Beyond Synthetic Noise: Deep Learning on Controlled Noisy Labels

Lu Jiang Di Huang Mason Liu Weilong Yang

ICML 2020

Background

Performing controlled experiments on noisy data is essential in understanding deep learning across noise levels

previous research has only examined deep learning on controlled synthetic label noise

- An approach that works well on artificial noise may not work well on real-world noise data sets

contribution

1. establish the first benchmark of controlled real-world label noise from the web

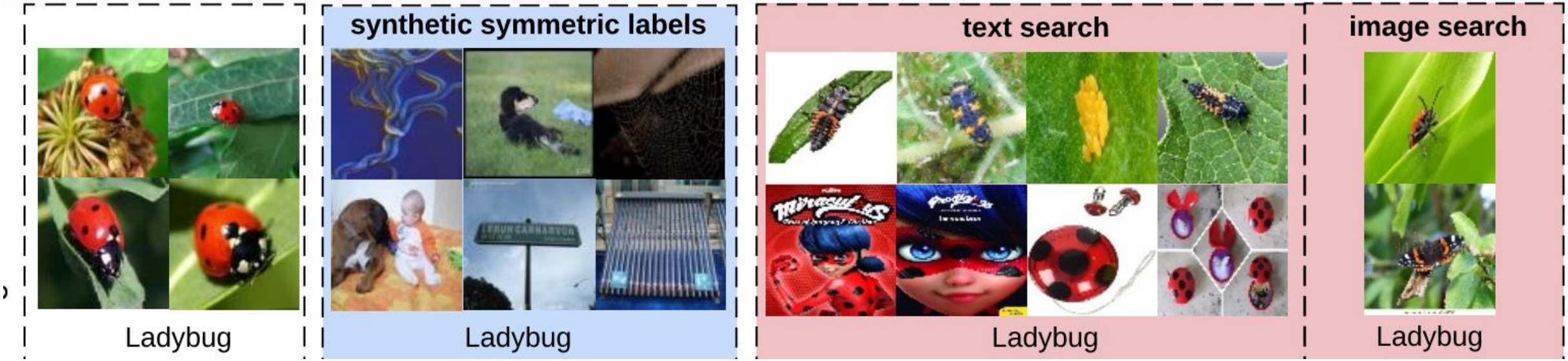
2. a simple but effective method to overcome both synthetic and real noisy labels

3. conduct the largest study into understanding deep neural networks trained on noisy labels across different noise levels, noise types, network architectures, and training settings

Background

The difference between synthetic label noise and real label noise

- images with noise from the web (or red noise) are more relevant (visually or semantically) to true positive images.
- synthetic noise (symmetric or asymmetric) is at class-level.
- images with noise from the web come from an open vocabulary outside the class vocabulary of Mini-ImageNet or Stanford Cars.



benchmark

Dataset Construction

- Based on Mini-ImageNet and Stanford Cars
- Replace clean training images in the original data set with mislabeled Web images

MentorMix

Mentornet

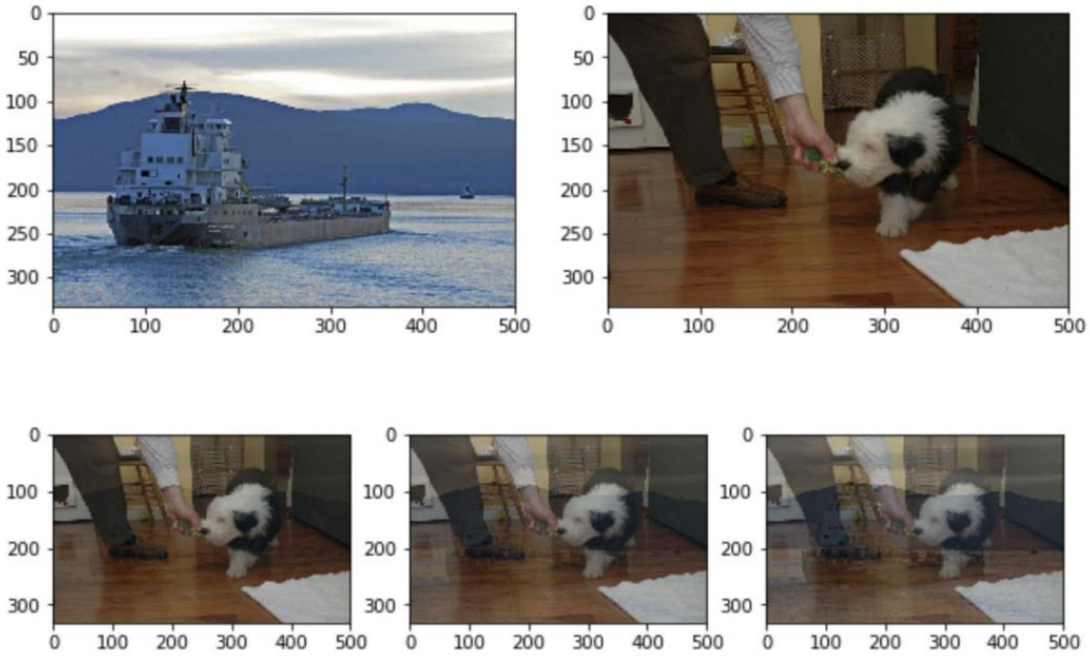
- noise labels get low weight, while clean labels get high weight
- MentorNet can be seen as a function. The input is the historical information of the loss generated by the current batch sample, and the output is the weight calculated according to the loss information

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}\in\mathbb{R}^d, \mathbf{v}\in[0,1]^n}{\operatorname{argmin}} \mathbb{F}(\mathbf{v}, \boldsymbol{w})$$

$$= \frac{1}{n}\sum_{i=1}^{n} v_i \ell(g_s(\mathbf{x}_i;\boldsymbol{w}), y_i) + \theta\|\boldsymbol{w}\|_2^2 + G(\mathbf{v};\gamma)$$

MentorMix

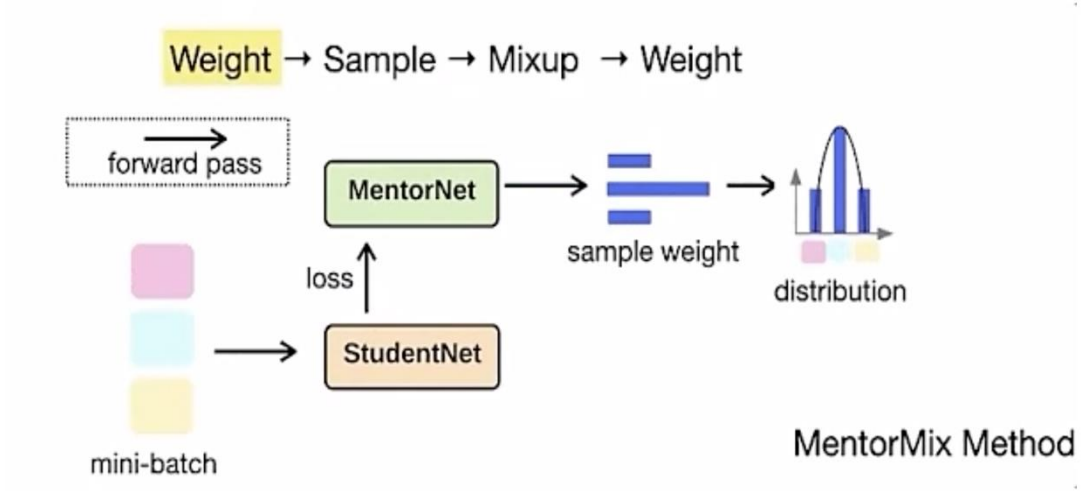
Mixup

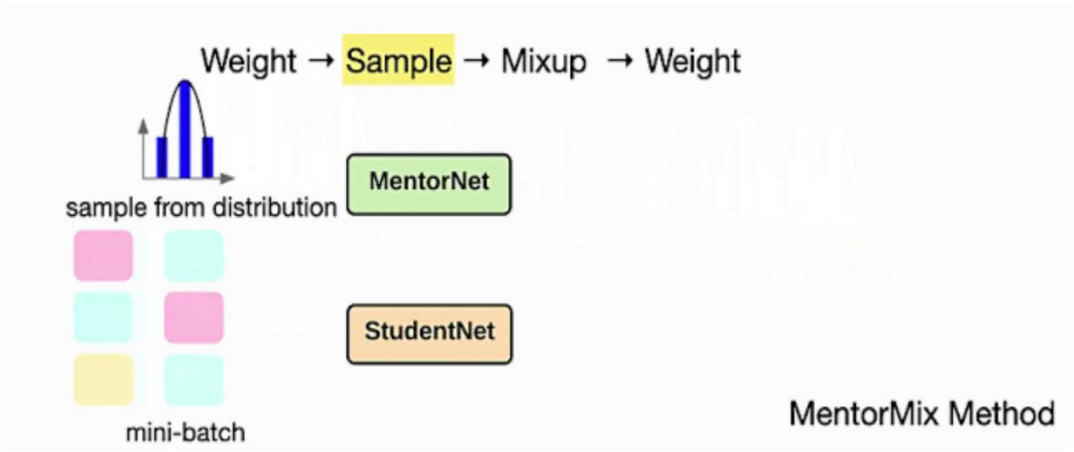


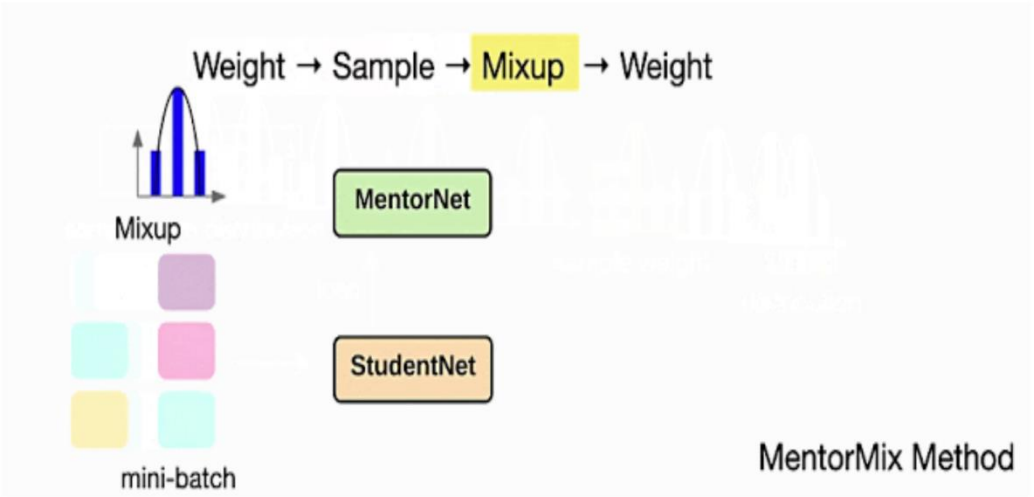
$$\tilde{\mathbf{x}}_{ij} = \lambda \mathbf{x}_i + (1-\lambda)\mathbf{x}_j$$
$$\tilde{\mathbf{y}}_{ij} = \lambda \mathbf{y}_i + (1-\lambda)\mathbf{y}_j$$

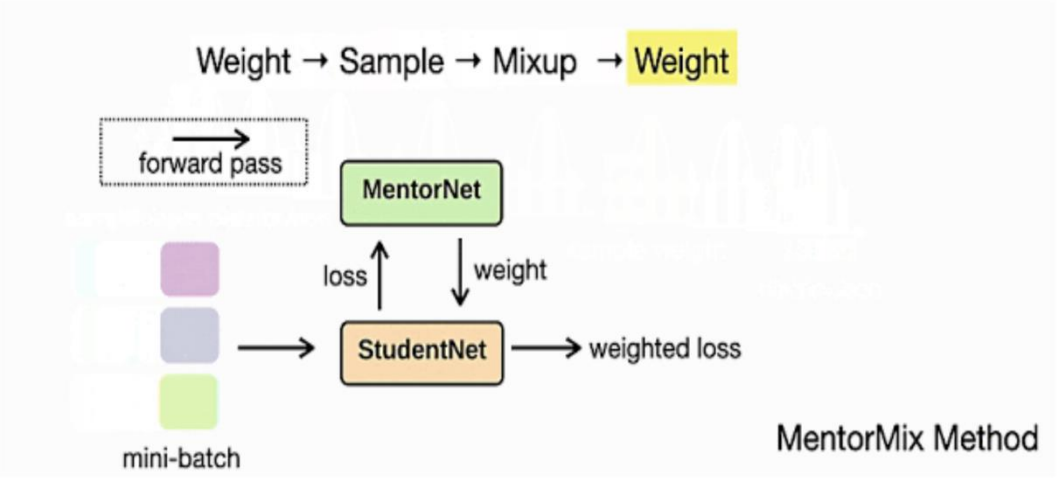
$$\boldsymbol{w}^* = \underset{\boldsymbol{w}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}\frac{1}{n}\sum_{j=1}^{n}\mathop{\mathbb{E}}_{\lambda}[\ell(g_s(\tilde{\mathbf{x}}_{ij};\boldsymbol{w}), \tilde{\mathbf{y}}_{ij})]$$

MentorMix


Weight → Sample → Mixup → Weight
forward pass
MentorNet
sample weight
distribution
loss
StudentNet
mini-batch
MentorMix Method


Weight → Sample → Mixup → Weight
sample from distribution
MentorNet
StudentNet
mini-batch
MentorMix Method


Weight → Sample → Mixup → Weight
Mixup
MentorNet
StudentNet
mini-batch
MentorMix Method


Weight → Sample → Mixup → Weight
forward pass
MentorNet
loss
weight
StudentNet
weighted loss
mini-batch
MentorMix Method

MentorMix

Algorithm 1 The proposed MentorMix method.

Input : mini-batch $\mathcal{D}_m$; two hyperparameters γ_p and α
Output : the loss of the mini-batch

1 For every $(\mathbf{x}_i, y_i)$ in $\mathcal{D}_m$ compute $\ell(\mathbf{x}_i, y_i)$
2 Set $\ell_p(\mathcal{D}_m)$ to be the γ_p-th percentile of the loss $\{\ell(\mathbf{x}_i, y_i)\}$.
3 $\gamma \leftarrow \text{EMA}(\ell_p(\mathcal{D}_m))$ `// update the moving average`
4 $v_i^* \leftarrow \text{MentorNet}(\ell(\mathbf{x}_i, y_i), \gamma)$ `// MentorNet weight`
5 Compute $P_{\mathbf{v}} = \text{softmax}(\mathbf{v}^*)$, where $\mathbf{v}^* = [v_1^*, \cdots, v_{|\mathcal{D}_m|}^*]$
6 Stop gradient
7 **foreach** $(\mathbf{x}_i, \mathbf{y}_i)$ **do**
8 Draw a sample $(\mathbf{x}_j, \mathbf{y}_j)$ with replacement from $P_{\mathbf{v}}$
9 $\lambda \leftarrow Beta(\alpha, \alpha)$
10 $\lambda \leftarrow v_i^* \max(\lambda, 1-\lambda) + (1 - v_i^*) \min(\lambda, 1-\lambda)$
11 $\tilde{\mathbf{x}}_{ij} \leftarrow \lambda \mathbf{x}_i + (1-\lambda)\mathbf{x}_j$
12 $\tilde{\mathbf{y}}_{ij} \leftarrow \lambda \mathbf{y}_i + (1-\lambda)\mathbf{y}_j$
13 Compute $\ell_i = \ell(\tilde{\mathbf{x}}_{ij}, \tilde{\mathbf{y}}_{ij})$
14 Weight ℓ_i using a separate MentorNet `// optional`
15 **end**
16 **return** $(1/|\mathcal{D}_m|) \sum_{i=1}^{|\mathcal{D}_m|} \ell_i$

- step2-4:weight
- step5-8:sample
- step9-12:mixup
- step14 :weight

Experiment

Setting

- training dataset split into 10 noise levels (0%->80%),clean validation dataset
- blue label noise(synthetic) red label noise(web)
- training from scratch and fine-tuning

Comparision with baseline

Table 2. Peak accuracy (%) of the best trial of each method averaged across 10 noise levels. – denotes the method failed to train.

Method	Mini-ImageNet				Stanford Cars			
	Fine-tuned		Trained from scratch		Fine-tuned		Trained from scratch	
	Blue	Red	Blue	Red	Blue	Red	Blue	Red
Vanilla	82.3±1.9	81.6±1.9	58.3±10.3	64.9±5.2	70.0±16.8	82.4±6.9	53.8±24.4	77.7±10.4
WeightDecay	81.9±1.8	81.5±1.8	—	—	72.2±17.5	84.3±6.6	—	—
Dropout	82.8±1.3	81.8±1.8	59.3±9.5	65.7±5.0	71.7±16.9	83.8±6.6	62.8±23.5	84.1±6.7
S-Model	82.3±1.8	82.0±1.9	58.7±10.2	64.6±5.1	69.7±16.8	82.4±7.1	53.9±23.5	77.6±10.2
Bootstrap	83.1±1.6	82.7±1.8	60.1±9.7	65.5±4.9	71.7±16.9	82.8±6.7	55.6±23.9	78.9±9.6
Mixup	81.7±1.8	82.4±1.7	60.7±9.8	66.0±4.9	73.1±16.6	85.0±6.2	64.2±21.6	82.5±8.0
MentorNet	82.9±1.7	82.4±1.7	61.8±10.3	65.1±5.0	75.9±16.8	82.6±6.6	56.8±23.1	78.9±8.9
Our MentorMix	**84.2±0.7**	**83.3±1.9**	**70.9±3.4**	**67.0±5.0**	**78.2±16.2**	**86.9±5.5**	**67.7±23.0**	**83.6±7.5**

Experiment

Comparision with SOTA

Table 3. Comparison with the state-of-the-art in terms of the validation accuracy on CIFAR-100 (top) and CIFAR-10 (bottom).

Data	Method	Noise level (%) 20	40	60	80
CIFAR100	Arazo et al. (2019)	73.7	70.1	59.5	39.5
	Zhang & Sabuncu (2018)	67.6	62.6	54.0	29.6
	MentorNet (2018)	73.5	68.5	61.2	35.5
	Mixup (2018)	73.9	66.8	58.8	40.1
	Huang et al. (2019)	74.1	69.2	39.4	-
	Ours (MentorMix)	**78.6**	**71.3**	**64.6**	**41.2**
CIFAR10	Arazo et al. (2019)	94.0	92.8	90.3	74.1
	Zhang & Sabuncu (2018)	89.7	87.6	82.7	67.9
	Lee et al. (2019)	87.1	81.8	75.4	-
	Chen et al. (2019)	89.7	-	-	52.3
	Huang et al. (2019)	92.6	90.3	43.4	-
	MentorNet (2018)	92.0	91.2	74.2	60.0
	Mixup (2018)	94.0	91.5	86.8	76.9
	Ours (MentorMix)†	**95.6**	**94.2**	**91.3**	**81.0**

Table 4. Comparison with the state-of-the-art on the clean validation set of ILSVRC12 and WebVision. The number outside (inside) the parentheses denotes the top-1 (top-5) classification accuracy(%). † marks the method trained using extra clean labels.

Data	Method	ILSVRC12	WebVision
Full	Lee et al. (2018)†	61.0(82.0)	69.1(86.7)
Full	Vanilla	61.7(82.4)	70.9(88.0)
Full	MentorNet (2018)†	64.2(84.8)	72.6(88.9)
Full	Guo et al. (2018)†	64.8(84.9)	72.1(89.2)
Full	Saxena et al. (2019)	—	67.5(—)
Full	Ours (MentorMix)	**67.5(87.2)**	**74.3(90.5)**
Mini	MentorNet (2018)	63.8(85.8)	—
Mini	Chen et al. (2019)	61.6(85.0)	65.2(85.3)
Mini	Ours (MentorMix)	**72.9(91.1)**	**76.0(90.2)**

Understanding

DNNs generalize much better on red label noise

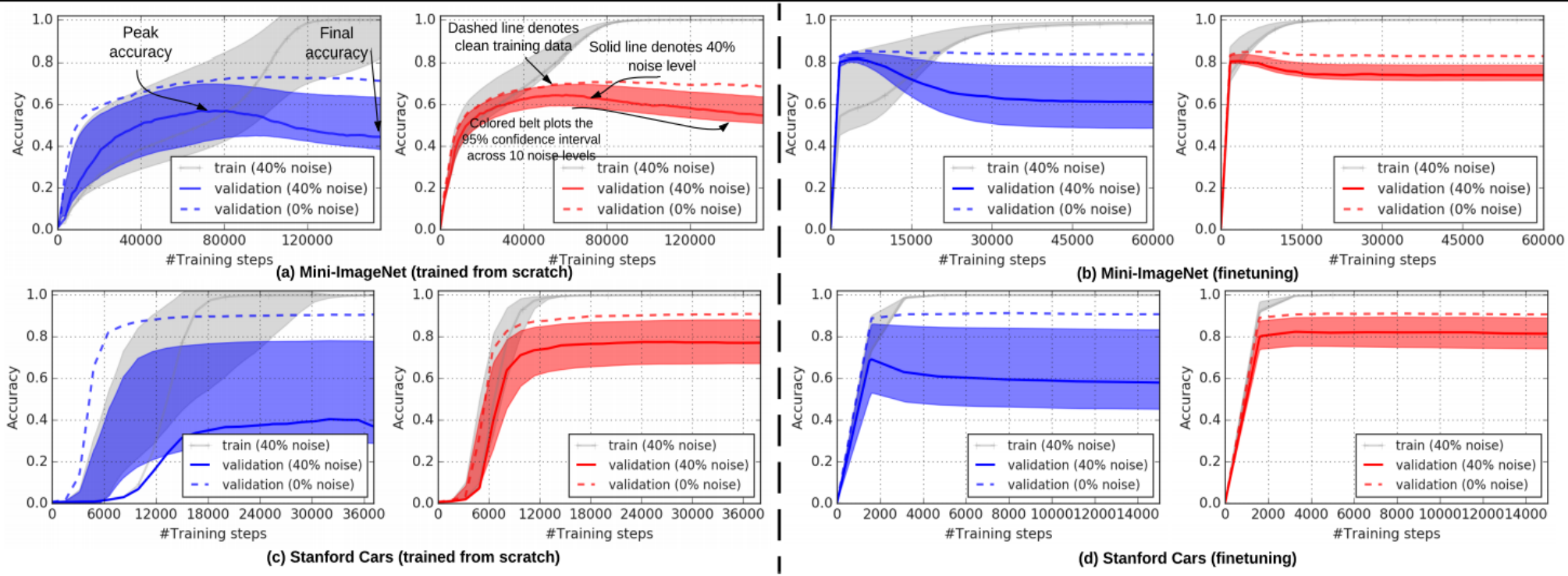
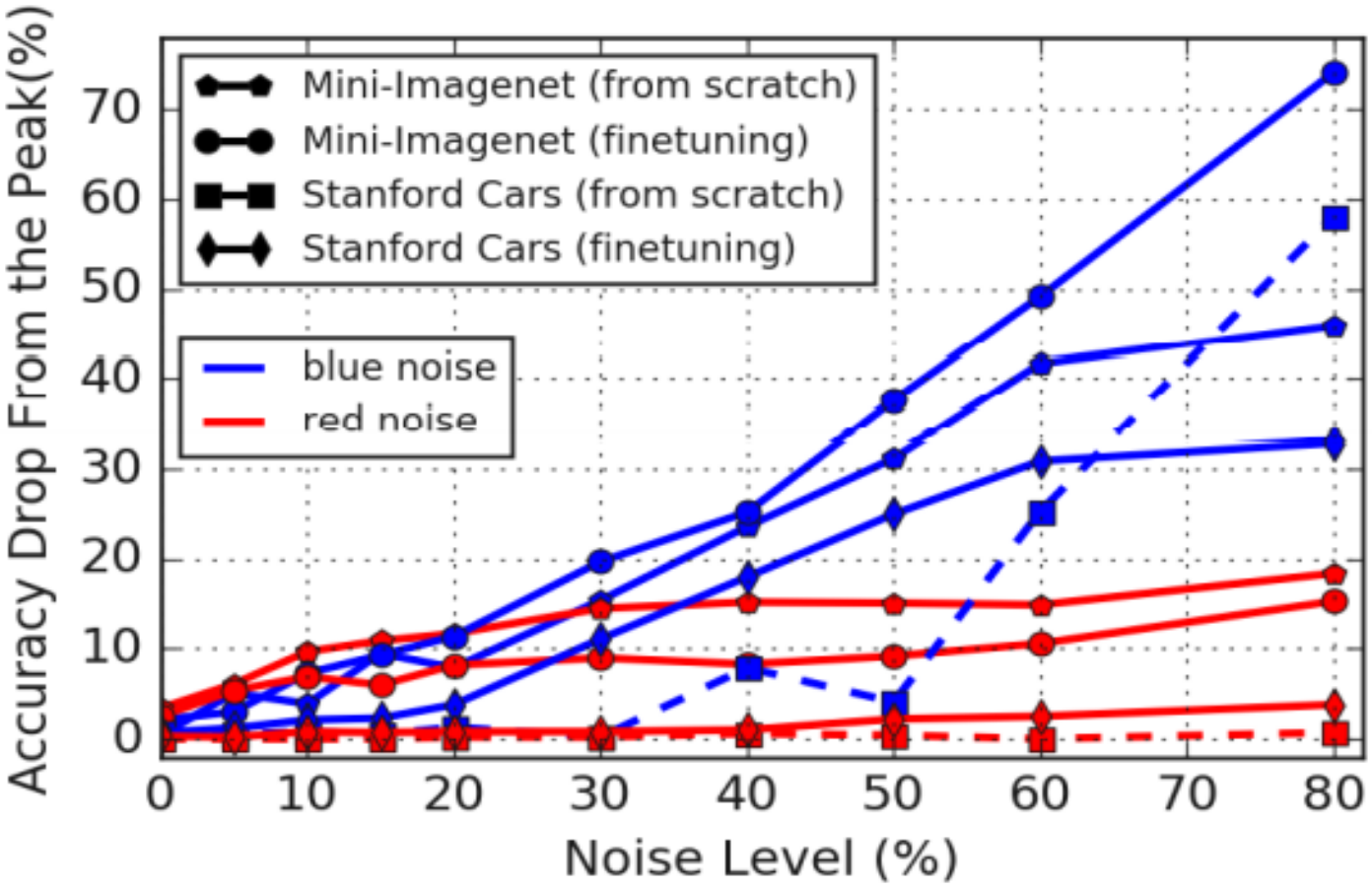


Figure 2. DNNs trained on synthetic (blue) and web label noise (red) on Mini-ImageNet (top) and Stanford Cars (bottom).

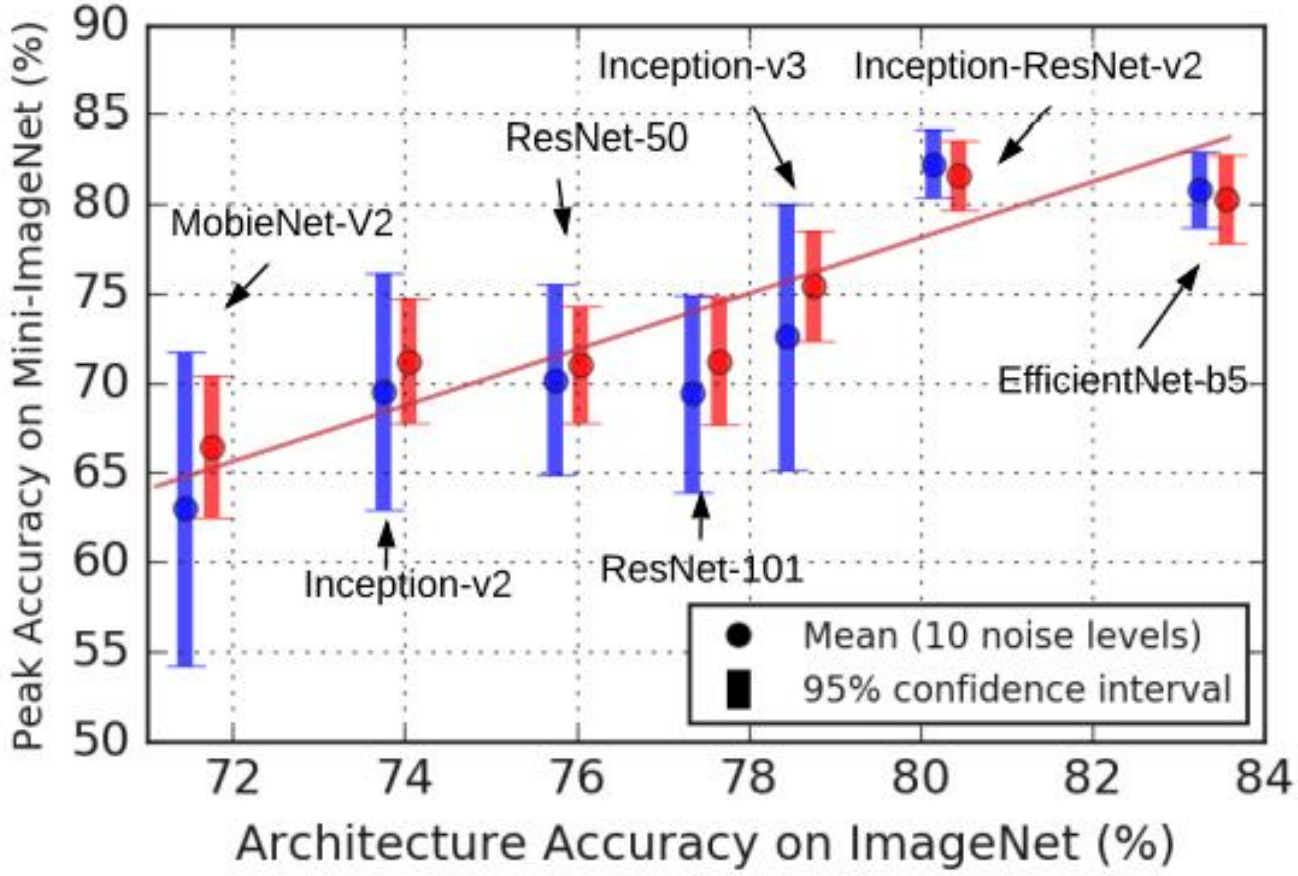
Understanding

DNNs may not learn patterns first on red label noise



Understanding

ImageNet architectures generalize on noisy labels when the networks are fine-tuned



(a) Mini-ImageNet ($r = 0.91$)

Thanks