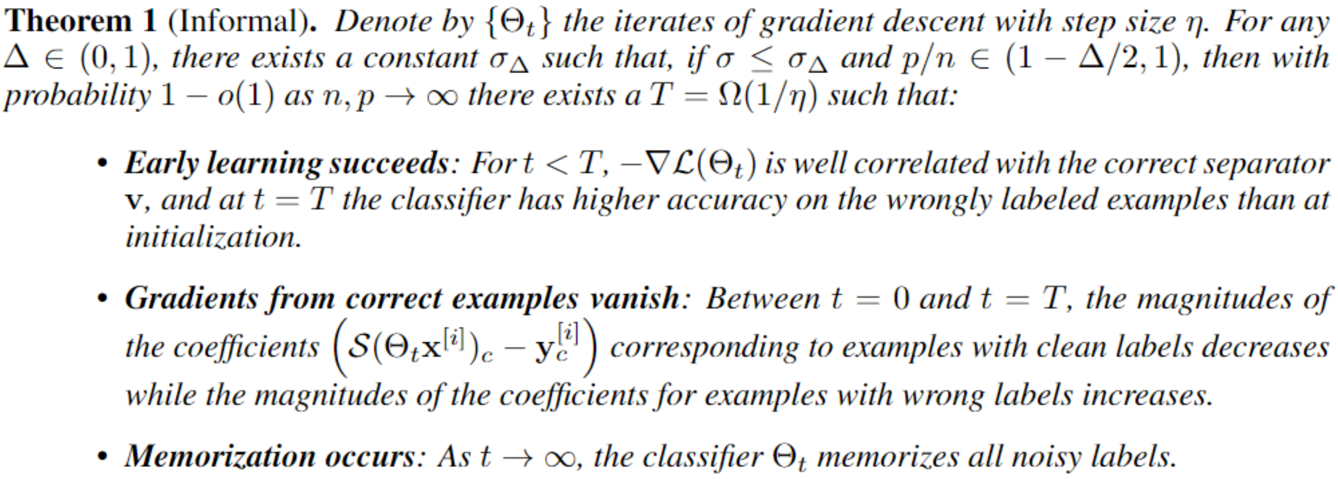
要点

- 详细分析了memorization effect 的形成过程
- 运用semi-supervised 的方法设计出了有记忆性的正则化
- 用了该方法后就不用early stopping，网络会抑制wrong label的fitting

方法

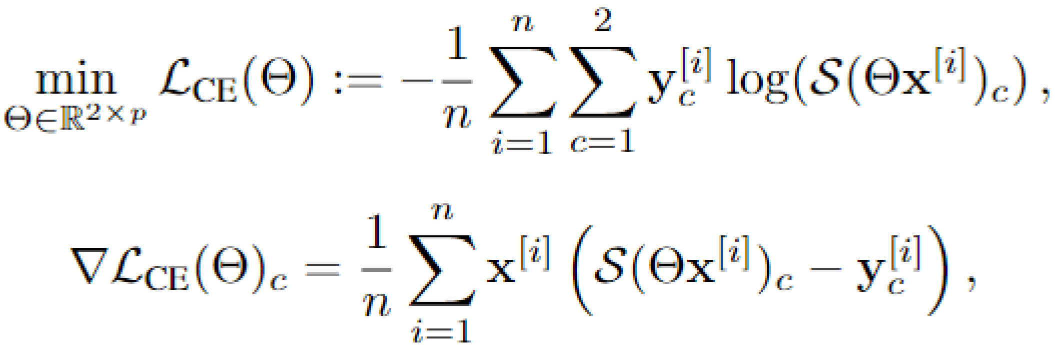
模型的memorization effect



Theorem 1 (Informal)**.** *Denote by $\{\Theta_t\}$ the iterates of gradient descent with step size η. For any $\Delta \in (0,1)$, there exists a constant σ_Δ such that, if $\sigma \leq \sigma_\Delta$ and $p/n \in (1-\Delta/2, 1)$, then with probability $1-o(1)$ as $n, p \to \infty$ there exists a $T = \Omega(1/\eta)$ such that:*

- ***Early learning succeeds****: For $t < T$, $-\nabla\mathcal{L}(\Theta_t)$ is well correlated with the correct separator $\mathbf{v}$, and at $t = T$ the classifier has higher accuracy on the wrongly labeled examples than at initialization.*
- ***Gradients from correct examples vanish****: Between $t = 0$ and $t = T$, the magnitudes of the coefficients $\left(\mathcal{S}(\Theta_t\mathbf{x}^{[i]})_c - \mathbf{y}_c^{[i]}\right)$ corresponding to examples with clean labels decreases while the magnitudes of the coefficients for examples with wrong labels increases.*
- ***Memorization occurs****: As $t \to \infty$, the classifier Θ_t memorizes all noisy labels.*

从线性模型的损失函数可以定性证明以上结论（S为softmax）



$$\min_{\Theta \in \mathbb{R}^{2\times p}} \mathcal{L}_{\mathrm{CE}}(\Theta) := -\frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{2}\mathbf{y}_c^{[i]}\log(\mathcal{S}(\Theta\mathbf{x}^{[i]})_c)\,,$$

$$\nabla\mathcal{L}_{\mathrm{CE}}(\Theta)_c = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}^{[i]}\left(\mathcal{S}(\Theta\mathbf{x}^{[i]})_c - \mathbf{y}_c^{[i]}\right),$$

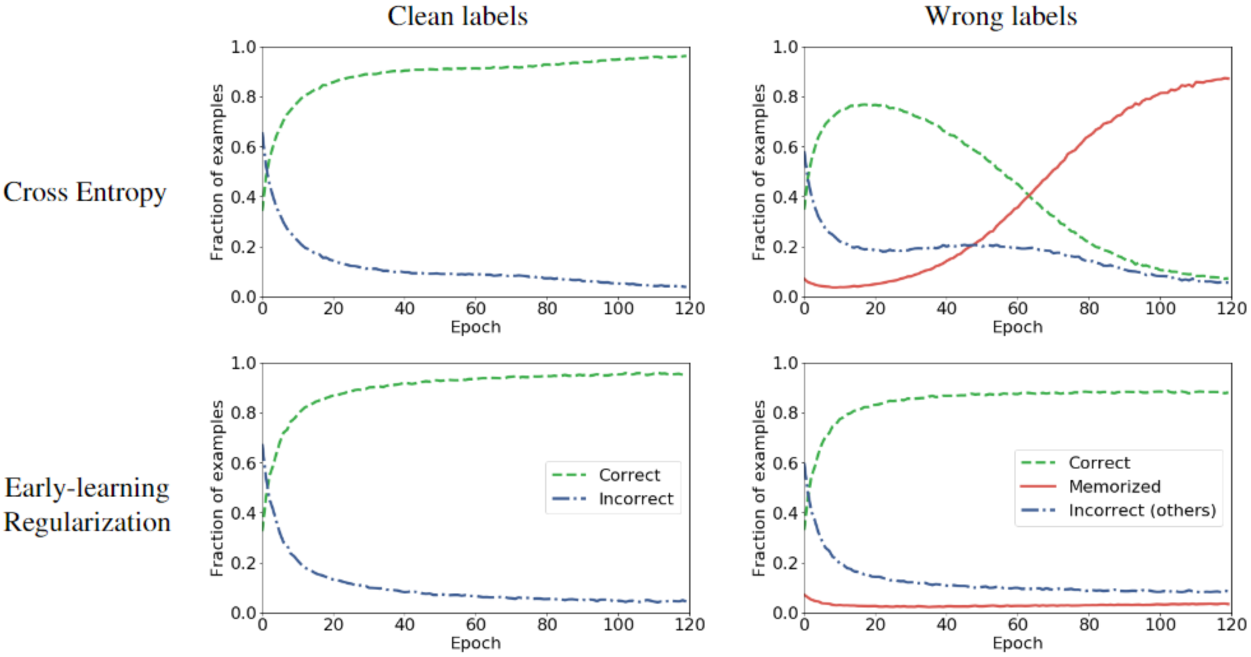
分析真实类别是c的clean label和wrong label，clean label 占多数

1.当训练早期， Θ还没有与v相关联时，$S(\Theta x_c^{[i]}) - y_c^{[i]}$对于任何样本都很大的，因此$\bigtriangledown L_c$大致指向全体样本的平均方向，也就是简单模式

2.当 Θ与v相关联时，此时正确标签的梯度可以忽略，因此错误标签的梯度开始起主导作用，同时错误标签的梯度会指向正确梯度的垂直方向，最终记住noisy label。注意此时并不会影响clean label的梯度消失。

$$\nabla \mathcal{L}_{\mathrm{CE}}(\Theta)=\frac{1}{n} \sum_{i=1}^{n} \nabla \mathcal{N}_{\mathbf{x}^{[i]}}(\Theta)\left(\mathbf{p}^{[i]}-\mathbf{y}^{[i]}\right),$$

对于神经网络亦然。并且可以发现，梯度主要受限于 $p^{[i]}-y^{[i]}$



正则化项

因此，根据之前的分析。我们的目标是：第二阶段正确标签的梯度不消失，仍然起主导作用

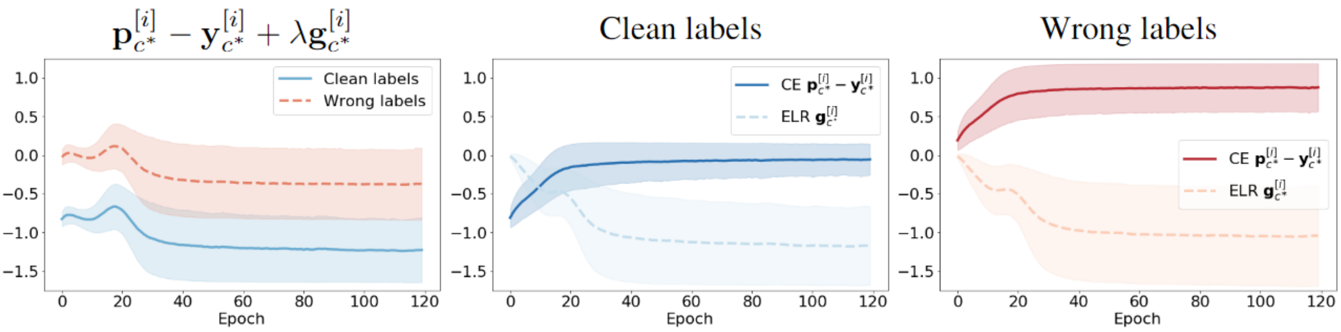


Figure 2: Illustration of the effect of the regularization on the gradient of the ELR loss (see Lemma 2) for the same deep-learning model as in Figure 1. On the left, we plot the entry of $\mathbf{p}^{[i]} - \mathbf{y}^{[i]} + \lambda \mathbf{g}^{[i]}$ corresponding to the true class, denoted by c^*, for training examples with clean (blue) and wrong (red) labels. The center image shows the c^*th entry of the cross-entropy (CE) term $\mathbf{p}^{[i]} - \mathbf{y}^{[i]}$ (dark blue) and the regularization term $\mathbf{g}^{[i]}$ (light blue) separately for the examples with clean labels. During early learning the CE term dominates, but afterwards it vanishes as the model learns the clean labels (i.e. $\mathbf{p}^{[i]} \approx \mathbf{y}^{[i]}$). However, the regularization term compensates for this, forcing the model to continue learning mainly on the examples with clean labels. On the right, we show the CE and the regularization term (dark and light red respectively) separately for the examples with wrong labels. The regularization cancels out the CE term, preventing memorization. In all plots the curves represent the mean value, and the shaded regions are within one standard deviation of the mean.

$$\mathcal{L}_{\text{ELR}}(\Theta) := \mathcal{L}_{\text{CE}}(\Theta) + \frac{\lambda}{n} \sum_{i=1}^{n} \log\left(1 - \langle \mathbf{p}^{[i]}, \mathbf{t}^{[i]} \rangle\right).$$

Lemma 2 (Gradient of the ELR loss). *The gradient of the loss defined in Eq. (6) is equal to*

$$\nabla \mathcal{L}_{ELR}(\Theta) = \frac{1}{n} \sum_{i=1}^{n} \nabla \mathcal{N}_{\mathbf{x}^{[i]}}(\Theta) \left(\mathbf{p}^{[i]} - \mathbf{y}^{[i]} + \lambda \mathbf{g}^{[i]}\right) \tag{7}$$

where the entries of $\mathbf{g}^{[i]} \in \mathbb{R}^C$ *are given by*

$$\mathbf{g}_c^{[i]} := \frac{\mathbf{p}_c^{[i]}}{1 - \langle \mathbf{p}^{[i]}, \mathbf{t}^{[i]} \rangle} \sum_{k=1}^{C} (\mathbf{t}_k^{[i]} - \mathbf{t}_c^{[i]}) \mathbf{p}_k^{[i]}, \qquad 1 \le c \le C. \tag{8}$$

这个是N是雅可比矩阵，但是他给的是x对于theta求导的雅可比矩阵，一般都是theta对于x求导的矩阵

$$\mathbf{t}^{[i]}(k) := \beta \mathbf{t}^{[i]}(k-1) + (1-\beta)\mathbf{p}^{[i]}(k),$$

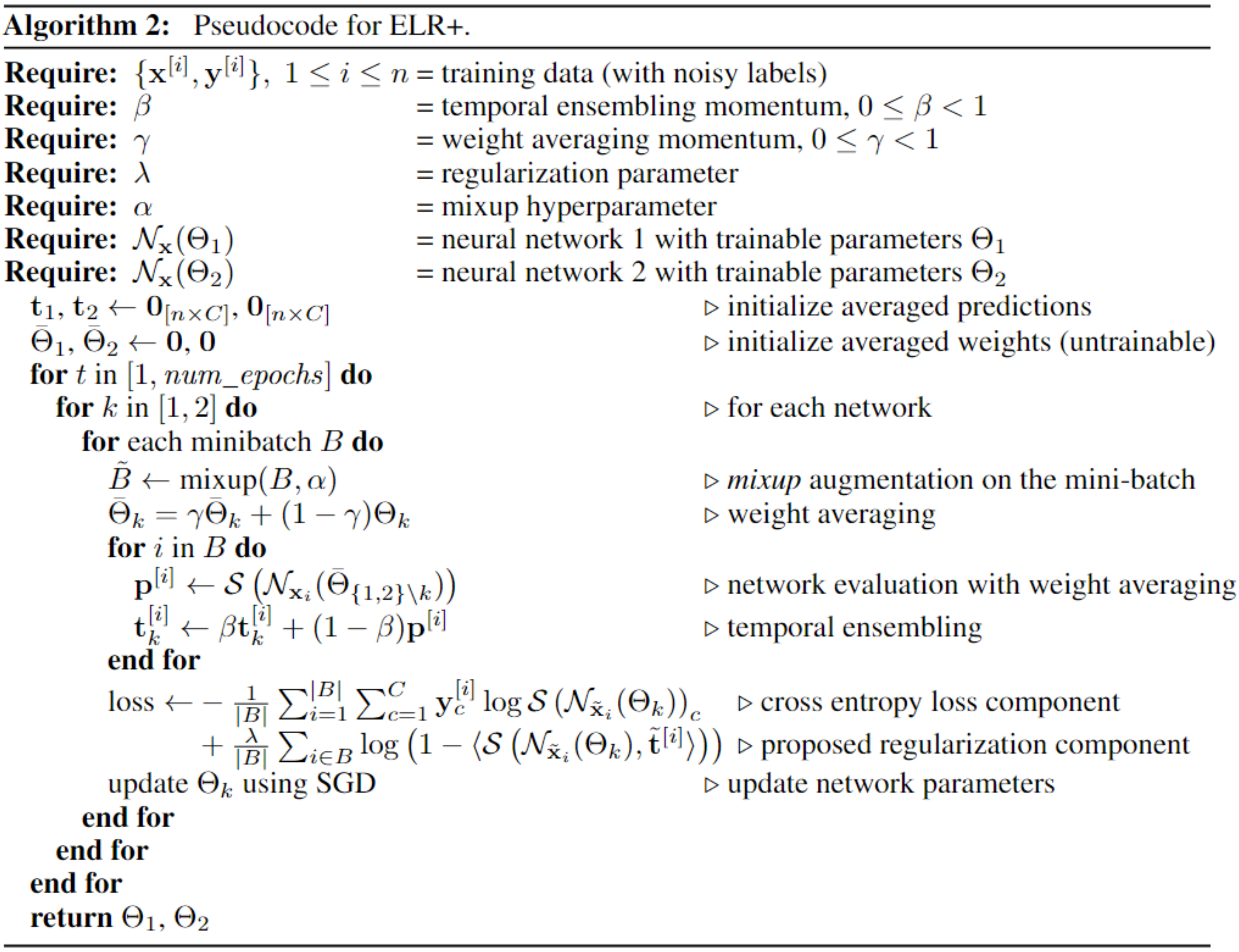
加入这个正则化项后：

early learning时t_{c^*}为主导地位，因此g_{c^*}为负。

随着clean label 的p_{c^*}越来越大，g_{c^*}更小，抑制了clean label的继续学习。虽然一定程度上抑制了，但是多个clean label的梯度之和仍能占据主要地位。

wrong label的p_{c^*}在memorization effect之前都是接近于0，因此t_{c^*}几乎为0，g_{c^*}为正，一定程度上“骗过了”损失函数。

加入一些trick的强化模型ELR+



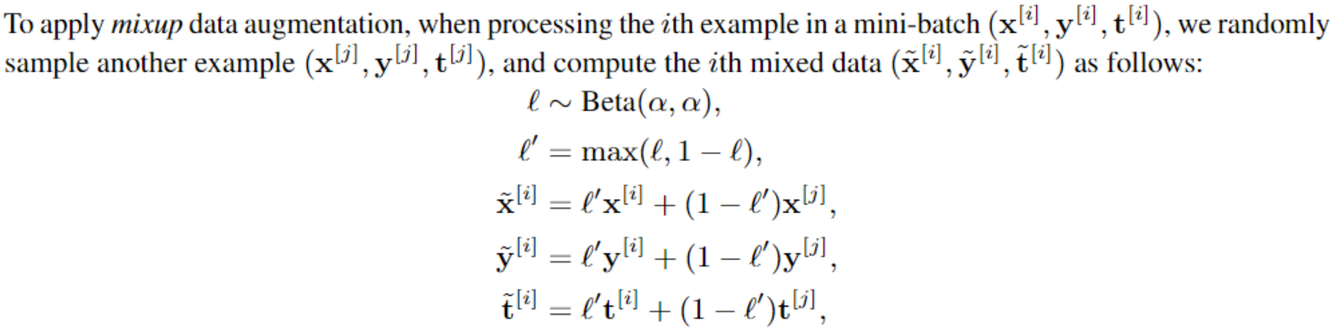
Algorithm 2: Pseudocode for ELR+.

```
Require: ${\mathbf{x}^{[i]}, \mathbf{y}^{[i]}}, 1 \leq i \leq n$ = training data (with noisy labels)
Require: $\beta$ = temporal ensembling momentum, $0 \leq \beta < 1$
Require: $\gamma$ = weight averaging momentum, $0 \leq \gamma < 1$
Require: $\lambda$ = regularization parameter
Require: $\alpha$ = mixup hyperparameter
Require: $\mathcal{N}_{\mathbf{x}}(\Theta_1)$ = neural network 1 with trainable parameters $\Theta_1$
Require: $\mathcal{N}_{\mathbf{x}}(\Theta_2)$ = neural network 2 with trainable parameters $\Theta_2$
  $\mathbf{t}_1, \mathbf{t}_2 \leftarrow \mathbf{0}_{[n \times C]}, \mathbf{0}_{[n \times C]}$      ▷ initialize averaged predictions
  $\bar{\Theta}_1, \bar{\Theta}_2 \leftarrow \mathbf{0}, \mathbf{0}$      ▷ initialize averaged weights (untrainable)
  for $t$ in $[1, num\_epochs]$ do
    for $k$ in $[1, 2]$ do      ▷ for each network
      for each minibatch $B$ do
        $\tilde{B} \leftarrow \text{mixup}(B, \alpha)$      ▷ mixup augmentation on the mini-batch
        $\bar{\Theta}_k = \gamma \bar{\Theta}_k + (1 - \gamma)\Theta_k$      ▷ weight averaging
        for $i$ in $B$ do
          $\mathbf{p}^{[i]} \leftarrow \mathcal{S}\left(\mathcal{N}_{\mathbf{x}_i}(\bar{\Theta}_{\{1,2\} \setminus k})\right)$      ▷ network evaluation with weight averaging
          $\mathbf{t}_k^{[i]} \leftarrow \beta \mathbf{t}_k^{[i]} + (1 - \beta)\mathbf{p}^{[i]}$      ▷ temporal ensembling
        end for
        loss $\leftarrow -\frac{1}{|B|}\sum_{i=1}^{|B|}\sum_{c=1}^{C} \mathbf{y}_c^{[i]} \log \mathcal{S}\left(\mathcal{N}_{\tilde{\mathbf{x}}_i}(\Theta_k)\right)_c$      ▷ cross entropy loss component
            $+ \frac{\lambda}{|B|}\sum_{i \in B} \log\left(1 - \langle \mathcal{S}\left(\mathcal{N}_{\tilde{\mathbf{x}}_i}(\Theta_k)\right), \tilde{\mathbf{t}}^{[i]} \rangle\right)$      ▷ proposed regularization component
        update $\Theta_k$ using SGD      ▷ update network parameters
      end for
    end for
  end for
  return $\Theta_1, \Theta_2$
```

1. 双网络训练，权重平均后预测

2. 网络的权重也采用 temporal ensembling

3. mixup data augmentation:

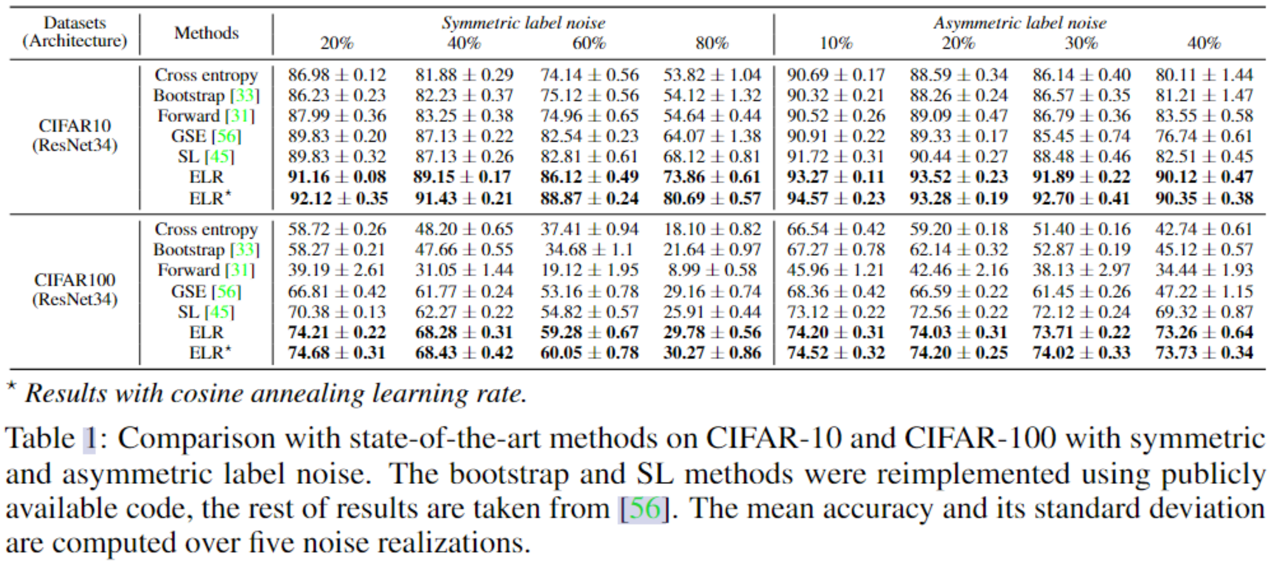


To apply *mixup* data augmentation, when processing the ith example in a mini-batch $(\mathbf{x}^{[i]}, \mathbf{y}^{[i]}, \mathbf{t}^{[i]})$, we randomly sample another example $(\mathbf{x}^{[j]}, \mathbf{y}^{[j]}, \mathbf{t}^{[j]})$, and compute the ith mixed data $(\tilde{\mathbf{x}}^{[i]}, \tilde{\mathbf{y}}^{[i]}, \tilde{\mathbf{t}}^{[i]})$ as follows:

$$\begin{aligned}
\ell &\sim \text{Beta}(\alpha, \alpha), \\
\ell' &= \max(\ell, 1 - \ell), \\
\tilde{\mathbf{x}}^{[i]} &= \ell' \mathbf{x}^{[i]} + (1 - \ell')\mathbf{x}^{[j]}, \\
\tilde{\mathbf{y}}^{[i]} &= \ell' \mathbf{y}^{[i]} + (1 - \ell')\mathbf{y}^{[j]}, \\
\tilde{\mathbf{t}}^{[i]} &= \ell' \mathbf{t}^{[i]} + (1 - \ell')\mathbf{t}^{[j]},
\end{aligned}$$

实验

测试模型在不同类型噪声、不同噪声率下的准确率

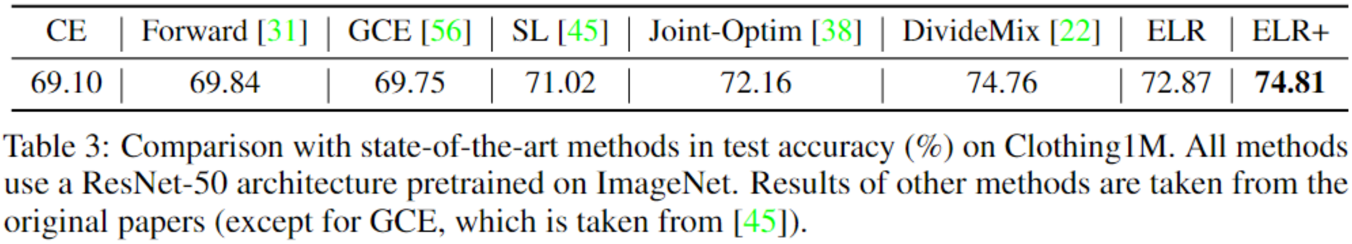


Datasets (Architecture)	Methods	*Symmetric label noise* 20%	40%	60%	80%	*Asymmetric label noise* 10%	20%	30%	40%
CIFAR10 (ResNet34)	Cross entropy	86.98 ± 0.12	81.88 ± 0.29	74.14 ± 0.56	53.82 ± 1.04	90.69 ± 0.17	88.59 ± 0.34	86.14 ± 0.40	80.11 ± 1.44
	Bootstrap [33]	86.23 ± 0.23	82.23 ± 0.37	75.12 ± 0.56	54.12 ± 1.32	90.32 ± 0.21	88.26 ± 0.24	86.57 ± 0.35	81.21 ± 1.47
	Forward [31]	87.99 ± 0.36	83.25 ± 0.38	74.96 ± 0.65	54.64 ± 0.44	90.52 ± 0.26	89.09 ± 0.47	86.79 ± 0.36	83.55 ± 0.58
	GSE [56]	89.83 ± 0.20	87.13 ± 0.22	82.54 ± 0.23	64.07 ± 1.38	90.91 ± 0.22	89.33 ± 0.17	85.45 ± 0.74	76.74 ± 0.61
	SL [45]	89.83 ± 0.32	87.13 ± 0.26	82.81 ± 0.61	68.12 ± 0.81	91.72 ± 0.31	90.44 ± 0.27	88.48 ± 0.46	82.51 ± 0.45
	ELR	**91.16 ± 0.08**	**89.15 ± 0.17**	**86.12 ± 0.49**	**73.86 ± 0.61**	**93.27 ± 0.11**	**93.52 ± 0.23**	**91.89 ± 0.22**	**90.12 ± 0.47**
	ELR*	**92.12 ± 0.35**	**91.43 ± 0.21**	**88.87 ± 0.24**	**80.69 ± 0.57**	**94.57 ± 0.23**	**93.28 ± 0.19**	**92.70 ± 0.41**	**90.35 ± 0.38**
CIFAR100 (ResNet34)	Cross entropy	58.72 ± 0.26	48.20 ± 0.65	37.41 ± 0.94	18.10 ± 0.82	66.54 ± 0.42	59.20 ± 0.18	51.40 ± 0.16	42.74 ± 0.61
	Bootstrap [33]	58.27 ± 0.21	47.66 ± 0.55	34.68 ± 1.1	21.64 ± 0.97	67.27 ± 0.78	62.14 ± 0.32	52.87 ± 0.19	45.12 ± 0.57
	Forward [31]	39.19 ± 2.61	31.05 ± 1.44	19.12 ± 1.95	8.99 ± 0.58	45.96 ± 1.21	42.46 ± 2.16	38.13 ± 2.97	34.44 ± 1.93
	GSE [56]	66.81 ± 0.42	61.77 ± 0.24	53.16 ± 0.78	29.16 ± 0.74	68.36 ± 0.42	66.59 ± 0.22	61.45 ± 0.26	47.22 ± 1.15
	SL [45]	70.38 ± 0.13	62.27 ± 0.22	54.82 ± 0.57	25.91 ± 0.44	73.12 ± 0.22	72.56 ± 0.22	72.12 ± 0.24	69.32 ± 0.87
	ELR	**74.21 ± 0.22**	**68.28 ± 0.31**	**59.28 ± 0.67**	**29.78 ± 0.56**	**74.20 ± 0.31**	**74.03 ± 0.31**	**73.71 ± 0.22**	**73.26 ± 0.64**
	ELR*	**74.68 ± 0.31**	**68.43 ± 0.42**	**60.05 ± 0.78**	**30.27 ± 0.86**	**74.52 ± 0.32**	**74.20 ± 0.25**	**74.02 ± 0.33**	**73.73 ± 0.34**

* *Results with cosine annealing learning rate.*

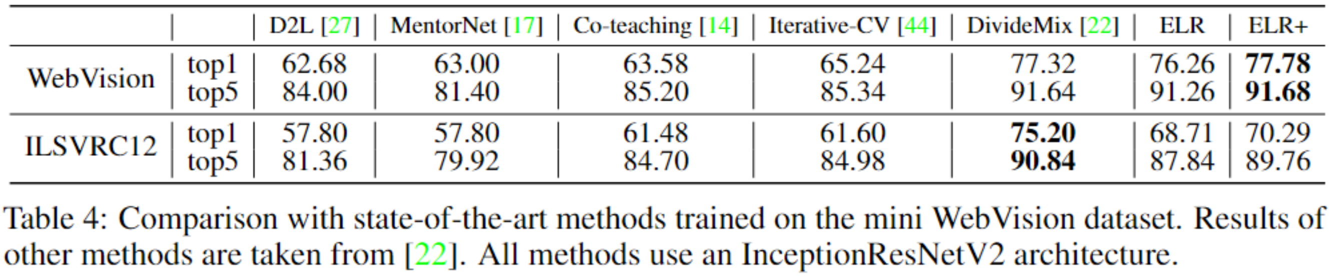
Table 1: Comparison with state-of-the-art methods on CIFAR-10 and CIFAR-100 with symmetric and asymmetric label noise. The bootstrap and SL methods were reimplemented using publicly available code, the rest of results are taken from [56]. The mean accuracy and its standard deviation are computed over five noise realizations.

在现实数据集(Clothing 1M, WebVision)中的对比



CE	Forward [31]	GCE [56]	SL [45]	Joint-Optim [38]	DivideMix [22]	ELR	ELR+
69.10	69.84	69.75	71.02	72.16	74.76	72.87	**74.81**

Table 3: Comparison with state-of-the-art methods in test accuracy (%) on Clothing1M. All methods use a ResNet-50 architecture pretrained on ImageNet. Results of other methods are taken from the original papers (except for GCE, which is taken from [45]).



		D2L [27]	MentorNet [17]	Co-teaching [14]	Iterative-CV [44]	DivideMix [22]	ELR	ELR+
WebVision	top1	62.68	63.00	63.58	65.24	77.32	76.26	**77.78**
	top5	84.00	81.40	85.20	85.34	91.64	91.26	**91.68**
ILSVRC12	top1	57.80	57.80	61.48	61.60	**75.20**	68.71	70.29
	top5	81.36	79.92	84.70	84.98	**90.84**	87.84	89.76

Table 4: Comparison with state-of-the-art methods trained on the mini WebVision dataset. Results of other methods are taken from [22]. All methods use an InceptionResNetV2 architecture.

缺省实验

			40%		80%	
			Weight Averaging		Weight Averaging	
			✓	✗	✓	✗
1 Network	mixup	✓	93.04 ± 0.12	91.05 ± 0.13	87.23 ± 0.30	81.43 ± 0.52
		✗	92.09 ± 0.08	90.83 ± 0.07	76.50 ± 0.65	72.54 ± 0.35
2 Networks	mixup	✓	93.68 ± 0.51	93.51 ± 0.47	88.62 ± 0.26	84.75 ± 0.26
		✗	92.95 ± 0.05	91.86 ± 0.14	80.13 ± 0.51	73.49 ± 0.47