FixMatch : Simplifying Semi-Supervised Learning with Consistency and Confidence

Abstract

- SSL提供了一种有效的方法去使用无标签数据, 近期这个领域发展的很快. 但代价是需要更加复杂的模型 (at the cost of requiring more complex methods)
- 本文提出一种简单但有效的方法, 首先对weak augmentation的数据输出pseudo-label, 并且只保留high-confidence label的数据. 然后使用同一张图片strong augmentation的数据对模型进行训练
- 虽然简单, 但是FixMatch在很多SSL benchmark中取得了SOTA, 包括在CIFAR-10上使用250个label的数据达到94.93%, 40个数据(每类4个) 达到了88.61% (2020NIPS)

Introduction

- 深度学习的成功主要归功于大量的数据集, 然而有标注的数据集需要大量的人力物力制作, 特别是给数据打标签需要专家知识
- 一个解决这个问题的方法是使用半监督学习Semi-supervised learning方法. SSL通过使用采集到的无标签数据来解决需要大量数据需求的问题
- 一个主流的方法是使用模型去对无标签的数据进行预测, 使用预测结果作为artificial label. 比如pseudo-labeling, consistency regularization
- 本文中作者不跟随近期结合越来越多复杂机制的趋势, 提出一种更加简单, 但也更加准确的方法FixMatch. FixMatch同时使用consistency regularization和pseudo labeling产生artificial label. 关键的是, 仅由weakly-augmented的数据产生artificial label, 然后将这些标记作为strongly-augmented version的target. 从UDA和ReMixMatch受到启发, 本文使用Cutout, CTAugment和RandAugment做strong augmentation. 然后仅保留具有high-confidence的label. 如图

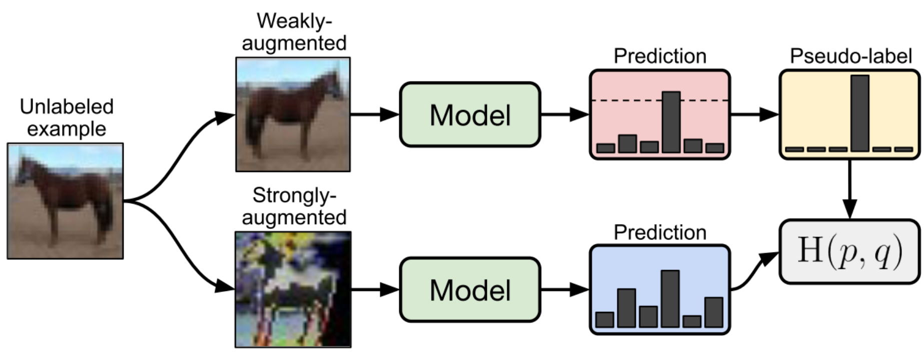

Figure 1: Diagram of FixMatch. A weakly-augmented image (top) is fed into the model to obtain predictions (red box). When the model assigns a probability to any class which is above a threshold (dotted line), the prediction is converted to a one-hot pseudo-label. Then, we compute the model's prediction for a strong augmentation of the same image (bottom). The model is trained to make its prediction on the strongly-augmented version match the pseudo-label via a cross-entropy loss.

- 尽管FixMatch比较简单, 但在大部分benchmark上获得了SOTA效果, 并且具有更少的超参数量

FixMatch

FixMatch是两个SSL方法的结合 :Consistency regularization和pseudo-labeling. 主要想法在于进行consistency regularization时使用了分开的weak augmentation和strong augmentation.

- Consistency regularization依赖的想法 : 模型对于相同但经过perturbed的图像应该输出相似的预测. 这个方法在[2]中首次提出, 在[24, 46]中推广, 并且模型对于有标签数据在标准supervised classification loss中训练, 无标签数据在如下loss函数中训练:

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

  其中α是随机的增强函数, 因此两个值不相同.

- pseudo-labeling在本文中方法所使用的公式为

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

  其中, 只有大于阈值的预测才会被计算cross-entropy loss

- FixMatch

  在有标签的数据集上使用标准的cross-entropy loss,特别的, 指在weakly augmented labeled examples上:

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

FixMatch首先计算每个unlabeledexample的artificial label, 然后在这上面使用标准的cross-entropy. 为了计算artificial label, 首先使用模型计算对wealy-augment version无标签数据的class distribution, 然后取最大值作为pseudo-label, 然后使用cross-entropy对strongly augment version

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

整体算法的loss即为两项加起来

$$\ell_s + \lambda_u \ell_u$$

- 整体流程如下

We present the complete algorithm for FixMatch in algorithm 1.

Algorithm 1 FixMatch algorithm.

Input: Labeled batch $\mathcal{X} = \{(x_b, p_b) : b \in (1, \dots, B)\}$, unlabeled batch $\mathcal{U} = \{u_b : b \in (1, \dots, \mu B)\}$, confidence threshold τ, unlabeled data ratio μ, unlabeled loss weight λ_u.
$\ell_s = \frac{1}{B} \sum_{b=1}^{B} \mathrm{H}(p_b, \alpha(x_b))$ {*Cross-entropy loss for labeled data*}
for $b = 1$ **to** μB **do**
  $q_b = p_{\mathrm{m}}(y \mid \alpha(u_b); \theta)$ {*Compute prediction after applying weak data augmentation of* u_b}
end for
$\ell_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}\{\max(q_b) > \tau\} \mathrm{H}(\arg\max(q_b), p_{\mathrm{m}}(y \mid \mathcal{A}(u_b))$ {*Cross-entropy loss with pseudo-label and confidence for unlabeled data*}
return $\ell_s + \lambda_u \ell_u$

- Augmentation in FixMatch

  FixMatch使用weak和strong做augmentation. 在本文所有实验中, weak augmentation采用标准的flip-and-shift augmentation strategy. 除了在SVHN上randomly translate image by up to 12.5% vertically and horizontally, 其他所有数据集都是randomly flip horizontally with probability of 50%

  strong augmentation使用CTAugment

- Additional important factors

  考察了其他影响结果的因素, 主要发现regularization非常重要, 因此使用了weight decay regularization. Adam optimizer会导致更差的结果, 因此使用SGD with momentum替代. 对于学习率, 使用cosine learning rate decay. 最后报告结果使用exponential moving average of model parameters

Related work

- 一些当前主要方法

Algorithm	Artificial label augmentation	Prediction augmentation	Artificial label post-processing	Notes
TS / Π-Model	Weak	Weak	None	
Temporal Ensembling	Weak	Weak	None	Uses model from earlier in training
Mean Teacher	Weak	Weak	None	Uses an EMA of parameters
Virtual Adversarial Training	None	Adversarial	None	
UDA	Weak	Strong	Sharpening	Ignores low-confidence artificial labels
MixMatch	Weak	Weak	Sharpening	Averages multiple artificial labels
ReMixMatch	Weak	Strong	Sharpening	Sums losses for multiple predictions
FixMatch	Weak	Strong	Pseudo-labeling	

Table 1: Comparison of SSL algorithms which include a form of consistency regularization and which (optionally) apply some form of post-processing to the artificial labels. We only mention those components of the SSL algorithm relevant to producing the artificial labels (for example, Virtual Adversarial Training additionally uses entropy minimization [17], MixMatch and ReMixMatch also use MixUp [59], UDA includes additional techniques like training signal annealing, etc.).

Experiments

- 按照标准的SSL benchmark评估. 主要使用CIFAR-10 CIFAR-100 SVHN STL-10 and ImageNet, 并且额外测试了在extremely label-scarce setting的性能. 并且使用同样的超参数设置

  对于CIFAR-10, 模型使用Wide ResNet-28-2, WRN-28-8 for CIFAR-100, WRN-37-2 for STL-10. 使用π-Model, Mean Teacher, Pseudo-Label, Mix Match, UDA, ReMixMatch作为baseline. 另外先前工作没有考虑过比25个有标签数据更小的情况, 因此我们进行了每类仅有4个标签图像的情况

- 在实验中, FixMatch超过了大多数方法, 但在CIFAR-100上ReMixMatch比FixMatch好, 通过分析得出结果是因为distribution alignment.

	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±0.97	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

Table 2: Error rates for CIFAR-10, CIFAR-100, SVHN and STL-10 on 5 different folds. FixMatch (RA) uses RandAugment [11] and FixMatch (CTA) uses CTAugment [3] for strong-augmentation. All baseline models (Π-Model [43], Pseudo-Labeling [25], Mean Teacher [51], MixMatch [4], UDA [54], and ReMixMatch [3]) are tested using the same codebase.

个人总结

本文提出一种更加简单的方法, 即仅使用consistency regularization加上pseudo label, 注意都是在weak augmentation上计算的, 然后对于strong augmentation也使用相同的pseudo label. 目前大多SSL方法属实很复杂, 结合了很多方法一起. 这个确实实验做的很详细, 比较全面 应该是投NIPS2021