Cyclic Annealing Training (CAT) CNNs for Image Classification with Noisy Labels

JiaWei Li, Tao Dai, QingTao Tang, YeLi Xing, Shu-Tao Xia
Tsinghua University

li-jw15@mails.tsinghua.edu.cn

October 8, 2018

Noisy Labels
- Noisy Labels Problem
- Noise Modeling with EM

Speedup the training in M-cycle
- Cyclic Annealing Training

Aggregate M-cycle CNNs at test time
- Bagging CNNs

Algorithm Description
- CAT on Noisy Labels

Experiments
- Performance on MNIST
- Robustness on CIFAR

Noisy Labels Problem:

- Labeling image dataset is a cubersome work and easily induce noise. It has a large impact on learning.

Figure 1: left-middle[1]-right might be labeled as **dog**, **seal**, and **seal**.

[1] Copyright: `http://www.dianliwenmi.com/postimg_3364775_6.html`

Noise patterns:

- Image **x** has a noisy label **z**, its true label **y** is unknown.

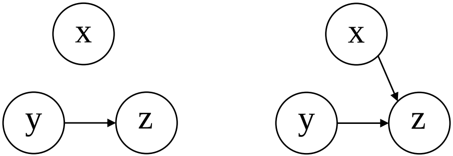
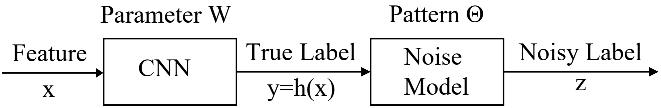


Figure 2: Two different noise patterns.

- Left: noisy label **z** only depends on true label **y**.
- Right: **z** depends on both of true label **y** and feature **x**.

Noise Modeling:



$$L(W,\theta) = \sum_{t=1}^{n} log(\sum_{i=1}^{k} p(z_t|y_t = i;\theta)p(y_t = i|x_t;W))$$

Figure 3: A typical label noise modeling procedure.

Learning with EM:

- E-step: fix **W** and update the noise modeling parameter θ.
- M-step: use **z**, y=h(x, w), and θ to train **W**.

Cyclic Annealing Training (CAT):

- It abruptly raises the learning rate α and then quickly decreases it with a cosine function:

$$\alpha(t) = \frac{\alpha_0}{2}(\cos(\frac{\pi mod(t-1, \lceil T/C \rceil)}{\lceil T/C \rceil}) + 1)$$

- Align every annealing learning rate cycle to every M-step.
- Then use the obtained local minimal CNN models to update the following E-step.
- Almost ***C-times faster*** than original EM approaches.

CAT vs. standard training schedule:

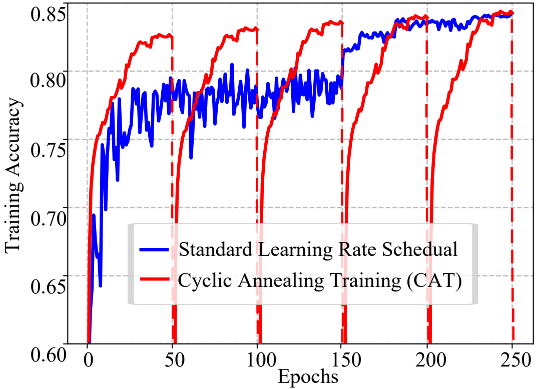


Figure 4: Training DenseNet-40 on CIFAR-10 with different schedule.

Aggregate M-cycle CNNs at test time:

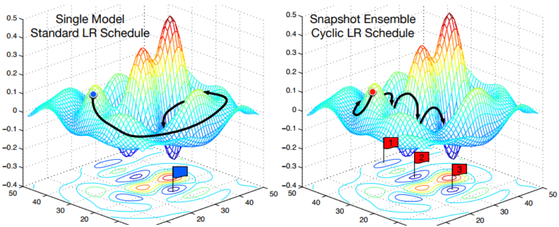


Figure 5: Using CAT for Snapshot Ensemble[1].

- Once the training finished, collect all local minimal CNNs.
- The aggregating output will be: $\hat{h}^{AVG}(x) = \frac{1}{C}\sum_{c=1}^{C}\hat{h}_c(x)$.

[1] ICLR 2017. Gao Huang, et al. Snapshot ensembles: Train 1, get m for free.

The log likelihood of model parameters are:

$$L(W, \theta) = \sum_{t=1}^{n} log(\sum_{i=1}^{k} p(z_t|y_t = i; \theta)p(y_t = i|x_t; W)).$$

1: Given n samples training data $X(x_1, ..., x_n)$ with noisy label $Z(z_1, ..., z_n)$, the true label $Y(y_1, ..., y_n)$ are unknown. The transfer probability between true label and noisy label is denoted as $\Theta(\theta_{ij} = p(z = j|y = i))$.
2: We first generate a random matrix $\hat{\Theta}_0$ to be the initialization of the noise pattern.
3: Then we repeatedly do C times the following:

(1) For every training cycle c ranges from 1 to C, initiate the learning rate with a constant value α_0.

(2) With the learning rate annealing from α_0 to 0 as function $\alpha(t) = \frac{\alpha_0}{2}(cos(\frac{\pi \cdot mod(t-1,T)}{T}) + 1)$, where t is current iteration number, train the CNN $p(y|x; W^c)$ with a fixed follow-up noise layer (linear or softmax) $p(z|y; \hat{\Theta}_{c-1})$ for total T iterations.

(3) Update the learned noise pattern $\hat{\Theta}_c$ with the closed-form function (3).

4: Once all of the training finished, drop the noise layer according to the final $\hat{\Theta}_C$. The remaining CNN parameters W^c will be used to predict the true labels, as $\hat{f}_c = p(y|x; W^c), c = 1, ..., C$.
5: For any prediction sample $\langle x_0, z_0 \rangle$ with a hidden true label y_0, the aggregating output $\hat{f}$ of bagging CNNs is the simple averaging $\hat{f}^{AVG}(x_0) = \frac{1}{C}\sum_{c=1}^{C} \hat{f}_c(x_0)$.
6: The prediction error is given by counting the proportion of prediction mistakes $\hat{f}(x_0) \neq y_0$ among the test dataset.

Algorithm 1: CAT on Noisy Labels.

Noise Setting on MNIST:

- We use the **label flipping operation** on MNIST dataset.

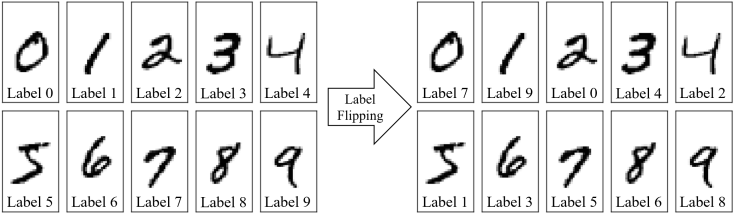


Figure 6: Label flipping with noise pattern [7,9,0,4,2,1,3,5,6,8].

Performance on MNIST:

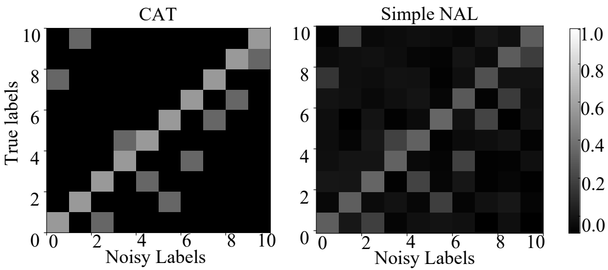


Figure 7: The acquired transfer probability $\hat{\theta}$ of CAT and Simple NAL.

- 46% noisy labels with noise pattern [7,9,0,4,2,1,3,5,6,8].
- The simple NAL has a 99.68% classification accuracy and CAT achieves 99.77%.

Noise Setting on CIFAR-100:

- **z** depends on both of true label **y** and feature **x**.

Figure 8: Randomly selected images from the noisy-label CIFAR.

Robustness on CIFAR-100:

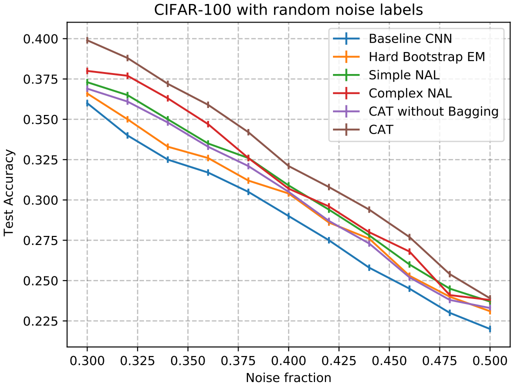


Figure 9: Compare the robustness of noise modeling methods.

Selected Reference:

1. TNNLS 2014. Classification in the Presence of Label Noise: a Survey.
2. ICLR 2015. Training convolutional networks with noisy labels.
3. ICLR 2015. Training deep neural networks on noisy labels with bootstrapping.
4. ICASSP 2016. Training deep neural-networks based on unreliable labels.
5. ICLR 2017. Snapshot ensembles: Train 1, get m for free.
6. ICLR 2017. Training DNNs Using a Noise Adaptation Layer.

Some New Progress:

1. JMLR 2018. A theory of learning with corrupted labels.
2. ICML 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels.
3. ICML 2018. Dimensionality-Driven Learning with Noisy Labels.
4. CVPR 2018. Iterative Learning with Open-set Noisy Labels.
5. ICLR 2019 submission. Pumpout: A Meta Approach for Robustly Training Deep Neural Networks with Noisy Labels.

Thanks for listening!