MixMatch

Abstract

- 半监督学习被证明是一个有效的方法去使用无标签的数据去解决大量labled dataset需求. 本文unify了semi-supervised现有的dominant方法, 提出了MixMatch
- MixMatch首先guessing low-entropy labels, 然后对unlabeled数据进行data augmentation, 最后再使用MixUp混合labeled和unlabeled数据集
- 实验表明, MixMatch取得了SOTA水平(2019)

Introduction

- 许多深度学习的成功来自于大型的, 有标记的数据集. 然而, 对于许多任务来说, 收集有标签的数据集代价非常高昂, 是time and resource consuming的. 更进一步, 数据可能包含有private sensitive information. 与之相对, 在很多任务中, 获取无标签的数据是非常容易的
- semi-supervised learning (SSL) 期望允许模型使用unlabeled data来极大的缓解对labeled data的需求. 很多近期SSL的工作添加了loss term在计算unlabeled data上, 以此鼓励模型能够更好generalize to unseen data. 这些loss term可以分为三类: 1) entropy minimization. 鼓励模型对unlabeled data进行confident prediction. 2) consistency regularization. 鼓励模型在输入有被扰动时 (perturbed) 输出相同的分布. 3) generic regularization. 鼓励模型有更好的泛华, 并且避免overfitting
- 本文提出MixMatch, 结合了当下dominant的SSL方法. 不同于先前的方法, MixMatch期望对于所有的properties一次性解决, 并且有如下的一些优点: 1) 实验表明MixMatch在所有standard image benchmark中取得了SOTA效果. 2) ablation study表明了MixMatch比组成部分的总和更好. 3) MixMatch对于differentially private learning有用. 是的PATE框架的students 取得了新的SOTA, 同时强化了隐私保障

Related Work

- Consistency Regularization

  在监督学习中, 一个通常的regularization technique是data augmentation, 对输入数据做变换, 并且不影响类别语义信息. 在SSL中, consistency regularization基于分类器对于无标签的样本, 即使在进行增强后也应该输出相同类别分布的想法, 将data augmentation应用到SSL. 设Augment( ) 是随机数据增强函数, 则有:

  $$\|\mathrm{p}_{\mathrm{model}}(y \mid \mathrm{Augment}(x); \theta) - \mathrm{p}_{\mathrm{model}}(y \mid \mathrm{Augment}(x); \theta)\|_2^2$$

  由于Augment( )是随机增强函数, 因此上式中两个Augment( )不同. 这个方法被用于很多图像分类的benchmark任务中, 比如旋转, 裁剪, 添加高斯噪声等. **Mean Teacher**方法使用exponential moving average of model parameter values与模型的输出一起改进了上式, 能够提供更加稳定的目标, 并且能够significantly improve result, 但是有drawback是, 这些方法使用domain-specifc的数据增强方法. **Virtual Adversarial Training** 解决这个问题的方法是计算能最大改变输出类别的分布的额外perturbation应用到输入上. MixMatch使用标准的数据增强consistency regularization形式

- Entropy Minimization

分类器的决策边界不应该穿过high-density regions of marginal data distribution这个假设在SSL中常见. 使得模型对这个假设成立的一个方法是强迫模型对无标签的数据输出low-entropy prediction. 这个在[17]中通过加上一个最小化无标签数据的预测的entropy的loss项显式的达成. 这个方法结合VAT获得了更好的效果. Pseudo-label通过从无标签数据high-confidence prediction中构造hard label, 并使用标准的cross-entropy loss来达成entropy minimization. MixMatch通过在target distribution使用sharpening function (temperature scaling?) 实现entropy minimization

- Traditional Regularization

  Regularization指通常对模型加上约束, 使得其更难以对训练数据进行记忆, 从而希望获得对unseen data更好的泛化性. 一种regularization的方法就是对模型的参数加上L2范数的loss项, 即类似与weight decay. 在本文中, 我们使用Adam作为我们的optimizer, 因此我们直接使用weight decay取代L2 loss项. 最近的MixUp regularizer被提出, 该方法在输入和标签的convex combination上训练模型. 通过要求模型对于individual input, 两个input的convex combination要相近与输出的convex combination, MixUp可以鼓励模型在样本间有strictly linear behavior. 本文使用MixUp同时作为regularizer和SSL方法.

MixMatch

- MixMatch是一个holistic (全面的) 方法, 集成了之前方法的想法和组成部分. 给定一个batch有标签的数据集和相应的one-hot目标, 以及equally-sized batch的无标签样本, MixMatch生成处理过的增强后的有标签样本和无标签样本以及对应的guessed label. 然后使用这两份数据分别计算有标签的和无标签对应的loss项, 形式化表述如下

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

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

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

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

  H是cross-entropy, T K α是超参数

  整体算法如下

Algorithm 1 MixMatch ingests a batch of labeled data $\mathcal{X}$ and a batch of unlabeled data $\mathcal{U}$ and produces a collection $\mathcal{X}'$ of processed labeled examples and a collection $\mathcal{U}'$ of processed unlabeled examples 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**
  $\hat{x}_b = \mathrm{Augment}(x_b)$ *// Apply data augmentation to* x_b
  for $k = 1$ **to** K **do**
    $\hat{u}_{b,k} = \mathrm{Augment}(u_b)$ *// Apply* k^{th} *round of data augmentation to* u_b
  end for
  $\bar{q}_b = \frac{1}{K} \sum_k \mathrm{p}_{\mathrm{model}}(y \mid \hat{u}_{b,k}; \theta)$ *// Compute average predictions across all augmentations of* u_b
  $q_b = \mathrm{Sharpen}(\bar{q}_b, T)$ *// 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*
$\mathcal{W} = \mathrm{Shuffle}\big(\mathrm{Concat}(\hat{\mathcal{X}}, \hat{\mathcal{U}})\big)$ *// Combine and shuffle labeled and unlabeled data*
$\mathcal{X}' = \big(\mathrm{MixUp}(\hat{\mathcal{X}}_i, \mathcal{W}_i); i \in (1, \dots, |\hat{\mathcal{X}}|)\big)$ *// Apply* MixUp *to labeled data and entries from* $\mathcal{W}$
$\mathcal{U}' = \big(\mathrm{MixUp}(\hat{\mathcal{U}}_i, \mathcal{W}_{i+|\hat{\mathcal{X}}|}); i \in (1, \dots, |\hat{\mathcal{U}}|)\big)$ *// Apply* MixUp *to unlabeled data and the rest of* $\mathcal{W}$
return $\mathcal{X}', \mathcal{U}'$

- Data Augmentation

  一个通常解决缺乏有标签数据的方法是使用数据增强, 数据增强引入一个产生随机变换的augment(x), 在变换数据的同时, 保持标签不变. 在SSL中, 通常对有标签和无标签的数据同时做数据增强, 这些每一个无标签数据增强后的K个数据也被用于生成guessed label

- Label Guessing

  对于每个无标签的数据, MixMatch使用模型的预测产生一个guess label. 这个guessed label之后用于计算unsupervised loss term. 使用K个增强数据集的模型预测均值作为这一个 (包括K个增强) 样本的标签 (每一类概率的均值)

- Sharpening

  从近期SSL中entropy minimization的成功受到启发, 在MixMatch中加入了sharpening function减少label distribution的entropy. 我们使用adjusting temperature的方法来sharpening, 公式如下

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

  上式中, P是类别分布, T为超参数. 使用sharpening可以使得模型输出lower-entropy prediction

- MixUp

  使用MixUp同时在有标签数据和无标签数据+guessed label. 不同于之前MixUp的工作, 本文混合了有标签的样本和无标签的样本一起, 发现效果更好. 在我们设计的loss function中, 分别使用不同的loss项计算有标签和无标签数据. 使用MixUp分别计算有标签的batch和无标签batch

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

- Loss Function

  使用标准的SSL loss function作为目标

Experiment

采用标准的SSL benchmark进行效果对比, ablation study作为测试每一个MixMatch部分的贡献, 另外还考虑了privacy-preserving learning

- Implementation Details

  除非特别说明, 所有实验都使用 Wide ResNet-28 模型(from [33]). 训练过程也和[33]一样, 仅做了一些微小修改 : 1) 使用exponential moving average with decay rate of 0.999, 2) 使用weight decay 作为regularization decaying weights by 0.02 at each update, 3) 每隔一段时间保存一次checkpoint, 取最后20个checkpoint中位数的错误率作为结果

- Semi-Supervised Learning

  在CIFAR-10, CIFAR-100, SVHN, STL-10数据集上进行实验, 这也是4个standard benchmark datasets. 前三个是通常用于supervised learning, 对于SSL, 使用大部分的数据当做无标签, 小部分视为有标签数据. STL-10专门设计为SSL, 有5000个标签数据和10000个无标签数据, 无标签数据来自略微不同与有标签数据的分布

- Baseline Method

  使用Π-Model, Mean Teacher, Virtual Adversarial Training, Pseudo Label以及MixUp本身作为baseline方法. 使用MixUp需要对于SSL做一些小改动. 在同样的codebase和同样的模型下进行实验. 对每个baseline模型都微调了一下超参数, 以取得更好效果

- Ablation Study

  因为MixMatch结合了很多SSL方法, 因此对每一个方法进行有效性评估是有意义的

- 实验结果

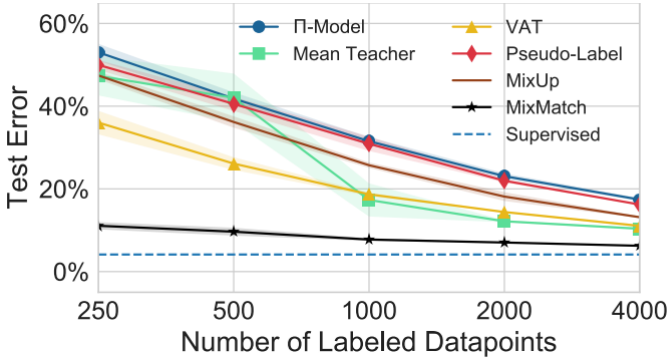

Figure 2: Error rate comparison of MixMatch to baseline methods on CIFAR-10 for a varying number of labels. Exact numbers are provided in table 5 (appendix). "Supervised" refers to training with all 50000 training examples and no unlabeled data. With 250 labels MixMatch reaches an error rate comparable to next-best method's performance with 4000 labels.

Method	CIFAR-10	CIFAR-100
Mean Teacher [42]	6.28	-
SWA [2]	5.00	28.80
MixMatch	4.95 ± 0.08	25.88 ± 0.30

Table 1: CIFAR-10 and CIFAR-100 error rate comparison with larger (26 million parameter) models.

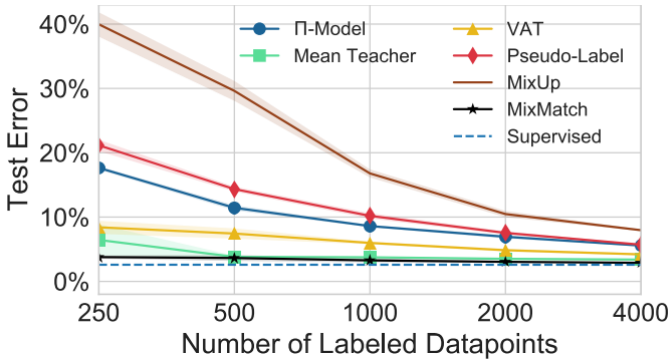

Figure 3: Error rate comparison of MixMatch to baseline methods on SVHN for a varying number of labels. Exact numbers are provided in table 6 (appendix). "Supervised" refers to training with all 73257 training examples and no unlabeled data. With 250 examples MixMatch nearly reaches the accuracy of supervised training for this model.

Method	1000 labels	5000 labels
CutOut [12]	-	12.74
IIC [20]	-	11.20
SWWAE [47]	25.70	-
CC-GAN^2 [11]	22.20	-
MixMatch	10.18 ± 1.46	5.59

Table 2: STL-10 error rate using 1000-label splits or the entire 5000-label training set.

Labels	250	500	1000	2000	4000	All
SVHN	3.78 ± 0.26	3.64 ± 0.46	3.27 ± 0.31	3.04 ± 0.13	2.89 ± 0.06	2.59
SVHN+Extra	2.22 ± 0.08	2.17 ± 0.07	2.18 ± 0.06	2.12 ± 0.03	2.07 ± 0.05	1.71

Table 3: Comparison of error rates for SVHN and SVHN+Extra for MixMatch. The last column ("All") contains the fully-supervised performance with all labels in the corresponding training set.