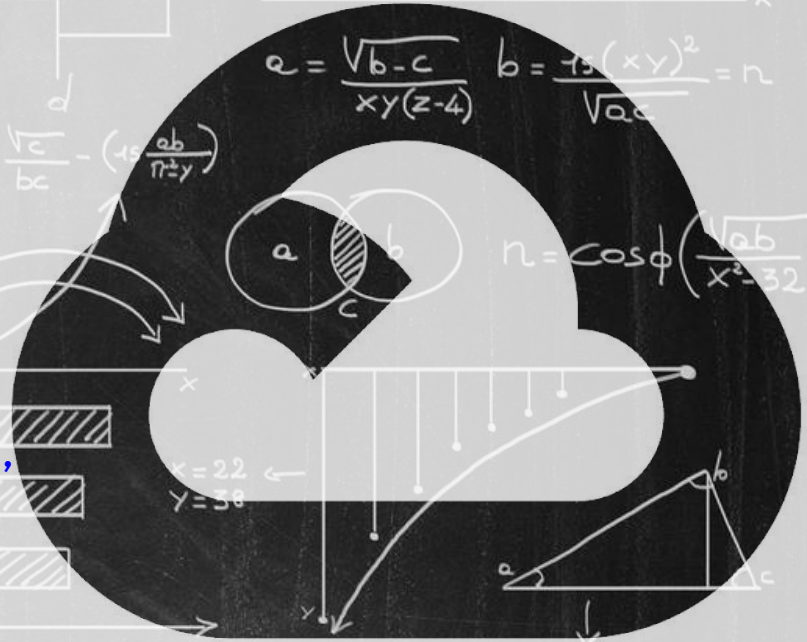
ICML | 2018

Thirty-fifth International Conference on Machine Learning

MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Jia Li, Fei-Fei Li.



Google AI

Google Cloud

Stanford University

Introduction

Deep Learning with Weak Supervision

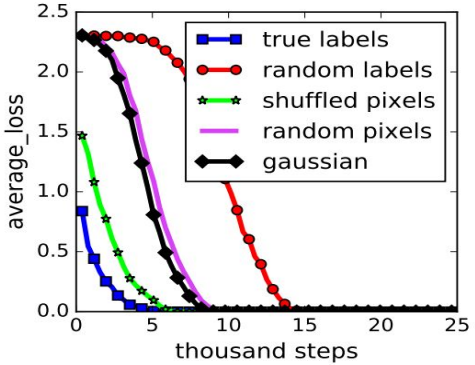
Problem: big data often come with noisy or corrupted labels.

YFCC100M

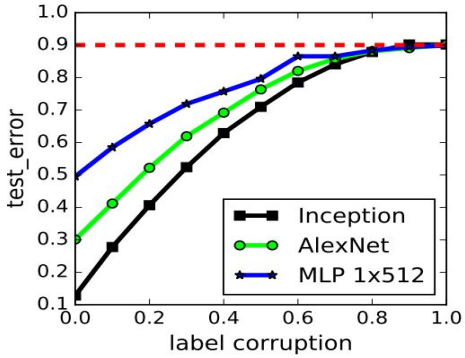
Deep Networks is able to memorize noisy labels

Zhang et al. (2017) showed that:

Deep networks are very good at overfitting (memorizing) the noisy labels.



(a) learning curves



(c) generalization error growth

Zhang, Chiyuan, et al. "Understanding deep learning requires rethinking generalization." *ICLR* (2017).

We study how to train very deep networks on noisy data to improve the generalization performance.

Very deep CNN (**ResNet-101** and **Inception-resnet-v2**)

- a few hundred layers
- tens of millions of model parameters
- takes days to train on 100+ GPUs

We study how to train **very deep networks** on noisy data to improve the generalization performance.

Inception Resnet V2 Network

Training labels are noisy or corrupted:

- incorrect labels (false positives)
- missing labels (false negatives)

We study how to train very deep networks on **noisy data** to improve the generalization performance.

Related Work

- A robust loss to model “prediction consistency” (Reed et al. 2014)
- Add a layer to model noise labels (Goldberger et al .2017)
- Regularization AIR (Azadi et al. 2016)
- Forgetting larger dropout rate (Kawaguchi et al. 2017)
- “Clean up” noisy labels using the clean labels (Veit. et al. 2017, Lee et al. 2017)
- Knowledge distillation (Li et al. 2017)
- Fidelity-weighting (Dehghani 2017)
- Robust Discriminative Neural Network (Vahdat 2017)
- … ..

Reed, Scott, et al. "Training deep neural networks on noisy labels with bootstrapping." arXiv preprint arXiv:1412.6596(2014).
S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus. Training convolutional networks with noisy labels. arXiv preprint arXiv:1406.2080, 2014
J. Goldberger and E. Ben-Reuven. Training deep neural networks using a noise adaptation layer. In ICLR, 2017.
S. Azadi, J. Feng, S. Jegelka, and T. Darrell. Auxiliary image regularization for deep cnns with noisy labels. ICLR, 2016.
A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, and S. Belongie. Learning from noisy large-scale datasets with minimal supervision. In CVPR, 2017.
Y. Li, J. Yang, Y. Song, L. Cao, J. Li, and J. Luo. Learning from noisy labels with distillation. In ICCV, 2017.
Lee, Kuang-Huei, et al. "CleanNet: Transfer Learning for Scalable Image Classifier Training with Label Noise." arXiv preprint arXiv:1711.07131 (2017).
Kawaguchi, Kenji, Leslie Pack Kaelbling, and Yoshua Bengio. "Generalization in deep learning." arXiv preprint arXiv:1710.05468 (2017).
Dehghani, Mostafa, et al. "Fidelity-Weighted Learning." arXiv preprint arXiv:1711.02799 (2017).
Vahdat, Arash. "Toward robustness against label noise in training deep discriminative neural networks." NIPS. 2017.

Related Work

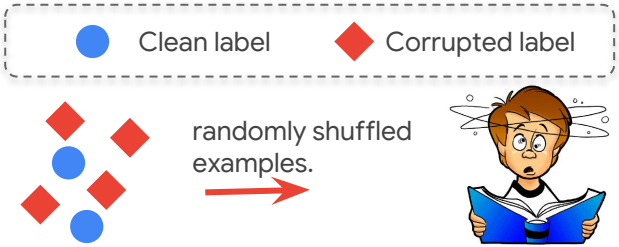
- A robust loss to model “prediction consistency” (Reed et al. 2014)
- Add a layer to model noise labels (Goldberger et al .2017)
- Regularization AIR (Azadi et al. 2016)
- Forgetting larger dropout rate (Kawaguchi et al. 2017)
- “Clean up” noisy labels using the clean labels (Veit. et al. 2017, Lee et al. 2017)
- Knowledge distillation (Li et al. 2017)
- Fidelity-weighting (Dehghani 2017)
- Robust Discriminative Neural Network (Vahdat 2017)
- … ..

We focus on training **very deep CNN from scratch.**

Reed, Scott, et al. "Training deep neural networks on noisy labels with bootstrapping." arXiv preprint arXiv:1412.6596(2014).
S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus. Training convolutional networks with noisy labels. arXiv preprint arXiv:1406.2080, 2014
J. Goldberger and E. Ben-Reuven. Training deep neuralnetworks using a noise adaptation layer. In ICLR, 2017.
S. Azadi, J. Feng, S. Jegelka, and T. Darrell. Auxiliary image regularization for deep cnns with noisy labels. ICLR, 2016.
A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, and S. Belongie. Learning from noisy large-scale datasets with minimal supervision. In CVPR, 2017.
Y. Li, J. Yang, Y. Song, L. Cao, J. Li, and J. Luo. Learning from noisy labels with distillation. In ICCV, 2017.
Lee, Kuang-Huei, et al. "CleanNet: Transfer Learning for Scalable Image Classifier Training with Label Noise." arXiv preprint arXiv:1711.07131 (2017).
Kawaguchi, Kenji, Leslie Pack Kaelbling, and Yoshua Bengio. "Generalization in deep learning." arXiv preprint arXiv:1710.05468 (2017).
Dehghani, Mostafa, et al. "Fidelity-Weighted Learning." arXiv preprint arXiv:1711.02799 (2017).
Vahdat, Arash. "Toward robustness against label noise in training deep discriminative neural networks." NIPS. 2017.

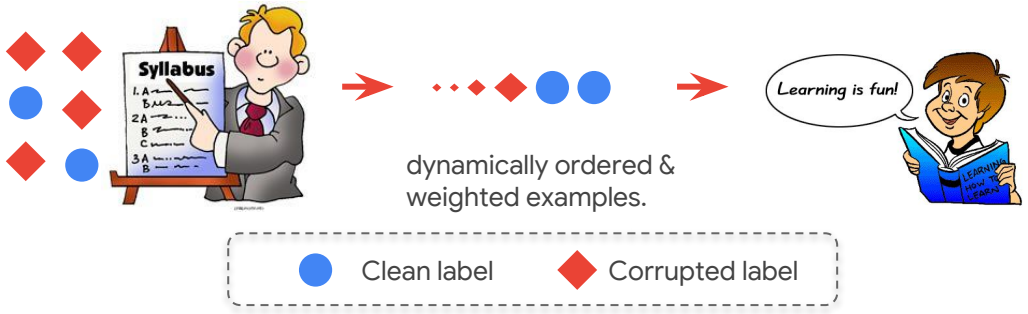
Backgrounds on Curriculum Learning

Traditional Learning



Curriculum Learning

Method: Curriculum learning (learning examples with focus).
Introduce a **teacher** to determine the weight and timing to learn every example.



Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In ICML, 2009.
M. P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In NIPS,2010.
Jiang, Lu, et al. "Self-Paced Curriculum Learning." AAAI. Vol. 2. No. 5.4. 2015.

Background (Curriculum Learning)

latent weight variable

Regularizer G

$$\min_{\mathbf{w}\in\mathbb{R}^d,\mathbf{v}\in[0,1]^{n\times m}} \mathbb{F}(\mathbf{w},\mathbf{v}) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{v}_i^T \mathbf{L}(\mathbf{y}_i, g_s(\mathbf{x}_i,\mathbf{w})) + G(\mathbf{v};\lambda) + \theta\|\mathbf{w}\|_2^2$$

Background (Curriculum Learning)

$$\min_{\mathbf{w}\in\mathbb{R}^d,\mathbf{v}\in[0,1]^{n\times m}} \mathbb{F}(\mathbf{w},\mathbf{v}) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{v}_i^T \mathbf{L}(\mathbf{y}_i, g_s(\mathbf{x}_i,\mathbf{w})) + G(\mathbf{v};\lambda) + \theta\|\mathbf{w}\|_2^2$$

An example: self-paced (Kumar et al 2010)

$$G(\mathbf{v}) = -\|\mathbf{v}\|_1 \rightarrow v_i^* = \begin{cases} 1 & \ell_i < \lambda \\ 0 & \ell_i \geq \lambda \end{cases},$$

regularizer

weighting scheme

Favor examples of smaller loss

$$\ell_i = L(\mathbf{y}_i, g_s(\mathbf{x}_i,\mathbf{w}))$$

$$\mathbf{v}_i \Rightarrow v_i$$

M. P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In NIPS,2010.

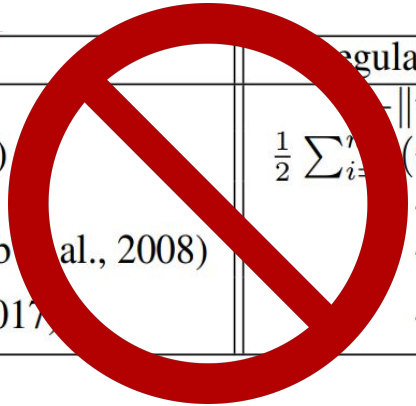
Existing studies define a curriculum as **a function**:

Strategy	Regularizer G	Weights v^*
Self paced (Kumar et al., 2010)	$-\|\mathbf{v}\|_1$	$v_i^* = \mathbb{1}(\ell_i \leq \lambda)$
Linear weighting (Jiang et al., 2015)	$\frac{1}{2}\sum_{i=1}^{n}(v_i^2 - 2v_i)$	$v_i^* = \max(0, 1 - \frac{1}{\lambda}\ell_i)$
Focal loss (Lin et al., 2017)	-	$v_i^* = [1 - \exp\{-\ell_i\}]^2$
Hard negative mining (Felzenszwalb et al., 2008)	-	$v_i^* = \mathbb{1}(\ell_i > \lambda(1 - y_i))$
Prediction variance (Chang et al., 2017)	-	$v_i^* = \frac{1}{Z}\sqrt{\mathrm{Var}(\ell_i) + \frac{\mathrm{Var}(\ell_i)}{\|\ell_i\|}}$

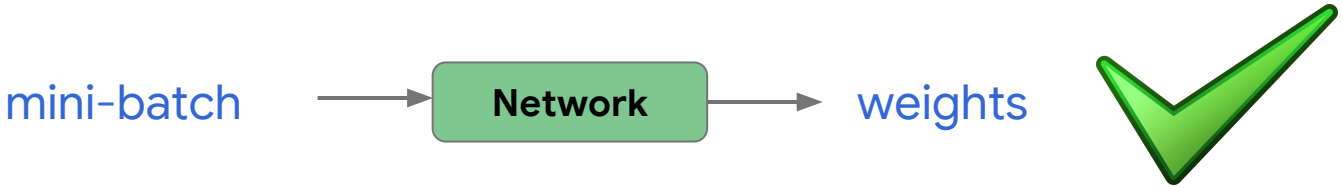
Regularizer

weighting scheme: how to compute the weight

Strategy	egularizer G	Weights v^*
Self paced (Kumar et al., 2010)	$-\|\mathbf{v}\|_1$	$v_i^* = \mathbb{1}(\ell_i \leq \lambda)$
Linear weighting (Jiang et al., 2015)	$\frac{1}{2}\sum_{i=}^{} (v_i^2 - 2v_i)$	$v_i^* = \max(0, 1 - \frac{1}{\lambda}\ell_i)$
Focal loss (Lin et al., 2017)	-	$v_i^* = [1 - \exp\{-\ell_i\}]^2$
Hard negative mining (Felzenszwalb al., 2008)	-	$v_i^* = \mathbb{1}(\ell_i > \lambda(1 - y_i))$
Prediction variance (Chang et al., 2017)	-	$v_i^* = \frac{1}{Z}\sqrt{\mathrm{Var}(\ell_i) + \frac{\mathrm{Var}(\ell_i)}{\|\ell_i\|}}$



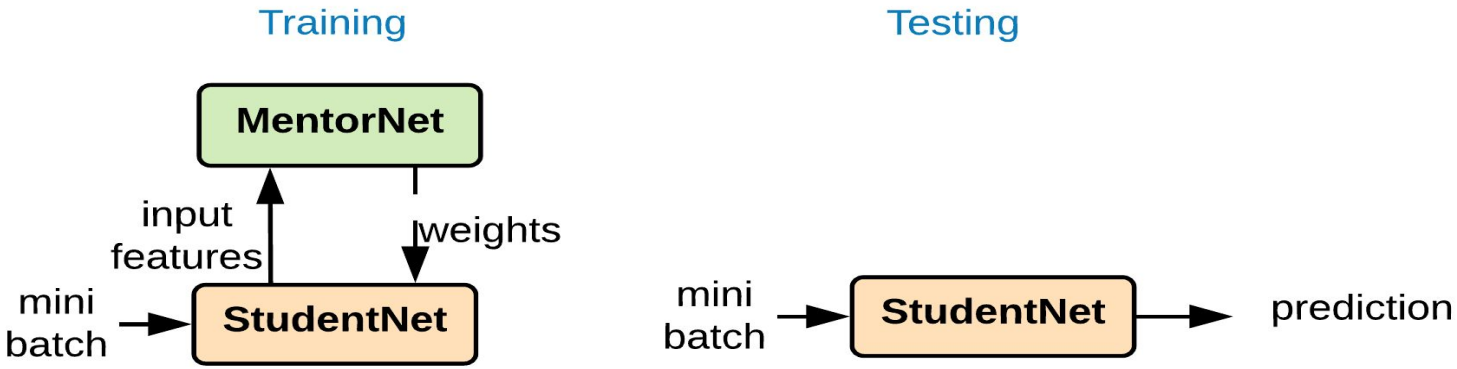
Learn a curriculum by a neural network from data?



Method

MentorNet

Each curriculum is implemented as a network (called MentorNet)

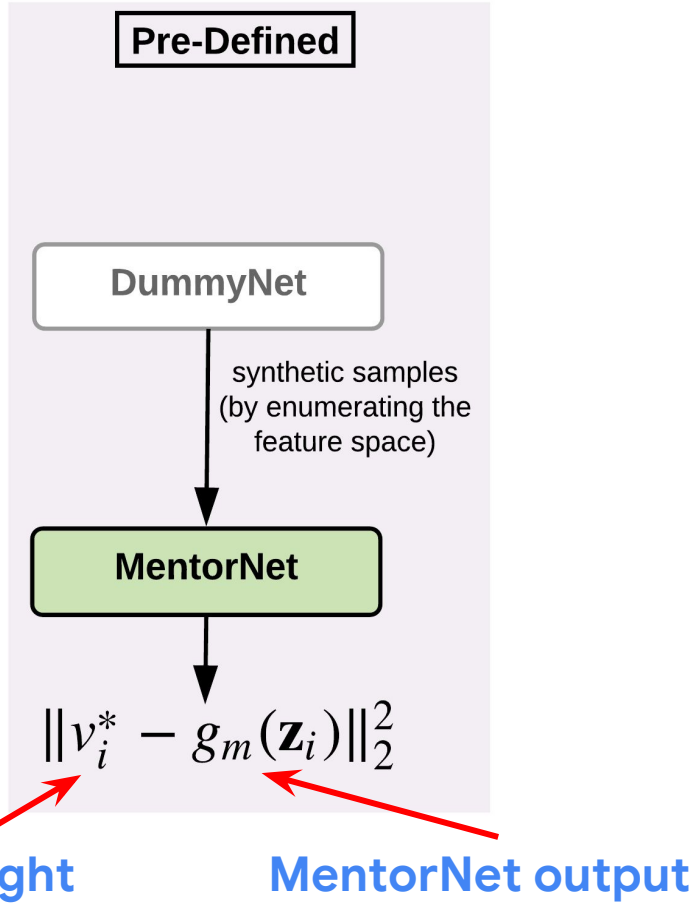


Learning the MentorNet

Two ways to learn a MentorNet

- **Pre-Defined**: approximate existing curriculums.

Strategy	Regularizer G	Weights v^*
Self paced (Kumar et al., 2010)	$-\|\mathbf{v}\|_1$	$v_i^* = \mathbb{1}(\ell_i \leq \lambda)$
Linear weighting (Jiang et al., 2015)	$\frac{1}{2}\sum_{i=1}^{n}(v_i^2 - 2v_i)$	$v_i^* = \max(0, 1 - \frac{1}{\lambda}\ell_i)$
Focal loss (Lin et al., 2017)	-	$v_i^* = [1 - \exp\{-\ell_i\}]^2$
Hard negative mining (Felzenszwalb et al., 2008)	-	$v_i^* = \mathbb{1}(\ell_i > \lambda(1 - y_i))$
Prediction variance (Chang et al., 2017)	-	$v_i^* = \frac{1}{Z}\sqrt{\mathrm{Var}(\ell_i) + \frac{\mathrm{Var}(\ell_i)}{\|\ell_i\|}}$

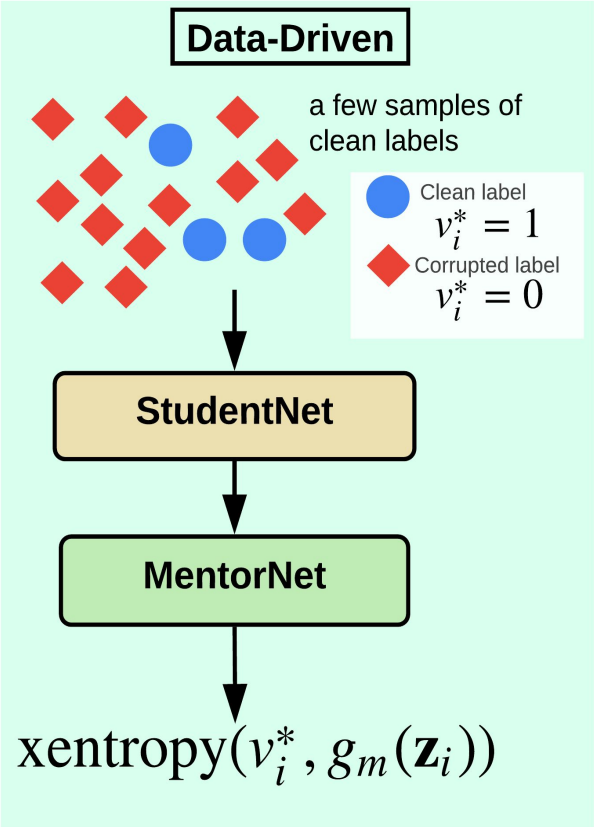


Learning the MentorNet

Two ways to learn a MentorNet

- **Data-Driven**: learn new curriculums from a small set (~10%) with clean labels.

Learn the MentorNet of CIFAR-10 and use it on CIFAR-100.



Algorithm (Training MentorNet with StudentNet)

Existing curriculum/self-paced learning is optimized by alternating batch GD.

We propose a **mini-batch SGD** and show good property on **model convergence** under standard assumptions.

Theorem 1. *Let the objective* $\mathbb{F}(\mathbf{w}, \mathbf{v})$ *defined in Eq.* (1) *be differentiable,* $L(\cdot)$ *be Lipschitz continuous in* $\mathbf{w}$ *and* $\nabla_{\mathbf{v}} G(\cdot)$ *be Lipschitz continuous in* $\mathbf{v}$. *Let* $\mathbf{w}^t, \mathbf{v}^t$ *be iterates from Algorithm 1 and* $\sum_{t=0}^{\infty} \alpha_t = \infty, \sum_{t=0}^{\infty} \alpha_t^2 < \infty$. *Then,* $\lim_{t\to\infty} \mathbb{E}[\|\nabla_{\mathbf{w}} \mathbb{F}(\mathbf{w}^t, \mathbf{v}^t)\|_2^2] = 0$.

Algorithm 1 SPADE for minimizing Eq. (1)

Input : Dataset $\mathcal{D}$, a predefined G or a learned $g_m(\cdot; \Theta)$
Output : The model parameter $\mathbf{w}$ of StudentNet.

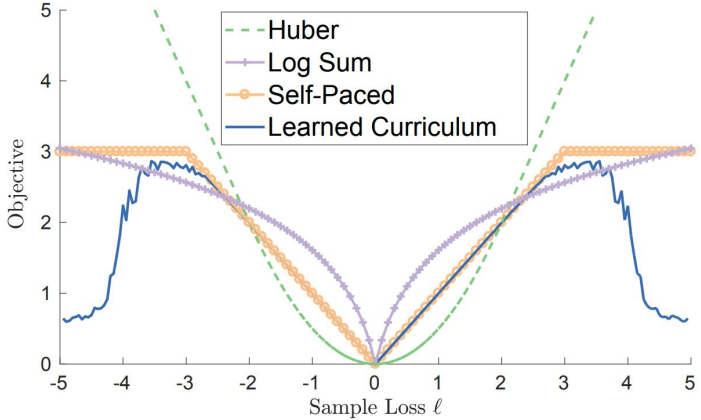
Initialize $\mathbf{w}^0, \mathbf{v}^0, t = 0$
while *Not Converged* **do**
  Fetch a mini-batch Ξ_t uniformly at random
  For every $(\mathbf{x}_i, y_i)$ in Ξ_t compute $\phi(\mathbf{x}_i, y_i, \mathbf{w}^t)$
  if *update curriculum* **then**
    $\Theta \leftarrow \Theta^*$, where Θ^* is learned in Sec. 3.1
  end
  if G *is used* **then**
    $\mathbf{v}_{\Xi}^t \leftarrow \mathbf{v}_{\Xi}^{t-1} - \alpha_t \nabla_{\mathbf{v}} \mathbb{F}(\mathbf{w}^{t-1}, \mathbf{v}^{t-1})|_{\Xi_t}$
  end
  else $\mathbf{v}_{\Xi}^t \leftarrow g_m(\phi(\Xi_t, \mathbf{w}^{t-1}); \Theta)$;
  $\mathbf{w}^t \leftarrow \mathbf{w}^{t-1} - \alpha_t \nabla_{\mathbf{w}} \mathbb{F}(\mathbf{w}^{t-1}, \mathbf{v}^t)|_{\Xi_t}$
  $t \leftarrow t + 1$
end
return $\mathbf{w}^t$

Why does it work?

We can show that MentorNet plays a role similar to the **robust M-estimator**:

- Huber (Huber et al., 1964)
- The log-sum penalty (Candes et al., 2008).

The robust objective function is **beneficial** for noisy labels.



Proposition 1. *Suppose* $(\mathbf{x}, y)$ *denotes a training sample and its corrupted label. For simplicity, let the MentorNet input* $\phi(\mathbf{x}, y, \mathbf{w}) = \ell$ *be the loss computed by the StudentNet model parameter* $\mathbf{w}$*. The MentorNet* $g_m(\ell; \Theta) = v$*, where* v *is the sample weight. If* g_m *decreases with respect to* ℓ*, then there exists an underlying robust objective* F*:*

$$F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \rho(\ell_i),$$

where $\rho(\ell_i) = \int_0^{\ell_i} g_m(x; \Theta)dx$*. In the special cases,* $\rho(\ell)$ *degenerates to the robust M-estimator: Huber (Huber et al., 1964) and the log-sum penalty (Candes et al., 2008).*

Experiments

Setups

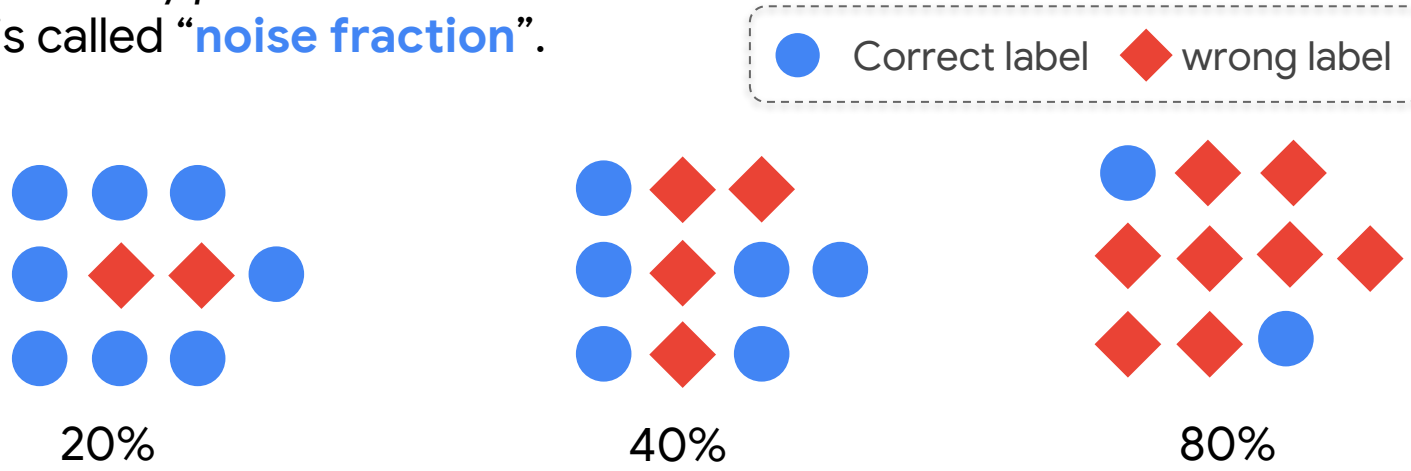
Table 1. StudentNet and their accuracies on the clean training data.

Dataset	Model	#para	train acc	val acc
CIFAR10	inception	1.7M	0.83	0.81
	resnet101	84M	1.00	0.96
CIFAR100	inception	1.7M	0.64	0.49
	resnet101	84M	1.00	0.79
ImageNet	inception_resnet	59M	0.88	0.77

Controlled Corrupted Label

First, following (Zhang et al. 2017), we test on controlled corrupted labels:

- Change each label of an image dependently to a uniform random class with probability *p*.
- *p* is called "**noise fraction**".



Zhang, Chiyuan, et al. "Understanding deep learning requires rethinking generalization." *ICLR* (2017).

Results on CIFAR

Baseline comparisons on **CIFAR-10 & CIFAR-100**
(under 20%, 40%, and 80% noise fractions)

	Resnet-101 StudentNet						Inception StudentNet					
	CIFAR-100			CIFAR-10			CIFAR-100			CIFAR-10		
Method	0.2	0.4	0.8	0.2	0.4	0.8	0.2	0.4	0.8	0.2	0.4	0.8
FullModel	0.60	0.45	0.08	0.82	0.69	0.18	0.43	0.38	0.15	0.76	0.72	0.42
Forgetting	0.61	0.44	0.16	0.78	0.63	0.35	0.42	0.37	0.17	0.76	0.71	0.44
Self-paced	0.70	0.55	0.13	0.89	0.85	0.28	0.44	0.38	0.14	**0.80**	0.74	0.33
Focal Loss	0.59	0.44	0.09	0.79	0.65	0.28	0.43	0.38	0.15	0.77	0.74	0.40
Reed Soft	0.62	0.46	0.08	0.81	0.63	0.18	0.42	0.39	0.12	0.78	0.73	0.39
MentorNet PD	0.72	0.56	0.14	0.91	0.77	0.33	0.44	0.39	0.16	0.79	0.74	0.44
MentorNet DD	**0.73**	**0.68**	**0.35**	**0.92**	**0.89**	**0.49**	**0.46**	**0.41**	**0.20**	0.79	**0.76**	**0.49**

Significant improvement over baselines.
Data-**D**riven performs better than **P**re-**D**efined MentorNet.

Results on ImageNet

ImageNet under 40% noise fraction

Method	P@1	P@5
NoReg	0.538	0.770
NoReg+WeDecay	0.560	0.809
NoReg+Dropout	0.575	0.807
NoReg+DataAug	0.522	0.772
NoReg+MentorNet	0.590	0.814
FullModel	0.612	0.844
Forgetting(FullModel)	0.628	0.845
MentorNet(FullModel)	**0.651**	**0.859**

NoReg: Vanilla model with no regularization
FullModel: Added weight decay, dropout, and data augmentation.

Results on **WebVision**

2.4 million images of noisy labels crawled by Flickr/Google Search.
1,000 classes defined in ImageNet ILSVRC 2012.
Real-world noisy labels.

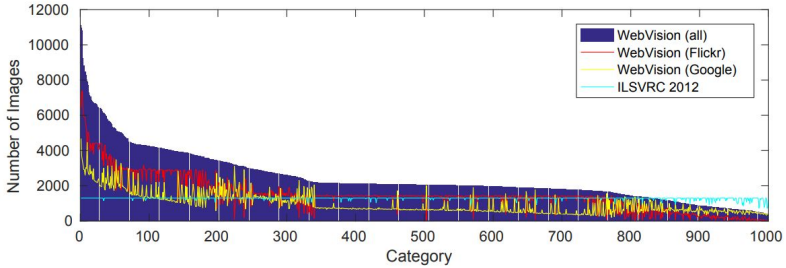


Figure 2. Number of images per category of the WebVison dataset.

Li, Wen, et al. "WebVision Database: Visual Learning and Understanding from Web Data." arXiv preprint arXiv:1708.02862 (2017).

Results on WebVision

Dataset	Method	ILSVRC12	WebVision
Entire	Li *et al.* (2017a)	0.476 (0.704)	0.570 (0.779)
Entire	Forgetting	0.590 (0.808)	0.666 (0.856)
Entire	Lee *et al.* (2017)*	0.602 (0.811)	0.685 (0.865)
Entire	MentorNet	0.625 (0.830)	0.708 (0.880)
Entire	**MentorNet***	**0.642 (0.848)**	**0.726 (0.889)**

The best-published result on the WebVision benchmark!

Substantiate that MentorNet is **beneficial** for training very deep networks on noisy data.

Li, Wen, et al. "WebVision Database: Visual Learning and Understanding from Web Data." arXiv preprint arXiv:1708.02862 (2017).
Lee, Kuang-Huei, et al. "CleanNet: Transfer Learning for Scalable Image Classifier Training with Label Noise." CVPR (2018).

Conclusions

Take-home messages

We discussed a novel way to train very deep networks on noisy data.

- The generalization performance can be improved by learning another network (called **MentorNet**) to supervise the training of the base network.
- Proposed an algorithm to perform curriculum learning and self-paced learning for deep networks via mini-batch SGD.

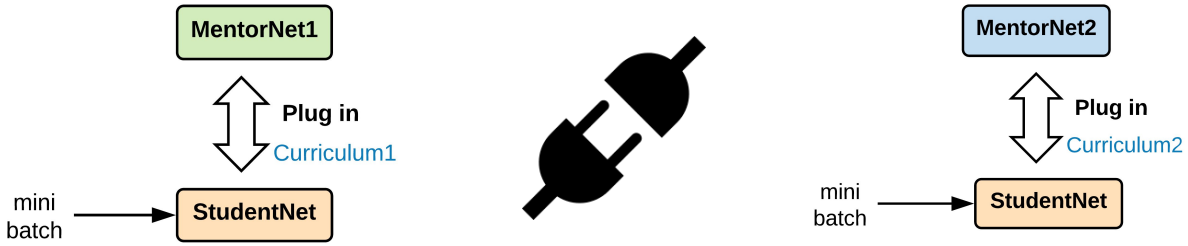
Code will be available soon on Google Github.

Please come to our Poster: **#113 (Hall B)**

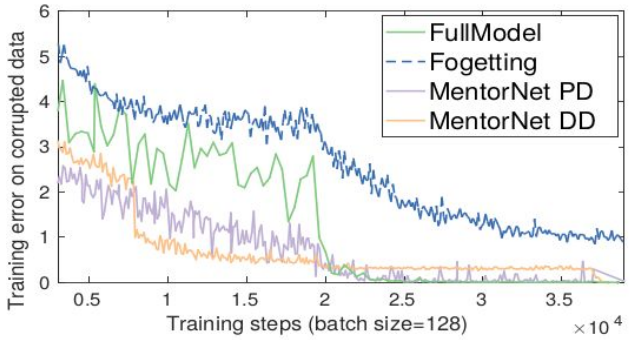
Flexible Framework for Curriculum Learning

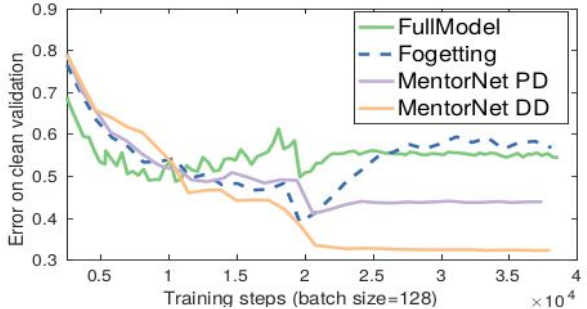
MentorNet is a general and flexible framework:

- Each curriculum is implemented as a network.
  - Approximate existing curriculums
  - Discover new curriculums from data
- Plug in/out a MentorNet to change the curriculum



Results



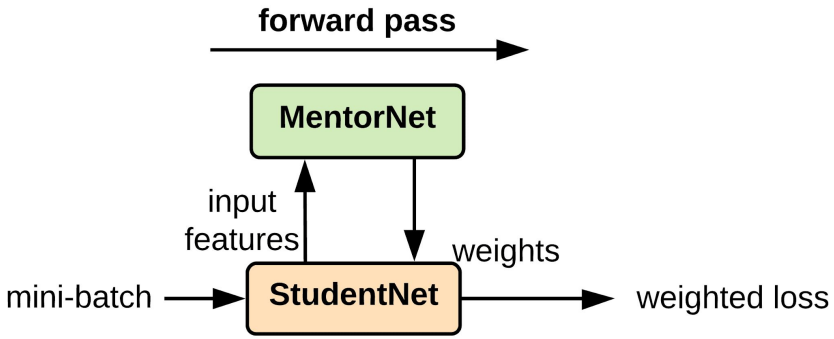


(a) Test error on clean validation

Our method overcomes the "memorization" problem of deep networks.

Algorithm (Training MentorNet with StudentNet)

Training MentorNet and StudentNet via mini-batch SGD.



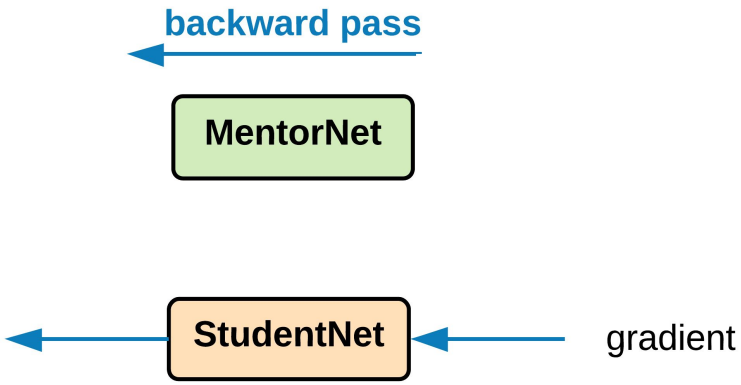
Algorithm 1 SPADE for minimizing Eq. (1)

Input : Dataset $\mathcal{D}$, a predefined G or a learned $g_m(\cdot;\Theta)$
Output : The model parameter $\mathbf{w}$ of StudentNet.

Initialize $\mathbf{w}^0, \mathbf{v}^0, t = 0$
while *Not Converged* **do**
  Fetch a mini-batch Ξ_t uniformly at random
  For every $(\mathbf{x}_i, y_i)$ in Ξ_t compute $\phi(\mathbf{x}_i, y_i, \mathbf{w}^t)$
  if *update curriculum* **then**
    $\Theta \leftarrow \Theta^*$, where Θ^* is learned in Sec. 3.1
  end
  if G *is used* **then**
    $\mathbf{v}_\Xi^t \leftarrow \mathbf{v}_\Xi^{t-1} - \alpha_t \nabla_{\mathbf{v}} \mathbb{F}(\mathbf{w}^{t-1}, \mathbf{v}^{t-1})|_{\Xi_t}$
  end
  else $\mathbf{v}_\Xi^t \leftarrow g_m(\phi(\Xi_t, \mathbf{w}^{t-1}); \Theta)$;
  $\mathbf{w}^t \leftarrow \mathbf{w}^{t-1} - \alpha_t \nabla_{\mathbf{w}} \mathbb{F}(\mathbf{w}^{t-1}, \mathbf{v}^t)|_{\Xi_t}$
  $t \leftarrow t + 1$
end
return $\mathbf{w}^t$

Algorithm (Training MentorNet with StudentNet)

Training MentorNet and StudentNet via mini-batch SGD.

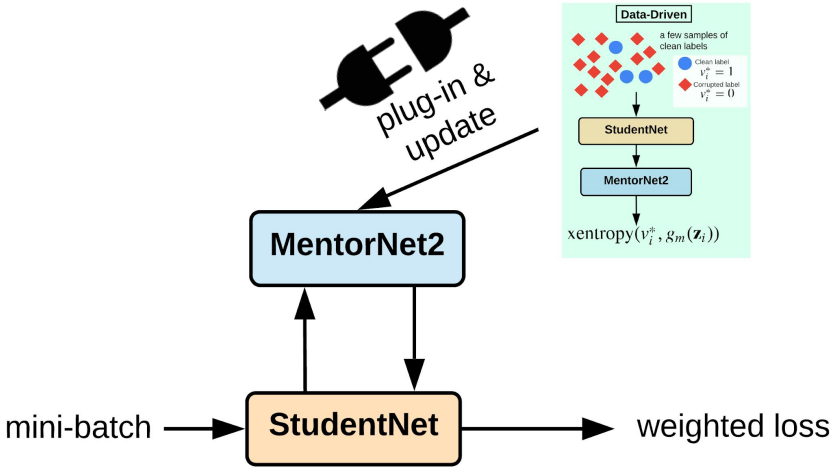


Algorithm 1 SPADE for minimizing Eq. (1)

Input : Dataset $\mathcal{D}$, a predefined G or a learned $g_m(\cdot;\Theta)$
Output : The model parameter $\mathbf{w}$ of StudentNet.

Initialize $\mathbf{w}^0$, $\mathbf{v}^0$, $t = 0$
while *Not Converged* **do**
  Fetch a mini-batch Ξ_t uniformly at random
  For every $(\mathbf{x}_i, y_i)$ in Ξ_t compute $\phi(\mathbf{x}_i, y_i, \mathbf{w}^t)$
  if *update curriculum* **then**
    $\Theta \leftarrow \Theta^*$, where Θ^* is learned in Sec. 3.1
  end
  if G *is used* **then**
    $\mathbf{v}_\Xi^t \leftarrow \mathbf{v}_\Xi^{t-1} - \alpha_t \nabla_{\mathbf{v}} \mathbb{F}(\mathbf{w}^{t-1}, \mathbf{v}^{t-1})|_{\Xi_t}$
  end
  else $\mathbf{v}_\Xi^t \leftarrow g_m(\phi(\Xi_t, \mathbf{w}^{t-1}); \Theta)$;
  $\mathbf{w}^t \leftarrow \mathbf{w}^{t-1} - \alpha_t \nabla_{\mathbf{w}} \mathbb{F}(\mathbf{w}^{t-1}, \mathbf{v}^t)|_{\Xi_t}$
  $t \leftarrow t + 1$
end
return $\mathbf{w}^t$

Algorithm (Training MentorNet with StudentNet)



Algorithm 1 SPADE for minimizing Eq. (1)

Input : Dataset $\mathcal{D}$, a predefined G or a learned $g_m(\cdot;\Theta)$
Output : The model parameter $\mathbf{w}$ of StudentNet.

Initialize $\mathbf{w}^0, \mathbf{v}^0, t = 0$
while *Not Converged* **do**
  Fetch a mini-batch Ξ_t uniformly at random
  For every $(\mathbf{x}_i, y_i)$ in Ξ_t compute $\phi(\mathbf{x}_i, y_i, \mathbf{w}^t)$
  if *update curriculum* **then**
    $\Theta \leftarrow \Theta^*$, where Θ^* is learned in Sec. 3.1
  end
  if G *is used* **then**
    $\mathbf{v}_\Xi^t \leftarrow \mathbf{v}_\Xi^{t-1} - \alpha_t \nabla_{\mathbf{v}} \mathbb{F}(\mathbf{w}^{t-1}, \mathbf{v}^{t-1})|_{\Xi_t}$
  end
  else $\mathbf{v}_\Xi^t \leftarrow g_m(\phi(\Xi_t, \mathbf{w}^{t-1}); \Theta)$;
  $\mathbf{w}^t \leftarrow \mathbf{w}^{t-1} - \alpha_t \nabla_{\mathbf{w}} \mathbb{F}(\mathbf{w}^{t-1}, \mathbf{v}^t)|_{\Xi_t}$
  $t \leftarrow t + 1$
end
return $\mathbf{w}^t$