

# 论文阅读报告

撰写人：19212010023 杜清华

## 一、 论文题目及来源

题目：Understanding and Utilizing Deep Neural Networks Trained with Noisy Labels

作者：Pengfei Chen Benben Liao Guangyong Chen Shengyu Zhang

来源：International Conference on Machine Learning 2019

## 二、 论文概述

噪声标签在现实世界的数据集中是无处不在的，但是深度神经网络（DNN）通常具有很高的能力来记住噪声标签，因此在论文中提出了在噪声标签的干扰下如何强健地训练 DNN。在论文中，作者发现可以根据数据集中的噪声比来对测试精度进行定量表征，特别是在对称噪声的情况下，测试精度是噪声比的二次函数。根据文中作者的分析，作者将交叉验证应用于随机分裂的嘈杂数据集，该数据集可识别大多数具有正确标签的样本。然后采用 Co-teaching 策略，该策略可以充分利用已识别的样本来针对噪声标签对 DNN 进行鲁棒性训练。与当前的一些最新方法相比，作者的策略在一定程度上提高了 DNN 在合成和真实训练噪声下的泛化性能。

## 三、 研究的问题及目标

### 3.1 问题研究背景

DNN 在监督学习任务上结果的准确性在很大程度上需要大量带有准确标签的训练样本，然而正确标记大量数据的成本太高，因此不可避免的会使用含有一定噪声标签的数据。带有太多噪声标签的训练会降低 DNN 的泛化性能，因为网络很容易在损坏的标签上过度拟合。

引出问题 1：如何利用大量的噪声数据了解噪声标签如何影响 DNN 的训练和泛化。

针对噪声标签的处理，目前已经提出几种方法，一种是专注于估计噪声转换矩阵并相应的校正目标函数，例如 Patrini 等人在 2017 年提出的向前或向后校正方法、Goldberger 和 Ben-Reuven 提出的 S-模型等，但是估计噪声矩阵相对困难不易实现。另一种方法是对选定或加权的样本进行训练，例如 Malach 和 Shalev-Shwartz 在 2017 年提出的解耦方法，但是问题是需要设计一个可靠且令人信服的选定或加权样本的标准。还有一种方法是使用 DNN 的预测来校正标签，例如 Tanaka 等人在 2018 提

出的 Joint Optimization, Ma 等人在 2018 年提出的 D2L, 但是这些方法容易受过拟合的影响。

引出问题 2: 如何处理数据的噪声标签或者说如何在有噪声标签的情况下使训练的 DNN 具有一定的鲁棒性。

Zhang 等人在 2017 的实验中证明了 DNN 具有很高的适应随机标签的能力, 但是泛化误差较大, 同时也发现了泛化误差与噪声比之间存在正相关, 这意味着 DNN 确实从噪声数据中得到了一些信息, 然而仍无法定量的表明其中关系。

引出问题 3: 如何定量的描述泛化误差与标签噪声的关系, 根据发现的定量关系如何去训练具有一定抗噪声性的 DNN。

### 3.2 问题研究目标

针对上面所引出的问题, 主要需要解决的问题包含如下:

1. 在理论上将 DNN 的泛化性能与噪声相关联
2. 定量的阐明使用噪声标签训练的 DNN 的泛化性能;
3. 开发一种针对噪声标签训练 DNN 的特定实用算法。

## 四、 研究的方法

### 4.1 相关表示

①c 分类数据集  $D = \{x_t, y_t\}_{t=1}^n$ , 其中  $x_t$  是第  $t$  个样本, 其观察到的标签为  $y_t \in [c] := \{1, \dots, c\}$ ,  $y$  表示真实标签。

② 噪声转换矩阵  $T \in \mathbb{R}^{c \times c}$  来描述集合  $D$  的破坏过程, 其中  $T_{ij} = P(y = j | \hat{y} = i)$  表示将第  $i$  类样本被标记为  $j$  的概率。

③  $f(x; \omega)$  表示由  $\omega$  参数化的神经网络, 而  $y^f \in [c]$  表示由网络  $f(x; \omega)$  给出的  $x$  的预测标签。

### 4.2 命题声明

Zhang 等人在 2017 发现在处理带有噪声标签的 DNN 时,  $y^f = y$  可能不会在每个测试样例  $x$  上评估时保持不变, 也就是说会有较大的泛化误差。然而作者发现泛化在某种分布仍然有意义上, 即分布的泛化, 该断言如下所示:

**断言1:**

网络  $f(x; \omega)$  在  $D_1$  上训练并在  $D_2$  上测试, 如果我们假设

- (i) 观察到的输入样例  $x$  在集合  $D$  中独立同分布
- (ii)  $f$  具有足够高的容量

则在  $D_2$  上, 预测第  $i$  类测试样本为  $j$  的概率为

$$P(y^f = j | \hat{y} = i) = T_{ij} \quad (1)$$

其中 $T_{ij} := P(y^f = j | \hat{y} = i)$  表示 $D_1$ 和 $D_2$ 共享的噪声转换矩阵。

该断言表明预测 $y^f$ 和真实的测试标签  $y$  具有相同的分布。在论文 5.1 节中的实验证明了断言的合理性。

### 命题 1:

令 $D_1$ 和 $D_2$ 为两个具有相同噪声过渡矩阵  $T$  的数据集,  $f(x; \omega)$ 为在 $D_1$ 上训练并在 $D_2$ 上进行测试的网络。根据断言中的假设, 对于任意类别 $i \in [c]$ 的测试精度为

$$P(y^f = y | \hat{y} = i) = \sum_{j=1}^c T_{ij}^2 \quad (2)$$

证明: 该命题是基于上述断言提出的, 因为 $y^f$ 和  $y$  具有以  $T$  为特征的同分布。假设标签损坏过程是独立的, 那么在测试集上有

$$P(y^f = j, y = k | \hat{y} = i) = P(y^f = j | \hat{y} = i)P(y = k | \hat{y} = i) = T_{ij}T_{ik} \quad (3)$$

因此公式(2)遵从于 $P(y^f = y | \hat{y} = i) = \sum_{j=1}^c P(y^f = j, y = j | \hat{y} = i)$

因为论文中涉及到有关对称噪声与非对称噪声这两种噪声类型, 所以作者对其进行了有关的定义:

### 定义 1:

对于比率为 $\varepsilon$ ,  $\forall i \in [c]$ 的对称噪声的情况下, 定义 $T_{ii} = 1 - \varepsilon$ , 并且 $\forall j \neq i T_{ij} = \varepsilon/(c - 1)$ 。

对于比率为 $\varepsilon$ ,  $\forall i \in [c]$ 的非对称噪声的情况下, 定义 $T_{ii} = 1 - \varepsilon$ , 对于某些  $j \neq i$  定义 $T_{ij} = \varepsilon$ , 否则定义 $T_{ij} = 0$ 。

作者在文章中提出, 在对称和非对称噪声的情况下可以使用噪声比 $\varepsilon$  量化 DNN 的测试精度, 这些 DNN 分别在前面提到的噪声数据集 $D_1$ 和 $D_2$ 上进行训练和测试。结合命题 1 到以下推论:

### 推论 1.1:

对于比率为 $\varepsilon$ 的对称噪声, 测试精度为

$$P(y^f = y) = (1 - \varepsilon)^2 + \frac{\varepsilon^2}{c-1} \quad (4)$$

对于比率为 $\varepsilon$ 的非对称噪声, 测试精度为

$$P(y^f = y) = (1 - \varepsilon)^2 + \varepsilon^2 \quad (5)$$

证明: 根据命题 1 有

$$P(y^f = y) = \sum_{i=1}^c P(\hat{y} = i)P(y^f = y | \hat{y} = i) = \sum_{i=1}^c P(\hat{y} = i) \sum_{j=1}^c T_{ij}^2$$

对于对称和非对称噪声,  $\forall i \in [c]$ ,  $\sum_{j=1}^c T_{ij}^2$ 是由 $\varepsilon$  给出的常数。因此, 通过将 $\varepsilon$  代入等式, 即可得到所需的结果。

需要注意的是，利用上述的等式 (4) 能够根据实验测试的准确性估算数据集的噪声比。

### 4.3 方法提出及算法设计

在论文中所开发的方法是在 Co-teaching 的基础上开发的，在之前的实验中发现，当训练集的噪声比增加时，Co-teaching 的性能会严重下降并变得不稳定，为了解决这个问题，论文中选择一个样本子集，该子集的噪声比将会比原始数据集小得多。

如果样本  $(x, y)$  的观察标记  $y$  等于  $f(x; \omega)$  给出的预测标记  $y^f$ ，那么可以将其识别为干净的。为了获取一个干净的样本子集作为训练数据集，因此提出噪声交叉验证 (NCV) 方法。

---

#### 算法 1：嘈杂的交叉验证 (NCV)：从嘈杂的样本中选择干净的样本

---

**输入：** 噪声数据集  $D$ ，训练周期  $E$

- 1:  $S = \emptyset$ ，初始化网络  $f(x; \omega)$
- 2: 将  $D$  随机分为两等分  $D_1$  和  $D_2$
- 3: 在数据集  $D_1$  上训练  $f(x; \omega)$ ，周期为  $E$
- 4: 选择样本  $S_1$ ，满足  $S_1 = \{(x, y) \in D_2: y^f = y\}$
- 5: 重新初始化网络  $f(x; \omega)$
- 6: 在数据集  $D_2$  上训练  $f(x; \omega)$ ，迭代周期为  $E$
- 7: 选择样本  $S_2$ ，满足  $S_2 = \{(x, y) \in D_1: y^f = y\}$
- 8:  $S = S_1 \cup S_2$

**输出：** 选定的集合  $S$

---

论文中使用 Label Precision (LP) 和 Label Recall (LR) 证明算法的有效性。LP 代表  $S$  中纯净样品的比例，而 LR 代表  $S$  的纯净样品的占  $D$  中所有纯净样品的比例。因此可以表示为

$$LP := \frac{|\{(x, y) \in S: y = \hat{y}\}|}{|S|} \quad LR := \frac{|\{(x, y) \in S: y = \hat{y}\}|}{|\{(x, y) \in D: y = \hat{y}\}|} \quad (6)$$

#### 命题 2:

使用算法 1 去选择干净的样本，那么  $\forall i \in [c]$ ，有

$$LP_i = \frac{T_{ii}^2}{\sum_{j=1}^c T_{ij}^2}, \quad LR_i = T_{ii} \quad (8)$$

由于  $\forall i, \sum_{j=1}^c T_{ij} = 1$ ，公式 (8) 又可以推出

$$\frac{T_{ii}^2}{T_{ii}^2 + (1 - T_{ii})^2} \leq LP_i \leq \frac{T_{ii}^2}{T_{ii}^2 + \frac{(1 - T_{ii})^2}{c-1}} \quad (9)$$

上面的公式 (9)，对于对称噪声得到了不等式 (9) 的上界，对于非对称噪声不等式 (9) 的下界。同时在对称和非对称噪声的情况下，还有  $LP = LP_1 = \dots = LP_c$ ， $LR =$

$LR_1 = \dots = LR_c$ 。因此可以重新构造 LP 和 LR,

对于对称噪声比 $\varepsilon$ , 有

$$LP = \frac{(1-\varepsilon)^2}{(1-\varepsilon)^2 + \varepsilon^2 / (c-1)}, \quad LR = 1 - \varepsilon \quad (10)$$

对于非对称噪声比 $\varepsilon$ , 有

$$LP = \frac{(1-\varepsilon)^2}{(1-\varepsilon)^2 + \varepsilon^2}, \quad LR = 1 - \varepsilon \quad (11)$$

根据 LP 的定义, 所选子集 S 的噪声比为 $\varepsilon_S = 1 - LP$ 。在实际情况下 ( $\forall i, T_{ii}$  在  $T_{ij}$  中最大,  $j \in [c]$ ), 算法 1 始终会产生一个噪声比 $\varepsilon_S < \varepsilon$  较小的子集。

#### 4.4 算法改进

虽然由算法 1 选择的子集通常具有比原始集合小的噪声比, 但是对于 DNN 来说, 其鲁棒训练可能需要更多的训练样本。为了解决此问题, 论文改进算法 1 并提出了算法 2——迭代噪声交叉验证 (INCV) 方法, 该方法可以通过应用算法 1 迭代增加选择的样本数。除了选择干净的样本外, INCC 还会删除每次迭代中分类交叉熵损失较大的样本, 去除率  $r$  决定将去除多少样品。

---

#### 算法 2: 迭代噪声交叉验证 (INCV): 从噪声样本中选择干净的样本

---

**输入:** 噪声数据集 D, 迭代次数 N, 训练周期 E, 去除率  $r$

- 1:  $S = \emptyset$ , 设置候选集  $C = D$
- 2: for  $i = 1, \dots, N$  do
- 3:   初始化网络  $f(x; \omega)$
- 4:   将 C 随机分为两等分  $C_1$  和  $C_2$
- 5:   在数据集  $S \cup C_1$  上训练  $f(x; \omega)$ , 周期为 E
- 6:   选择样本  $S_1$ , 满足  $S_1 = \{(x, y) \in C_2: y^f = y\}$
- 7:   确定  $n = r|S_1|$  将被移除的样本:  
 $R_1 = \{\#n \arg \max_{C_2} \mathcal{L}(y, f(x; \omega))\}$
- 8:   如果  $i=1$ , 使用公式 (4) 估计噪声比  $\varepsilon$
- 9:   重新初始化网络  $f(x; \omega)$
- 10:   在数据集  $S \cup C_2$  上训练  $f(x; \omega)$ , 周期为 E
- 11:   选择样本  $S_2$ , 满足  $S_2 = \{(x, y) \in C_1: y^f = y\}$
- 12:   确定  $n = r|S_2|$  将被移除的样本:  
 $R_2 = \{\#n \arg \max_{C_1} \mathcal{L}(y, f(x; \omega))\}$
- 13:    $S = S \cup S_1 \cup S_2$ ,  $C = C - S_1 \cup S_2 \cup R_1 \cup R_2$
- 14: end for

**输出:** 选定的集合 S, 剩余候选集 C, 估计噪声比  $\varepsilon$

---

在使用算法 2 详细剖析了嘈杂的数据集 D 后, 可以进一步改善 Co-teaching, 以充分利用所选集 S 和候选集 C 的优势。具体来说, 让这两个网络在第一个  $E_0$  周期集中于所选集合 S, 然后合并候选集合 C。因此训练稳定性和测试准确性均得到

改善。针对噪声标签训练 DNN 的鲁棒性算法如下所示。

---

**算法 3: 针对噪声标签训练 DNN 的鲁棒性**

---

**输入:** 选择数据集  $S$ , 候选数据集  $C$  和来自算法 2 的估计噪声比  $\epsilon$ 。预热周期  $E_0$ ,

总周期  $E_{max}$

```

1: 初始化两个网络  $f_1(x; \omega_1)$  和  $f_2(x; \omega_2)$ 
2: for  $e=1, \dots, E_{max}$  do
3:   for batches  $(B_S, B_C)$  in  $(S, C)$  do
4:     if  $t > E_0$  then  $B = B_S \cup B_C$ , else  $B = B_S$ 
5:      $B_1 = \{\#n(e) \arg \min_B \mathcal{L}(y, f_1(x; \omega_1))\}$ 
6:      $B_2 = \{\#n(e) \arg \min_B \mathcal{L}(y, f_2(x; \omega_2))\}$ 
7:     使用  $B_2$  更新  $f_1$ 
8:     使用  $B_1$  更新  $f_2$ 
9:   end for
10: end for

```

**输出:**  $f_1(x; \omega_1)$  和  $f_2(x; \omega_2)$

---

## 五、研究的实验及结果

实验过程主要包括三个部分。首先通过实验验证了前面的断言及相关命题。然后, 演示算法2中的INCV方法, 证明该方法可以识别更多具有正确标签的样品。最后, 证明算法3的可靠性, 该方法可以针对噪声标签训练DNN的鲁棒性, 并且优于目前最新方法。

### 5.1 实验准备

1. 数据集: (1) CIFAR-10 数据集 (2) WebVision 数据集

2. 数据处理: 对 CIFAR-10 中的原始标签随机破坏产生合成噪声标签, 其中主要包含两种代表性的噪声类型: 对称噪声和非对称噪声。实验使用 WebVision 数据集验证在真实噪声标签上的方法, 该数据集包含从网站抓取的 240 万幅图像。WebVision 的训练集包含许多没有人为注释的真实噪声标签。

### 5.2 实验过程

#### 实验一. 使用噪声标签训练DNN

在理论上通过以下指标来表征使用噪声标签训练的 DNN: ①等式(4) (5)给出的测试准确性。②等式(10) (11)给出的 LP。③等式(10) (11)中给出的 LR。在本次实验中评估了这三个指标, 并通过实验结果证明了之前的理论分析。

在给定嘈杂的数据集  $D$  上, 作者通过交叉验证将  $D$  随机分为两半  $D_1, D_2$ , 然后在  $D_1$  上训练网络 ResNet-110 并在  $D_2$  上进行测试。

实验结果如下图所示:

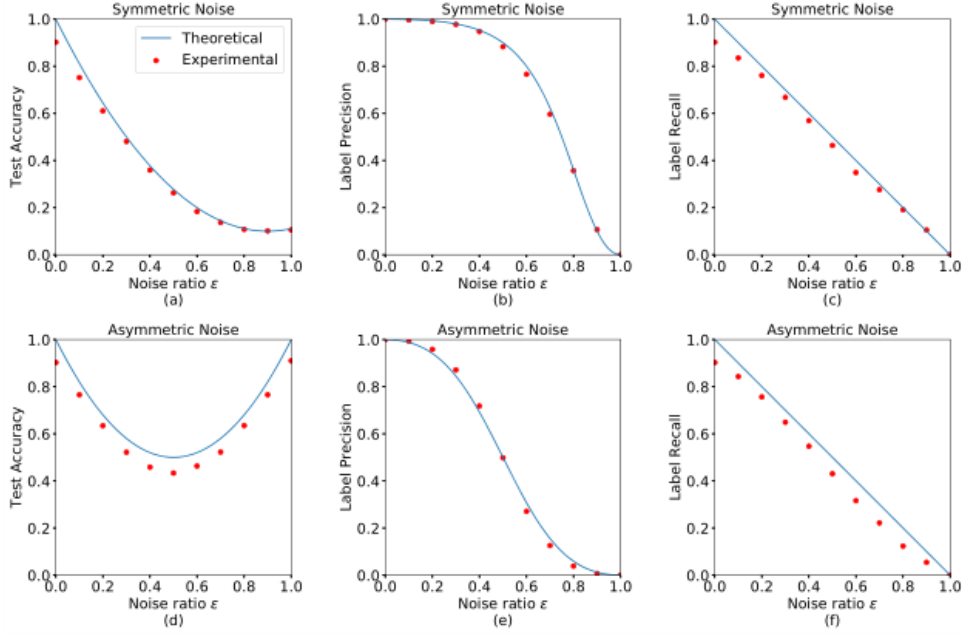


图 1. 实验一噪声率 $\epsilon$ 关系图

图 1 是关于实验一中相关指标的图示。其中包含测试精度, 标签精度(LP) 和标签召回率 (LR) 分别与噪声比的关系。第一行对应对称噪声, 第二行对应非对称噪声。实验结果与理论估计一致, 特别是, 图 1 (a) 显示出测试精度高度依赖于噪声比, 论文在式 (4) 中明确表示——测试精度是噪声比的二次函数。在图 1 (b) 和 (e) 中, 实验 LP 由我们的公式精确给出, LR 类似。实验结果基本与理论曲线吻合。

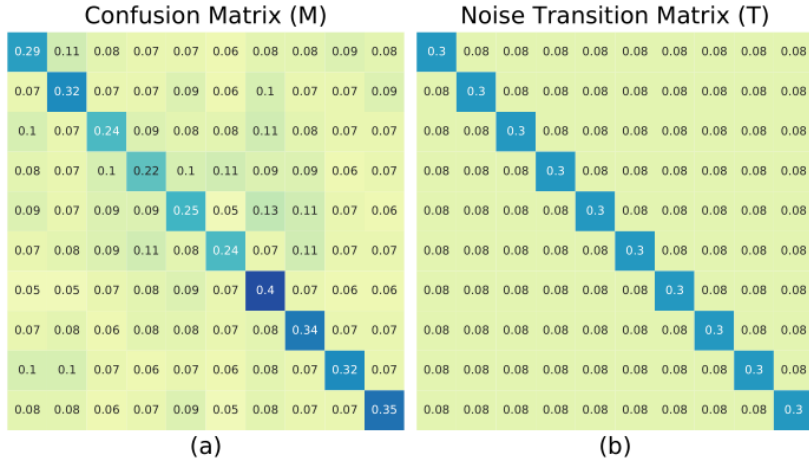


图 2. RseNet-110 混淆矩阵(M) 以及噪声转换矩阵(T)

为了进一步研究使用噪声标签训练的 DNN 的预测行为, 我们定义了一个混淆矩阵  $M$ , 其第  $ij$  个条目表示预测第  $i$  类测试样本为  $j$  的概率,

$$M_{ij} := P(y^f = j | \hat{y} = i)$$

图 2 展示了在对称噪声比为 0.7 的 CIFAR-10 上训练的 DNN 的混淆矩阵, 可以发现  $M \approx T$  满足断言 1 中的陈述。

## 实验二. 使用INCV鉴定干净的样本

图 1 (b) 和 (e) 验证了算法 1 选择的子集通常具有比原始设置小的噪声比。有时, 训练 DNN 需要大量训练样本。这次实验演示了算法 2 (INCV) 可以通过迭代识别更多干净的样本。为了提高效率, 实验使用 ResNet-32 并设置  $N=4$ ,  $E=50$  而不进行微调。所有实验中 $\epsilon$ 是使用公式 (4) 自动估算的。

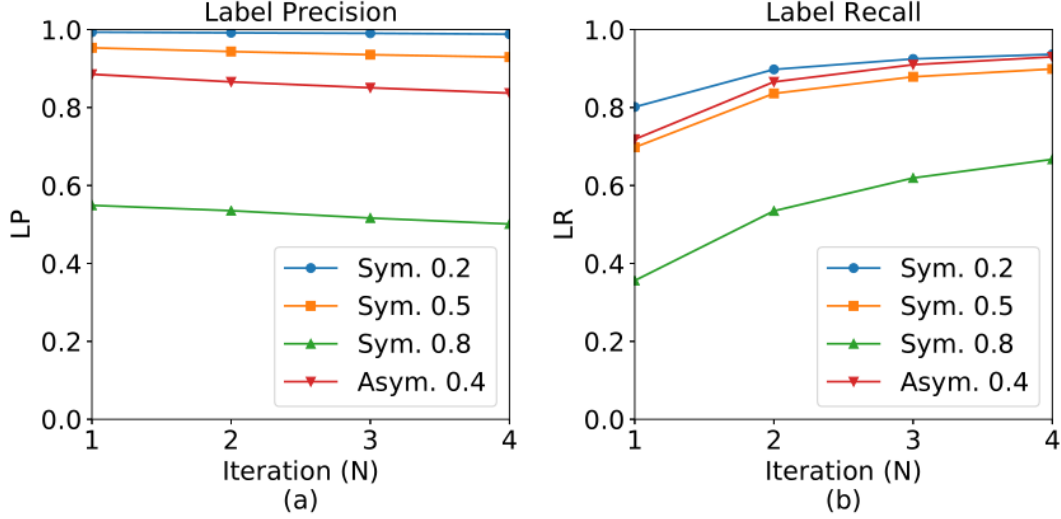


图 3. 人工噪声 CIFAR-10 数据集上 INCV 的 LP 和 LR

在每个图中, 四个曲线分别对应于比率为 0.2、0.5、0.8 的对称噪声和比率为 0.4 的非对称噪声。

INCV 可以准确识别大多数干净样品。图 3 展示出了算法 2 通过重复实验 5 次计算的平均 LP 和 LR 值。如图所示, 即使经过一次迭代, LP 和 LR 也比理论下限更好。与实验一使用的 ResNet-110 相比, 本次实验使用较为简单的 ResNet-32。一个简单得多的模型自然会消除过拟合问题, 从而产生更好的 LP 和 LR。此外, 图 4 还表明 LR 随着迭代的增加而大大增加, 而 LP 则略有下降。经过四次迭代, INCV 可以准确识别出大多数干净的样品。在比率为 0.5 的对称噪声下, 它将选择大约 90% (LR) 的干净样本, 并且所选集合的噪声比率会降低到大约 10% ( $1-LP$ )。

## 实验三. 针对噪声标签训练DNN的鲁棒性

本次实验是在算法三的基础上进行的, 算法中的数据处理等步骤在实验里的具体处理作如下说明:

合理的设置从数据集  $C$  和  $S$  提取的小批次数据  $B_C$  和  $B_S$ 。但是, 当  $C$  很大时, 会导致从  $C$  提取太多样本, 这会损害训练过程, 因为  $C$  通常包含许多损坏的样本。因此可以通过设置  $|B_C|/|B_S| = \min(0.5, |C|/|S|)$  来稍微调整策略。在本次实验中设置批次大小  $|B_S|$  到 128, 然后相应地计算  $|B_C|$ 。

每个小批量样品中应保留的样品数是这样计算的: 在每个小批量中使用训练损失较小的  $n(e)$  个样本来更新网络, 其中  $e$  是当前周期。设置  $n(e) = |B_S|(1 -$



$\varepsilon_S \min\left(\frac{e}{10}, 1\right)$ ), 这意味着实验是在前 10 个周期线性地从  $|B_S|$  中减去  $n(e)$  到  $|B_S|(1 - \varepsilon_S)$ , 然后再去调整。同时, 为了证明实验的效果, 本次实验还和几个比较流行的算法进行效果比较, 算法有: (1)F-校正 (2)解耦 (3)Co-teaching (4)MentorNet (5)D2L。

首先通过在数据集 CIFAR-10 上手动制造带有不同类型噪声的标签来评估所有方法。对于对称噪声, 我们设置测试噪声比为 0.2, 0.5 和 0.8。对于非对称噪声, 我们设置噪声比 0.4, 因为大于 0.5 的非对称噪声是微不足道。实验仍使用 ResNet-32 并将所有实验重复五次。

表 1. 在不同噪声类型和噪声比下的平均测试准确度

Method	Sym.			Asym.
	0.2	0.5	0.8	0.4
F-correction	85.08 $\pm 0.43$	76.02 $\pm 0.19$	34.76 $\pm 4.53$	83.55 $\pm 2.15$
Decoupling	86.72 $\pm 0.32$	79.31 $\pm 0.62$	36.90 $\pm 4.61$	75.27 $\pm 0.83$
Co-teaching	89.05 $\pm 0.32$	82.12 $\pm 0.59$	16.21 $\pm 3.02$	84.55 $\pm 2.81$
MentorNet	88.36 $\pm 0.46$	77.10 $\pm 0.44$	28.89 $\pm 2.29$	77.33 $\pm 0.79$
D2L	86.12 $\pm 0.43$	67.39 $\pm 13.62$	10.02 $\pm 0.04$	85.57 $\pm 1.21$
Ours	<b>89.71</b> $\pm 0.18$	<b>84.78</b> $\pm 0.33$	<b>52.27</b> $\pm 3.50$	<b>86.04</b> $\pm 0.54$

如表 1 所示, 在所有情况下, 算法 3 的方法始终可以达到最佳测试精度 (以黑体字标出)。即使对于大多数方法都具有挑战性的比率为 0.8 的对称噪声, 算法 3 也可以达到良好的测试精度。

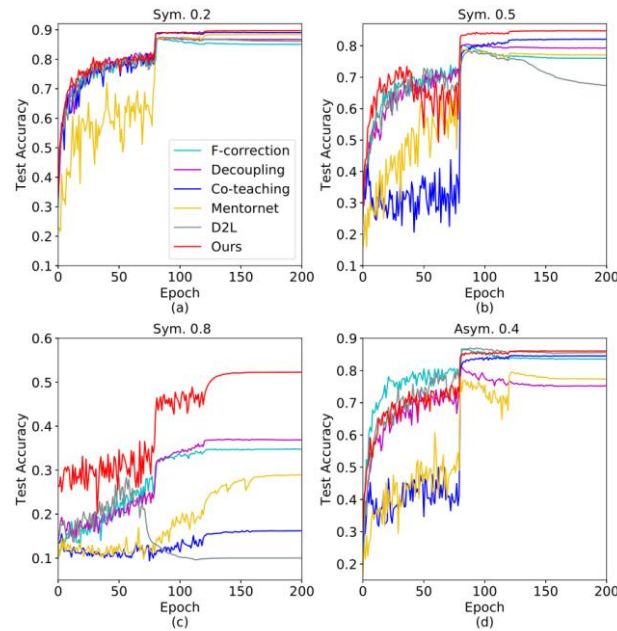


图 4. 在不同的噪音类型和噪音比率下的测试精度图

图 4 展示了在每个训练时期之后，在干净测试集上所有方法的测试准确性。可以发现，算法 3 在所有设置下都能令人印象深刻地达到最佳测试精度，而某些其他方法在训练的后期阶段会过度拟合，例如图 4（b）和（d）中所示的 F 校正，去耦和 MentorNet。

表 2. WebVision 验证集和 ImageNet ILSVRC12 验证集的验证准确性

Method	WebVision Val.	ILSVRC2012 Val.
F-correction	61.12 (82.68)	57.36 (82.36)
Decoupling	62.54 (84.74)	58.26 (82.26)
Co-teaching	63.58 (85.20)	61.48 (84.70)
MentorNet	63.00 (81.40)	57.80 (79.92)
D2L	62.68 (84.00)	57.80 (81.36)
Ours	<b>65.24 (85.34)</b>	<b>61.60 (84.98)</b>

为了验证算法 3 在实际嘈杂标签上的实际用法，作者使用了 WebVision 数据集 1.0，其训练集包含许多真实世界的嘈杂标签。由于数据集很大，为了进行快速实验，实验使用 Inception-Resnet v2 比较了 Google 图像子集的前 50 个类的所有方法。实验在人工注释的 WebVision 验证集和 ILSVRC12 验证集上测试训练后的模型。如表 2 所示，在测试准确性方面，算法 3 始终优于其他最新技术。

## 六、 研究结论

这篇论文首先通过针对噪声标签训练的 DNN 的泛化提出了一些发现，然后通过理论分析和广泛的实验证明论文中陈述的正确性。 作者根据自己的发现提出了 INCV 方法，该方法将噪声数据集随机划分，然后利用交叉验证来识别干净的样本。本文为 INCV 提供了理论上的保证，然后通过实验证明它能够准确识别大多数清洁样品。最后采用提出的算法 3 利用已识别的样本来针对噪声标签对 DNN 进行鲁棒性训练。 通过与较多的其他方法进行比较，证明了算法 3 在干净的测试数据上达到了最新的测试精度。

## 七、 阅读收获

在机器学习领域中，常见的一类工作是使用带标签数据训练神经网络实现分类、回归或其他目的。在这种需要标签的监督学习中，训练数据所对应的标签质量对于学习效果至关重要。同时，深度学习使用的神经网络往往结构复杂，为了得到良好的学习效果，对于带标签的训练数据的数量也有较高要求。特别是对于深层神经网络 (DNN)，一旦标签出现较多噪声，则训练很容易导致模型无法准确的进行泛化预测。

作者针对此问题提出了本篇论文中的几个观点，那就是噪声率与 DNN 泛化关系

之间是存在某种二次函数关系，然后作者针对此情况提出了 INCV 算法去识别筛选噪声标签小的数据集，并最后提出了提高 DNN 训练鲁棒性的算法，最后通过实验证明其理论和算法的正确性。

从该篇论文中，可以看出作者在写这篇文章的逻辑性，从发现问题到提出理论观点再到实验证明，相关部分都是紧紧呼应，在这种思维方式上是需要学习的。对于作者在论文中所提出的方法，也是针对当前比较现实的问题入手。但是在实验过程中所训练的 DNN 过于单一，而且训练数据的噪声是人工制造的，显得没那么具有说服力。不过最后通过在真实噪声下的实验结果证明算法的可靠性，也是弥补这一不足。