}{K}\sum_k \text{p}_{\text{model}}(y \mid \hat{u}_{b,k}; \theta)$ *// Compute average predictions across all augmentations of u_b*
  $q_b = \text{Sharpen}(\bar{q}_b, T)$ *// Apply temperature sharpening to the average prediction (see eq. (7))*
end for
$\hat{\mathcal{X}} = \big((\hat{x}_b, p_b); b \in (1, \dots, B)\big)$ *// Augmented labeled examples and their labels*
$\hat{\mathcal{U}} = \big((\hat{u}_{b,k}, q_b); b \in (1, \dots, B), k \in (1, \dots, K)\big)$ *// Augmented unlabeled examples, guessed labels*
$\mathcal{W} = \text{Shuffle}\big(\text{Concat}(\hat{\mathcal{X}}, \hat{\mathcal{U}})\big)$ *// Combine and shuffle labeled and unlabeled data*
$\mathcal{X}' = \big(\text{MixUp}(\hat{\mathcal{X}}_i, \mathcal{W}_i); i \in (1, \dots, |\hat{\mathcal{X}}|)\big)$ *// Apply MixUp to labeled data and entries from $\mathcal{W}$*
$\mathcal{U}' = \big(\text{MixUp}(\hat{\mathcal{U}}_i, \mathcal{W}_{i+|\hat{\mathcal{X}}|}); i \in (1, \dots, |\hat{\mathcal{U}}|)\big)$ *// Apply MixUp to unlabeled data and the rest of $\mathcal{W}$*
return $\mathcal{X}', \mathcal{U}'$

Loss Function

$$\mathcal{L}_{\mathcal{X}} = \frac{1}{|\mathcal{X}'|}\sum_{x,p \in \mathcal{X}'} \text{H}(p, \text{p}_{\text{model}}(y \mid x; \theta))$$

$$\mathcal{L}_{\mathcal{U}} = \frac{1}{L|\mathcal{U}'|}\sum_{u,q \in \mathcal{U}'} \|q - \text{p}_{\text{model}}(y \mid u; \theta)\|_2^2$$

$$\mathcal{L} = \mathcal{L}_{\mathcal{X}} + \lambda_{\mathcal{U}}\mathcal{L}_{\mathcal{U}}$$

$$\text{Sharpen}(p, T)_i := p_i^{\frac{1}{T}} \Big/ \sum_{j=1}^{L} p_j^{\frac{1}{T}}$$

$$\lim_{T \to 0} \text{Sharpen}(p, T) = \text{Dirac (i.e., "one-hot") distribution}$$

Berthelot, David, et al. "Mixmatch: A holistic approach to semi-supervised learning." *arXiv preprint arXiv:1905.02249* (2019).

Self-training with Noisy Student improves ImageNet classification

Require: Labeled images $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ and unlabeled images $\{\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_m\}$.

1: Learn teacher model θ_*^t which minimizes the cross entropy loss on labeled images

$$\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f^{noised}(x_i, \theta^t))$$

2: Use an unnoised teacher model to generate soft or hard pseudo labels for unlabeled images

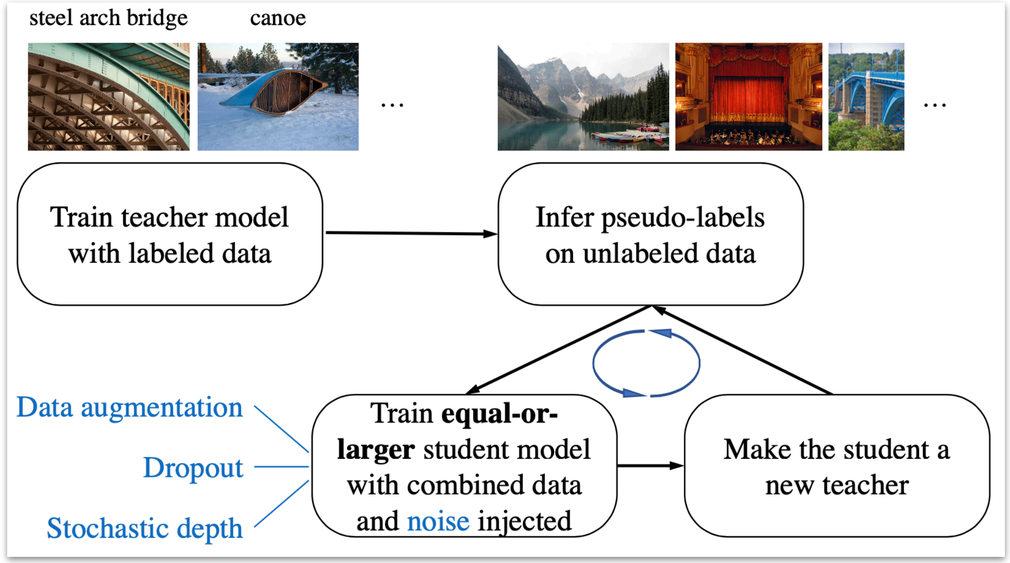
$$\tilde{y}_i = f(\tilde{x}_i, \theta_*^t), \forall i = 1, \cdots, m$$

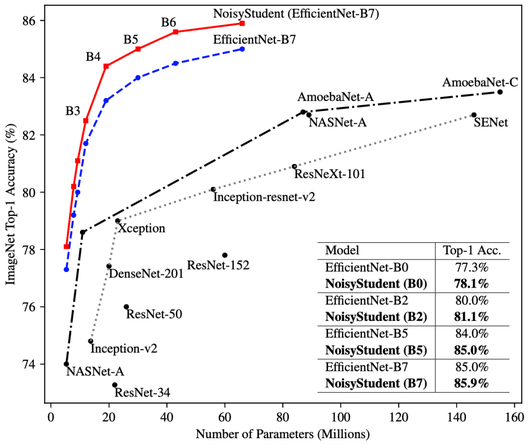
3: Learn an **equal-or-larger** student model θ_*^s which minimizes the cross entropy loss on labeled images and unlabeled images with **noise** added to the student model

$$\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f^{noised}(x_i, \theta^s)) + \frac{1}{m}\sum_{i=1}^{m} \ell(\tilde{y}_i, f^{noised}(\tilde{x}_i, \theta^s))$$

4: Iterative training: Use the student as a teacher and go back to step 2.

	ImageNet top-1 acc.	ImageNet-A top-1 acc.	ImageNet-C mCE	ImageNet-P mFR
Prev. SOTA	86.4%	61.0%	45.7	27.8
NoisyStudent	**88.4%**	**83.7%**	**28.3**	**12.2**



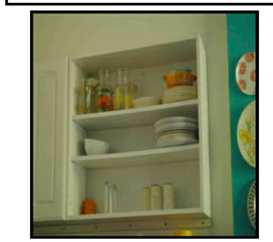


ImageNet-C and ImageNet-P test sets include images with common corruptions and perturbations such as blurring, fogging, rotation and scaling. ImageNet-A test set consists of difficult images that cause significant drops in accuracy to state-of-the-art models. These test sets are considered as "robustness" benchmarks.

mCE (mean corruption error) is the weighted average of error rate on different corruptions, with AlexNet's error rate as a baseline (lower is better). mFR (mean flip rate) measures the model's probability of flipping predictions under perturbations with AlexNet as a baseline (lower is better).

(a) ImageNet-A

(b) ImageNet-C

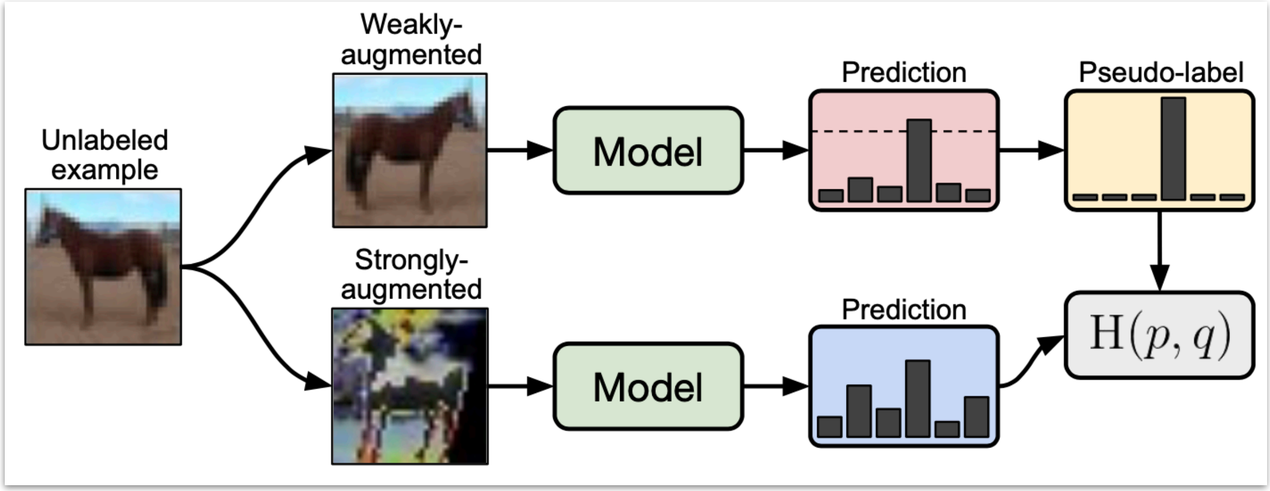


(c) ImageNet-P

Xie, Qizhe, et al. "Self-training with noisy student improves imagenet classification." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence

Semi-Supervised Learning (SSL): Leveraging unlabeled data to improve a model's performance



$L \rightarrow$ number of classes

$\mathcal{X} = \{(x_b, p_b) : b = 1, \ldots, B\} \rightarrow$ batch of B labeled examples

$x_b \rightarrow$ training example

$p_b \rightarrow$ one-hot-labels

$\mathcal{U} = \{u_b : b = 1, \ldots, \mu B\} \rightarrow$ batch of μB unlabeled examples

$\mu \rightarrow$ determines the relative size of $\mathcal{X}$ and $\mathcal{U}$

$p_{\mathrm{m}}(y|x) \rightarrow$ predicted class distribution produced by the model for input x

$H(p, q) \rightarrow$ cross-entropy between two probability distributions p and q

$\mathcal{A}(\cdot) \rightarrow$ strong augmentation (autoaugment/randaugment + cutout)

$\alpha(\cdot) \rightarrow$ weak augmentation (flip and shift)

Consistency Regularization

$\sum_{b=1}^{\mu B} \|p_{\mathrm{m}}(y|\,\alpha(u_b)) - p_{\mathrm{m}}(y|\,\alpha(u_b))\|_2^2 \rightarrow$ both α and p_{m} are stochastic functions, so the two terms in this equation will indeed have different values

Extensions of the consistency regularization idea

– using an adversarial transformation in place of α

– using a running average or past model predictions for one invocation of p_{m}

– using a cross-entropy loss in place of the squared l^2 loss

– using stronger forms of augmentation

Pseudo-labeling

$q_b = p_{\mathrm{m}}(y|u_b)$

$$\frac{1}{\mu B}\sum_{b=1}^{\mu B} \mathbb{1}(\max(q_b) \geq \tau)\, \mathrm{H}(\hat{q}_b, q_b)$$

(τ: predefined threshold)

$\hat{q}_b = \arg\max(q_b) \rightarrow$ one-hot (hard-label)

encourages model predictions to be low-entropy (i.e., high-confidence) on unlabeled data

FixMatch

$\ell_s \rightarrow$ supervised loss

$\ell_u \rightarrow$ unsupervised loss

$$\ell_s = \frac{1}{B}\sum_{b=1}^{B} \mathrm{H}(p_b, p_{\mathrm{m}}(y \mid \alpha(x_b)))$$

$$\ell_u = \frac{1}{\mu B}\sum_{b=1}^{\mu B} \mathbb{1}(\max(q_b) \geq \tau)\, \mathrm{H}(\hat{q}_b, p_{\mathrm{m}}(y \mid \mathcal{A}(u_b)))$$

$q_b = p_{\mathrm{m}}(y \mid \alpha(u_b))$ $\quad \hat{q}_b = \arg\max(q_b) \rightarrow$ pseudo-label $\quad \ell_s + \lambda_u \ell_u$

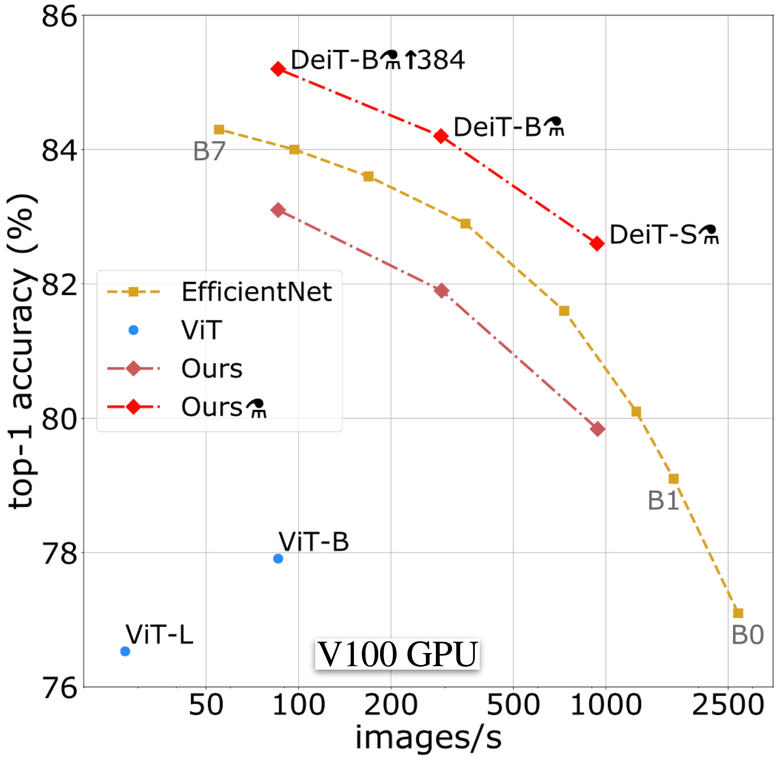
FixMatch reaches 78% CIFAR-10 accuracy using only above 10 labeled images.

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

Training Data-Efficient Image Transformers & Distillation Through Attention

DeiT: Data-Efficient Image Transformers

ImageNet Data Only (No External Data such as JFT-300M)



⚗ → models trained with our transformer-specific distillation

Fixing the positional encoding across resolutions

Use a lower training resolution and fine-tune the network at a larger resolution

Keep the image patch sizes the same

$\implies$ N (sequence length) changes

$\implies$ need to adapt positional encodings (use bicubic interpolation)

Pre-training	Fine-tuning	Rand-Augment	AutoAug	Mixup	CutMix	Erasing	Stoch. Depth	Repeated Aug.	Dropout	Exp. Moving Avg.	pre-trained 224	fine-tuned 384
adamw	adamw	✓	✗	✓	✓	✓	✓	✓	✗	✗	$81.8_{\pm 0.2}$	$83.1_{\pm 0.1}$
SGD	adamw	✓	✗	✓	✓	✓	✓	✓	✗	✗	74.5	77.3
adamw	SGD	✓	✗	✓	✓	✓	✓	✓	✗	✗	81.8	83.1
adamw	adamw	✗	✗	✓	✓	✓	✓	✓	✗	✗	79.6	80.4
adamw	adamw	✗	✓	✓	✓	✓	✓	✓	✗	✗	81.2	81.9
adamw	adamw	✓	✗	✗	✓	✓	✓	✓	✗	✗	78.7	79.8
adamw	adamw	✓	✗	✓	✗	✓	✓	✓	✗	✗	80.0	80.6
adamw	adamw	✓	✗	✗	✗	✓	✓	✓	✗	✗	75.8	76.7
adamw	adamw	✓	✗	✓	✓	✗	✓	✓	✗	✗	4.3*	0.1
adamw	adamw	✓	✗	✓	✓	✓	✗	✓	✗	✗	3.4*	0.1
adamw	adamw	✓	✗	✓	✓	✓	✓	✗	✗	✗	76.5	77.4
adamw	adamw	✓	✗	✓	✓	✓	✓	✓	✓	✗	81.3	83.1
adamw	adamw	✓	✗	✓	✓	✓	✓	✓	✗	✓	81.9	83.1

Distillation through attention

Teacher Model: A strong image classifier (e.g., RegNetY ConvNet)

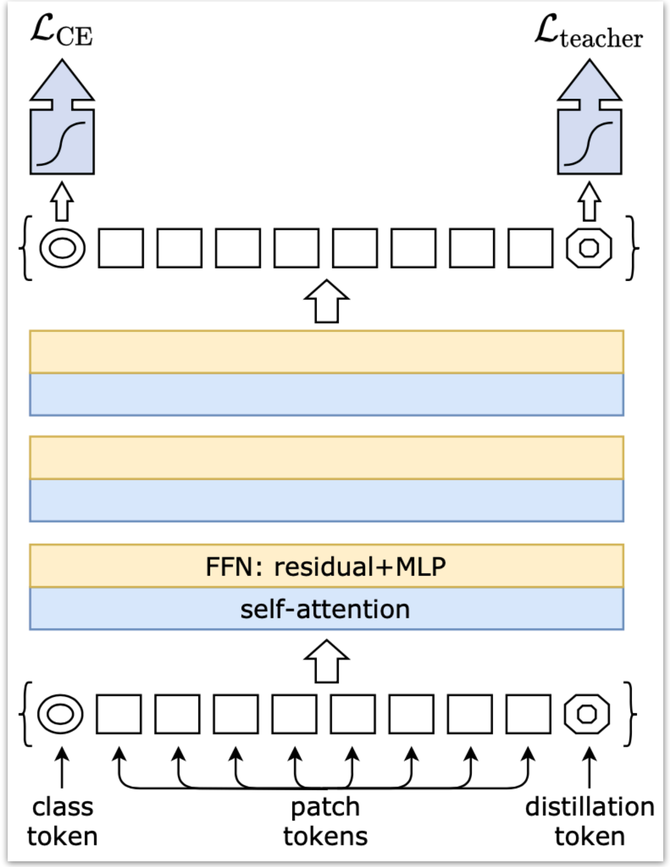
Soft Distillation

$$\mathcal{L}_{\text{global}} = (1-\lambda)\mathcal{L}_{\text{CE}}(\psi(Z_s), y) + \lambda\tau^2 \text{KL}(\psi(Z_s/\tau), \psi(Z_t/\tau))$$

$\psi \rightarrow$ softmax

$\tau \rightarrow$ temperature

$Z_s, Z_t \rightarrow$ student and teacher logits



Hard-label Distillation

$$y_t = \text{argmax}_c Z_t(c)$$

$$\mathcal{L}^{\text{hardDistill}}_{\text{global}} = \frac{1}{2}\mathcal{L}_{\text{CE}}(\psi(Z_s), y) + \frac{1}{2}\mathcal{L}_{\text{CE}}(\psi(Z_s), y_t)$$

Touvron, Hugo, et al. "Training data-efficient image transformers & distillation through attention." *International Conference on Machine Learning*. PMLR, 2021.

Boulder

Questions?