MixMatch: A Holistic Approach to Semi-Supervised Learning

David Berthelot, Nicholas Carlini, Ian Goodfellow
Avital Oliver, Nicolas Papernot, Colin Raffel

Google Research

NIPS 2019, 193회 인용

Park, MinKyu

2020.07.08

Dongguk University

Artificial Intelligence Laboratory

Variations on MixMatch

- These include:

  1) ReMixMatch: Semi-Supervised Learning with Distribution Alignment and Augmentation Anchoring https://arxiv.org/abs/1911.09785

  2) MetaMixUp: Learning Adaptive Interpolation Policy of MixUp with Meta-Learning https://arxiv.org/abs/1908.10059

  3) EnAET: Self-Trained Ensemble AutoEncoding Transformations for Semi-Supervised Learning https://arxiv.org/abs/1911.09265

  4) RealMix: Towards Realistic Semi-Supervised Deep Learning Algorithms https://arxiv.org/abs/1912.08766

- Key prerequisite reference
  - mixup: Beyond Empirical Risk Minimization https://arxiv.org/abs/1710.09412

Semi-Supervised Learning

- Using labelled as well as unlabelled data to perform certain learning tasks

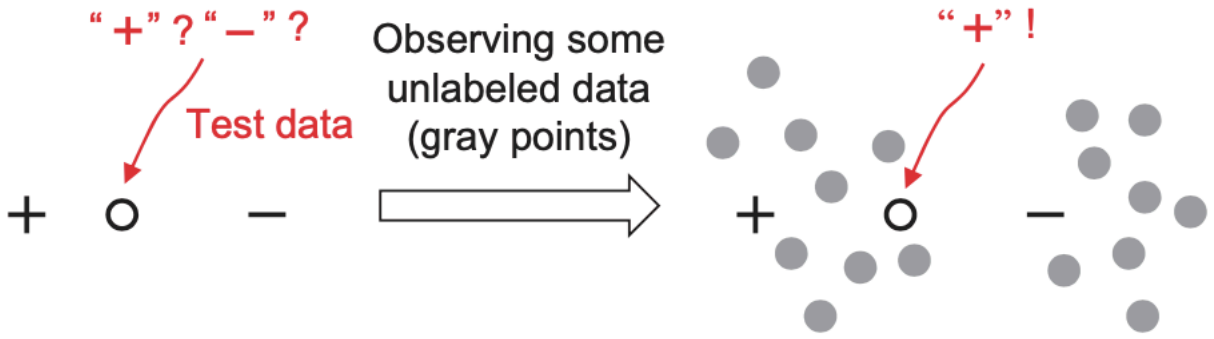


Figure 3. Illustration of the usefulness of unlabeled data.

SSL: Representative approaches

- **Generative methods**
  - Using a generative model for the classifier and employing EM to model the label estimation or parameter estimation process
- **S3VMs (Semi-Supervised SVMs)**
  - Using unlabeled data to adjust the decision boundary such that it goes through the less dense region
- **Graph-based methods**
  - Using unlabeled data to regularize the learning process via graph regularization
- **Disagreement-based methods**
  - multiple learners are trained for the task and the disagreements among the learners are exploited during the SSL process
- **Semi-supervised clustering**
  - a technique that partitions unlabeled data by making use of domain knowledge, usually expressed as pairwise constraints among instances or just as an additional set of labeled instances

Semi-Supervised Learning by Disagreement

- Co-training

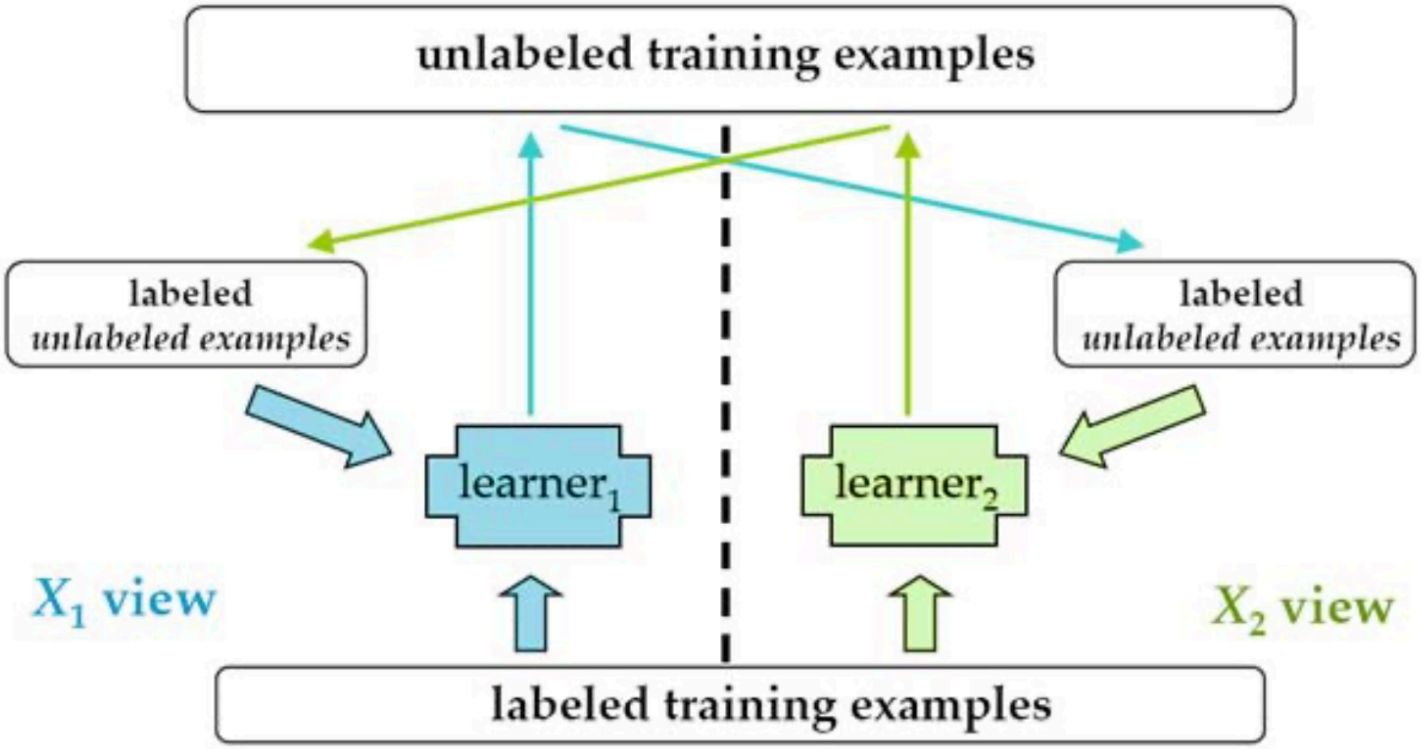
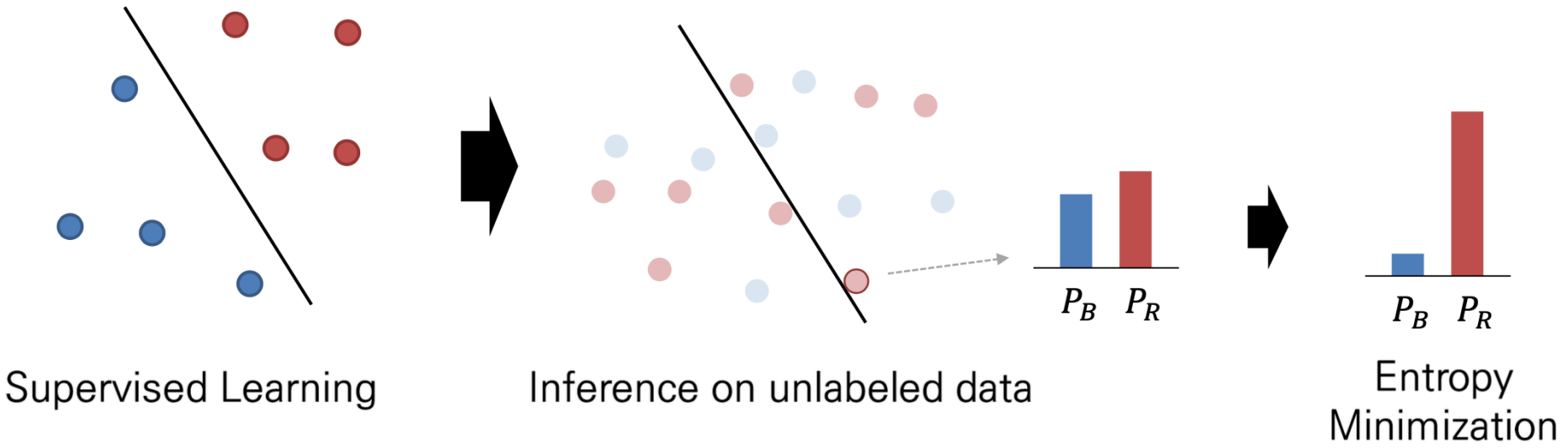


Fig. 1. An illustration of the co-training procedure

Recent works for SSL

- Entropy Minimization
  - encourages the model to output confident predictions on unlabeled data



출처 : https://www.youtube.com/watch?v=nSJP7bn2D1U

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

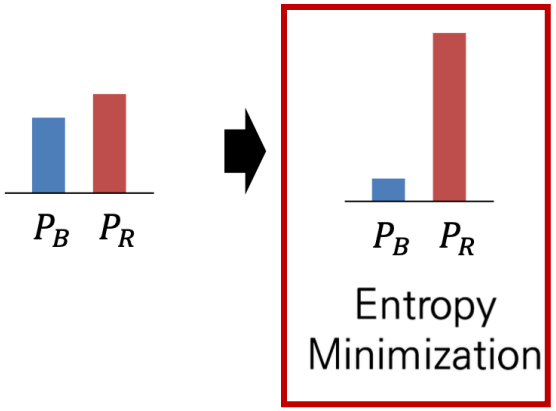
Recent works for SSL

- Entropy Minimization
  - encourages the model to output confident predictions on unlabeled data

Softmax Temperature

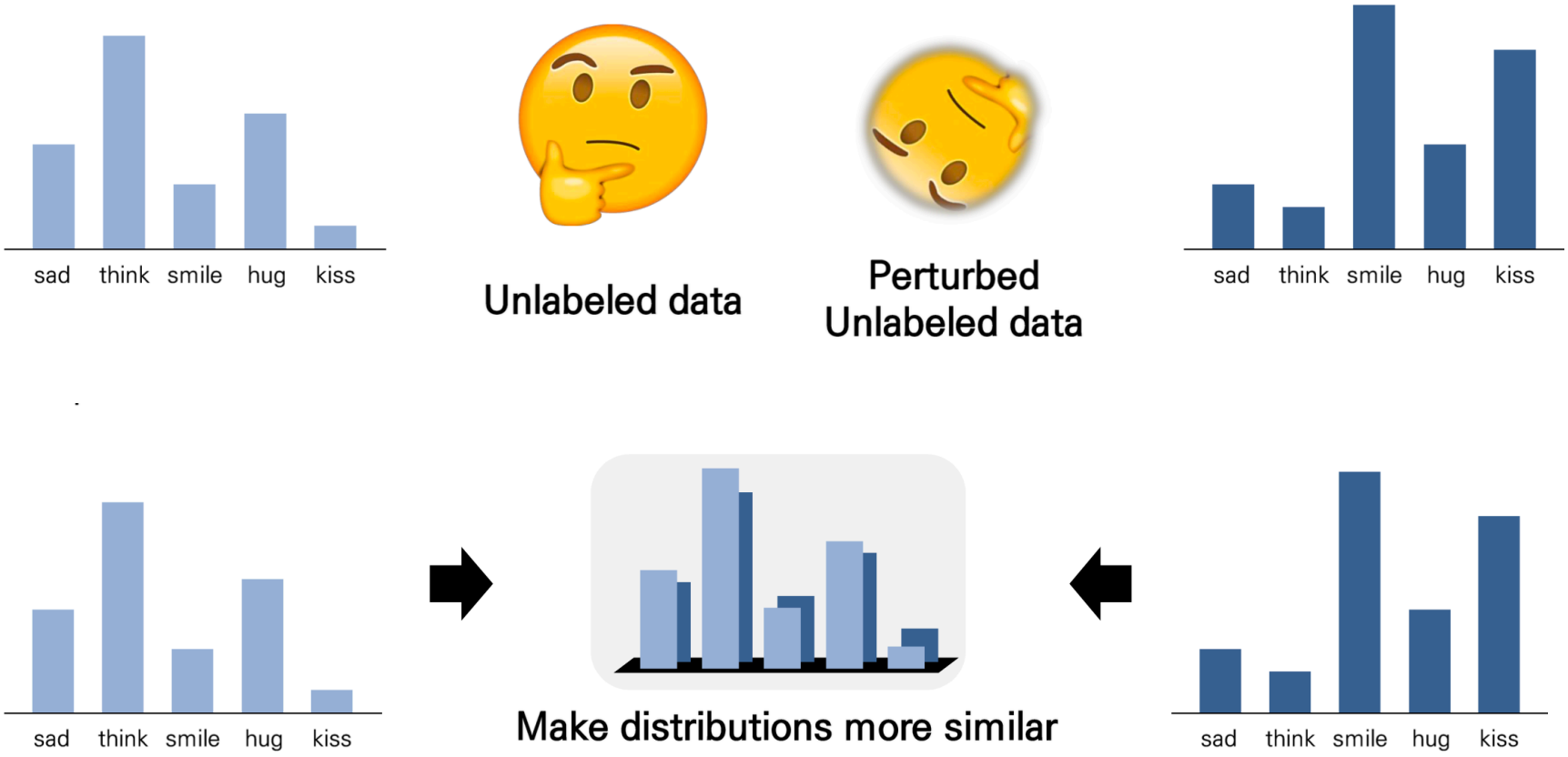
$$\frac{p_i}{\sum p_j} \Rightarrow \frac{p_i^{\frac{1}{T}}}{\sum p_j^{\frac{1}{T}}}$$

Low entropy
for low temperature (T → 0)



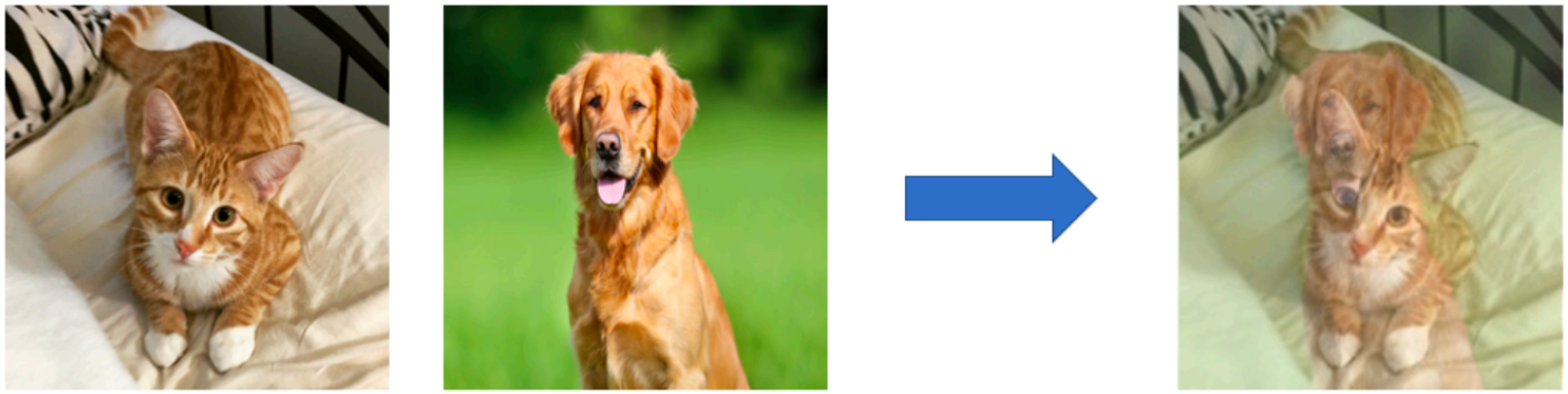
Recent works for SSL

- Consistency Regularization
  - encourages the model to produce the same output distribution when its inputs are perturbed



Recent works for SSL

- Generic Regularization (Traditional Regularization)
  - encourages the model to generalize well and avoid overfitting the training data.



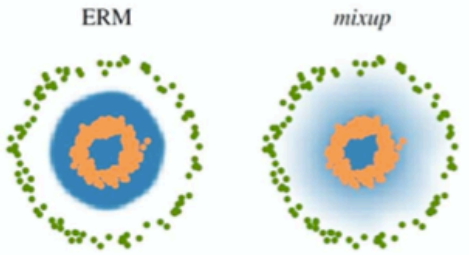
Recent works for SSL

- Generic Regularization (Traditional Regularization)
  - use weight decay which penalizes the L2 norm of the model parameters
  - MixUp : Encourage the model to have strictly linear behavior between examples.

$$\tilde{x} = \lambda x_i + (1-\lambda)x_j, \quad \text{where } x_i, x_j \text{ are raw input vectors}$$
$$\tilde{y} = \lambda y_i + (1-\lambda)y_j, \quad \text{where } y_i, y_j \text{ are one-hot label encodings}$$

```
# y1, y2 should be one-hot vectors
for (x1, y1), (x2, y2) in zip(loader1, loader2):
    lam = numpy.random.beta(alpha, alpha)
    x = Variable(lam * x1 + (1. - lam) * x2)
    y = Variable(lam * y1 + (1. - lam) * y2)
    optimizer.zero_grad()
    loss(net(x), y).backward()
    optimizer.step()
```

(a) One epoch of *mixup* training in PyTorch.



(b) Effect of *mixup* ($\alpha = 1$) on a toy problem. Green: Class 0. Orange: Class 1. Blue shading indicates $p(y = 1|x)$.

MixMatch: How to combine three concepts?

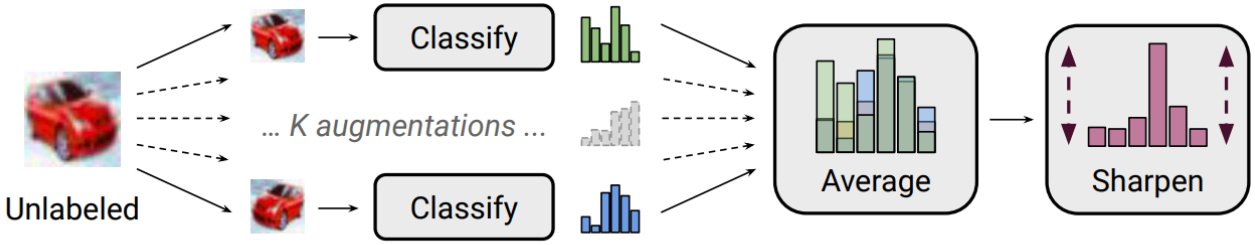


Figure 1: Diagram of the label guessing process used in MixMatch. Stochastic data augmentation is applied to an unlabeled image K times, and each augmented image is fed through the classifier. Then, the average of these K predictions is "sharpened" by adjusting the distribution's temperature. See algorithm 1 for a full description.

$$\mathcal{X}', \mathcal{U}' = \text{MixMatch}(\mathcal{X}, \mathcal{U}, T, K, \alpha) \tag{2}$$

$$\mathcal{L}_{\mathcal{X}} = \frac{1}{|\mathcal{X}'|} \sum_{x,p \in \mathcal{X}'} \text{H}(p, \text{p}_{\text{model}}(y \mid x; \theta)) \tag{3}$$

$$\mathcal{L}_{\mathcal{U}} = \frac{1}{L|\mathcal{U}'|} \sum_{u,q \in \mathcal{U}'} \|q - \text{p}_{\text{model}}(y \mid u; \theta)\|_2^2 \tag{4}$$

$$\mathcal{L} = \mathcal{L}_{\mathcal{X}} + \lambda_{\mathcal{U}} \mathcal{L}_{\mathcal{U}} \tag{5}$$

where $\text{H}(p, q)$ is the cross-entropy between distributions p and q, and T, K, α, and $\lambda_{\mathcal{U}}$ are hyperparameters described below. The full MixMatch algorithm is provided in algorithm 1, and a diagram of the label guessing process is shown in fig. 1. Next, we describe each part of MixMatch.

MixMatch: Algorithm

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 $\mathcal{X}', \mathcal{U}'$

MixMatch: Data Augmentation

- Stochastic transformation of the datapoint in such a way that its label remains unchanged.
- Apply data augmentation on both labeled and unlabeled data (Rotation, Flip, etc)

2: **for** $b = 1$ **to** B **do**
3: $\hat{x}_b = \text{Augment}(x_b)$ // *Apply data augmentation to* x_b
4: **for** $k = 1$ **to** K **do**
5: $\hat{u}_{b,k} = \text{Augment}(u_b)$ // *Apply* k^{th} *round of data augmentation to* u_b

MixMatch: Label Guessing

- compute the average of the model's predicted class distributions across all the K augmentations of $\mathcal{U}b$

$$\bar{q}_b = \frac{1}{K} \sum_{k=1}^{K} \mathrm{p}_{\mathrm{model}}(y \mid \hat{u}_{b,k}; \theta)$$

MixMatch: Sharpening

- Add Sharpening for entropy minimization
- apply a sharpening function to reduce the entropy of the label distribution
- use the common approach of adjusting the "temperature" of this categorical distribution

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

MixMatch: MixUp

- Apply MixUp both to labeled and unlabeled samples with label guessing.

$$\lambda \sim \mathrm{Beta}(\alpha, \alpha)$$
$$\lambda' = \max(\lambda, 1-\lambda)$$
$$x' = \lambda' x_1 + (1-\lambda') x_2$$
$$p' = \lambda' p_1 + (1-\lambda') p_2$$

$\hat{\mathcal{X}} = \big((\hat{x}_b, p_b); b \in (1, \ldots, B)\big)$ // *Augmented labeled examples and their labels*

$\hat{\mathcal{U}} = \big((\hat{u}_{b,k}, q_b); b \in (1, \ldots, B), k \in (1, \ldots, K)\big)$ // *Augmented unlabeled examples, guessed labels*

$\mathcal{W} = \mathrm{Shuffle}\big(\mathrm{Concat}(\hat{\mathcal{X}}, \hat{\mathcal{U}})\big)$ // *Combine and shuffle labeled and unlabeled data*

$\mathcal{X}' = \big(\mathrm{MixUp}(\hat{\mathcal{X}}_i, \mathcal{W}_i); i \in (1, \ldots, |\hat{\mathcal{X}}|)\big)$ // *Apply* MixUp *to labeled data and entries from* $\mathcal{W}$

$\mathcal{U}' = \big(\mathrm{MixUp}(\hat{\mathcal{U}}_i, \mathcal{W}_{i+|\hat{\mathcal{X}}|}); i \in (1, \ldots, |\hat{\mathcal{U}}|)\big)$ // *Apply* MixUp *to unlabeled data and the rest of* $\mathcal{W}$

MixMatch: Hyper-parameters

- T : Temperature for sharpening ($T = 0.5$)
- K : Number of data augmentation for unlabeled samples ($K = 2$)

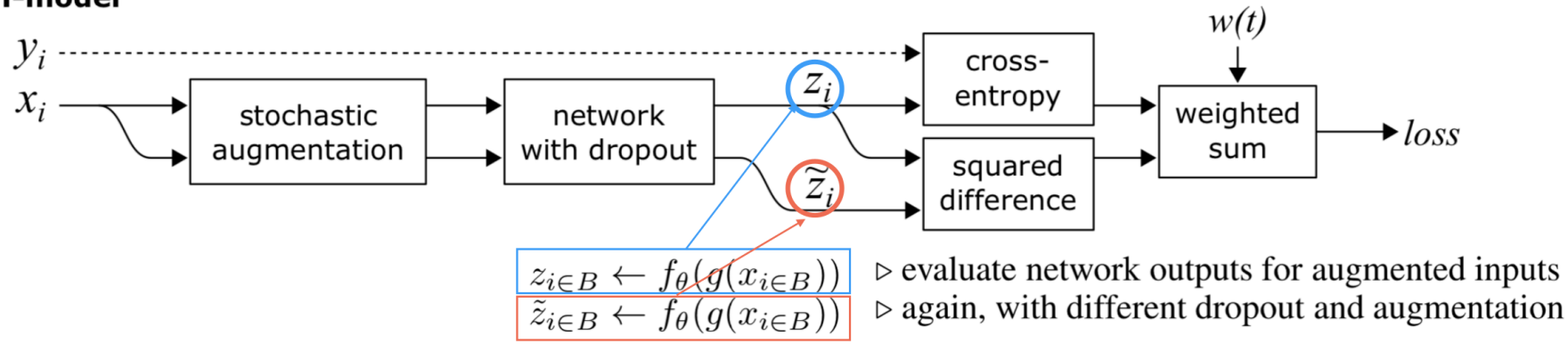
- λ_U : Unsupervised loss weigh ($\lambda_U = 100$)
- α : MixUp parameter ($\alpha = 0.75$)

Experiments

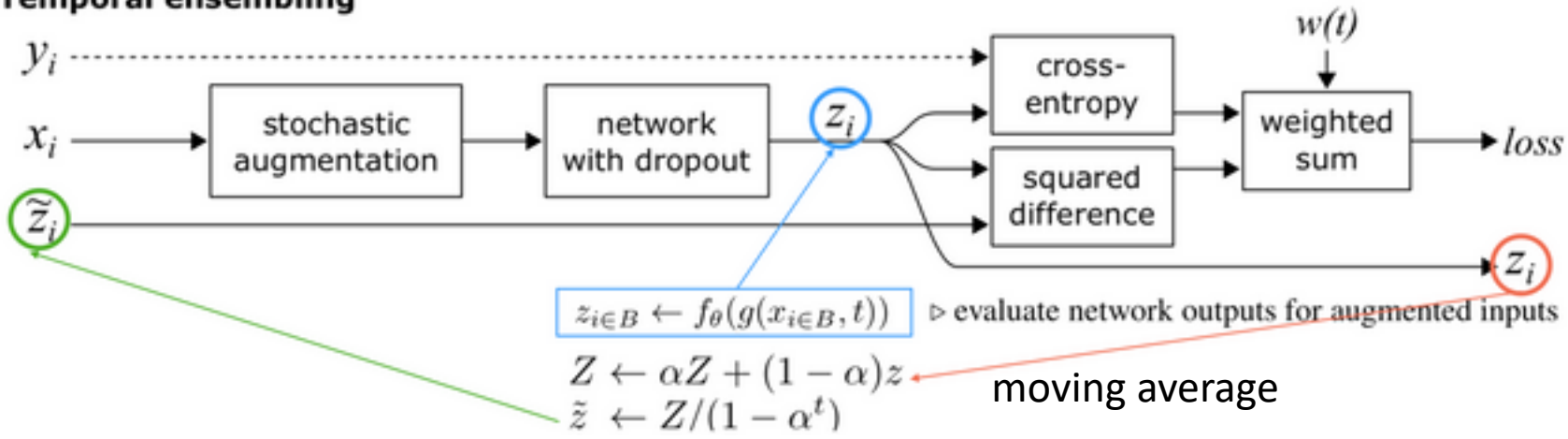
- Implementation details
  - in all experiments we use the “Wide ResNet-28” model
- four standard benchmark datasets
  - CIFAR-10 and CIFAR-100 , SVHN , and STL-10
- Baseline Methods
  - Π-Model (ICLR 2017) ,
  - Mean Teacher (NIPS 2017) ,
  - Virtual Adversarial Training (ICLR 2017) ,
  - Pseudo-Label
  - MixUp

Π-Model

Π-model



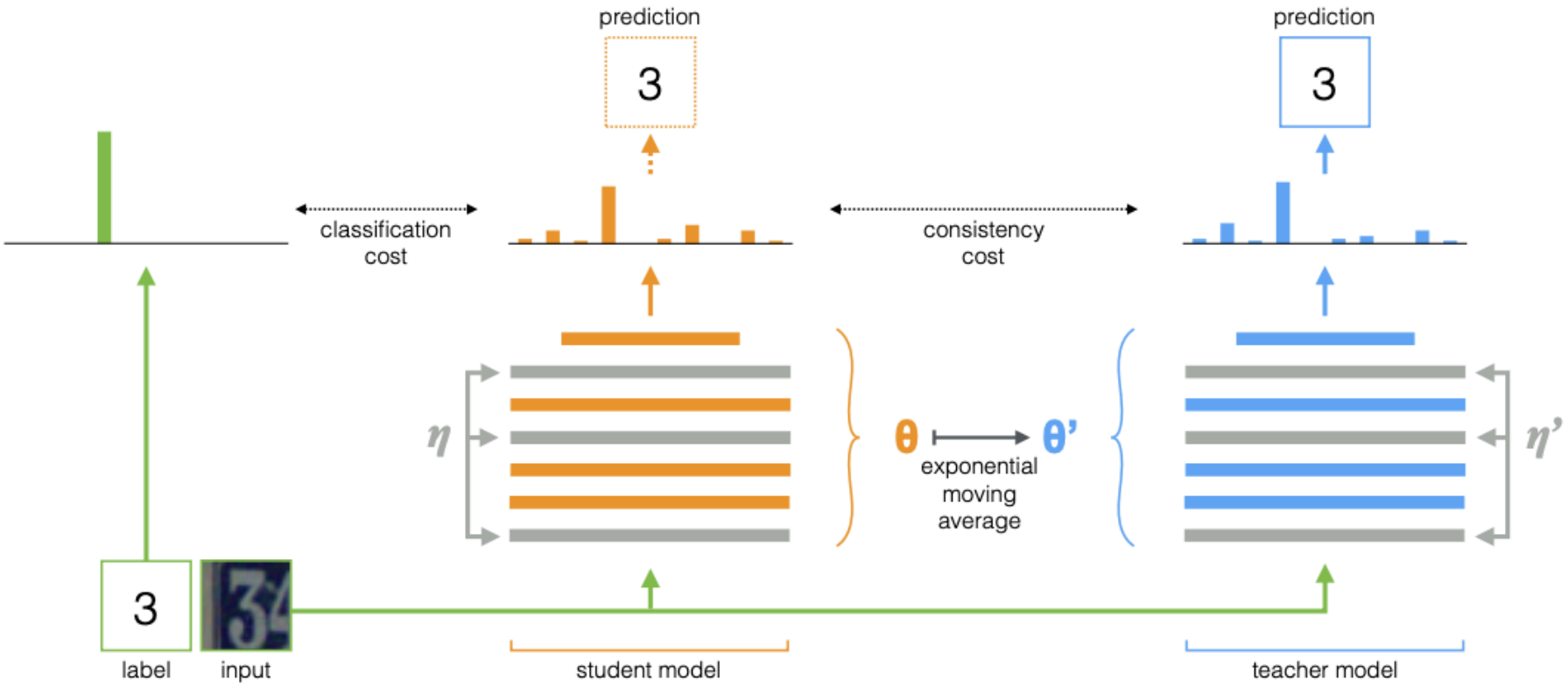
Temporal ensembling



source : [Laine & Aila, 2017]

Mean Teacher

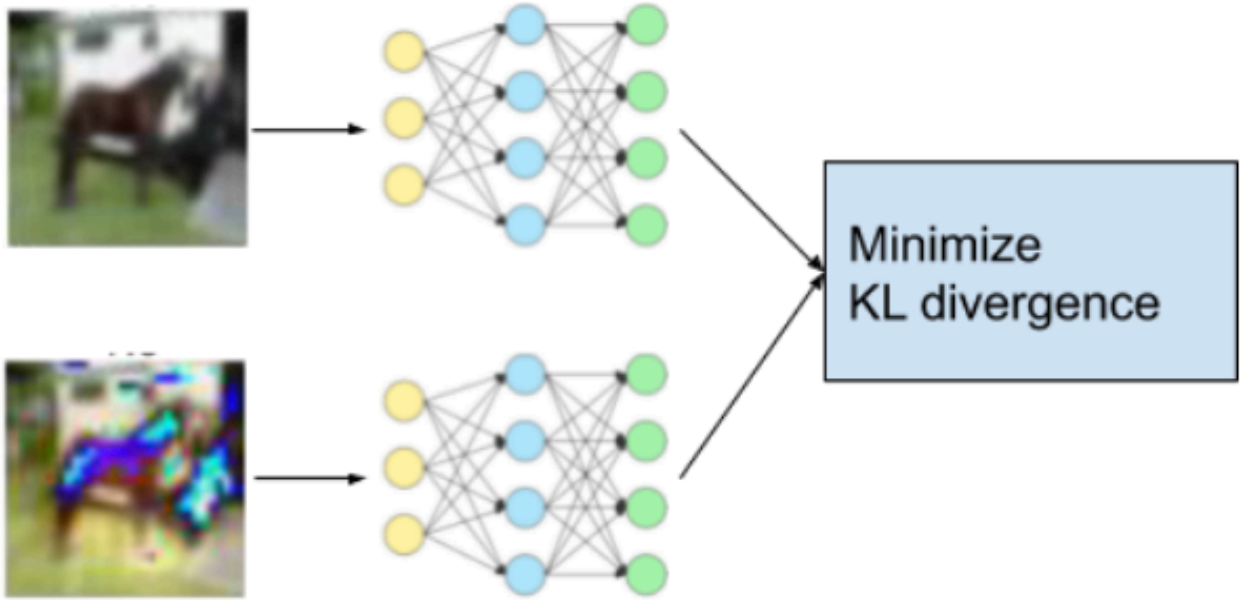
- averaging model weights instead of predictions



* source : [Tarvainen & Harri Valpola, 2017]

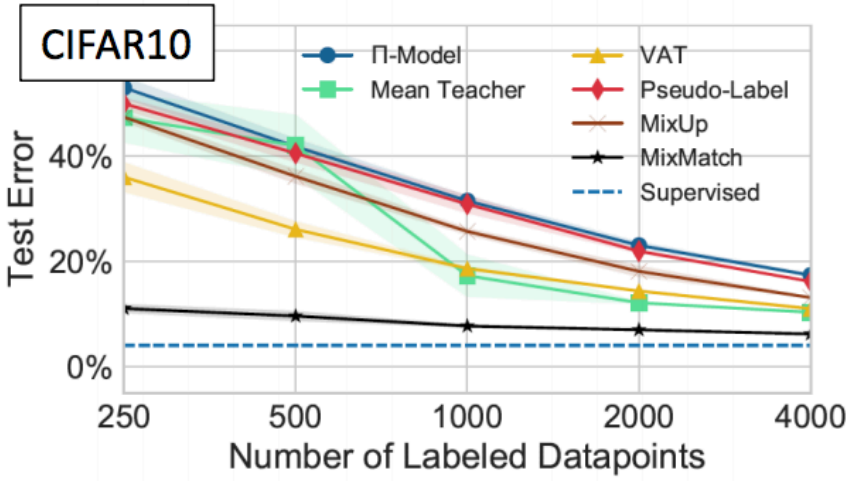
Virtual Adversarial Training(VAT)

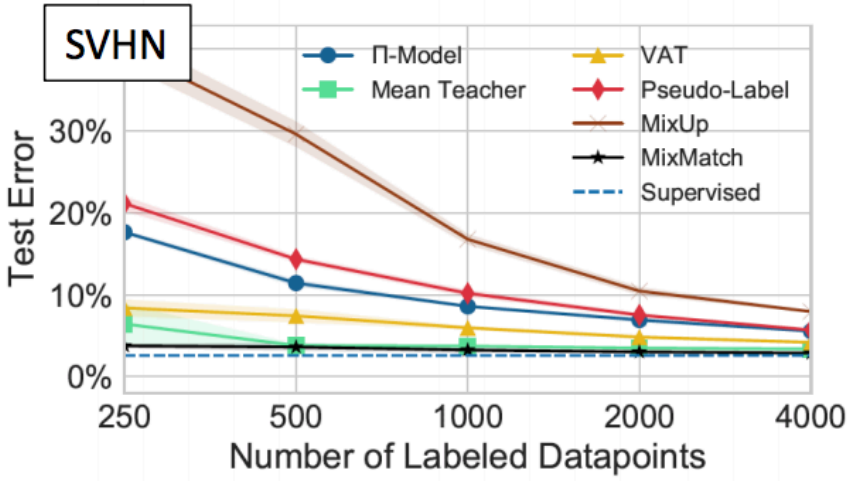
Step 1 : Generate the adversarial image



Step 2: Minimize the KL divergence

- This simple method achieves **state-of-the-art** performance on benchmark datasets





- Ablation study: **all components are important**

Ablation	250 labels	4000 labels
MixMatch	11.80	6.00
MixMatch without distribution averaging ($K = 1$)	17.09	8.06
MixMatch with $K = 3$	11.55	6.23
MixMatch with $K = 4$	12.45	5.88
MixMatch without temperature sharpening ($T = 1$)	27.83	10.59
MixMatch with parameter EMA	11.86	6.47
MixMatch without MixUp	39.11	10.97
MixMatch with MixUp on labeled only	32.16	9.22
MixMatch with MixUp on unlabeled only	12.35	6.83
MixMatch with MixUp on separate labeled and unlabeled	12.26	6.50
Interpolation Consistency Training [45]	38.60	6.81

Averaging

Mixup

Sharpening

References

[Laine & Aila, 2017] Temporal Ensembling for Semi-Supervised Learning, ICML2017

[Tarvainen & Harri Valpola, 2017] Mean teachers are better role models: Weight-averaged consistency targ ets improve semi-supervised deep learning results, NIPS 2017

[Miyato et al., 2017] Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Superv ised Learning, TPAMI 2019

[Berthelot et al., 2019] MixMatch: A Holistic Approach to Semi-Supervised Learning, NIPS 2019