Insight
Centre for Data Analytics

International Conference
on Machine Learning

Long Beach, June 2019

ICML

DCU

Unsupervised Label Noise Modeling and Loss Correction

Eric Arazo*, Diego Ortego*, Paul Albert, Noel O'Connor and Kevin McGuinness
eric.arazo@insight-centre.org, diego.ortego@insight-centre.org





Outline

- Motivation
- Observations
- Proposed method
  - Label noise modeling
  - Loss correction approach
- Results

Motivation: why label noise?

- Top performing DNN models: strong supervision
- Labeled data is a scarce resource
- Several alternatives to relax strong supervision

Motivation: why label noise?

- Top performing DNN models: strong supervision
- Labeled data is a scarce resource
- Several alternatives to relax strong supervision

Data

Semi-supervised learning

Unlabeled

Labeled

Motivation: why label noise?

- Top performing DNN models: strong supervision
- Labeled data is a scarce resource
- Several alternatives to relax strong supervision

Data

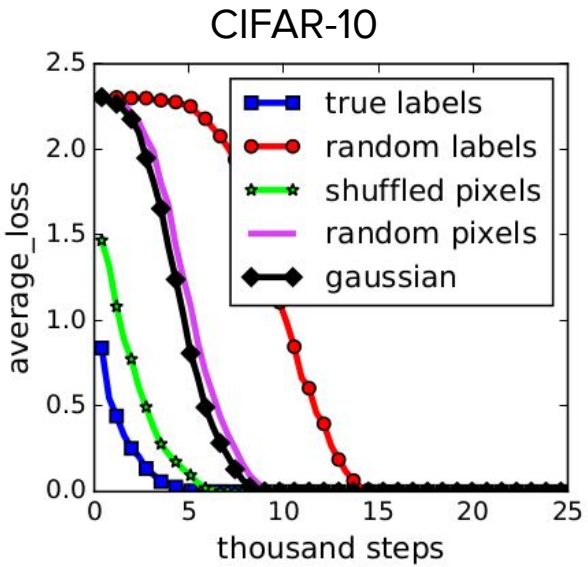
Automatic labeling (label noise)

Incorrectly labeled

Correctly Labeled

Observations

- “Deep neural networks easily fit random labels” [1]



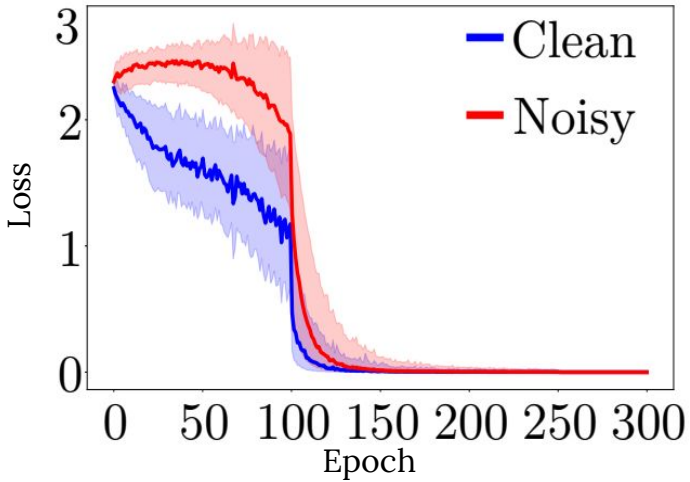
Source: [1]

[1] Zhang et al., “Understanding Deep Learning Requires Re-thinking Generalization”, ICLR 2017.

Observations

- Noisy samples take longer to learn
  - “Simple patterns are learned first” [2]
  - “Small loss” [3]
  - “High learning rate prevents memorization [4]”

CIFAR-10
80% label noise
Uniform label noise

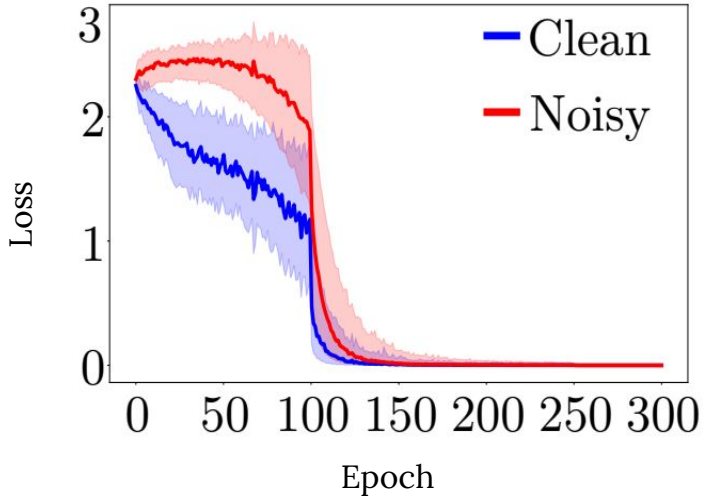


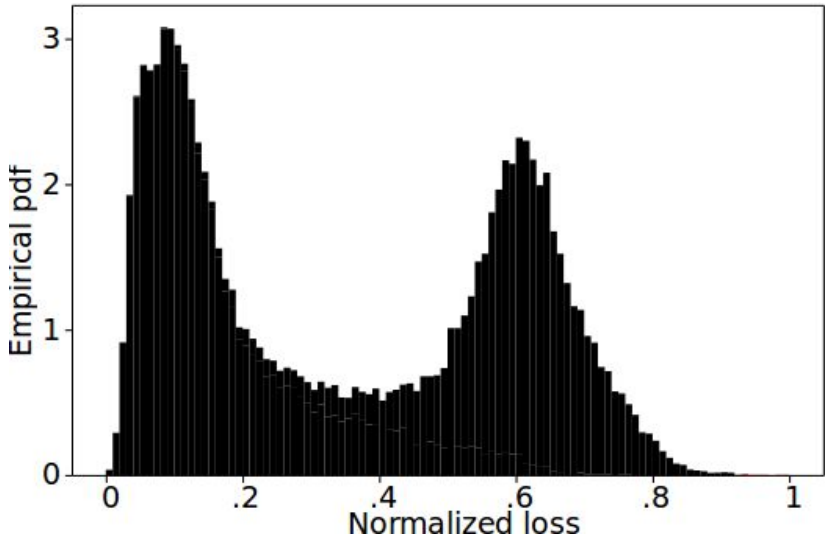
[2] Arpit et al., “A Closer Look at Memorization in Deep Networks”, ICML 2017.
[3] Yu et al., How does disagreement help against label corruption?, ICML 2019
[4] Tanaka et al., “Joint Optimization Framework for Learning with Noisy Labels”, CVPR 2018.

Label noise modeling

- Before label noise memorization: clean and noisy samples are (to some extent) distinguishable in the loss

- Two-component mixture model suits the problem

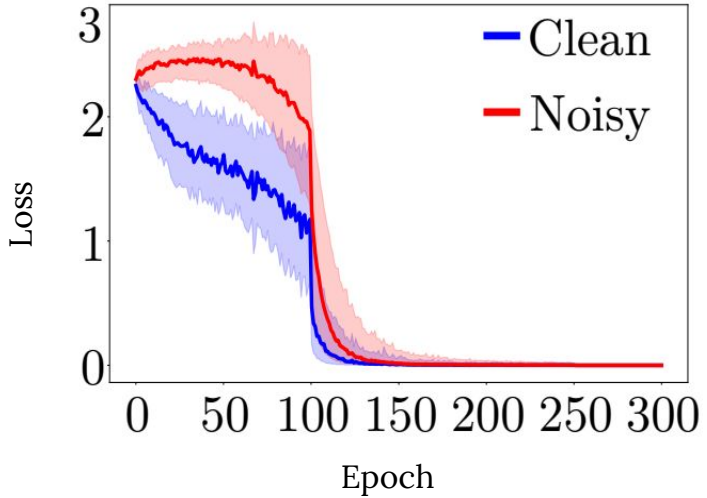


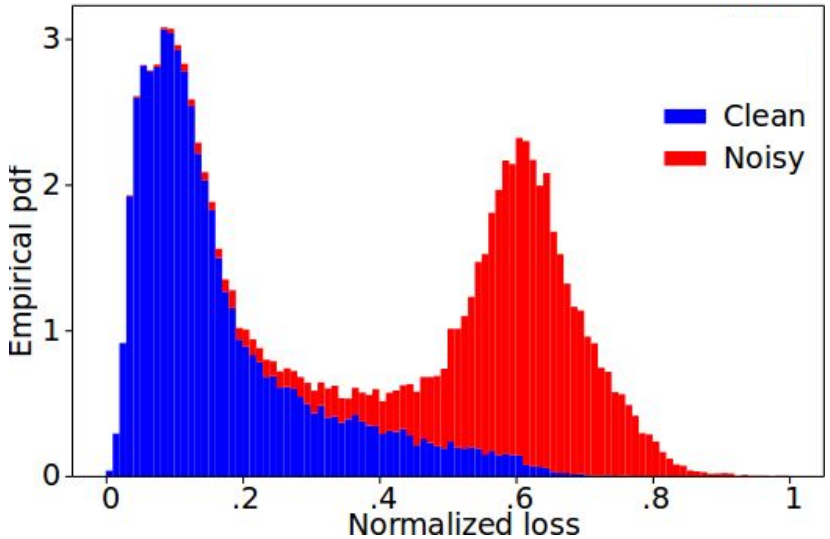


Label noise modeling

- Before label noise memorization: clean and noisy samples are (to some extent) distinguishable in the loss

- Two-component mixture model suits the problem

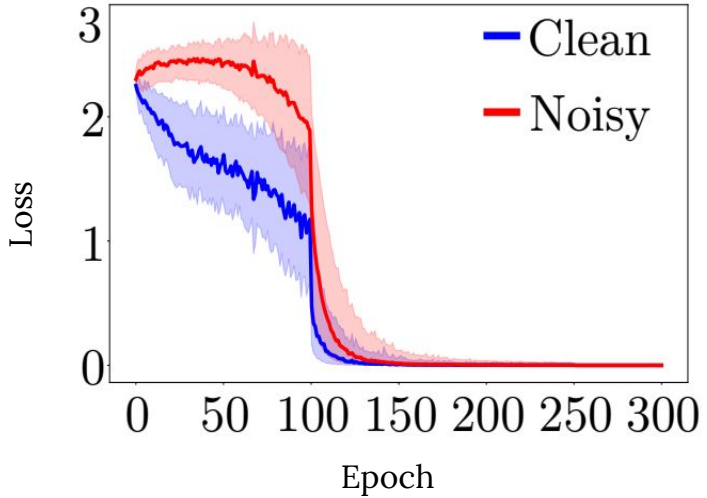


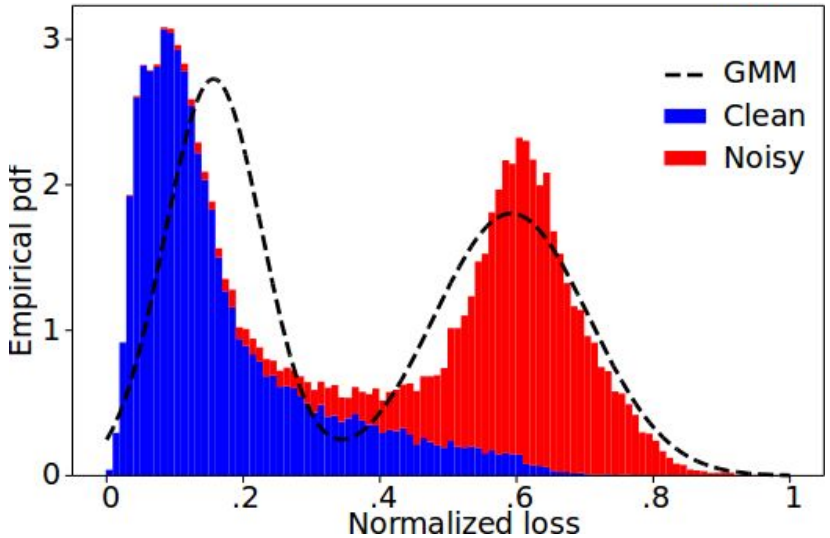


Label noise modeling

- Before label noise memorization: clean and noisy samples are (to some extent) distinguishable in the loss

- Two-component mixture model suits the problem

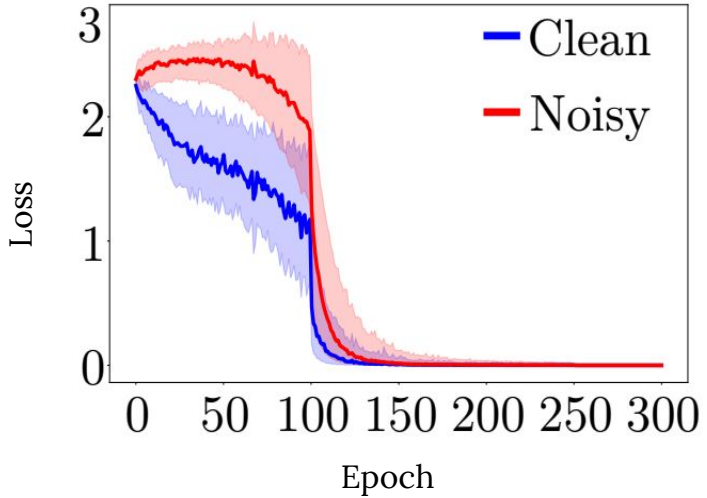


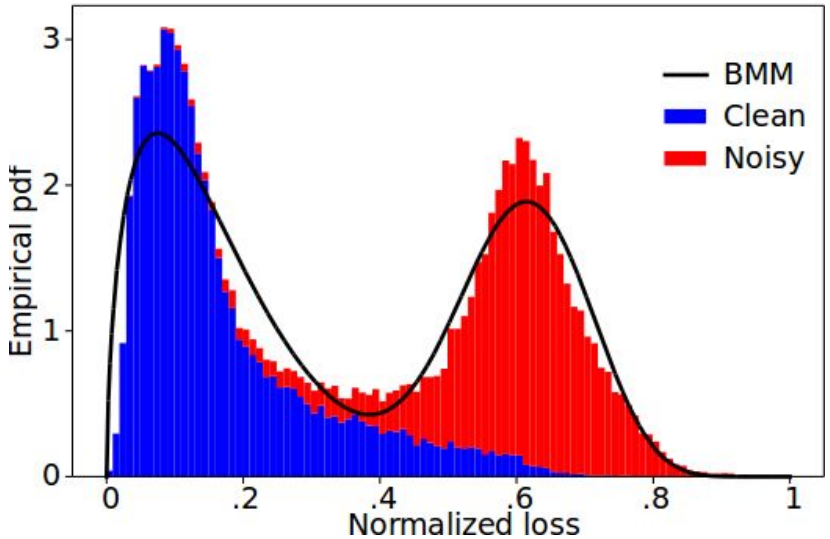


Label noise modeling

- Before label noise memorization: clean and noisy samples are (to some extent) distinguishable in the loss

- Two-component mixture model suits the problem





Loss correction approach

- Bootstrapping loss correction [5] + mixup data augmentation [6]

$$\ell^* = -\delta \left[\left(\left(1 - w_p\right) y_p + w_p z_p \right)^T \log\left(h\right) \right] - \left(1 - \delta\right) \left[\left(\left(1 - w_q\right) y_q + w_q z_q \right)^T \log\left(h\right) \right]$$

[5] Reed t al. “Training deep neural networks on noisy labels with bootstrapping”, ICLR 2015.
[6] Zhang et al., “mixup: Beyond Empirical Risk Minimization”, ICLR 2018.

Loss correction approach

- Bootstrapping loss correction [5] + mixup data augmentation [6]

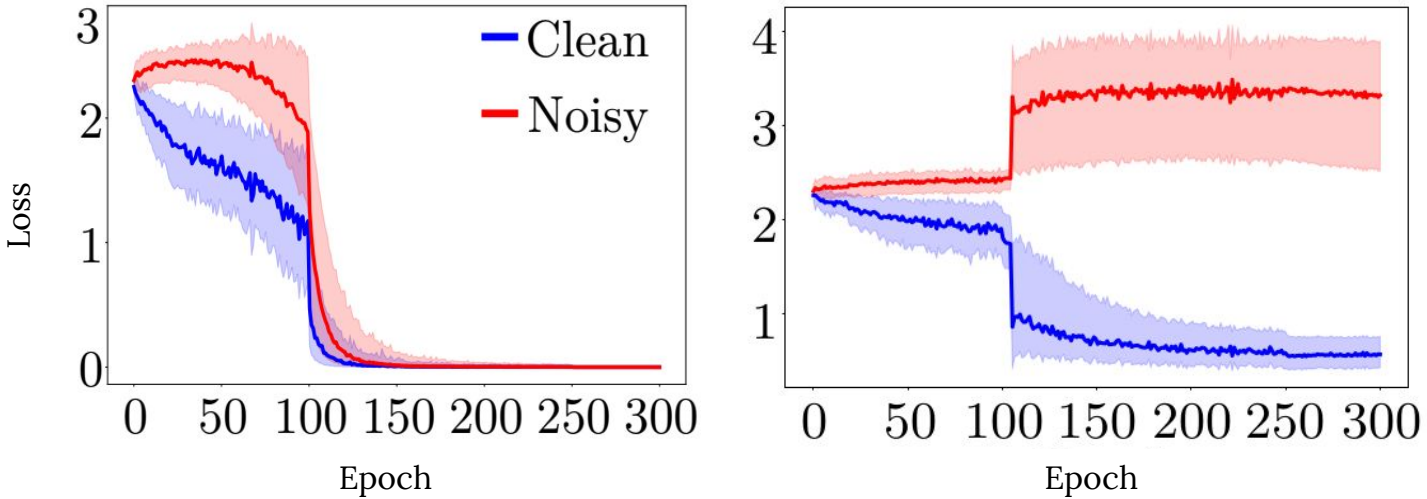
$$\ell^* = -\delta \left[\left(\left(1 - w_p\right) y_p + w_p z_p \right)^T \log\left(h\right) \right] - \left(1 - \delta\right) \left[\left(\left(1 - w_q\right) y_q + w_q z_q \right)^T \log\left(h\right) \right]$$

- Our Beta Mixture Model drives our learning approach a step further by:
  - **Preventing memorization**
  - **Correcting noisy labels to learn from them**

[5] Reed t al. “Training deep neural networks on noisy labels with bootstrapping”, ICLR 2015.
[6] Zhang et al., “mixup: Beyond Empirical Risk Minimization”, ICLR 2018.

Loss correction approach

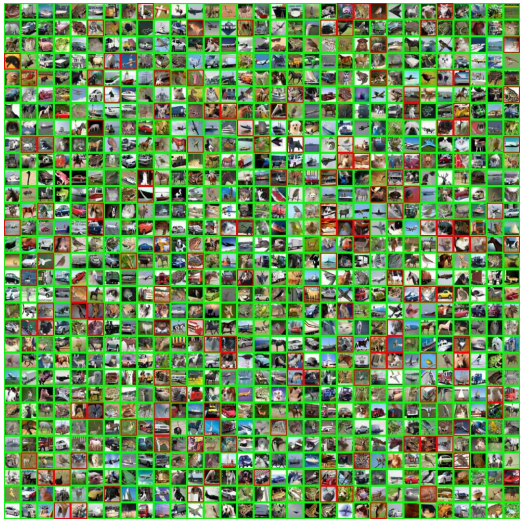
- Standard training (left) vs proposed training (right)



CIFAR-10, 80% label noise, uniform label noise

Loss correction approach

- Original labels training (left) vs predicted labels after training (right)



Results

CIFAR-10 results

Alg./Noise level (%)		0	20	50	80	90
(Reed et al., 2015)*	Best	94.7	86.8	79.8	63.3	42.9
	Last	94.6	82.9	58.4	26.8	17.0
(Patrini et al., 2017)*	Best	94.7	86.8	79.8	63.3	42.9
	Last	94.6	83.1	59.4	26.2	18.8
(Zhang et al., 2018)*	Best	**95.3**	**95.6**	87.1	71.6	52.2
	Last	**95.2**	92.3	77.6	46.7	43.9
M-DYR-H	Best	93.6	94.0	**92.0**	**86.8**	40.8
	Last	93.4	**93.8**	**91.9**	**86.6**	9.9
MD-DYR-SH	Best	93.6	93.8	90.6	82.4	**69.1**
	Last	92.7	93.6	90.3	77.8	**68.7**

Algorithm	Architecture	Noise level (%)			
		20	40	60	80
(Jiang et al., 2018b)	WRN-101	92.0	89.0	-	49.0
(Ma et al., 2018)	GCNN-12	85.1	83.4	72.8	-
(Ren et al., 2018)	WRN-28	-	86.9	-	-
(Wang et al., 2018b)	GCNN-7	81.4	78.2	-	-
M-DYR-H	PRN-18	**94.0**	**92.8**	**90.3**	46.3
MD-DYR-SH	PRN-18	93.8	92.3	86.1	**74.1**

Code on github: https://git.io/fjsvE

For more details and discussions...

Come to our poster!
(Pacific Ballroom #176)

Thanks!