Learning with Noisy Supervision

Part IV. Automated Learning from Noisy Labels (LNL)

Masashi Sugiyama[1 2], Tongliang Liu[3], Bo Han[4], Quanming Yao[5], Gang Niu[1]

[1]RIKEN, [2]University of Tokyo, [3]University of Sydney,
[4]Hong Kong Baptist University, [5]Tsinghua University

Email: qyaoaa@tsinghua.edu.cn

Outline

1. What is Automated Machine Learning (AutoML)?
    - What is Machine Learning?
    - What is Automated Machine Learning (AutoML)?
    - How to Use AutoML Techniques
2. Sample Selection for Learning with Noisy Labels (LNL)
3. Future Works & Summary

What is Machine Learning (ML)?

Applications

Search Engine
Recommender Systems
Loss Assessment

Image Classification

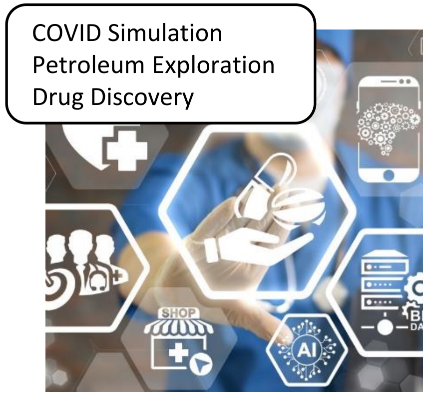
Predict the class of the object

Security Monitoring
Bio-payment
Flow Statistics

Face Recognition

Who is the person

COVID Simulation
Petroleum Exploration
Drug Discovery



Drug Design

Learn to make decisions

Better Performance

Higher Efficiency

Definition

Structure Samples

$$\min_x F(x; D)$$

Parameters

(iterative) optimization → Prediction Accuracy ↑

[1]. Machine Learning, Tom Mitchell, McGraw Hill, 1997.
[2]. 周志华 著. 机器学习, 北京: 清华大学出版社, 2016年

ML = Data + Knowledge

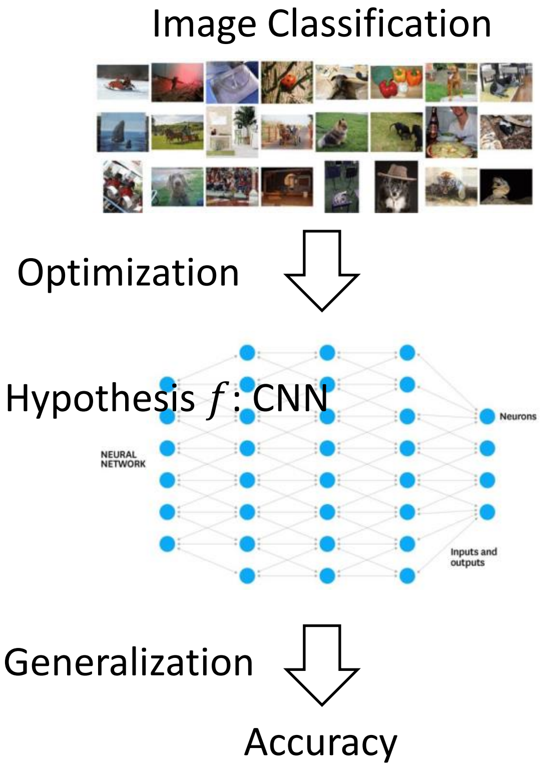


Image Classification

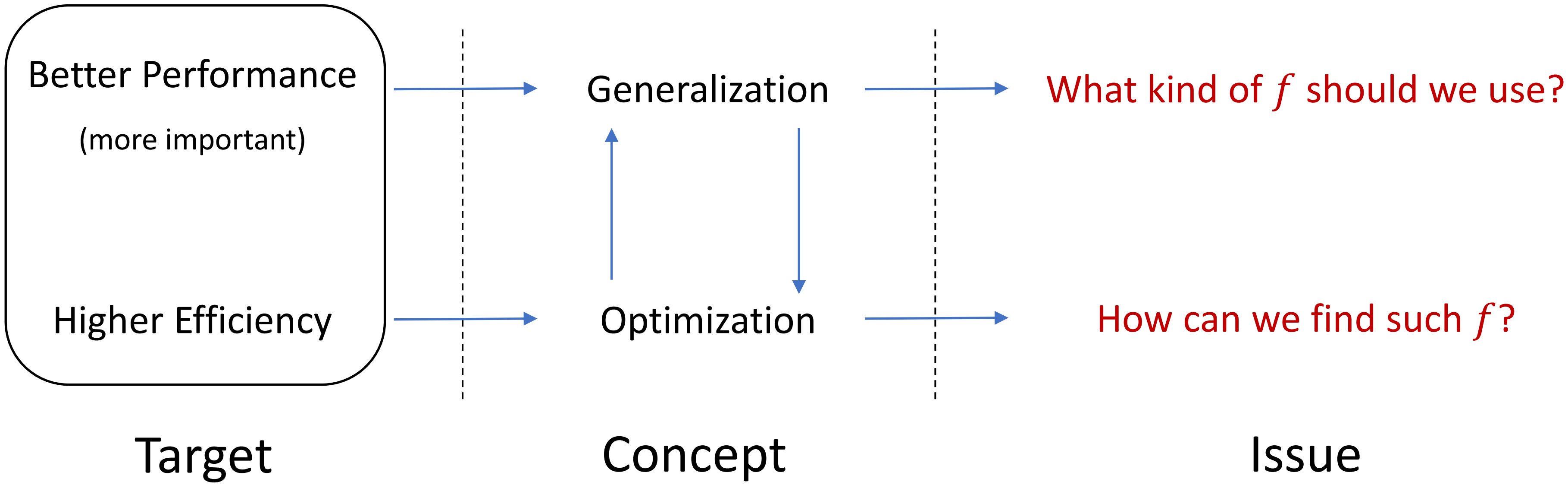
Optimization

Hypothesis f: CNN

Generalization

Accuracy

Design a **hypothesis (function) f** to perform the learning task


Better Performance

(more important)

Higher Efficiency

Generalization

Optimization

What kind of f should we use?

How can we find such f?

Target

Concept

Issue

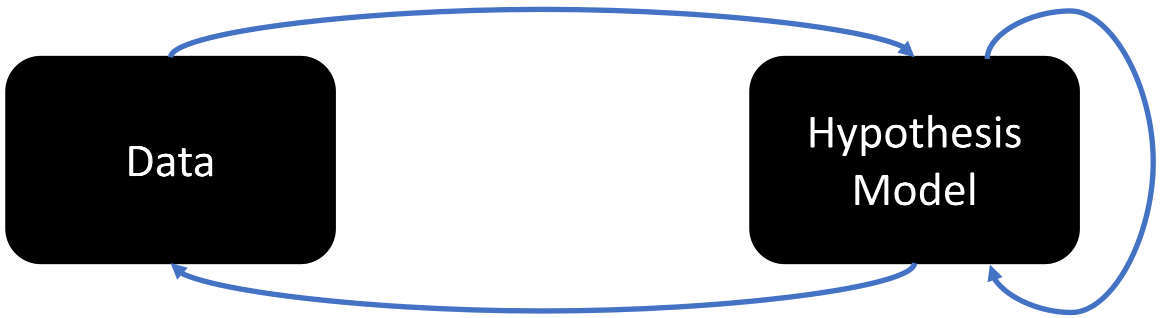
Not everything can be learnt

PAC-Learning (Definition 2.3 in [1]): What kind of problems can be solved in polynomial time

No Free Lunch Theorem (Appendix B [2]): No single algorithm can be good on all problems

[1]. M. Mohri, A. Rostamizadeh, A. Talwalkar. Foundations of machine learning. 2018
[2]. O. Bousquet, et.al. Introduction to Statistical Learning Theory. 2016

How to use ML Well?



Prior Knowledge

Optimization

Data

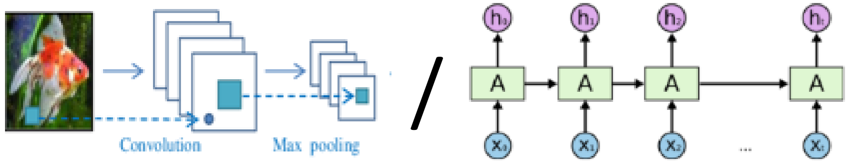
Hypothesis Model

Generalization Performance

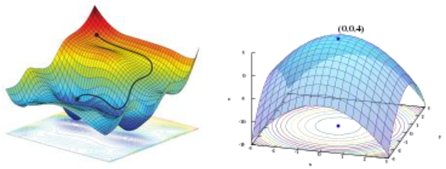
The Advancement of Learning

- An iteration between theory and practice
- A feedback loop

CNN / RNN



Generalization: What kind of f should we use?



SGD v.s. Adagrad[1]

Optimization: How can we find such f?

Prior knowledge

"All models are wrong, but some are useful"[2]

Better understanding of prior knowledge → Better hypothesis → Better generalization performance

---

[1]. Image Source: A. Amini et al. "Spatial Uncertainty Sampling for End-to-End Control". NeurIPS Bayesian Deep Learning 2018

[2] G. Box, Science and statistics, JASA 1976

Outline

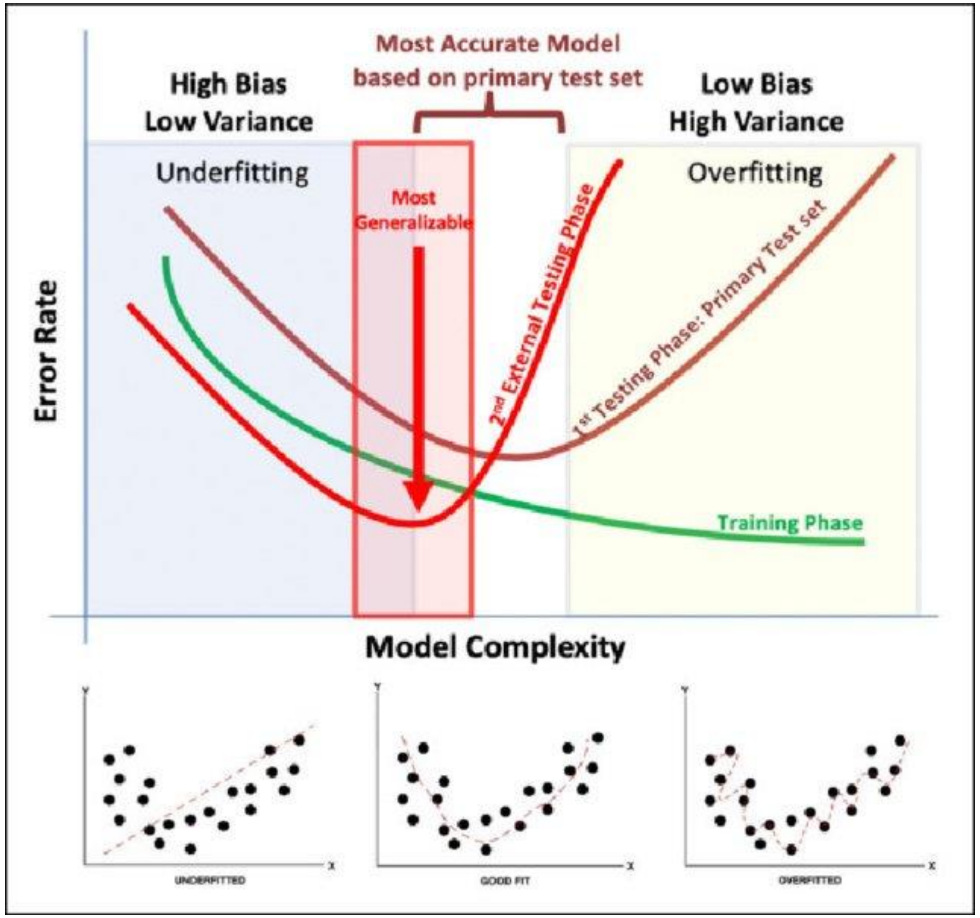
1. What is Automated Machine Learning (AutoML)?
    - What is Machine Learning?
    - What is Automated Machine Learning (AutoML)?
    - How to Use AutoML Techniques
2. Sample Selection for Learning with Noisy Labels (LNL)
3. Future Works & Summary

Simple Example – Tune hyper-parameter

Bi-level optimization

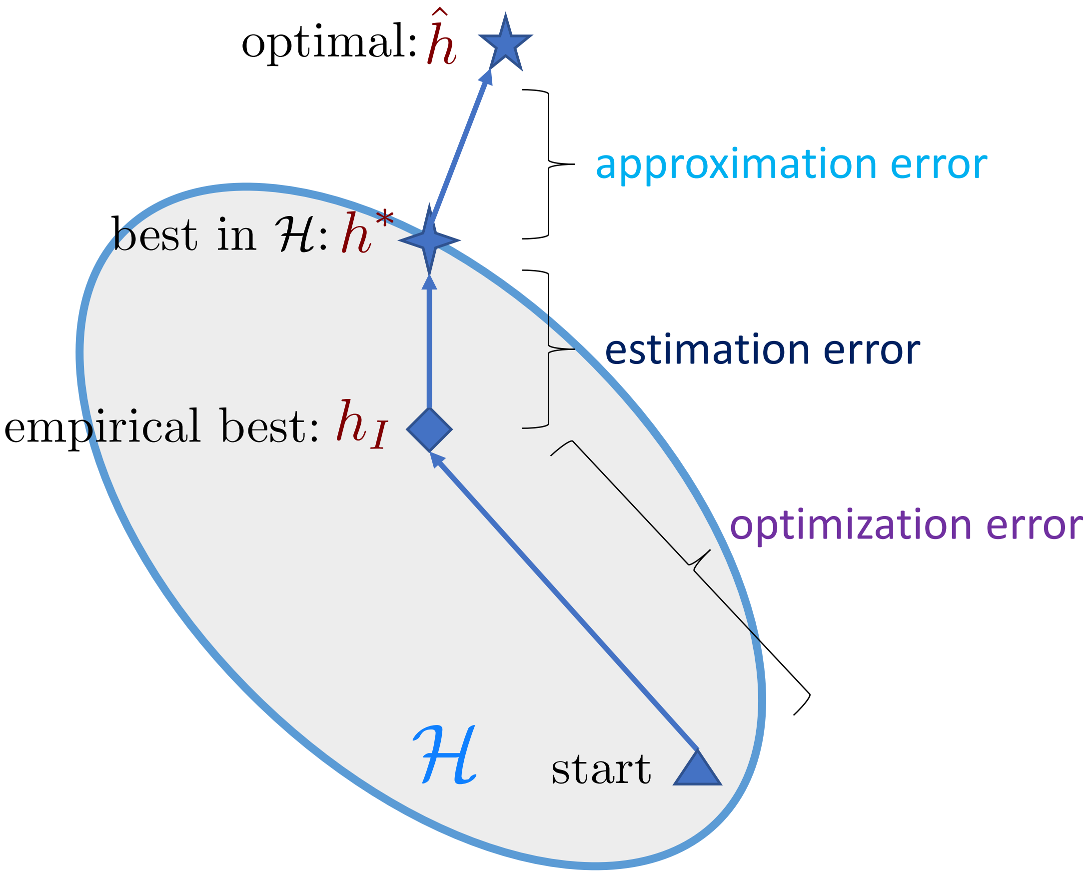
$$\underbrace{\max_{\lambda} \sum_j h(x_j; w^*)}_{\text{Validation Performance}} \quad \text{s.t.} \quad w^* = \underbrace{\min_{w} \sum_i f(x_i; w) + \overset{\text{Hyper-parameter}}{\lambda} \|w\|_1}_{\text{Training objective}}$$

- Large λ leads to sparse w^*
- Grid search: enumerating $\lambda \in \{1,2,4,8, \ldots\}$



[1]. Image source: Artificial Intelligence and Machine Learning in Pathology: The Present Landscape of Supervised Methods.

Mach. Learn – Error decomposition



Total error in machine learning

- Approximation error
  - Which classifier to be used
  - What are their hyper-parameters
  - Distribution changes
- Estimation error
  - Finite samples
  - Regularization hyper-parameter
- Optimization error
  - Which algorithm to be used
  - How to tune its step-size

Reduce

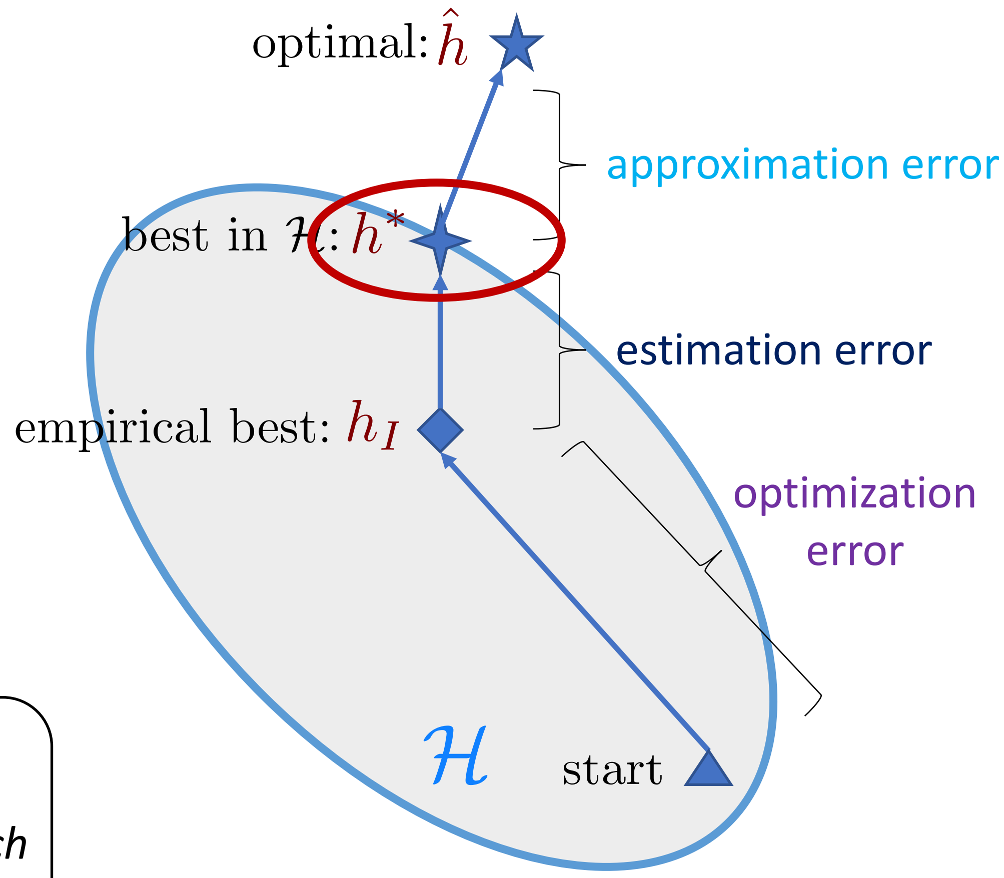
$$\min_{w} \sum_{i} f(x_i; w) + \boxed{\lambda} \|w\|_1$$

Image is from Y. Wang, et. al. Generalizing from a Few Examples: A Survey on Few-Shot Learning. CSUR 2020

Look Inside Error Decomposition

Automatically find h^* by bi-level optimization

$$\underbrace{\max_{\lambda} \sum\nolimits_j h(x_j; w^*)}_{\text{Validation Performance}} \quad \text{s.t.} \quad w^* = \underbrace{\min_{w} \sum\nolimits_i f(x_i; w) + \lambda \|w\|_1}_{\text{Training objective}}$$



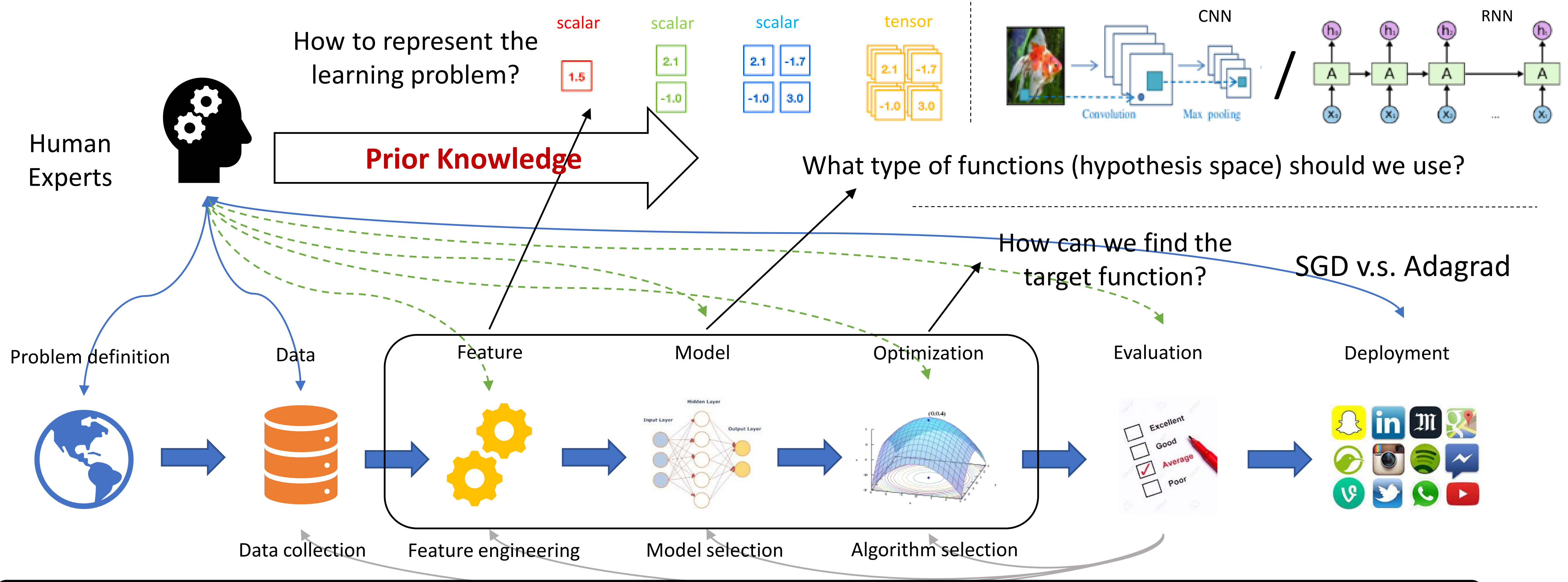
How to further improve the performance in an automatic manner (i.e., reduce the approximation error)?

- Feature can be weak → *Automatic feature engineering*
- Linear predictor can be too restrictive → *Neural architecture search*
- Grid search can be slow → *Search in a supernet*

AutoML

Figure is from Q. Yao et.al. Taking Human out of Learning Applications: A Survey on Automated Machine Learning. arXiv 2018

What is AutoML – Practical Viewpoint



How to represent the learning problem?

Human Experts

Prior Knowledge

What type of functions (hypothesis space) should we use?

How can we find the target function?

SGD v.s. Adagrad

Problem definition

Data

Feature

Model

Optimization

Evaluation

Deployment

Data collection

Feature engineering

Model selection

Algorithm selection

Parameterize (low-level) prior knowledge in the usage and design of machine learning

As a consequence

- Human participations can be naturally replaced by computation power
- total error of machine learning can be reduced (generalization can be improved)

Figure 4.1 & 4.2. M. Mohri, A. Rostamizadeh, A. Talwalkar. Foundations of machine learning. 2018
[1]. P. Battaglia. et.al. Relational inductive biases, deep learning, and graph networks. arXiv 2018.

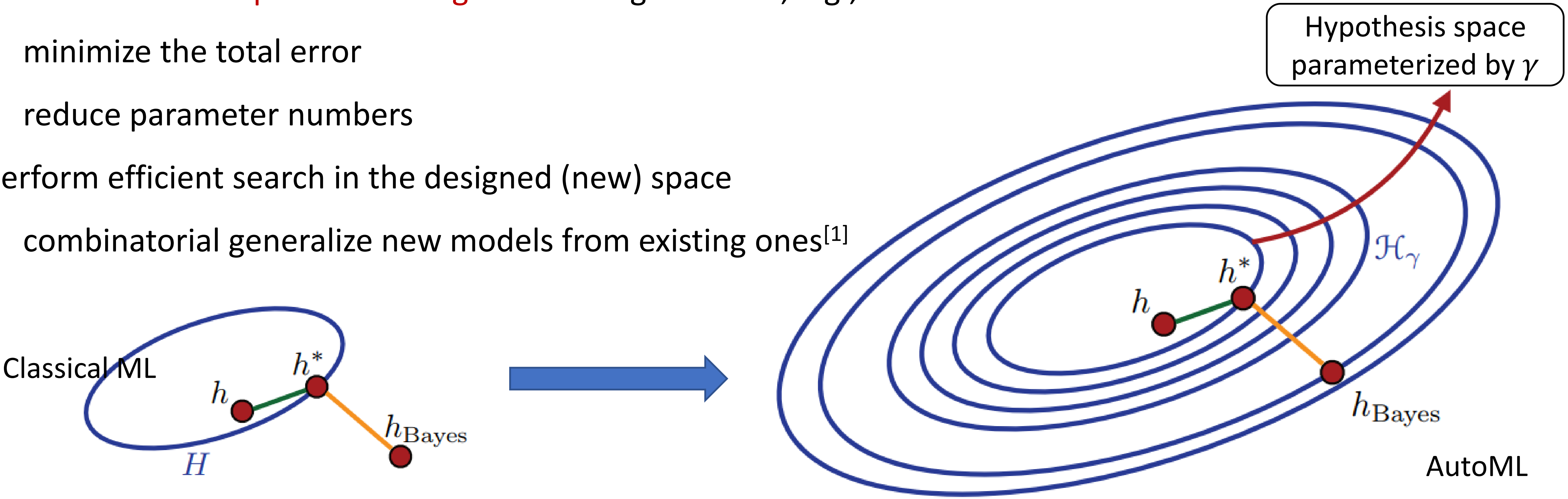
What is AutoML – Generalization Viewpoint

Parameterized the prior knowledge of learning methods, e.g.,

- minimize the total error
- reduce parameter numbers

Perform efficient search in the designed (new) space

- combinatorial generalize new models from existing ones[1]

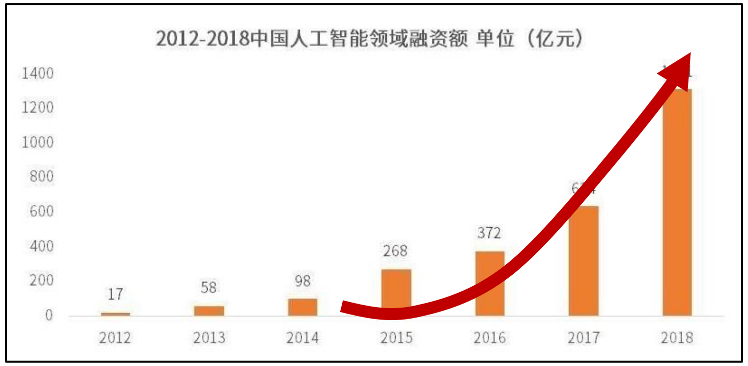


Parameterize (low-level) prior knowledge in the usage and design of machine learning

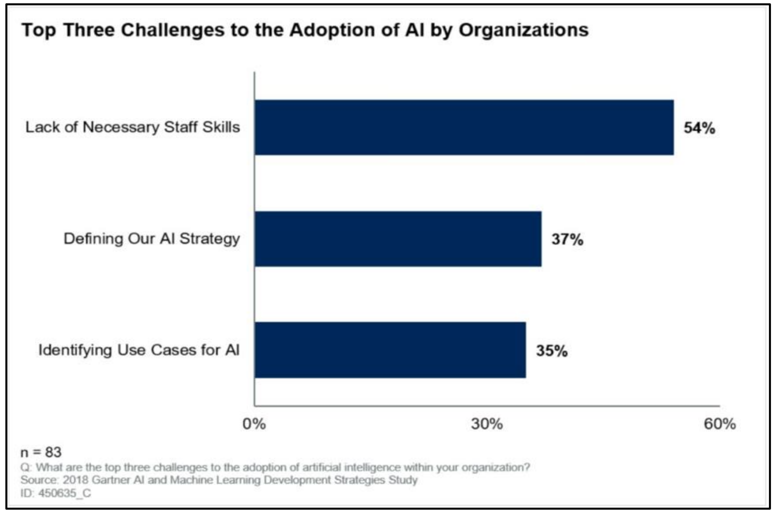
As a consequence

- Human participations can be naturally replaced by computation power
- total error of machine learning can be reduced (generalization can be improved)

Why We need AutoML?

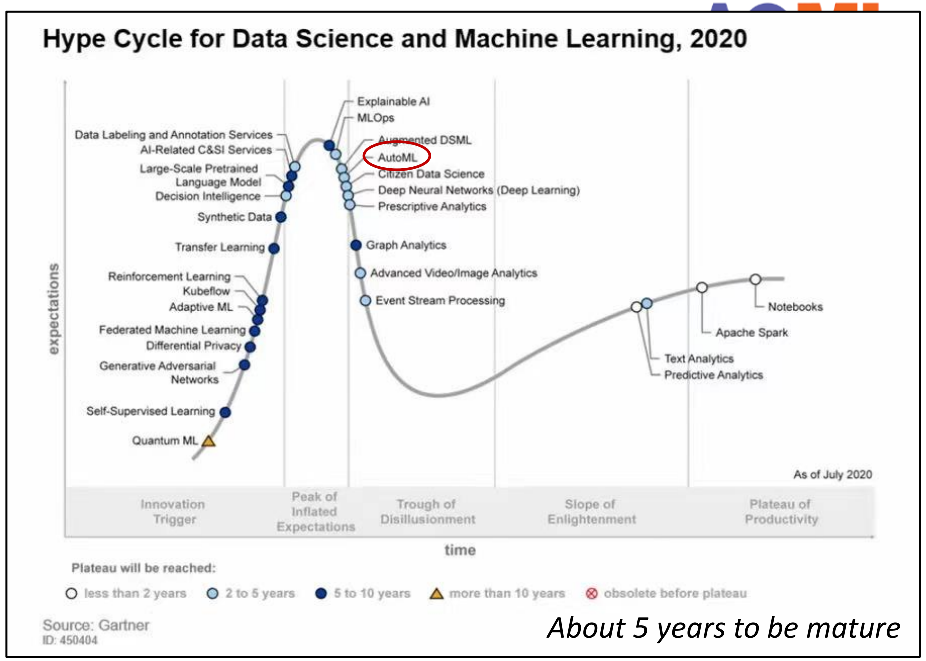


Investment in AI industry



n = 83
Q: What are the top three challenges to the adoption of artificial intelligence within your organization?
Source: 2018 Gartner AI and Machine Learning Development Strategies Study
ID: 450635_C

Practical needs



Source: Gartner
ID: 450404

About 5 years to be mature

Technical trends

- **Industry** – reduce the expense, increase usage coverage – huge market value [1]
- **Academy** – understanding data science on a higher level – great intelligence value [2,3]

[1]. Gartner: https://www.forbes.com/sites/janakirammsv/2020/03/02/key-takeaways-from-the-gartner-magic-quadrant-for-ai-developer-services/#a95b99ee3e5e
[2]. Y. Bengio: From System 1 Deep Learning to System 2 Deep Learning | NeurIPS 2019
[3]. F Hutter, L Kotthoff, J Vanschoren. Automated machine learning: methods, systems, challenges. Book 2019

Related Areas

Sub-areas

- Neural architecture search
- Hyper-parameter search
- Automated feature engineering
- Algorithms selection
- Model selection

Related areas

- Bi-level / Derivative-free optimization
  - Focus more on algorithm design
  - AutoML objective is one kind of objective where these algorithms can be applied
- Meta-learning
  - Focus on parameterize task distributions
  - Another kind of bi-level objective
  - Do not use validation set to update hyper-parameters

Outline

1. What is Automated Machine Learning (AutoML)?
   - What is Machine Learning?
   - What is Automated Machine Learning (AutoML)?
   - How to Use AutoML Techniques
2. Sample Selection for Learning with Noisy Labels (LNL)
3. Future Works & Summary

How to use AutoML

1. Search Space

2. Search Objective

Bi-level optimization

4. Training Objective

3. Search Constraints

1. Define an AutoML problem

- Derive a search space from insights in specific domains
- Search objective is usually validation performance
- Search constraint is usually resource budgets
- Training objective usually comes from classical learning models

$$\text{Search Space} \rightarrow \min_{\boxed{\lambda\in\mathcal{S}}} \boxed{M(F(w^*;\lambda), D_{\mathrm{val}})} \leftarrow \text{Search Objective}$$

$$\text{s.t.} \begin{cases} \boxed{\min_{w} L(F(w;\lambda), D_{\mathrm{tra}})} \leftarrow \text{Training Objective} \\ \boxed{G(\lambda) \leq C} \leftarrow \text{Search Constraints} \end{cases}$$

2. Design or select proper search algorithm

- Reduce model training cost (time to get w^*)

What is AutoML – Short Summary

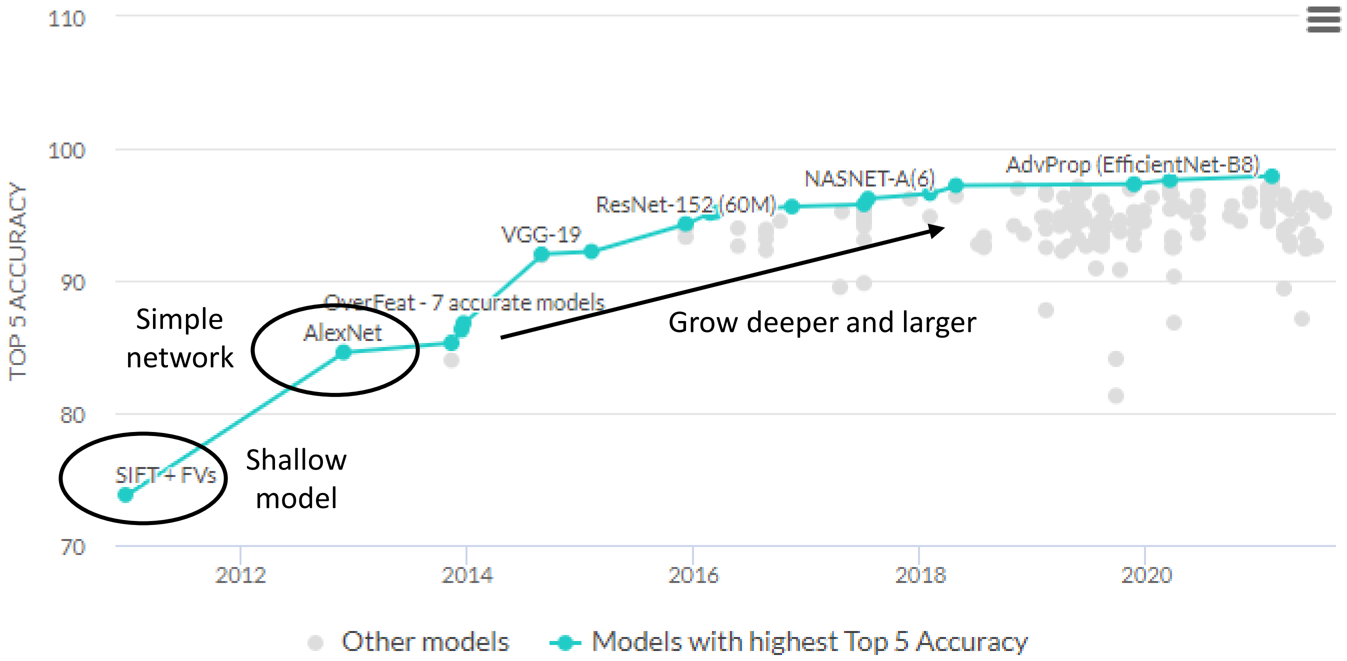
- Exploring prior knowledge is important in machine learning
  - Cost time and critical to generalization performance
- AutoML attempts to parameterize low-level prior knowledge
  - Human participations can be naturally replaced by computation power
  - total error can be reduced (generalization can be improved)
- To use well AutoML techniques
  - Exploring high-level domain knowledge when defining the AutoML problem
  - Reducing model training cost when design search algorithm

Outline

1. What is Automated Machine Learning (AutoML)?
2. Sample Selection for Learning with Noisy Labels (LNL)
    - What are Small-loss Samples
    - Co-teaching, its Variants and Limitations
    - Design Sample Selection Criterion by AutoML
3. Future Works & Summary

Success of Deep Networks

- 14197122 images
- 21841 classes indexed



Big & High-quality data is the fuel

Figure is from https://paperswithcode.com/sota/image-classification-on-imagenet?metric=Top%205%20Accuracy

Where does Big Data Comes from?

Crowd-sourcing

Incorrect ones

Correct ones

Hamburger

Hamburger with french fries and a beer

Course	Main course
Place of origin	Hamburg (Germany) or United States (disputed)
Created by	Multiple claims (see text)
Serving temperature	Hot
Main ingredients	Ground meat, bread

Cookbook: Hamburger
Media: Hamburger

Web crawler

take image as sample

take words from caption as labels

Where does Big Data Comes from?

Crowd-sourcing

- Workers may not be reliable
- There can be spammers or attackers

Web crawler

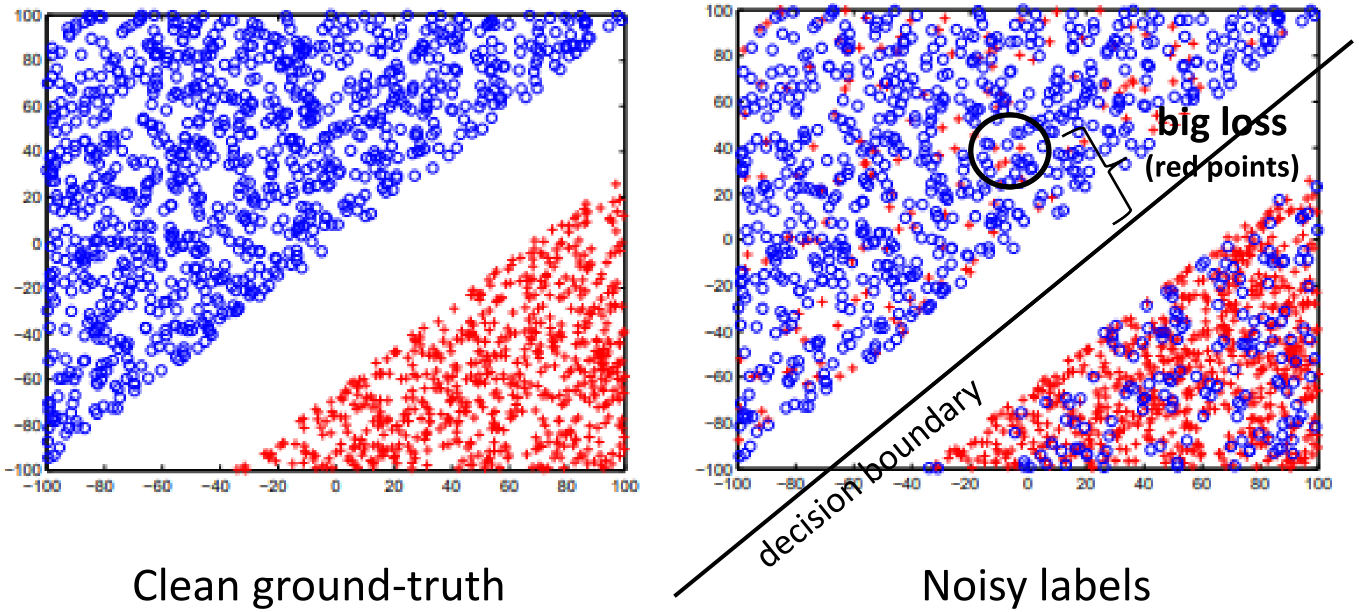
- The context can be complex
- Caption may not be relevant

Big & High-quality data: difficult & expensive

- Data: what we usually have in hand is a big data with noisy labels
- Performance: noisy labels degrade the accuracy of deep neural networks by 20% to 40%

Where does Big Data Comes from?

If the classifier A has the ability to predict, then A sample with noisy labels should have larger loss than sample with correct labels



Clean ground-truth

Noisy labels

Small-loss samples

→ Likely to be clean samples

Using hinge loss as an example

- Red points: zero loss
- Blue points: much larger than zero

Please check more explanations in theory in “*Part II: Statistical Learning with Noisy Supervision*” By Tongliang Liu

What is Special about Deep Networks?

Stochastic gradient descent (SGD) is a must for training deep networks

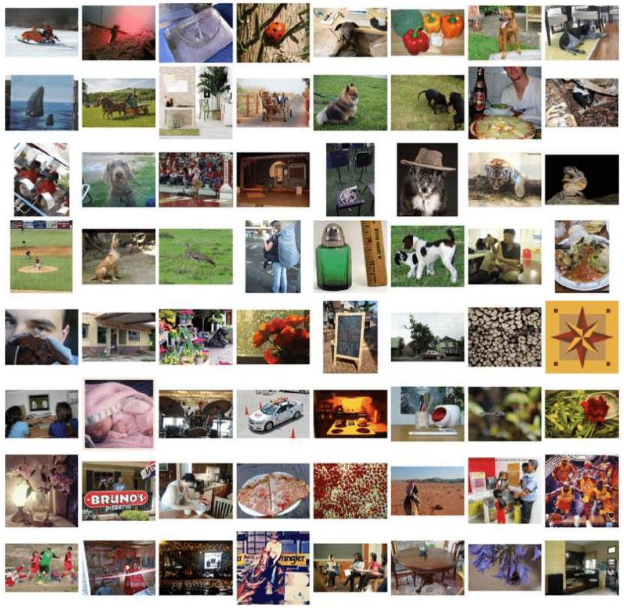
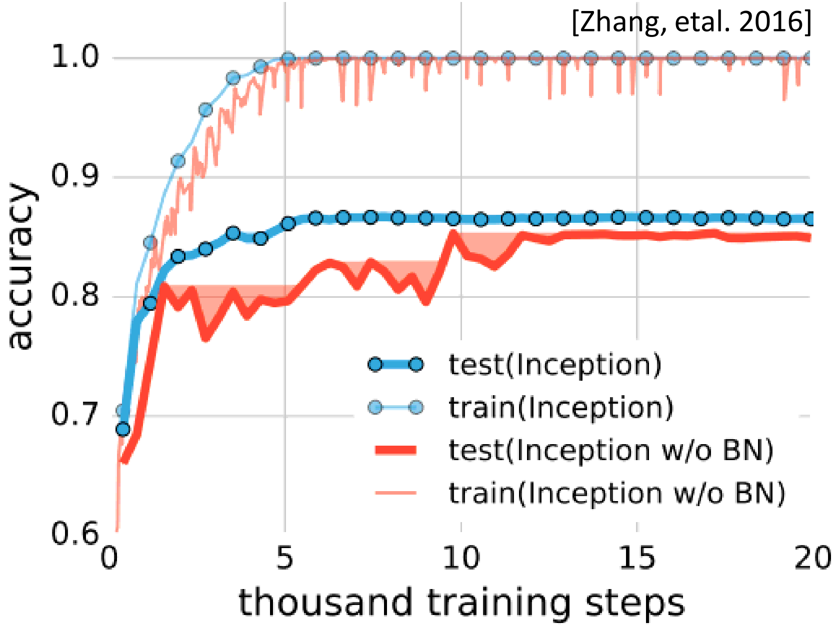


Image classification

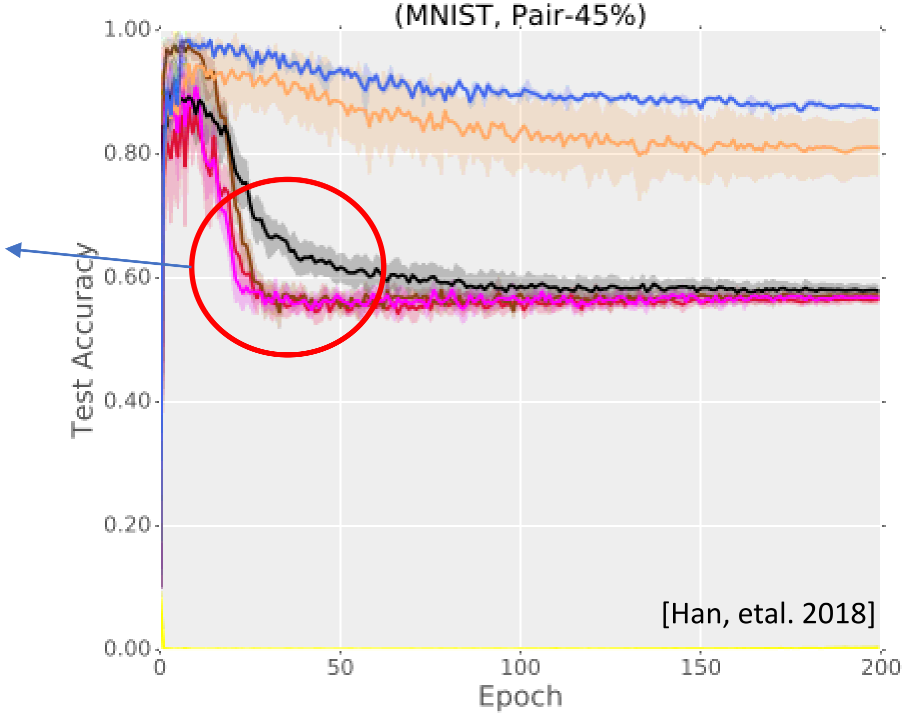


Train/test accuracy v.s. steps

What is Special about Deep Networks?

Noisy labels

Standard CNN



Test accuracy v.s. steps

Memorization effect: Learning easy patterns first, then (totally) over-fit noisy training data. Independent with network types and structures.

C. Zhang et.al. Understanding deep learning requires rethinking generalization. ICLR 2017
D Arpit et.al. A closer look at memorization in deep networks. NIPS 2017

How to Learn from Noisy Labels?

Fundamental properties

- SGD is almost a must for deep networks
- Deep networks have memorization effects

Facts

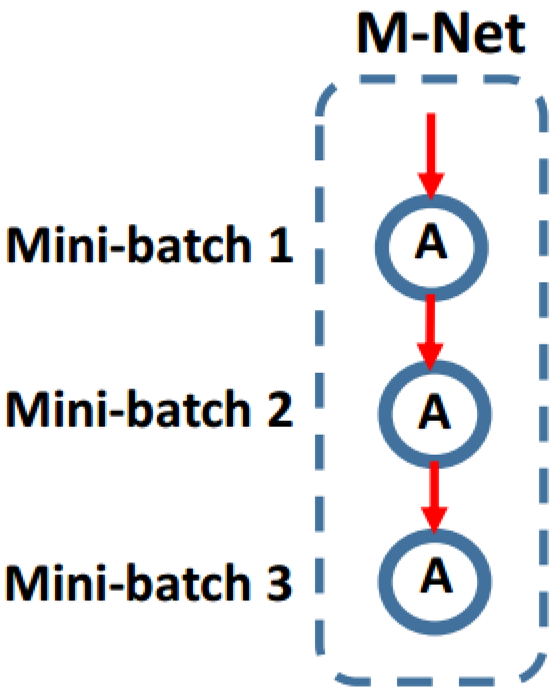
- Noisy labels has larger losses.

How can we robustly learn from noisy label utilizing above properties and fact?

Outline

1. What is Automated Machine Learning (AutoML)?
2. Sample Selection for Learning with Noisy Labels (LNL)
   - What are Small-loss Samples
   - Co-teaching, its Variants and Limitations
   - Design Sample Selection Criterion by AutoML
3. Future Works & Summary

Prior Work – Menter-net [Lu et.al. 2018]



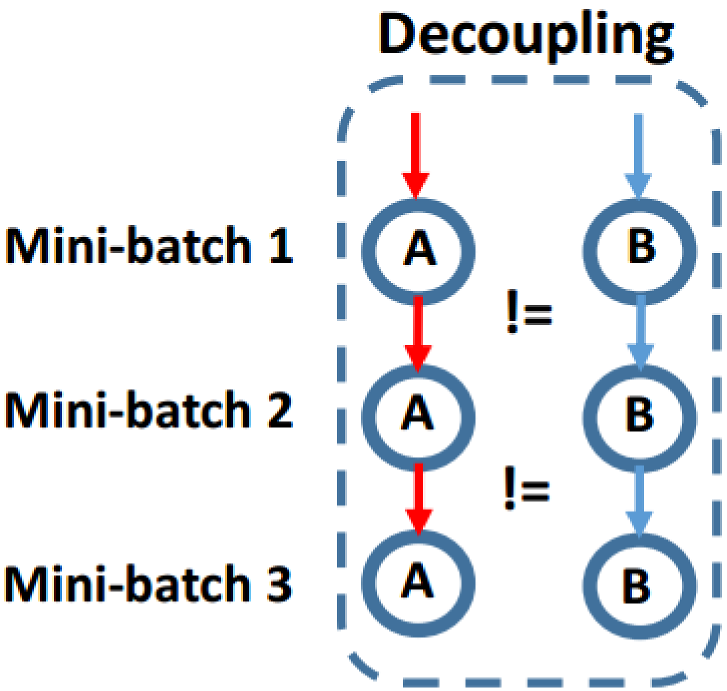
Deep networks are all based on

- Stochastic gradient descent
- Gradient is performed by mini-batch

Mentor-Net

- drop samples with large loss in each mini-batch, use small loss samples in each mini-batch to update parameters
- use one classifier to self-bootstrap

Prior Work – Decoupling [E. Malach and S. Shalev-Shwartz, 2017]



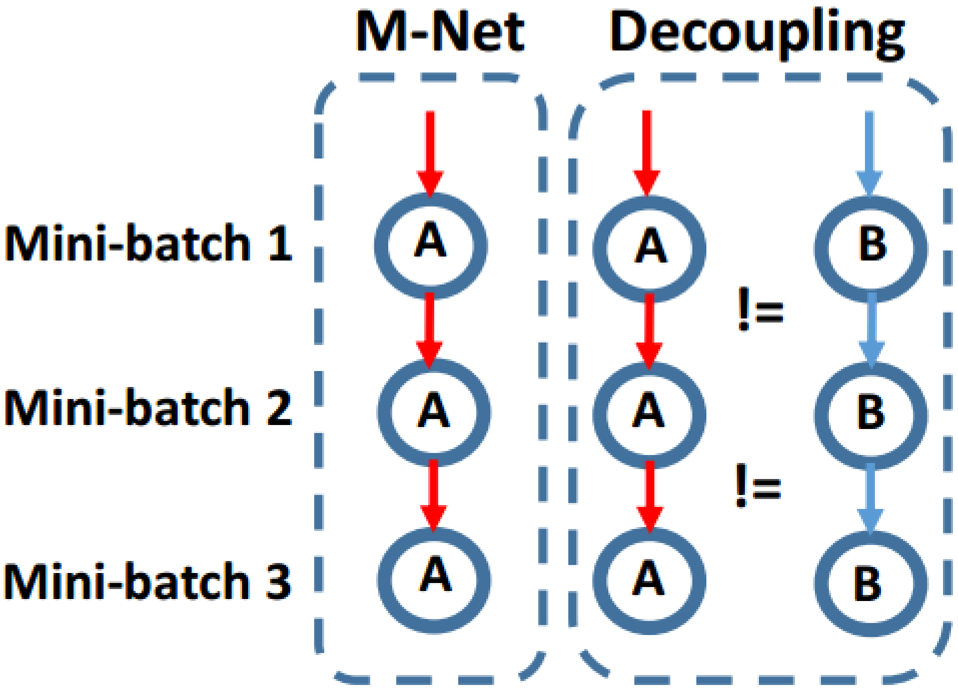
Easy samples

- Can be quickly learnt and classified (memorization)
- Have small gradients, which slow down network training

Decoupling

- Focus on hard examples, which can be more informative
- Use samples in each mini-batch that two classifiers have different predictions to update network

Message from Prior Works

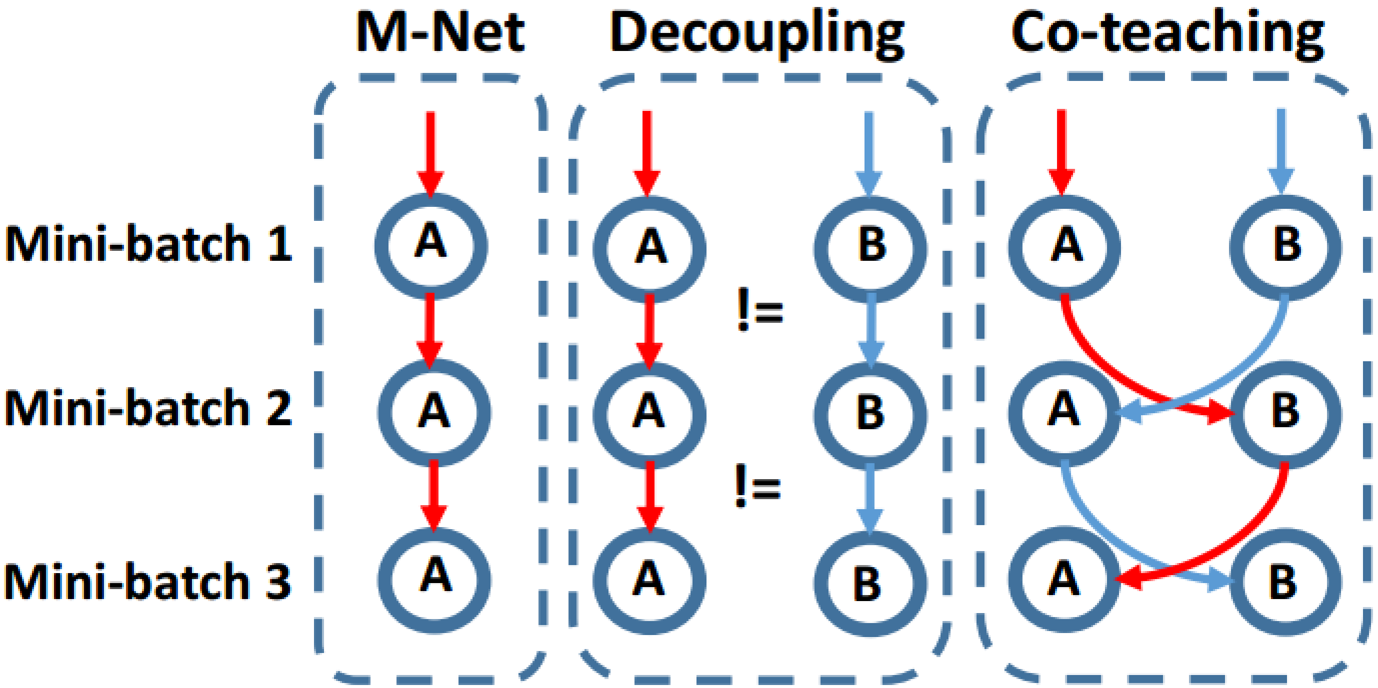


	Mentor-net	Decoupling
Small loss	YES	NO
Memorization	NO	YES
SGD	YES	YES

How can we robustly learn from noisy label utilizing (small loss, memorization and SGD)?

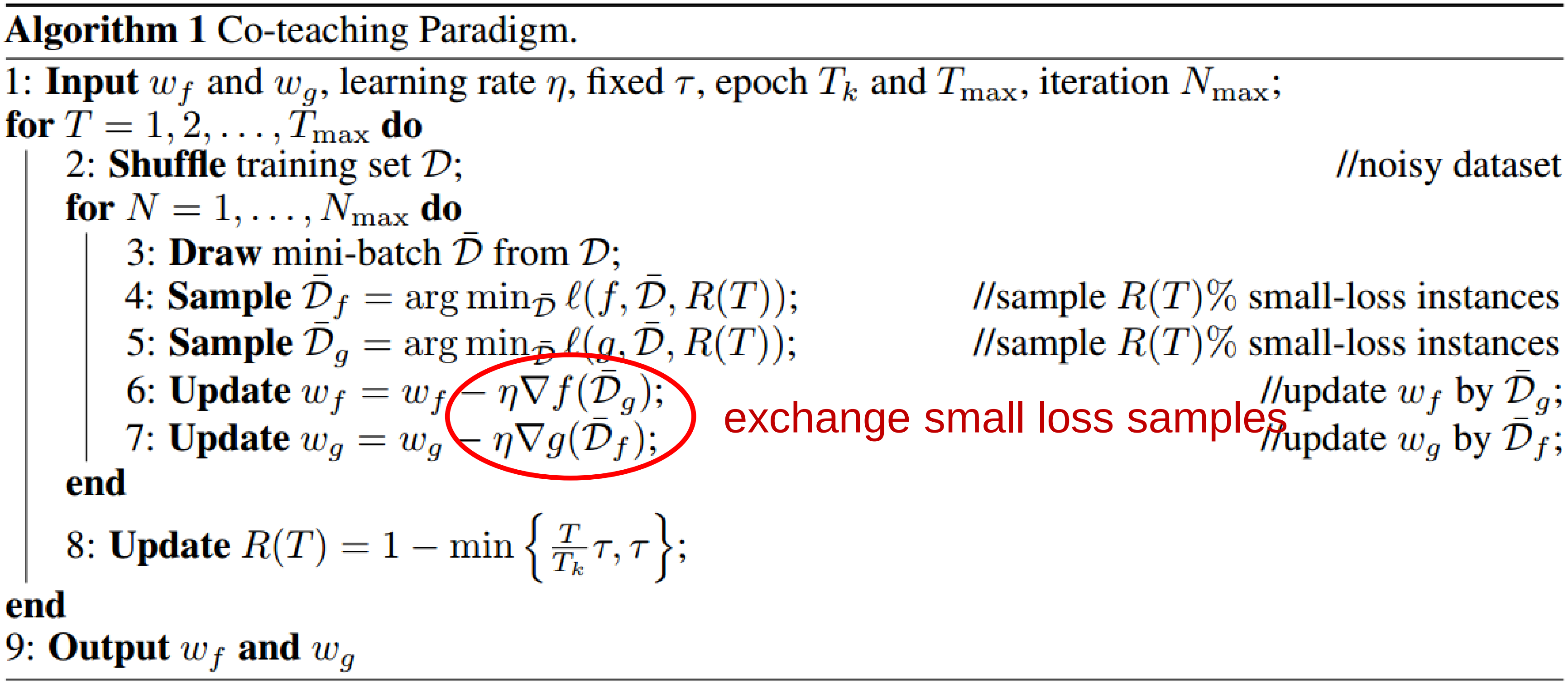
Co-teaching – Core idea

Why not **exchange** small loss in each mini-batch for two classifiers?



B. Han et.al. Co-teaching: Robust training deep neural networks with extremely. NeurIPS 2018

Co-teaching – Implementations



Algorithm 1 Co-teaching Paradigm.

```
1: Input $w_f$ and $w_g$, learning rate $\eta$, fixed $\tau$, epoch $T_k$ and $T_{\max}$, iteration $N_{\max}$;
for $T = 1, 2, \ldots, T_{\max}$ do
    2: Shuffle training set $\mathcal{D}$;                                   //noisy dataset
    for $N = 1, \ldots, N_{\max}$ do
        3: Draw mini-batch $\bar{\mathcal{D}}$ from $\mathcal{D}$;
        4: Sample $\bar{\mathcal{D}}_f = \arg\min_{\bar{\mathcal{D}}} \ell(f, \bar{\mathcal{D}}, R(T))$;   //sample $R(T)\%$ small-loss instances
        5: Sample $\bar{\mathcal{D}}_g = \arg\min_{\bar{\mathcal{D}}} \ell(g, \bar{\mathcal{D}}, R(T))$;   //sample $R(T)\%$ small-loss instances
        6: Update $w_f = w_f - \eta \nabla f(\bar{\mathcal{D}}_g)$;               //update $w_f$ by $\bar{\mathcal{D}}_g$;
        7: Update $w_g = w_g - \eta \nabla g(\bar{\mathcal{D}}_f)$;               //update $w_g$ by $\bar{\mathcal{D}}_f$;
    end
    8: Update $R(T) = 1 - \min\left\{\frac{T}{T_k}\tau, \tau\right\}$;
end
9: Output $w_f$ and $w_g$
```

exchange small loss samples

- Change the procedures in SGD algorithm

Co-teaching – Key questions

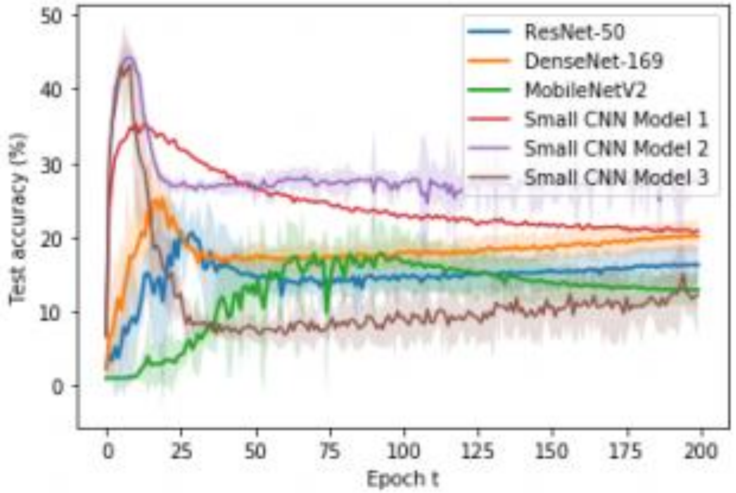
Q1. Why can sampling small-loss instances help find clean instances?

- When labels are correct, small-loss instances are more likely to be ones with correct labels
- However, the above requires that the classifier is reliable enough. The “memorization” effect of deep networks can exactly help us address this problem

Co-teaching – Key questions

Q2. How many samples to be kept?

- During the initial phase when the learning curve rises, the deep network is plastic and can learn easy patterns. One can allow a larger $R(t)$ as there is little risk of memorization.
- As training proceeds and the learning curve has peaked, the network starts to memorize and overfit the noisy samples. Hence, $R(t)$ should then decrease.

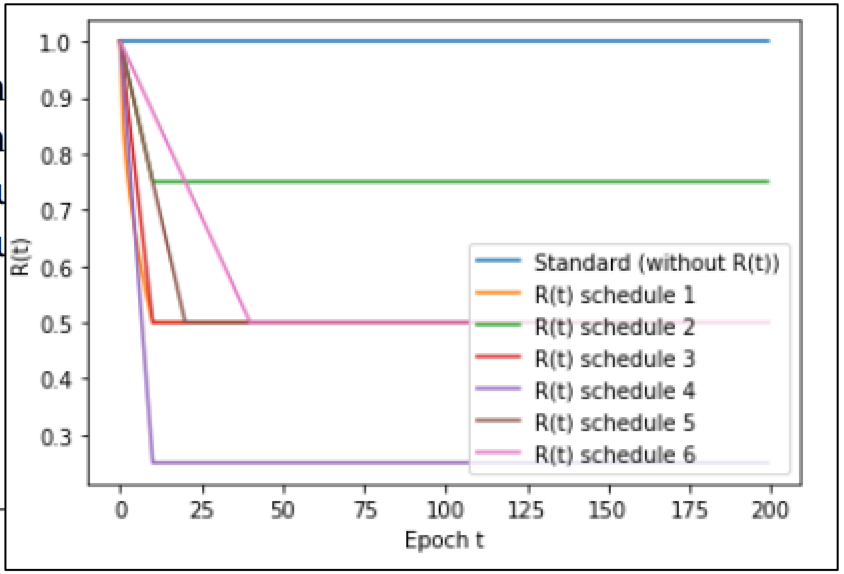


$$R(t) = 1 - \tau \cdot \min\left((t/t_k)^c, 1\right),$$

Co-teaching – Selection rule

```
Algorithm 1 Co-teaching Paradigm.
1: Input $w_f$ and $w_g$, learning rate $\eta$, fixed $\tau$, epoch $T_k$ and $T_{\max}$, iteration $N_{\max}$;
for $T = 1, 2, \ldots, T_{\max}$ do
    2: Shuffle training set $\mathcal{D}$;                                   //noisy dataset
    for $N = 1, \ldots, N_{\max}$ do
        3: Draw mini-batch $\bar{\mathcal{D}}$ from $\mathcal{D}$;
        4: Sample $\bar{\mathcal{D}}_f = \arg\min_{\bar{\mathcal{D}}} \ell(f, \bar{\mathcal{D}}, R(T))$;     //sample $R(T)\%$ sm
        5: Sample $\bar{\mathcal{D}}_g = \arg\min_{\bar{\mathcal{D}}} \ell(g, \bar{\mathcal{D}}, R(T))$;     //sample $R(T)\%$ sm
        6: Update $w_f = w_f - \eta \nabla f(\bar{\mathcal{D}}_g)$;                  //u
        7: Update $w_g = w_g - \eta \nabla g(\bar{\mathcal{D}}_f)$;                  //u
    end
    8: Update $R(T) = 1 - \min\left\{\frac{T}{T_k}\tau, \tau\right\}$;
end
9: Output $w_f$ and $w_g$
```

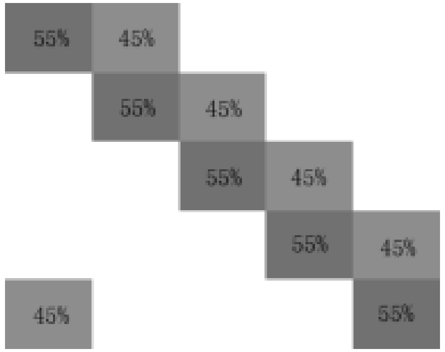
How many samples to be kept



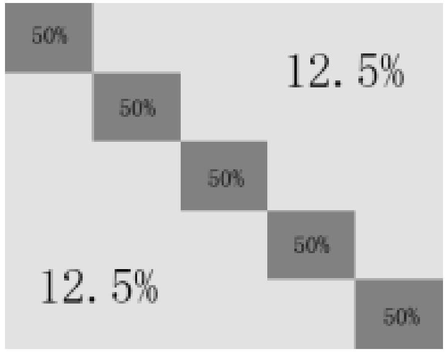
$$R(t) = 1 - \tau \cdot \min\left((t/t_k)^c, 1\right),$$

Experiments – Setup

	# of training	# of testing	# of class	image size
MNIST	60,000	10,000	10	28×28
CIFAR-10	50,000	10,000	10	32×32
CIFAR-100	50,000	10,000	100	32×32



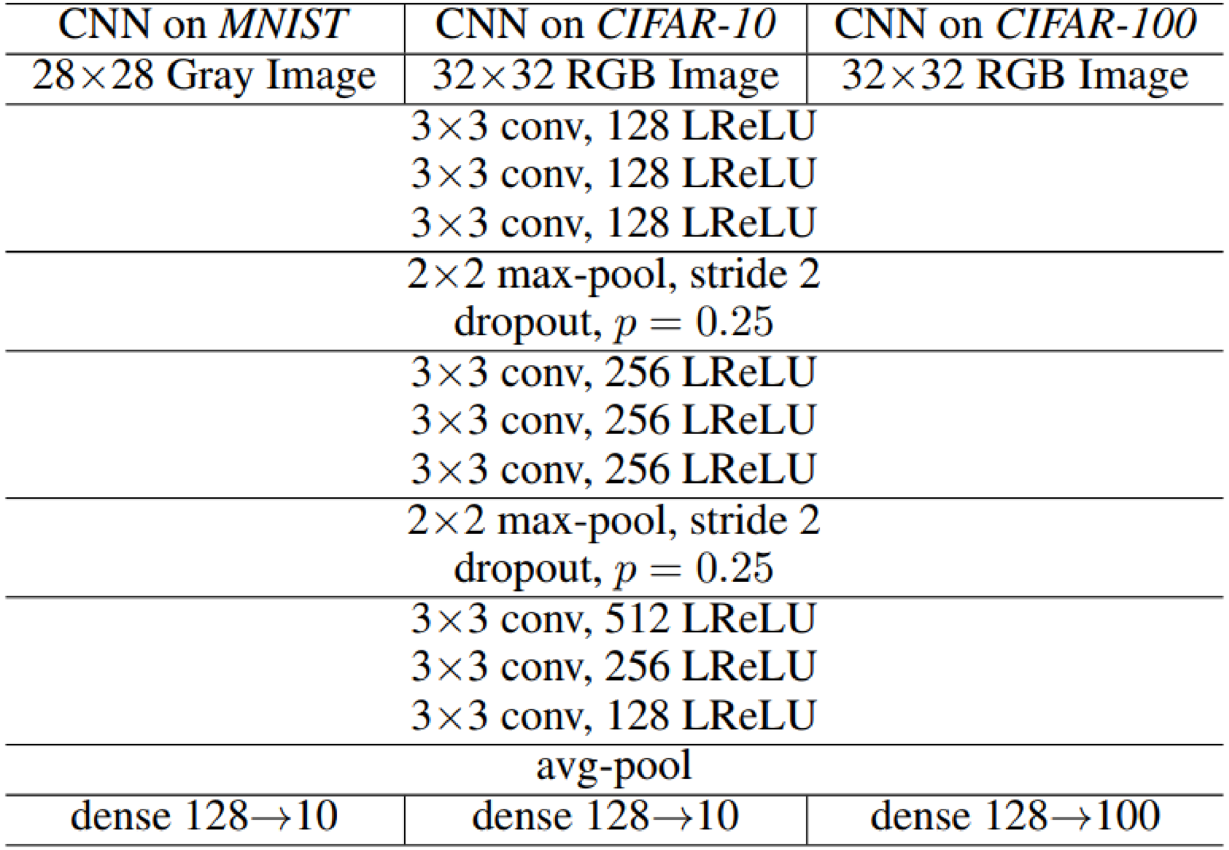
(a) Pair ($\epsilon = 45\%$).



(b) Symmetry ($\epsilon = 50\%$).

- Transition matrices of different noise types (using 5 classes as an example)
- *Pair* is much harder than *symmetry*

Experiments – Setup

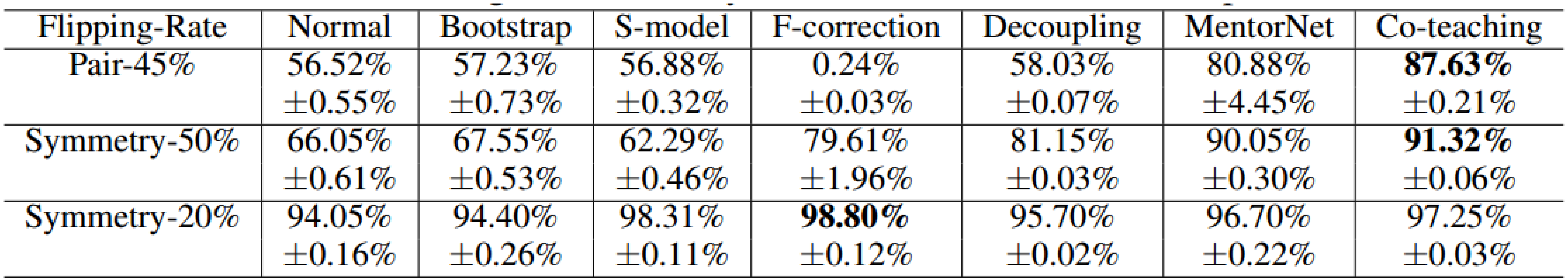


CNN on *MNIST*	CNN on *CIFAR-10*	CNN on *CIFAR-100*
28×28 Gray Image	32×32 RGB Image	32×32 RGB Image
3×3 conv, 128 LReLU		
3×3 conv, 128 LReLU		
3×3 conv, 128 LReLU		
2×2 max-pool, stride 2		
dropout, $p = 0.25$		
3×3 conv, 256 LReLU		
3×3 conv, 256 LReLU		
3×3 conv, 256 LReLU		
2×2 max-pool, stride 2		
dropout, $p = 0.25$		
3×3 conv, 512 LReLU		
3×3 conv, 256 LReLU		
3×3 conv, 128 LReLU		
avg-pool		
dense 128→10	dense 128→10	dense 128→100

- CNN models used on MNIST, CIFAR-10, and CIFAR-100. The slopes of all LReLU functions in the networks are set to 0.01
- These are not state-of-the-art models, but testbed for noisy labels [S. Laine and T. Aila, 2017]

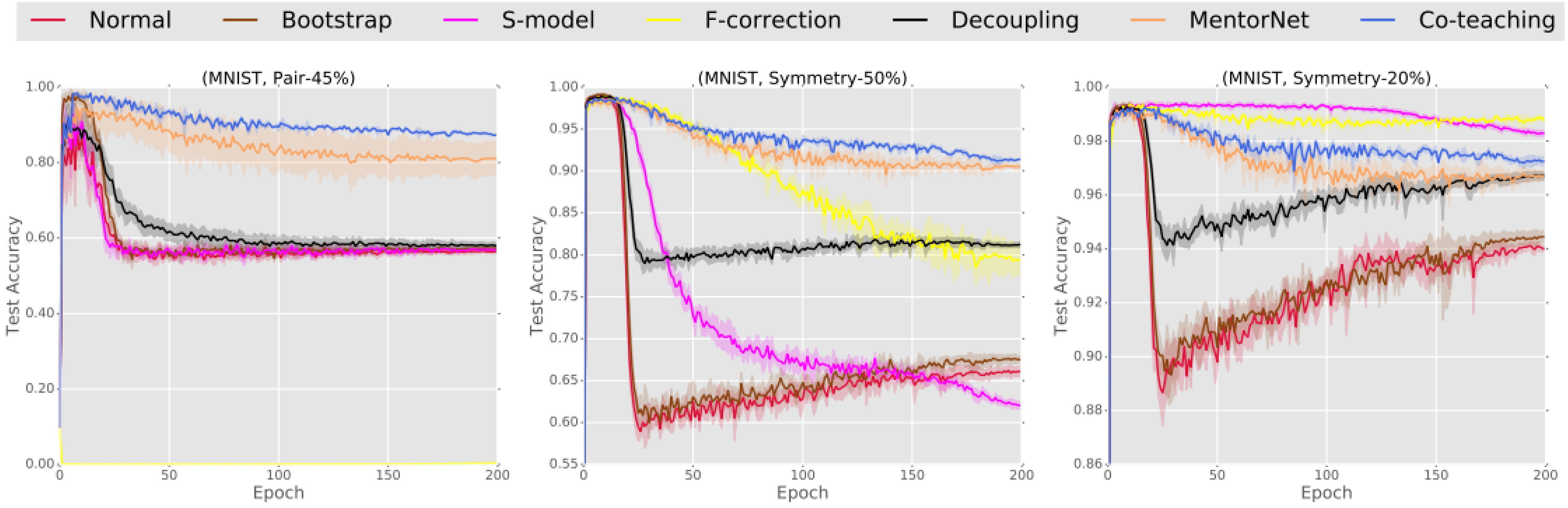
Experiments – MNIST

Average test accuracy on MNIST over the last ten epochs



Flipping-Rate	Normal	Bootstrap	S-model	F-correction	Decoupling	MentorNet	Co-teaching
Pair-45%	56.52% ±0.55%	57.23% ±0.73%	56.88% ±0.32%	0.24% ±0.03%	58.03% ±0.07%	80.88% ±4.45%	**87.63%** ±0.21%
Symmetry-50%	66.05% ±0.61%	67.55% ±0.53%	62.29% ±0.46%	79.61% ±1.96%	81.15% ±0.03%	90.05% ±0.30%	**91.32%** ±0.06%
Symmetry-20%	94.05% ±0.16%	94.40% ±0.26%	98.31% ±0.11%	**98.80%** ±0.12%	95.70% ±0.02%	96.70% ±0.22%	97.25% ±0.03%

Experiments – MNIST



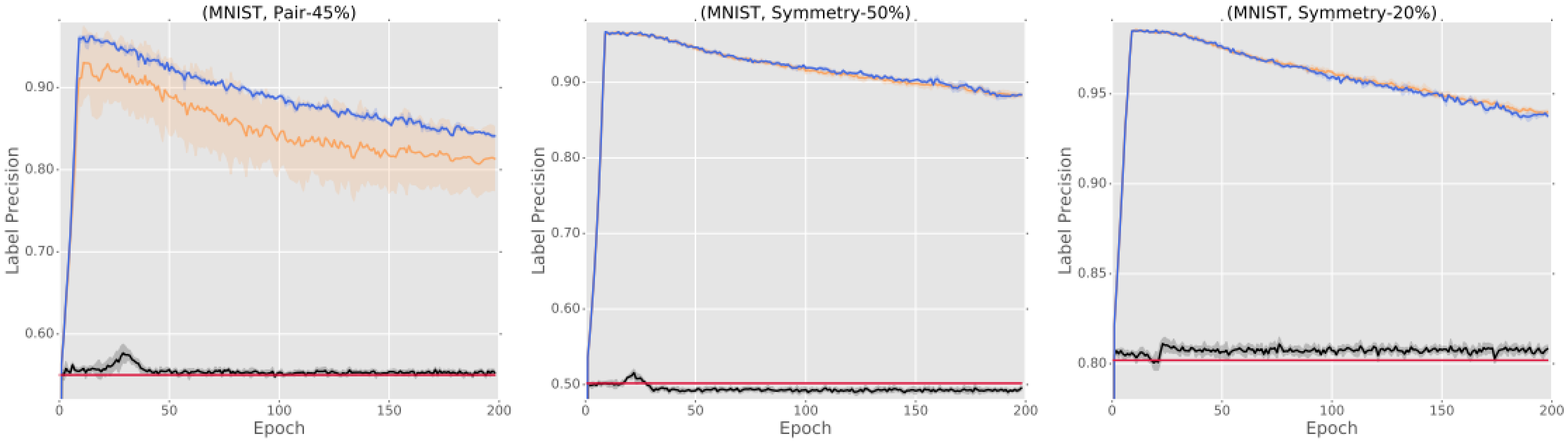
(a) Pair-45%.

(b) Symmetry-50%.

(c) Symmetry-20%.

Test accuracy vs number of epochs on MNIST dataset

Experiments – MNIST



(a) Pair-45%. (b) Symmetry-50%. (c) Symmetry-20%.

Label precision vs number of epochs on MNIST dataset.

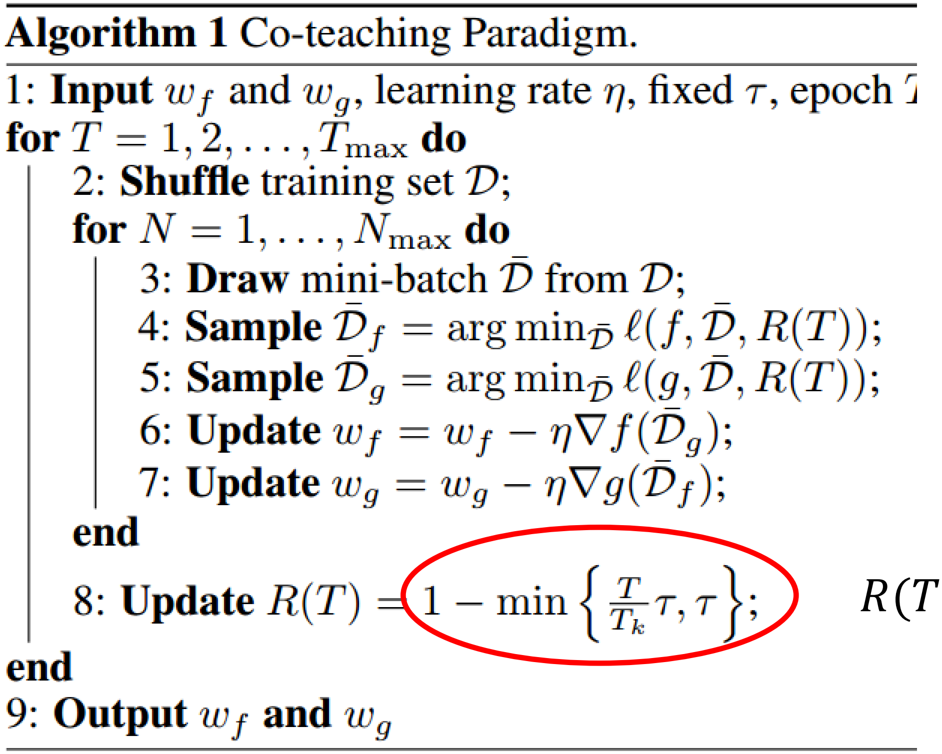
Experiments – $R(T)$

Impact of memorization

$$\boxed{R(T) = 1 - min\left\{\frac{T^c}{T_k}\tau, \tau\right\}}$$

Choices

- $c \in \{0.5, 1.0, 2\}$
- $T_k \in \{5, 10, 15\}$

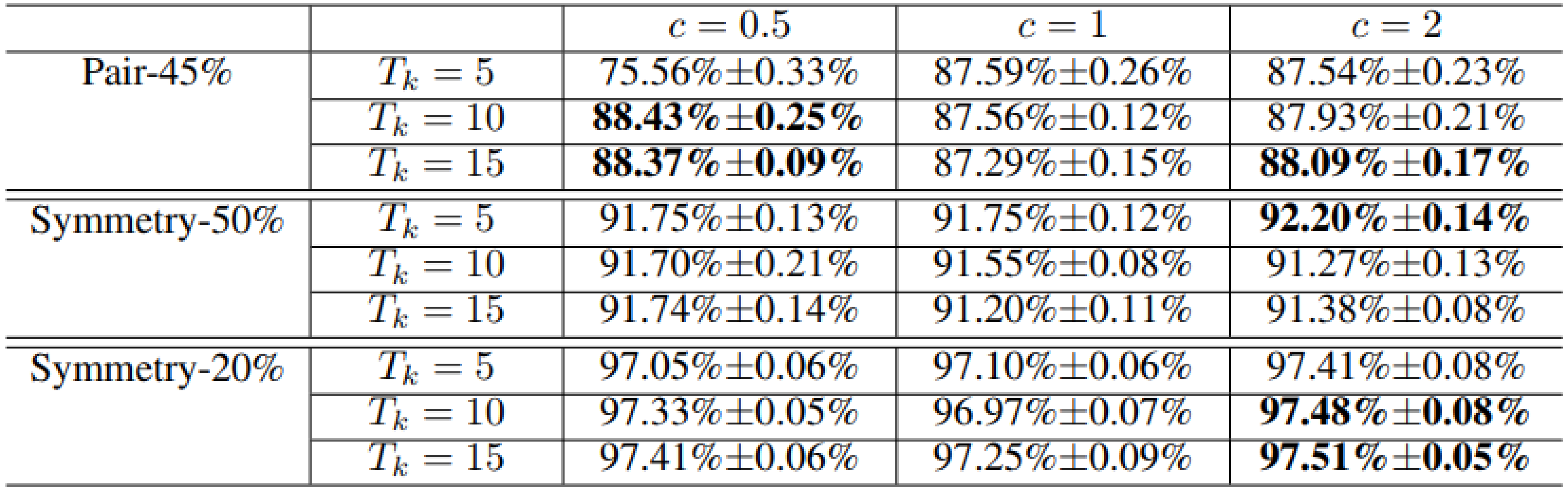


Algorithm 1 Co-teaching Paradigm.

```
1: Input $w_f$ and $w_g$, learning rate $\eta$, fixed $\tau$, epoch
for $T = 1, 2, \ldots, T_{\max}$ do
    2: Shuffle training set $\mathcal{D}$;
    for $N = 1, \ldots, N_{\max}$ do
        3: Draw mini-batch $\bar{\mathcal{D}}$ from $\mathcal{D}$;
        4: Sample $\bar{\mathcal{D}}_f = \arg\min_{\bar{\mathcal{D}}} \ell(f, \bar{\mathcal{D}}, R(T))$;
        5: Sample $\bar{\mathcal{D}}_g = \arg\min_{\bar{\mathcal{D}}} \ell(g, \bar{\mathcal{D}}, R(T))$;
        6: Update $w_f = w_f - \eta \nabla f(\bar{\mathcal{D}}_g)$;
        7: Update $w_g = w_g - \eta \nabla g(\bar{\mathcal{D}}_f)$;
    end
    8: Update $R(T) = 1 - \min\left\{\frac{T}{T_k}\tau, \tau\right\}$;
end
9: Output $w_f$ and $w_g$
```

$R(T)$: how fast drop

Experiments – $R(T)$



		$c = 0.5$	$c = 1$	$c = 2$
Pair-45%	$T_k = 5$	75.56%±0.33%	87.59%±0.26%	87.54%±0.23%
	$T_k = 10$	**88.43%±0.25%**	87.56%±0.12%	87.93%±0.21%
	$T_k = 15$	**88.37%±0.09%**	87.29%±0.15%	**88.09%±0.17%**
Symmetry-50%	$T_k = 5$	91.75%±0.13%	91.75%±0.12%	**92.20%±0.14%**
	$T_k = 10$	91.70%±0.21%	91.55%±0.08%	91.27%±0.13%
	$T_k = 15$	91.74%±0.14%	91.20%±0.11%	91.38%±0.08%
Symmetry-20%	$T_k = 5$	97.05%±0.06%	97.10%±0.06%	97.41%±0.08%
	$T_k = 10$	97.33%±0.05%	96.97%±0.07%	**97.48%±0.08%**
	$T_k = 15$	97.41%±0.06%	97.25%±0.09%	**97.51%±0.05%**

- $R(T)$ and τ can influence the performance
- However, their sensitive is not high, and they can be easily set
- In previous experiments, we set c = 1 and T_k = 10

Co-teaching – Variants

1. Utilize unlabeled data using semi-supervised learning
    - Li et al., ICLR 2020, Liu et al., NeurIPS 2020.
2. Stronger rule to select small-loss samples
    - Yu et al., ICML 2019, Arazo et al., ICML 2019, Y. Kim et al. CVPR 2019
3. Learn soft instead of hard weights for samples
    - J. Shu et at. NeurIPS 2019, J. Lu et al. ICML 2020

Outline

1. What is Automated Machine Learning (AutoML)?
2. Sample Selection for Learning with Noisy Labels (LNL)
    - What are Small-loss Samples
    - Co-teaching, its Variants and Limitations
    - Design Sample Selection Criterion by AutoML
3. Future Works & Summary

Search to Exploit Memorization Effect

- Key component to exploit memorization effect: *R(t)*
  - controls the percentage of small-loss samples
- Hard to set an appropriate R(t)
  - memorization effect is complex
  - depends on datasets, noise type, noise ratio, architecture, …
- We are encouraged to apply AutoML to this problem
  - “search” an appropriate R(t)

How?

Q. Yao et.al. Searching to Exploit Memorization Effect in Learning from Corrupted Labels. ICML 2020

Message on using AutoML

1. Search Space

2. Search Objective

3. Search Constraints

4. Training Objective

Bi-level optimization

1. Define an AutoML problem from insights in specific domains
2. Design a search algorithm reducing model training cost

$$\min_{\boxed{\lambda\in\mathcal{S}}} \boxed{M(F(w^*;\lambda),D_{\mathrm{val}})}$$

$$\text{s.t.}\begin{cases}\boxed{\min_{w} L(F(w;\lambda),D_{\mathrm{tra}})}\\ \boxed{G(\lambda)\le C}\end{cases}$$

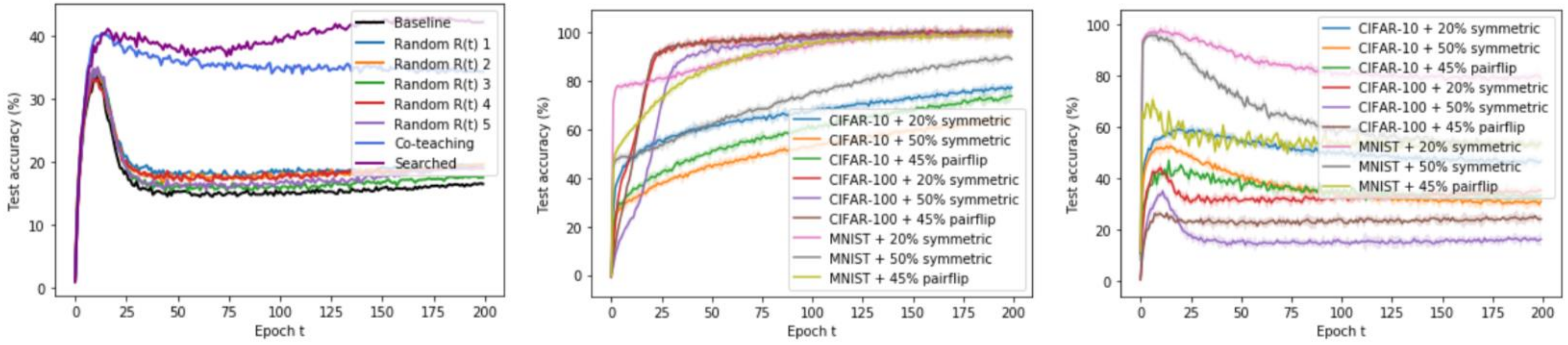
Search Space →

← ~~Search Objective~~

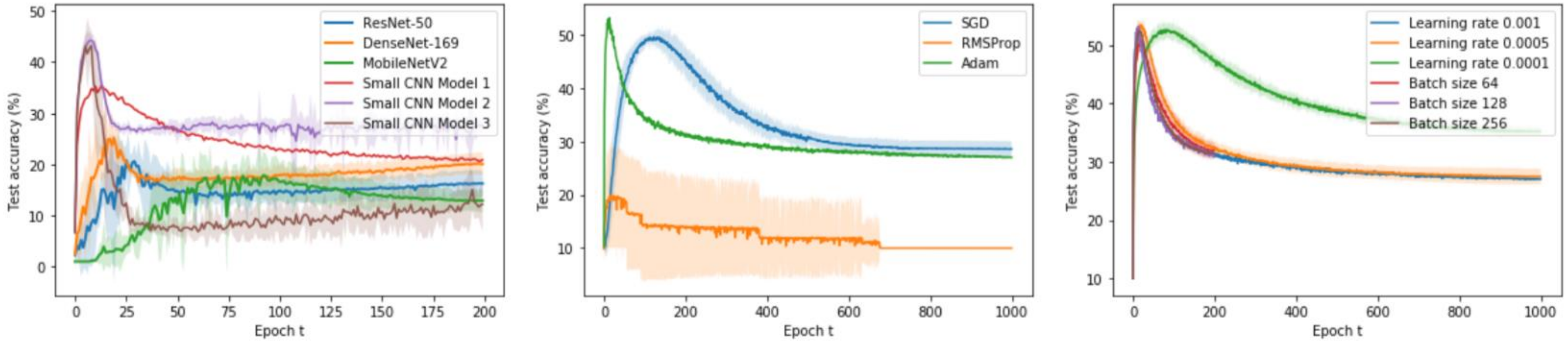
← Training Objective

← ~~Search Constraints~~

Revisit Memorization Effect



(a) Impact of $R(t)$. (b) Different data sets (training accuracy). (c) Different data sets (testing accuracy).



(d) Different architectures. (e) Different optimizers. (f) Different optimizer settings.

Figure 1. Training and testing accuracies on CIFAR-10, CIFAR-100, and MNIST using various architectures, optimizers, and optimizer settings. The detailed setup is in Appendix A.3.

Derive a Search Space

- During the initial phase when the learning curve rises, the deep network is plastic and can learn easy patterns from the data. In this phase, one can allow a larger $R(t)$ as there is little risk of memorization. Hence, at time $t = 0$, we can set $R(0) = 1$ and the entire noisy data set is used.
- As training proceeds and the learning curve has peaked, the network starts to memorize and overfit the noisy samples. Hence, $R(t)$ should then decrease.
- Finally, as the network gets less plastic and in case $R(t)$ drops too much at the beginning, it may be useful to allow $R(t)$ to slowly increase so as to enable learning some complex patterns.

Table 1: The four basis functions used to define the search space in the experiments. Here, a_i's are the hyperparameters.

$f_1(t; \boldsymbol{a})$	$e^{-a_2 t^{a_1}} + a_3 (\frac{t}{T})^{a_4}$
$f_2(t; \boldsymbol{a})$	$e^{-a_2 t^{a_1}} + a_3 \frac{\log(1+t^{a_4})}{\log(1+T^{a_4})}$
$f_3(t; \boldsymbol{a})$	$\frac{1}{(1+a_2 t)^{a_1}} + a_3 (\frac{t}{T})^{a_4}$
$f_4(t; \boldsymbol{a})$	$\frac{1}{(1+a_2 t)^{a_1}} + a_3 \frac{\log(1+t^{a_4})}{\log(1+T^{a_4})}$

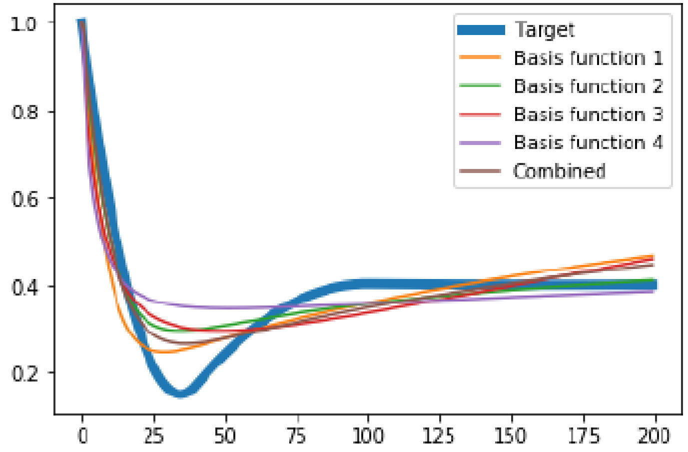


Figure 4: Plots of the basis functions in Table 1. An example $R(\cdot)$ to be learned is shown in blue.

Define an AutoML Problem

Bi-level objective

$$\bar{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}), \quad \text{s.t. } \bar{\boldsymbol{w}}(R_{\boldsymbol{x}}) = \arg\min\nolimits_{\boldsymbol{w}} \mathcal{L}_{\text{tr}}(\boldsymbol{w}, R_{\boldsymbol{x}}),$$

where

Search objective: $\mathcal{J}(\boldsymbol{\theta}) \equiv \mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{\theta}}(\boldsymbol{x})}[\mathcal{L}_{\text{val}}(\bar{\boldsymbol{w}}(R_{\boldsymbol{x}}))] = \int_{\boldsymbol{x}\in\mathcal{S}} \mathcal{L}_{\text{val}}(\bar{\boldsymbol{w}}(R_{\boldsymbol{x}}))p_{\boldsymbol{\theta}}(\boldsymbol{x})\, d\boldsymbol{x},$

- $R(t)$ is complexly coupled with training process gradient w.r.t. $R(t)$ is hard to obtain
- Stochastic relaxation is used gradient is taken w.r.t θ instead of $R(t)$

Search space: $R(t) \equiv \sum_{i=1}^{k} \alpha_i \cdot f^i(t;\boldsymbol{\beta}^i) : \{\boldsymbol{\alpha}, \{\boldsymbol{\beta}^i\}\} \in \mathcal{S},$

- $R(t)$ is derived based on memorization effect

Derive a Search Algorithm

The general idea is to introduce Hessian matrix to solve stochastic bi-level objective

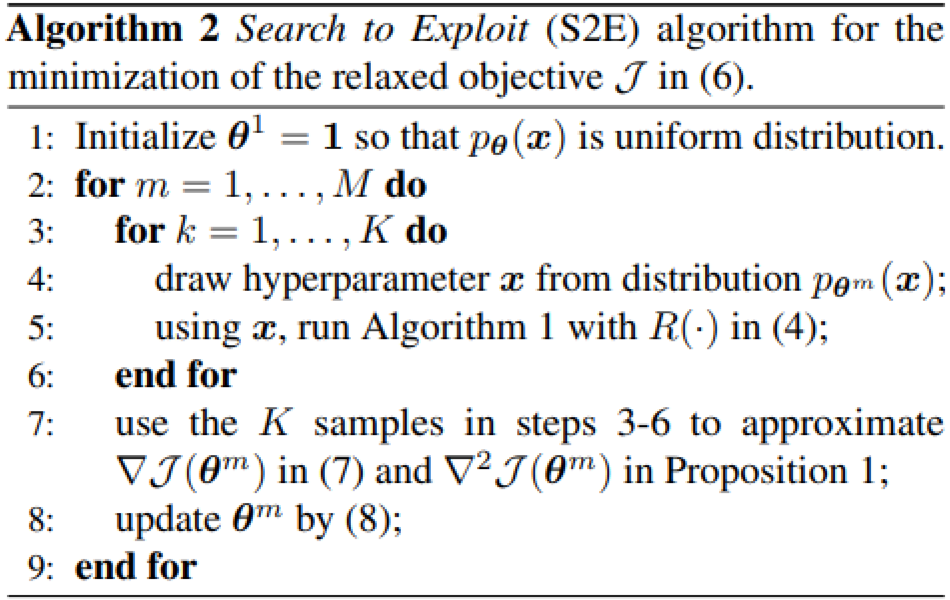
- Faster convergence → reduce the number of updates on θ → less time on model training

$$\bar{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}), \quad \text{s.t. } \bar{\boldsymbol{w}}(R_{\boldsymbol{x}}) = \arg\min\nolimits_{\boldsymbol{w}} \mathcal{L}_{\text{tr}}(\boldsymbol{w}, R_{\boldsymbol{x}}),$$

Gradient $\nabla \mathcal{J}(\boldsymbol{\theta}) = \int_{\boldsymbol{x} \in \mathcal{S}} \bar{f}(\boldsymbol{x}) \nabla p_{\boldsymbol{\theta}}(\boldsymbol{x}) d\boldsymbol{x}$

Hessian $\boldsymbol{H}(\boldsymbol{\theta}; \boldsymbol{x}) = \bar{f}(\boldsymbol{x})(\nabla^2 \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) + \nabla \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) \nabla \log p_{\boldsymbol{\theta}}(\boldsymbol{x})^\top)$

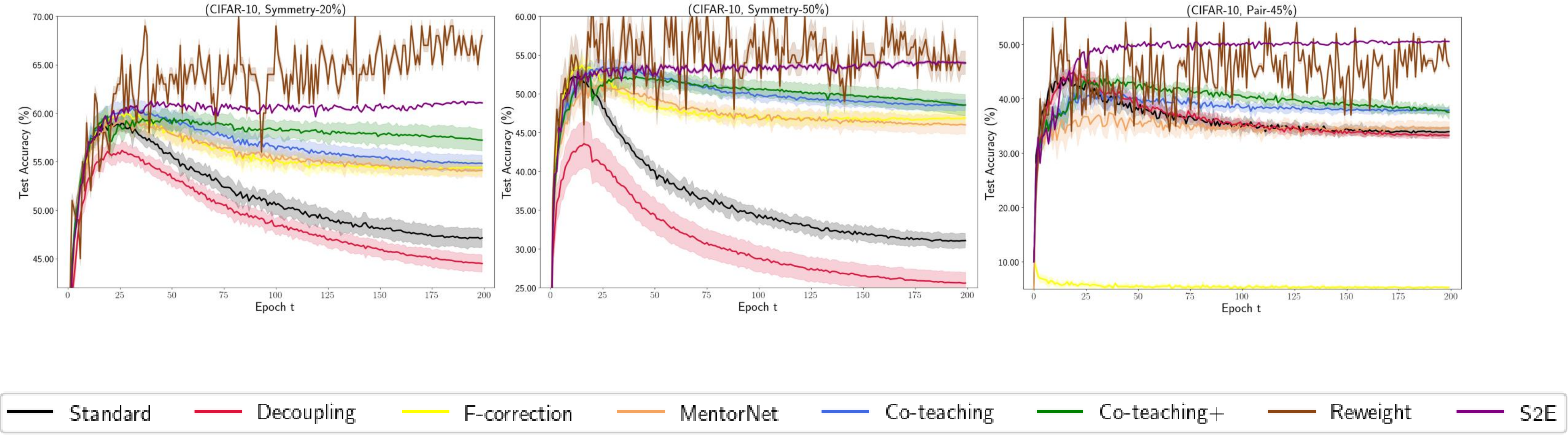
Can be faster than first-order method in AutoML



Algorithm 2 *Search to Exploit* (S2E) algorithm for the minimization of the relaxed objective $\mathcal{J}$ in (6).

```
1: Initialize $\boldsymbol{\theta}^1 = \mathbf{1}$ so that $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ is uniform distribution.
2: for $m = 1, \dots, M$ do
3:     for $k = 1, \dots, K$ do
4:         draw hyperparameter $\boldsymbol{x}$ from distribution $p_{\boldsymbol{\theta}^m}(\boldsymbol{x})$;
5:         using $\boldsymbol{x}$, run Algorithm 1 with $R(\cdot)$ in (4);
6:     end for
7:     use the $K$ samples in steps 3-6 to approximate $\nabla \mathcal{J}(\boldsymbol{\theta}^m)$ in (7) and $\nabla^2 \mathcal{J}(\boldsymbol{\theta}^m)$ in Proposition 1;
8:     update $\boldsymbol{\theta}^m$ by (8);
9: end for
```

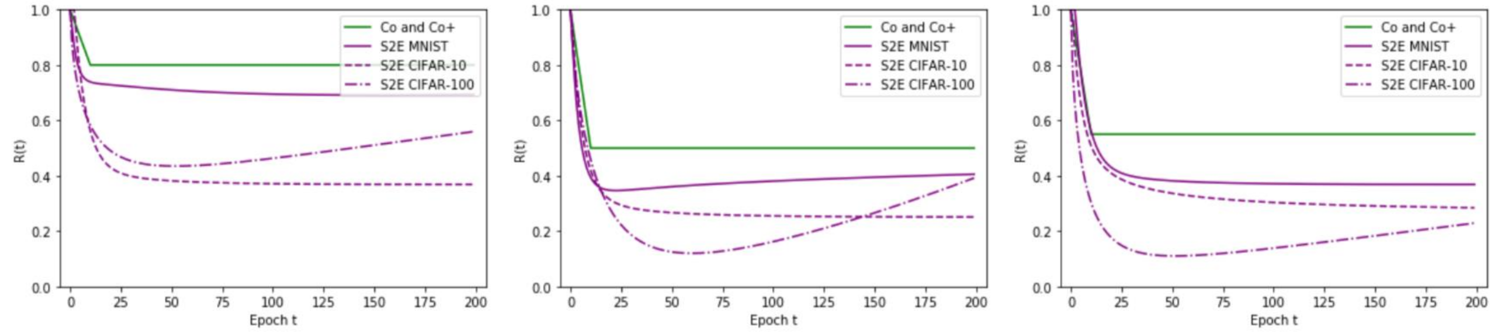
Experiments – Overall performance



CIFAR-10, same setup as Co-teaching

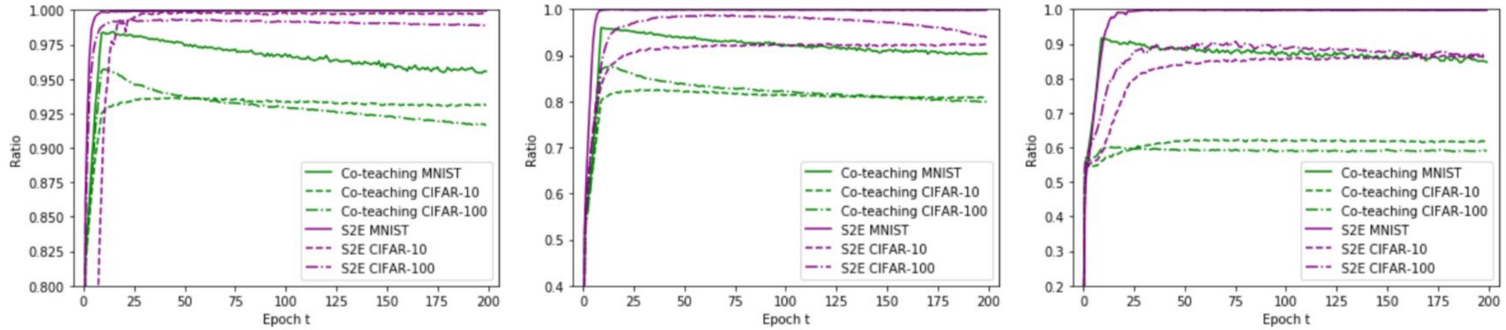
Experiments – Searched R(t)

- Our searched R(t)
  - more flexible
  - cleaner training set



(a) symmetry flipping (20%). (b) symmetry flipping (50%). (c) pair flipping (45%).

Figure 4. $R(\cdot)$ obtained by the sample selection methods. Note that *MentorNet* (*MN*), *Co-teaching* (*Co*) and *Co-teaching+* (*Co+*) all use the same $R(t)$.

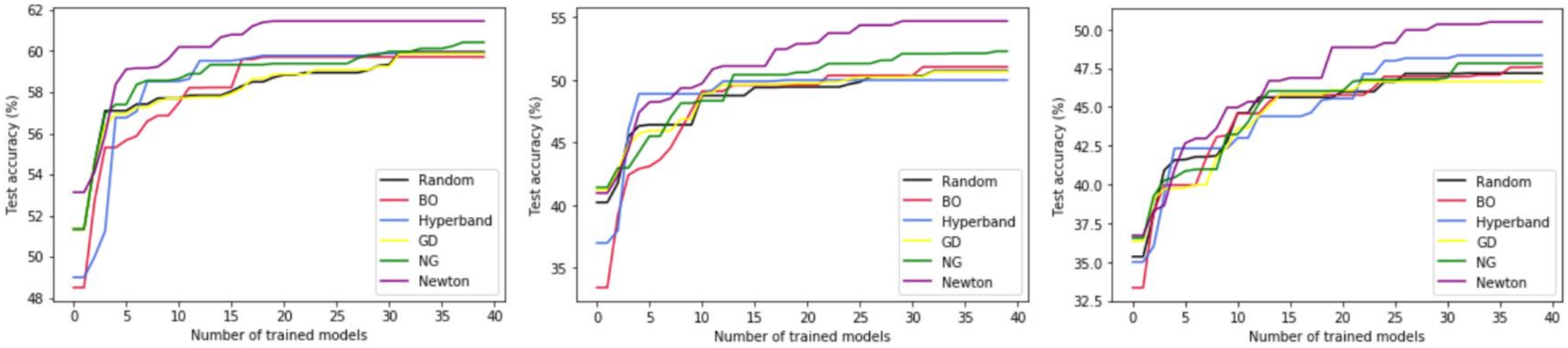


(a) symmetry flipping (20%). (b) symmetry flipping (50%). (c) pair flipping (45%).

Figure 5. Label precision of *S2E* and *Co-teaching*.

Experiments – Search Algorithm

- Search algorithm:
  - much more efficient



(a) symmetry flipping (20%).

(b) symmetry flipping (50%).

(c) pair flipping (45%).

Figure 6. Search efficiency of *S2E* and the other search algorithms.

Sample Selection for NNL – Short Summary

- Noisy label learning problem is important
- Small-loss based method is popular and empirical work well
    - Co-teaching is an exemplar work with many variants
    - Design sample selection rule is hard
- AutoML is a promising way to design sample selection rule
    - Good search space relies on memorization effect
    - Reduce model training times is important to reduce search cost

Outline

1. What is Automated Machine Learning (AutoML)?
2. Sample Selection for Learning with Noisy Labels (LNL)
3. Future Works & Summary

Future Works & Summary

AutoML is a meta-approach to

- improve learning performance
- understand domain information at a higher level

Your next work can be on “what else can be searched in NNL”.

- Robust loss functions is an example

Seek more opportunities from other tutor’s slides!

- Take S2E as an example.

Thanks!