FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence

•••

By Kihyuk Sohn, David Berthelot, Chun-Liang, Li Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang and Colin Raffel

Presented by Gwen Legate and Balaji Balasubramanian

Claims

- Less complex and more accurate
- Achieves 94:93% accuracy on CIFAR-10 with 250 labeled examples compared to the previous state-of-the-art of 93:73%[1]
- It's simplicity allows for fewer hyper parameters allowing for an extensive ablation study

Novel Ideas

- Combination of pseudo labelling and consistency regularization
- Uses separate weak/strong augmentation when doing consistency regularization

[1] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel. Remixmatch:Semi-supervised learning with distribution matching and augmentation anchoring. In Eighth International Conference on Learning Representations, 2020

Consistency Regularization

Relies on the assumption that a model should output similar predictions when passed perturbed versions of the same image

$$\sum_{b=1}^{\mu B} \|p_{\mathrm{m}}(y|\,\alpha(u_b)) - p_{\mathrm{m}}(y|\,\alpha(u_b))\|_2^2$$

Pseudo Labeling

Uses the model to obtain artificial labels

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

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

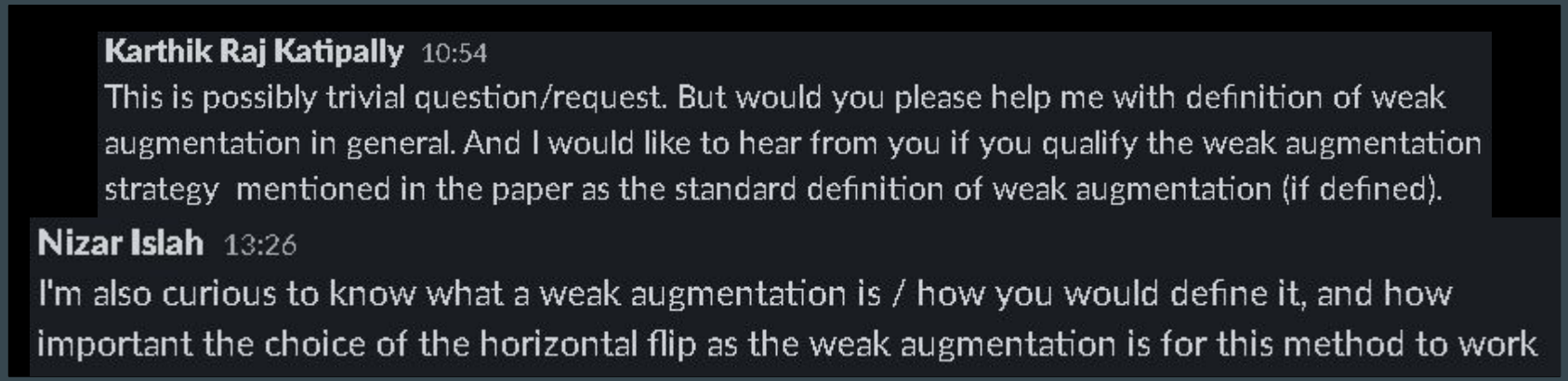
Augmentation

Weak Augmentation α(.)

- Flip images horizontally → p(0.5)
- Translate images vertically and horizontally by up to 12.5%

Strong Augmentation A(.)

- Use two methods based on AutoAugment[1] followed by Cutout[2]



[1]E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. Autoaugment: Learning augmentation strategies from data. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019
[2] T. DeVries and G. W. Taylor. Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552, 2017.

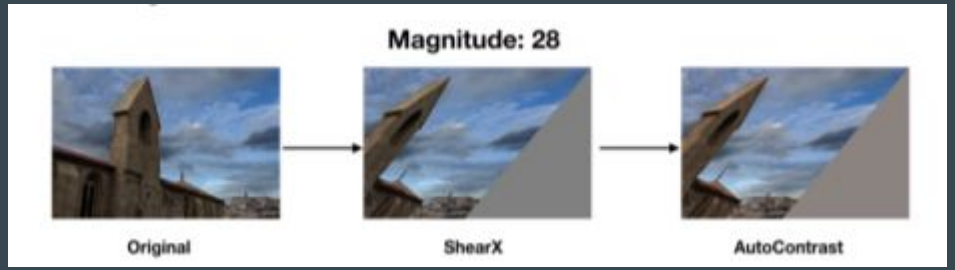
AutoAugment

AutoAugment uses reinforcement learning to find an augmentation policy

- Policy selected will consist of two augmentation types applied in sequence
- Requires labeled data → problem in SSL with limited labels

FixMatch uses variants of AutoAugment:

- RandAugment[3] → still applies two types of augmentation in sequence but replaces learned policy with random selection
- CTAugment[4] → randomly selects augmentations but infers magnitude of distortion



[3] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. Randaugment: Practical automated data augmentation with a reduced search space. arXiv preprint arXiv:1909.13719, 2019.
[4] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In Eighth International Conference on Learning Representations, 2020

Cutout

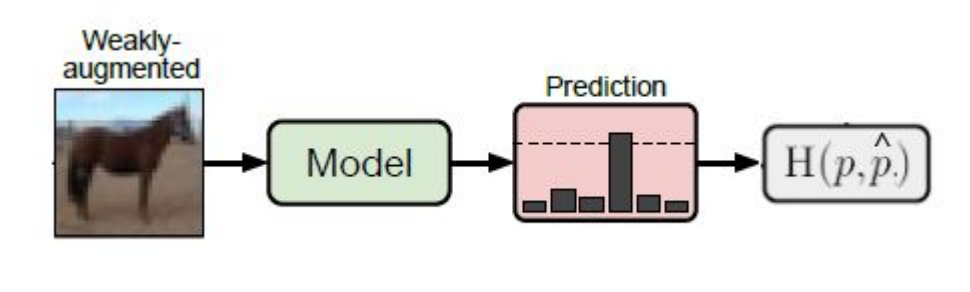
- Masks out arbitrary regions of the image during training
- Size of the cutout is a more important than the shape, so all cutouts are square patches
- A pixel is randomly selected as the center of the cutout

FixMatch Algorithm

- Calculate cross entropy loss for labeled data

labeled image set

$$\mathcal{X} = \{(x_b, p_b) : b \in (1, \ldots, B)\}$$

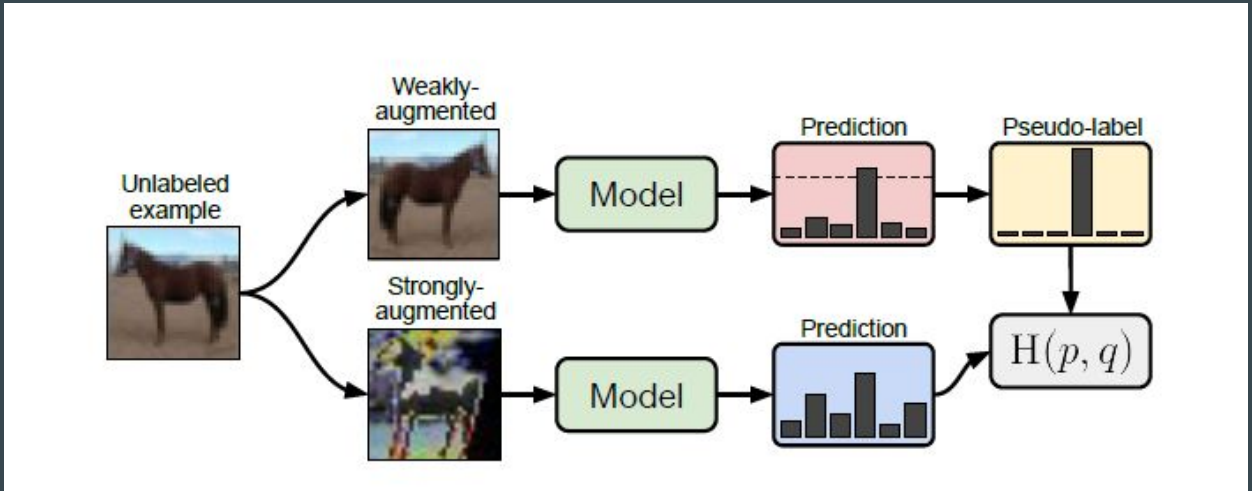


$$\ell_s = \frac{1}{B} \sum_{b=1}^{B} \mathrm{H}(p_b, p_{\mathrm{m}}(y \mid \alpha(x_b)))$$

FixMatch Algorithm

- Apply weak augmentation to each input u_b
- Compute pseudo labels q
- Calculate cross entropy loss for unlabeled data

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



unlabeled image set

$$\mathcal{U} = \{u_b : b \in (1, \ldots, \mu B)\}$$

Fixmatch Algorithm

- Compute l_s (1:B)

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

- For each U (1:μB):
  - Apply weak augmentation and get prediction
- Compute l_u (1: μB)

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

- Compute loss for mini batch iteration

$$\ell_s + \lambda_u \ell_u$$

FixMatch Algorithm

Reza Davari 15:07
What are your thoughts about doing this procedure in an iterative way. So kinda like Noisy Student paper, and progressively adding more noise (stronger augmentation) to the training.

2. The process of generating pseudo-labels on weak augmentation then predict on strong augmentation seems like curriculum learning. Is this right? If it is, then do you think repeating the process iteratively like noisy-student would bring better results?

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

Frequently below τ early in training

Thresholding

τ	mask rate	impurity	error rate
0.25	100.00	6.39	6.40
0.5	100.00	5.40	5.87
0.75	99.82	5.35	5.09
0.85	99.31	4.32	5.12
0.9	99.21	3.85	4.90
0.95	98.13	3.47	4.84
0.97	96.35	2.30	5.00
0.99	92.14	2.06	5.05

Mask rate is percentage of images with confidence > τ

Impurity is percentage of samples with incorrect predictions whose confidence was > τ

More on this in the ablation

Kavin Patel 13:30

Section B.2 indicates that since there is a tradeoff between quality and quantity of pseudo-labels, the confirmation error is introduced. I was wondering, if there can be some ways to mitigate this error? Is there any future work done on this issue?

Regularization

- Regularization is an important factor in FixMatch
- Cosine weight decay regularization is used

$$\eta' = \eta \cos \frac{7\pi k}{16K}$$

Effect of regularization is examined in the ablation study

Extensions: Augmentation Anchoring

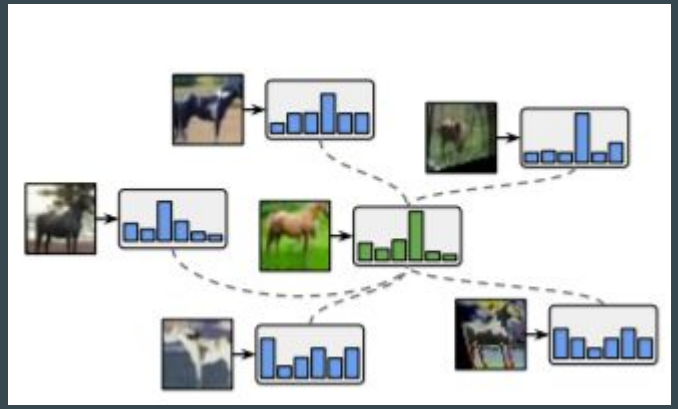
- Augmentation anchoring is applied for consistency regularization of the unlabeled examples[4]
- Hyperparameters: M=4, μ=4

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

With augmentation anchoring

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

Unaltered FixMatch



Reduces the error rate on CIFAR-10 with 250 labeled examples from 5:07% to 4:81%.

Extensions: Datatype-Agnostic Data Augmentation

Application to different problem domains will require a novel augmentation strategy.

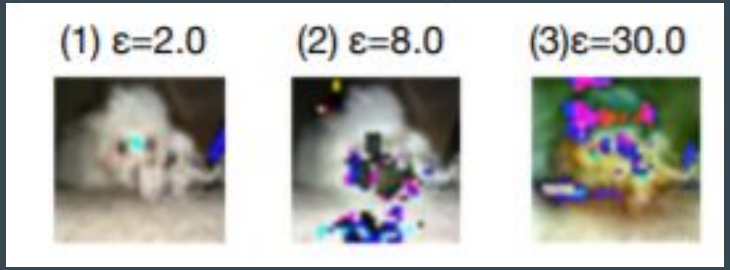
Data type-agnostic data augmentation methods:

- Virtual Adversarial Training (VAT)[5]
- MixUp[6]

[5] T. Miyato, S.-i. Maeda, S. Ishii, and M. Koyama. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. IEEE transactions on pattern analysis and machine intelligence, 2018
[6] Zhang, H., Cissé, M., Dauphin, Y., & Lopez-Paz, D. (2018). mixup: Beyond Empirical Risk Minimization. ArXiv, abs/1710.09412.

Virtual Adversarial Training (VAT)

- Requires pairs of data points close in the input space, but far in the model output space.
- For a given input, a perturbation is made that gives a different output.
- Then the model is penalized for sensitivity to the perturbation → makes outputs closer to each other



MixUp

For two data points (x_i, y_i) and (x_j, y_j)

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

λ=[0,1] defined as a function of α

[7] Kulakov, Artyom. “2 Reasons to Use Mixup When Training Your Deep Learning Models.” *Medium*, Towards Data Science, 30 Aug. 2020, https://towardsdatascience.com/2-reasons-to-use-mixup-when-training-yor-deep-learning-models-58728f15c559.

Extensions: Datatype-Agnostic Data Augmentation

For MixUp, instead of mixing both input and label, mixed random pairs of inputs

Hyperparameters

VAT	τ=0.5
MixUp	α=9

FixMatch is able to generalize to different data augmentation strategies.

FixMatch + Input MixUp	MixMatch	FixMatch + VAT	VAT
10.99±0.50	11.05±0.86	32.26±2.24	36.03±2.82

Related Work: Noisy Student

Arash Ash 12:28

I am really curious whether this has any benefit compared to noisy student paper? to me, it looks like FixMatch is just one iteration of the Noisy Student paper with less tricks. since this paper is coming after, I really wonder whats their contribution is compared to previous work? (edited)

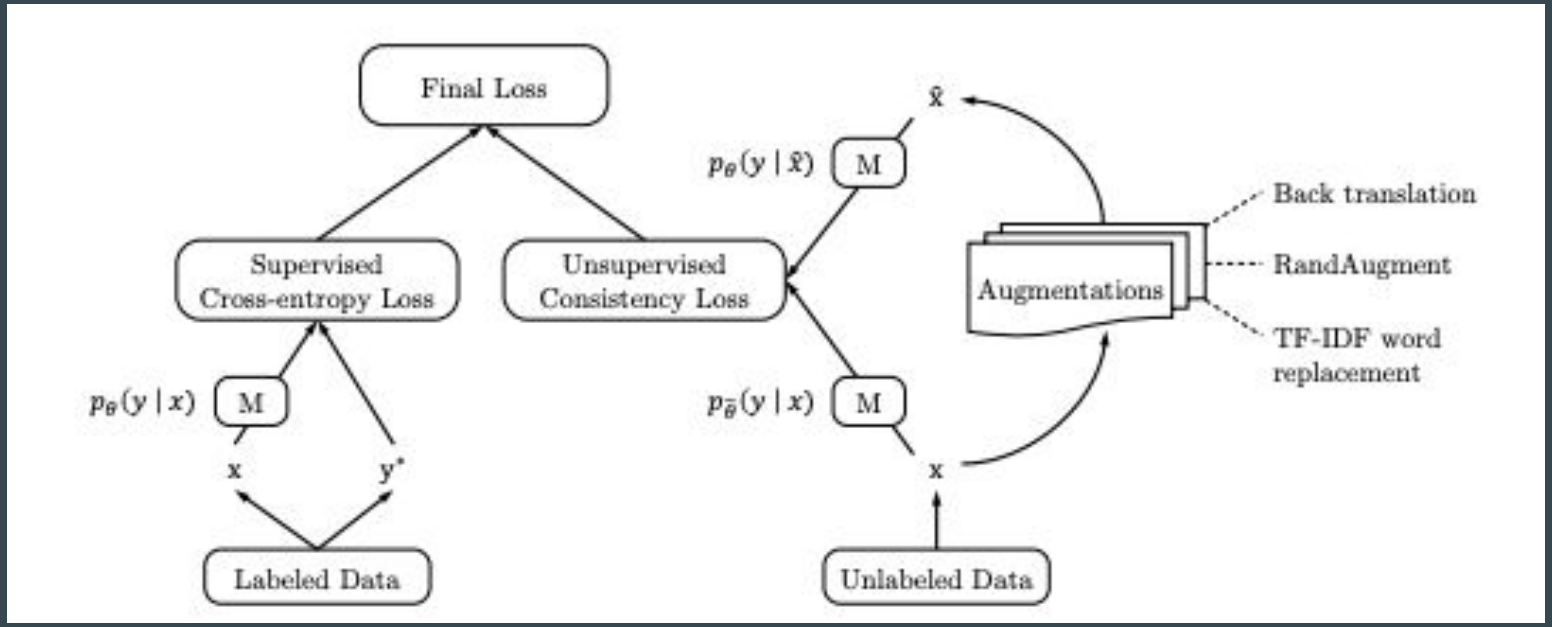
$$\ell = \frac{1}{B}\sum_{b=1}^{B} \mathrm{H}(p_b, p_{\mathrm{m}}(y \mid \alpha(x_b))) + \lambda \frac{1}{\mu B}\sum_{b=1}^{\mu B} \mathbb{1}(\max(q_b) \geq \tau)\, \mathrm{H}(\hat{q}_b, p_{\mathrm{m}}(y \mid \mathcal{A}(u_b)))$$

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

Noisy Student

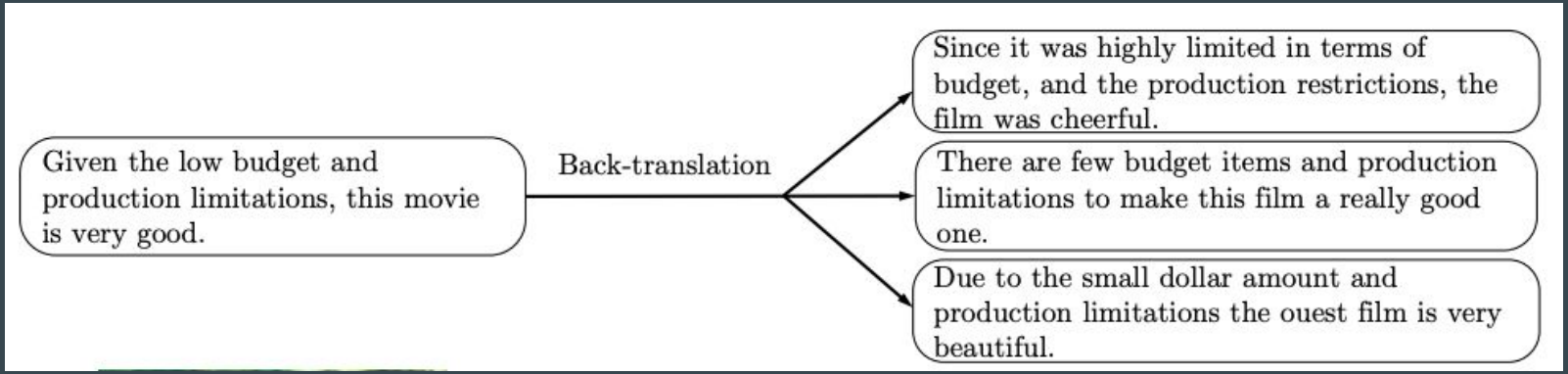
- data augmentation with RandAugment
- model noise: dropout and stochastic depth

UDA



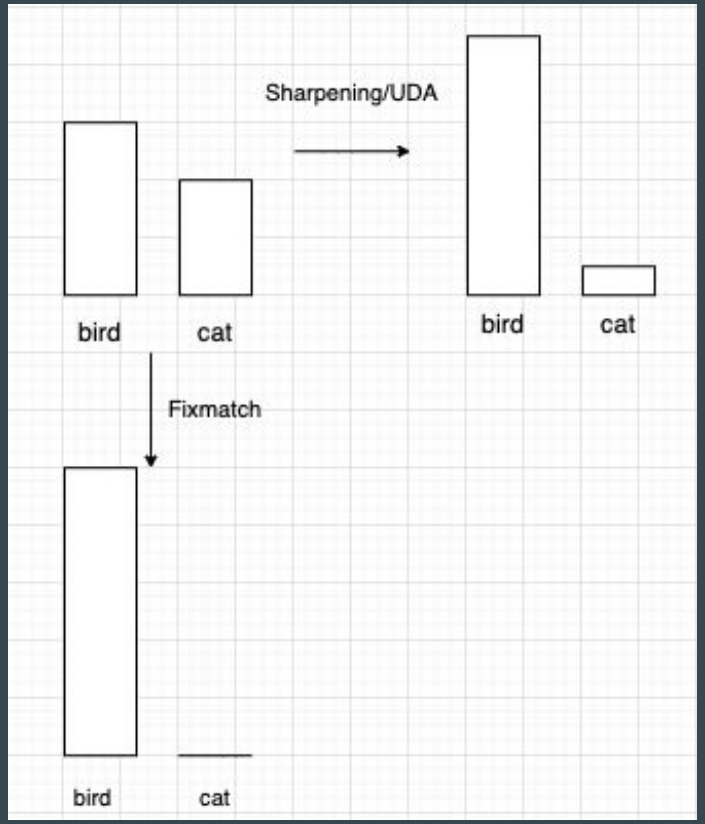
https://arxiv.org/pdf/1904.12848.pdf

Back-translation and TF-IDF Word Replacement



- Back-translation- English -> French -> English
- TF-IDF Word Replacement- Replace the low information (high frequency) words with a synonym and keep the high information (low frequency) words. Eg :- change production limitations to production restrictions.

https://arxiv.org/pdf/1904.12848.pdf



Sharpening

$$p_{\tilde{\theta}}^{(sharp)}(y \mid x) = \frac{\exp(z_y/\tau)}{\sum_{y'} \exp(z_{y'}/\tau)}$$

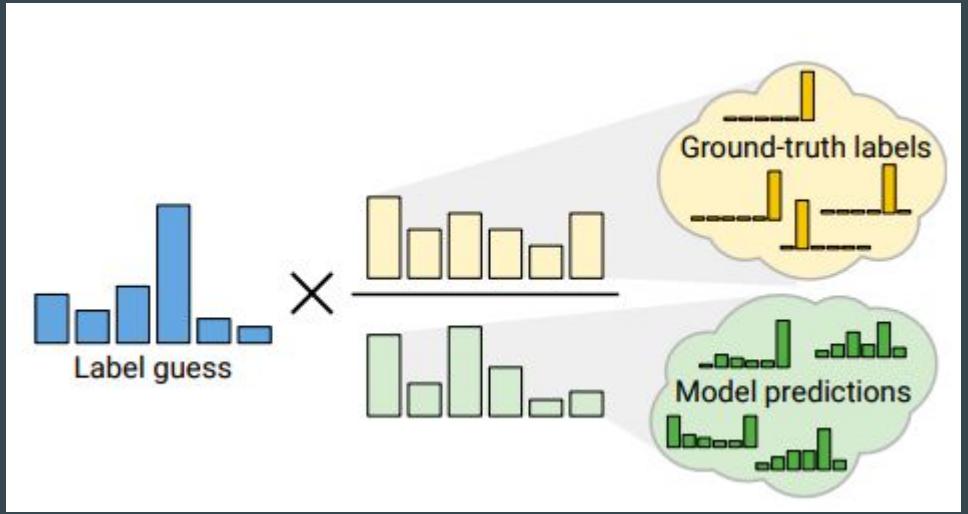
```
Probabilities = 0.6,0.4
T= 0.1, output= 0.88,0.12
T= 1, output= 0.55,0.45
T= 10, output= 0.505,0.495
```

https://arxiv.org/pdf/1904.12848.pdf

Can you describe what sharpening is and what does the temperature T in sharpening means?

ReMixMatch- MixUp, Distribution Alignment, Augmentation Anchoring

Distribution Alignment- Guessed label distribution is adjusted based on the ratio of ground-truth class distribution for the labelled data and moving average of model predictions on unlabeled data over the last 128 batches.



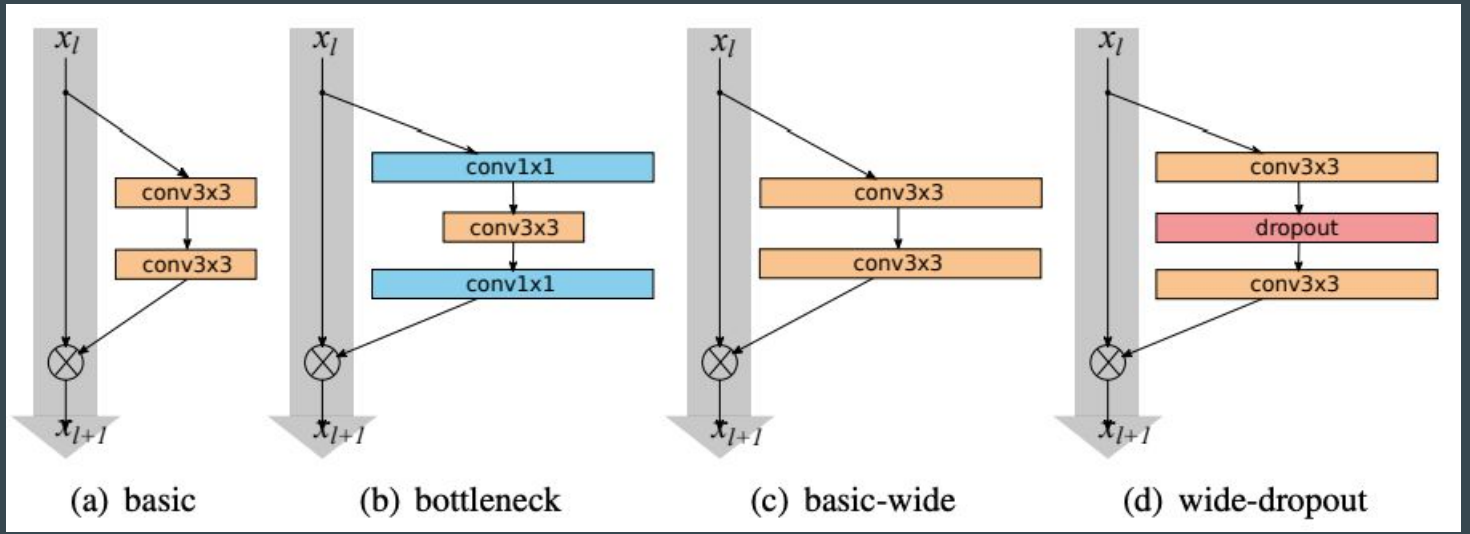
https://arxiv.org/pdf/1911.09785.pdf

Experiments

- The simplicity of FixMatch allows better hyperparameter tuning. This is the secret to the success of FixMatch.
- Experiments were performed on Cifar 10/100, STL-10, SVHN and Imagenet Datasets.
- This paper considered the model performance at setting with low number of labelled setting such as 25 and 4 images (older papers considered only 40 and 250 labelled image setting).

Wide Residual Networks

- Decrease the depth and increase the width of the model.
- A simple 16 Layer Wide Residual Network model outperformed all the previous state of resnet architectures including thousand layer ones (diminishing feature reuse).



https://arxiv.org/abs/1605.07146

Wide Residual Network Architectures

- Wide ResNet-28-2 was used for Cifar-10 and SVHN.
- Wide Resnet-28-8 for CIFAR-100. It has more channels due to higher number of labels.
- Wide Resnet-37-2 for STL-10. It has a larger depth due to larger image sizes.
- Cifar-10, SVHN and Cifar-100 have 32 x 32 images and STL-10 has 96 x 96 images.
- Resnet 50 was used for Imagenet dataset. Wide Resnet-50-k could have been tried.
- Efficient Net can be tried.

Hyperparameters used for Cifar 10/100, SVHN, STL 10

$$\lambda_u = 1,\ \eta = 0.03,\ \beta = 0.9,\ \tau = 0.95,\ \mu = 7,\ B = 64,\ K = 2^{20}$$

Similar hyperparameters were used for Cifar 10/100, STL-10 and SVHN datasets.

relative weight of unlabelled example to labelled example = 1

learning rate = 0.03 , Momentum = 0.9

Threshold = 0.95

Ratio of Unlabelled to labelled data = 7

Batch Size = 64

Total number of training steps = 2^20 = 1048576

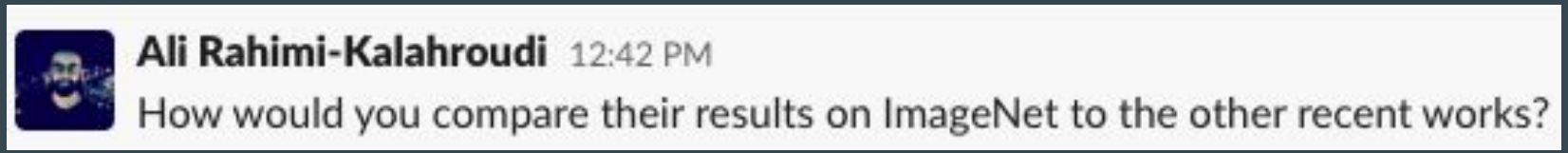
Result Table

	CIFAR-10			CIFAR-100			SVHN			STL-10
Method	40 labels	250 labels	4000 labels	400 labels	2500 labels	10000 labels	40 labels	250 labels	1000 labels	1000 labels
Π-Model	-	54.26±3.97	14.01±0.38	-	57.25±0.48	37.88±0.11	-	18.96±1.92	7.54±0.36	26.23±0.82
Pseudo-Labeling	-	49.78±0.43	16.09±0.28	-	57.38±0.46	36.21±0.19	-	20.21±1.09	9.94±0.61	27.99±0.83
Mean Teacher	-	32.32±2.30	9.19±0.19	-	53.91±0.57	35.83±0.24	-	3.57±0.11	3.42±0.07	21.43±2.39
MixMatch	47.54±11.50	11.05±0.86	6.42±0.10	67.61±1.32	39.94±0.37	28.31±0.33	42.55±14.53	3.98±0.23	3.50±0.28	10.41±0.61
UDA	29.05±5.93	8.82±1.08	4.88±0.18	59.28±0.88	33.13±0.22	24.50±0.25	52.63±20.51	5.69±2.76	**2.46**±0.24	7.66±0.56
ReMixMatch	**19.10**±9.64	**5.44**±0.05	4.72±0.13	**44.28**±2.06	**27.43**±0.31	**23.03**±0.56	**3.34**±0.20	**2.92**±0.48	2.65±0.08	**5.23**±0.45
FixMatch (RA)	**13.81**±3.37	**5.07**±0.65	**4.26**±0.05	48.85±1.75	28.29±0.11	**22.60**±0.12	**3.96**±2.17	**2.48**±0.38	**2.28**±0.11	7.98±1.50
FixMatch (CTA)	**11.39**±3.35	**5.07**±0.33	**4.31**±0.15	49.95±3.01	28.64±0.24	23.18±0.11	7.65±7.65	**2.64**±0.64	**2.36**±0.19	**5.17**±0.63

- FixMatch RA (RandAugment) and FixMatch CTA (CTAugment)- 40 label SVHN vs 40 Label Cifar-10
- FixMatch performs best for most cases except for Cifar-100 in which ReMixMatch performs the best due to Distribution Alignment (DA).

Results on Imagenet Dataset

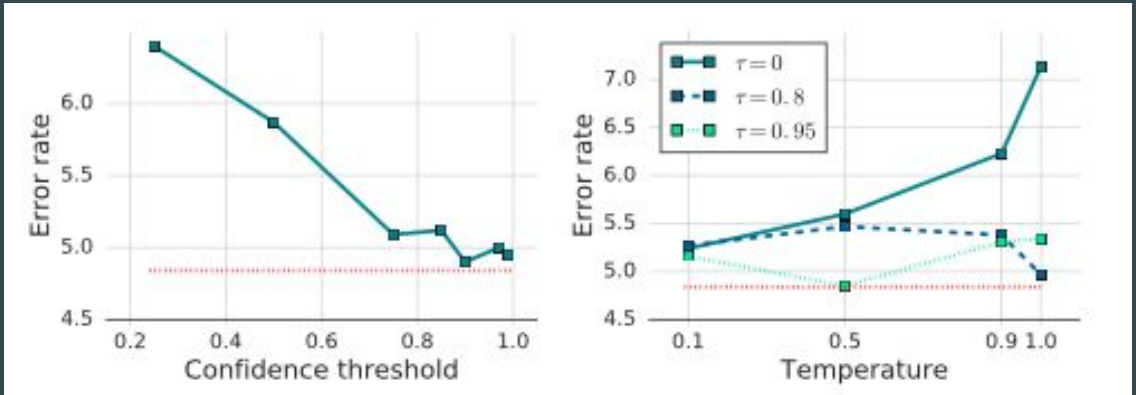
- 10% of the dataset was used as labelled data.
- Resnet-50 architecture with RandAugment.
- Fixmatch has top-1 error rate of 28.54 ± 0.52% which is 2.68% better than UDA.
- Noisy Student training has a top-1 error rate of 11.6%, but it was trained on 300M Unlabelled JFT images which is much larger than 14M Imagenet images.



Barely Supervised Learning

- Apply FixMatch to CIFAR-10 with only one example per class.
- Accuracy on the test set varies between 48.58% and 85.32% depending on the quality of the sample.
- Construct 8 dataset with samples in the decreasing order of quality (prototypical samples).
- Dataset with high quality labelled sample has an accuracy of 78% and the dataset with low quality labelled sample has an accuracy of 10%.

Ablation Study- Confidence Threshold and Temperature



Julia Hindel 3:42 PM

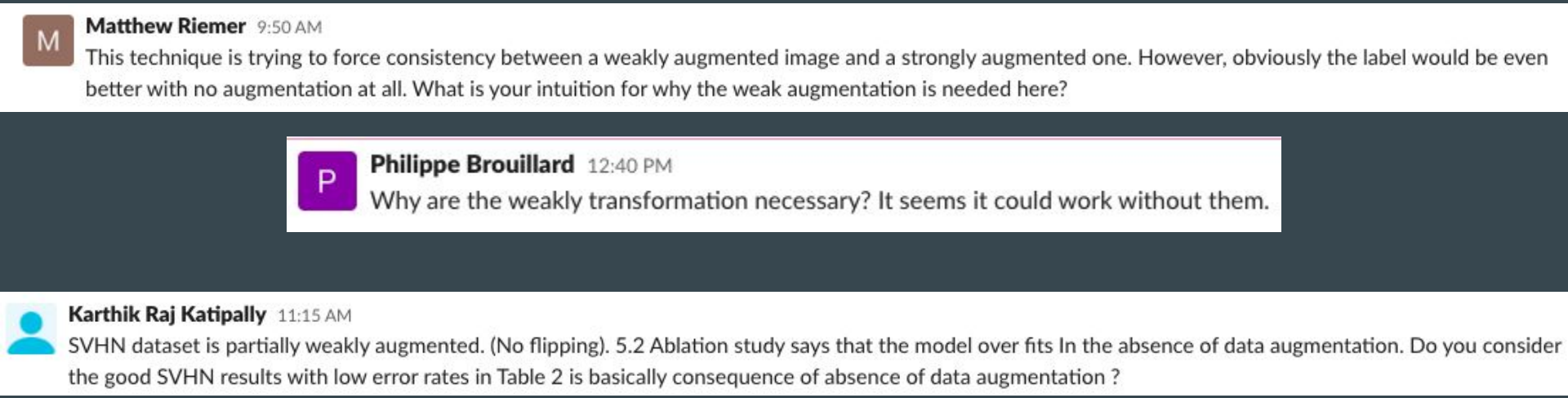
Do you have any reasoning behind how the temperature and confidence threshold are connected? I am a bit confused by the different shape of the graphs, in particular the increase and decrease in error rate with high temperatures.

$$p_{\tilde{\theta}}^{(sharp)}(y \mid x) = \frac{\exp(z_y/\tau)}{\sum_{y'} \exp(z_{y'}/\tau)}$$

```
Probabilities = 0.6,0.4
T= 0.1, output= 0.88,0.12
T= 1, output= 0.55,0.45
T= 10, output= 0.505,0.495
```

Ablation Study- Augmentation Strategy

- Replacing weak augmentation for pseudo labelling with no augmentation leads to overfitting, the model overfits to the pseudo labels because initially the model has few pseudo labels due to high confidence threshold.



Ablation Study- Augmentation Strategy

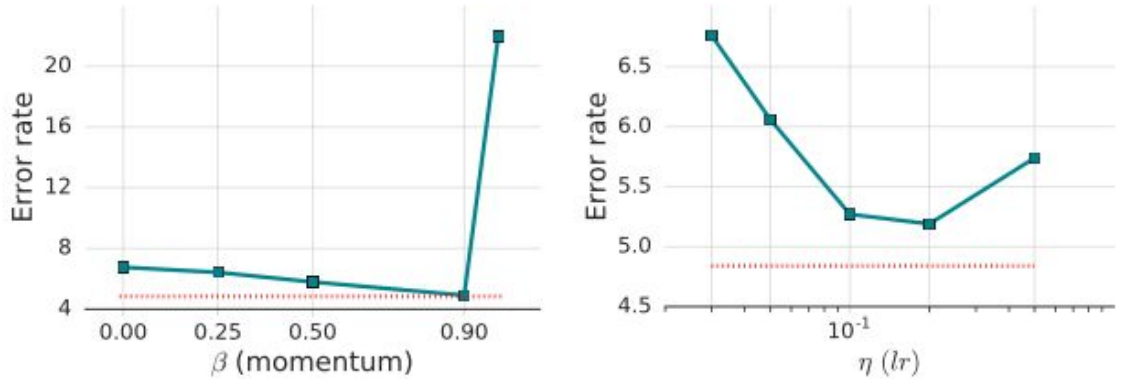
- Using strong augmentation for pseudo labelling performs badly because the model is not able to learn the basic features of the image classes well (no experiment results given in the paper).
- Replacing strong augmentation by weak augmentation leads to training collapse. The model reached 45% score initially but later collapsed to 12%. Strong augmentation helps to build robust features. Strong augmentation makes the prediction problem harder and improves generalization.

Abhinav Moudgil 5:15 PM

What do you think is the role of "strong" augmentation in this approach? Is it making the prediction problem harder (similar to hard negatives) which could improve performance or is it helping with correcting noise in pseudo labels?

Ablation Study- Optimizer

SGD	$\eta = 0.03$	$\beta = 0.90$		**4.86**
Adam	$\eta = 0.0003$	$\beta_1 = 0.9$	$\beta_2 = 0.999$	5.37



(a) (b)

Figure 4: Plots of ablation studies on optimizers. (a) Varying β. (b) Varying η with $\beta = 0$.

- Adam optimiser has poorer performance than SGD. Do you have a possible explanation?