

半监督图像分类

实验目的

关于半监督学习

神经网络模型通常需要大量标记好的训练数据来训练模型。然而，在许多情况下，获取大量标记好的数据可能是困难、耗时或昂贵的。这就是半监督学习的应用场景。半监督学习的核心思想是利用无标记数据的信息来改进模型的学习效果。在半监督学习中，我们使用少量标记数据进行有监督学习，同时利用大量无标记数据的信息。通过充分利用无标记数据的潜在结构和分布特征，半监督学习可以帮助模型更好地泛化和适应未标记数据。关于深度学习中半监督学习更全面的总结，可以参考深度学习半监督学习综述[1]

半监督图像分类

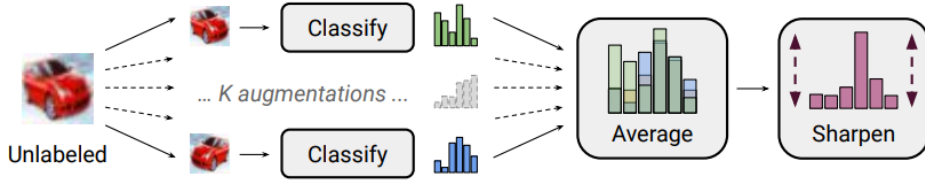
半监督学习在图像分类取得了非常大的进步，涌现了许多经典的半监督图像分类算法，如： π Model[2]，Mean Teacher[3]，MixMatch[4]，FixMatch[5]等。这些算法都取得了非常好的结果，能够在仅使用少量标注数据的情况下，实现高精度的图像分类，在 ImageNet，CIFAR-10，CIFAR-100 等数据集上都有非常不错的效果。TorchSSL[6]是微软发布的一个用于半监督深度学习的库，其中提供了许多半监督学习算法的实现，如下图所示：

CIFAR-10 and CIFAR-100

		CIFAR-10				CIFAR100	
	40	250	4000		400	2500	10000
FullySupervised	95.38±0.05	95.39±0.04	95.38±0.05		80.7±0.09	80.7±0.09	80.73±0.05
PiModel [1]	25.66±1.76	53.76±1.29	86.87±0.59		13.04±0.8	41.2±0.66	63.35±0.0
PseudoLabel [3]	25.39±0.26	53.51±2.2	84.92±0.19		12.55±0.85	42.26±0.28	63.45±0.24
PseudoLabel_Flex [9]	26.26±1.96	53.86±1.81	85.25±0.19		14.28±0.46	43.88±0.51	64.4±0.15
MeanTeacher [2]	29.91±1.6	62.54±3.3	91.9±0.21		18.89±1.44	54.83±1.06	68.25±0.23
VAT [4]	25.34±2.12	58.97±1.79	89.49±0.12		14.8±1.4	53.16±0.79	67.86±0.19
MixMatch [5]	63.81±6.48	86.37±0.59	93.34±0.26		32.41±0.66	60.24±0.48	72.22±0.29
ReMixMatch [7]	90.12±1.03	93.7±0.05	95.16±0.01		57.25±1.05	73.97±0.35	79.98±0.27
UDA [6]	89.38±3.75	94.84±0.06	95.71±0.07		53.61±1.59	72.27±0.21	77.51±0.23
UDA_Flex [9]	94.56±0.52	94.98±0.07	95.76±0.06		54.83±1.88	72.92±0.15	78.09±0.1
FixMatch [8]	92.53±0.28	95.14±0.05	95.79±0.08		53.58±0.82	71.97±0.16	77.8±0.12
FlexMatch [9]	95.03±0.06	95.02±0.09	95.81±0.01		60.06±1.62	73.51±0.2	78.1±0.15

实验内容

1. 基于 MixMatch 的 CIFAR-10 数据集半监督图像分类

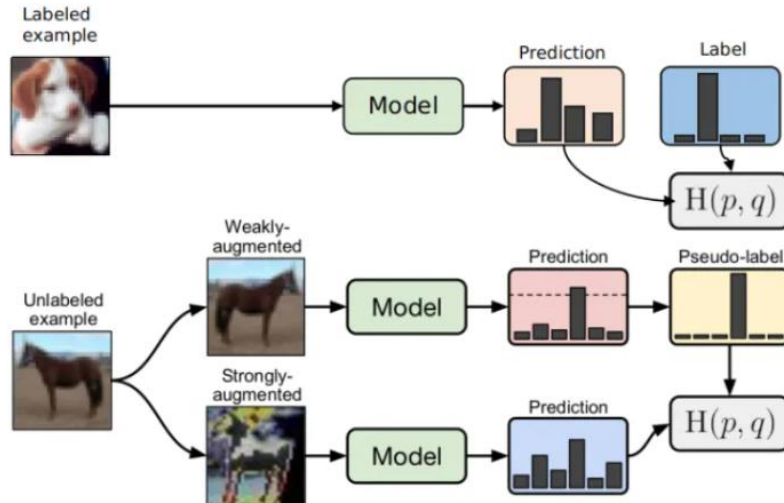


MixMatch 采用两种方式利用无标注数据，即熵最小化和一致性正则化。具体来说，MixMatch 首先对无标注数据进行 K 次增广，并获得对应的 K 个输出，并计算模型平均的预测分布，之后，MixMatch 使用一个"sharpening"函数来对无标注数据的平均预测进行锐化，并把锐化后的预测分布作为该无标注数据的标签。获得对无标注数据的标签后，将标注数据和无标注数据结合起来，通过 Mixup 的方式完成新数据集的构建，最后使用新的数据集进行训练。

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).

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

2. 基于 FixMatch 的 CIFAR-10 数据集半监督图像分类



FixMatch 结合了伪标签和一致性正则化来实现对无标注数据的高效利用，训练过程包括两个部分，有监督训练和无监督训练。有标注的数据，执行有监督训练，和普通分类任务训练没有区别。没有标注的数据，首先将其经过弱增强之后送到模型中推理获取伪标签，然后再将其经过强增强送到模型中进行预测，并利用生成的伪标签监督模型对强增强数据的预测。模型对弱增强数据的预测，只有大于一定阈值条件时才执行伪标签的生成，并使用该伪标签来进行无标注图像的训练。

Algorithm 1 FixMatch algorithm.

- 1: **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 .
 - 2: $\ell_s = \frac{1}{B} \sum_{b=1}^B H(p_b, \alpha(x_b))$ {Cross-entropy loss for labeled data}
 - 3: **for** $b = 1$ **to** μB **do**
 - 4: $q_b = p_m(y \mid \alpha(u_b); \theta)$ {Compute prediction after applying weak data augmentation of u_b }
 - 5: **end for**
 - 6: $\ell_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}\{\max(q_b) > \tau\} H(\arg \max(q_b), p_m(y \mid \mathcal{A}(u_b)))$ {Cross-entropy loss with pseudo-label and confidence for unlabeled data}
 - 7: **return** $\ell_s + \lambda_u \ell_u$
-

实验要求

1. 阅读原始论文和相关参考资料，基于 Pytorch 分别实现 MixMatch 和 FixMatch 半监督图像分类算法，按照原始论文的设置，MixMatch 和 FixMatch 均使用 WideResNet-28-2 作为 Backbone 网络，即深度为 28，扩展因子为 2，在 CIFAR-10 数据集上进行半监督图像分类实验，**报告算法在分别使用 40, 250, 4000 张标注数据的情况下的图像分类效果（标注数据随机选取指定数量）**
2. 使用 TorchSSL[6]中提供的 MixMatch 和 FixMatch 的实现进行半监督训练和测试，**对比自己实现的算法和 TorchSSL 中的实现的效果**
3. 提交源代码，不需要包含数据集，并提交实验报告，实验报告中应该包含代码的使用方法，对**数据集数据的处理步骤以及算法的主要实现步骤**，并**分析对比 MixMatch 和 FixMatch 的相同点和不同点**。

- [1]. A Survey on Deep Semi-supervised Learning 2021
- [2]. Temporal Ensembling for Semi-Supervised Learning
- [3]. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results
- [4]. Mixmatch: A holistic approach to semi-supervised learning
- [5]. Fixmatch: Simplifying semi-supervised learning with consistency and confidence
- [6]. <https://github.com/StephenStorm/TorchSSL>

作业提交时间：2024 年 6 月 24 日 24:00 前, 推迟一天成绩减 5 分
(不足一天按一天计算), 以有效作业提交的邮件时间戳为准。

提交方式：作业命名 “学号_姓名_模式识别_homework3.zip” , 交到课程邮箱 (邮箱: andypr2024@163.com)

注意：不得抄袭, 包括源代码!!!