ACML 2021
The 13th Asian Conference on Machine Learning
清華大學电子工程系
Department of Electronic Engineering, Tsinghua University

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

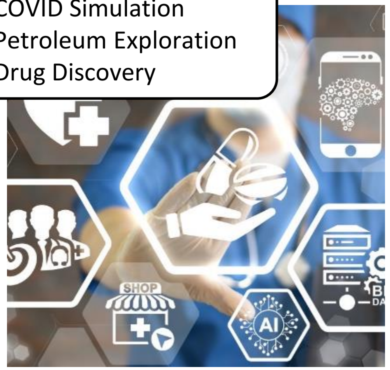
Predict the class of the object

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 ↑

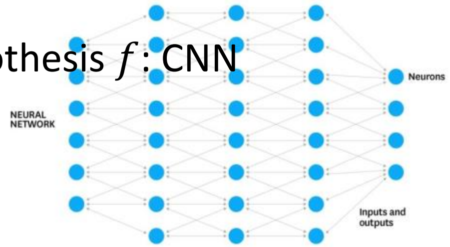
[1]. Machine Learning, Tom Mitchell, McGraw Hill, 1997.
[2]. 周志华 著. 机器学习, 北京: 清华大学出版社, 2016年

ML = Data + Knowledge

Image Classification

Optimization

Hypothesis f: CNN



Generalization

Accuracy

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

Target	Concept	Issue
Better Performance (more important)	Generalization	What kind of f should we use?
Higher Efficiency	Optimization	How can we find such f?

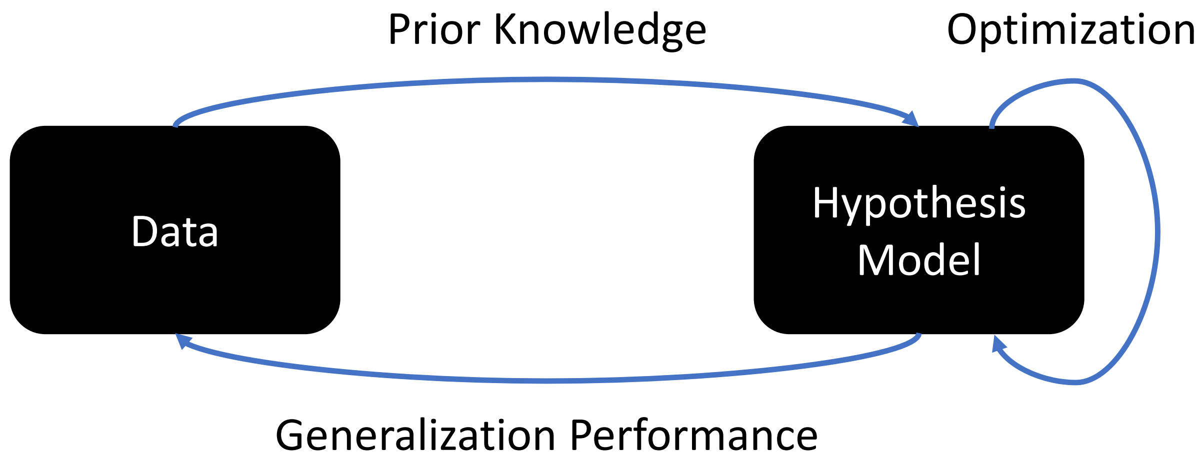
Not everything can be learnt

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?

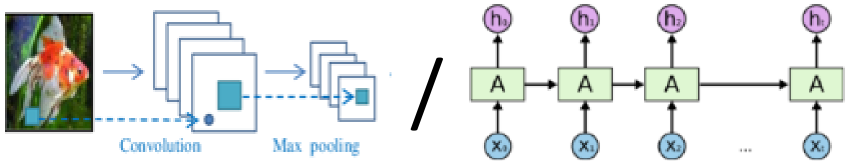


The Advancement of Learning

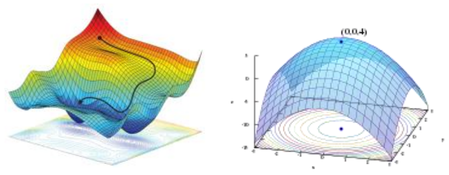
- An iteration between theory and practice

- 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

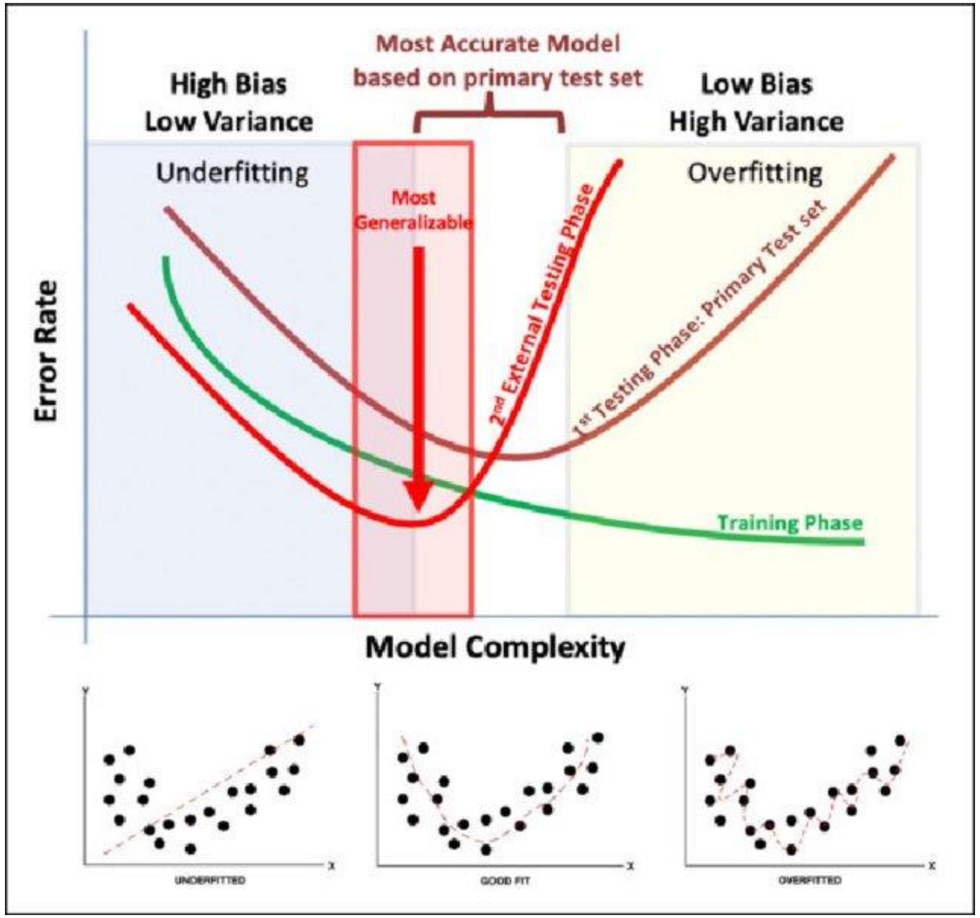
1. What is Automated Machine Learning (AutoML)?
   - What is Machine Learning?
   - What is Automated Machine Learning (AutoML)?
   - How to Use AutoML Techniques
2. Sample Selection for Learning with Noisy Labels (LNL)
3. Future Works & Summary

Simple Example – Tune hyper-parameter

Bi-level optimization

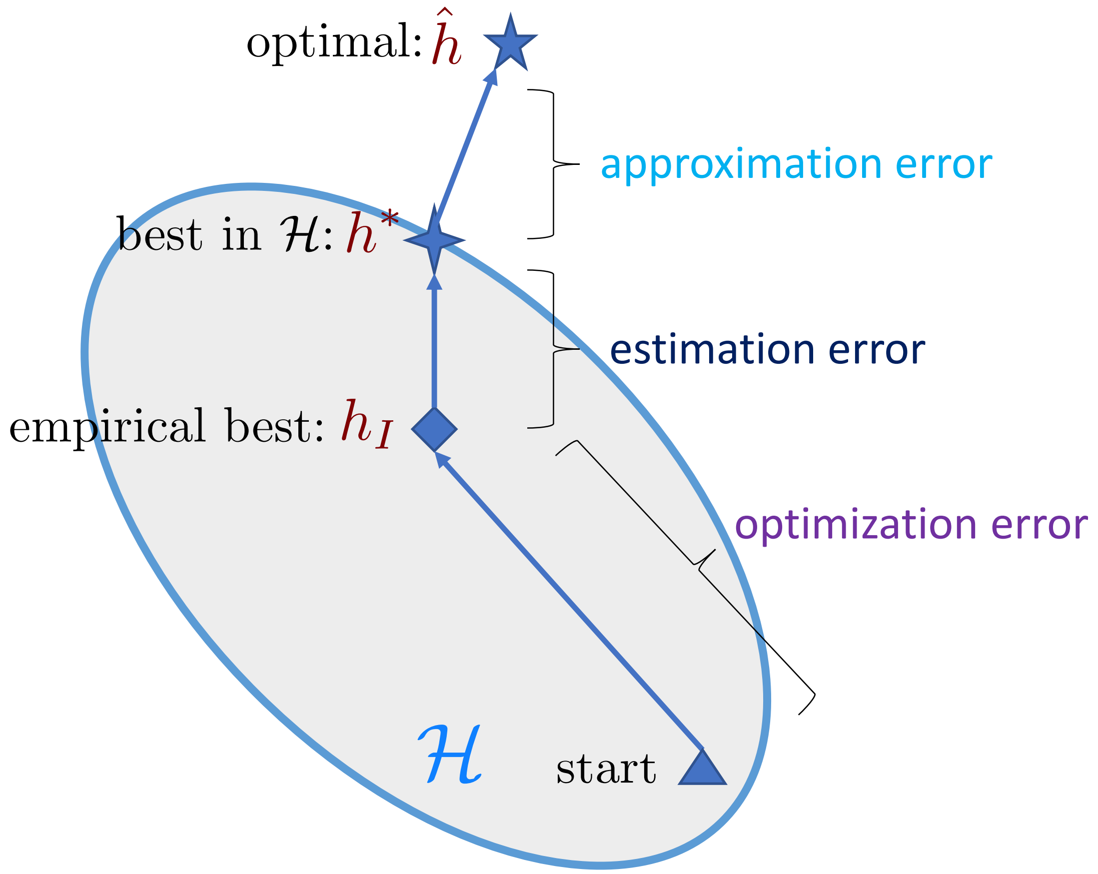
$$\underbrace{\max_{\lambda} \sum_{j} h(x_j; w^*)}_{\text{Validation Performance}} \quad \text{s.t.} \quad \underbrace{w^* = \min_{w} \sum_{i} f(x_i; w) + \overset{\text{Hyper-parameter}}{\lambda} \|w\|_1}_{\text{Training objective}}$$

- 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

Reduce

$$\min_w \sum_i f(x_i; w) + \lambda \|w\|_1$$

- Optimization error
  - Which algorithm to be used
  - How to tune its step-size

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

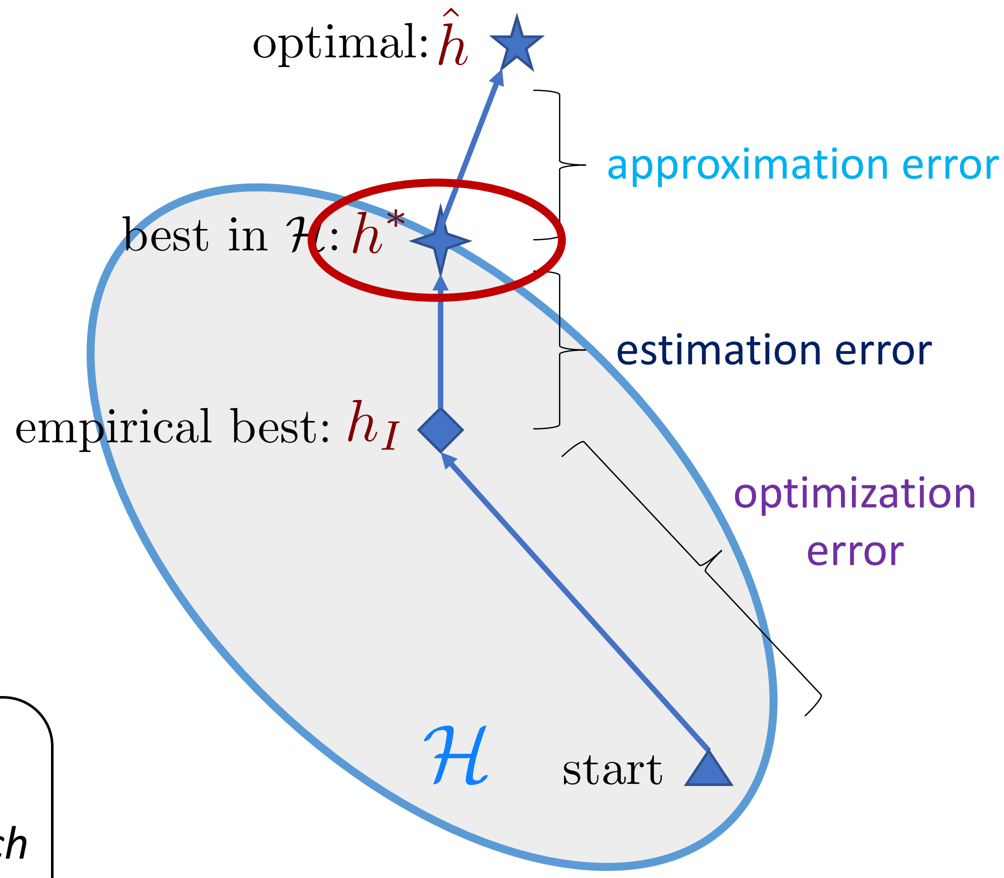
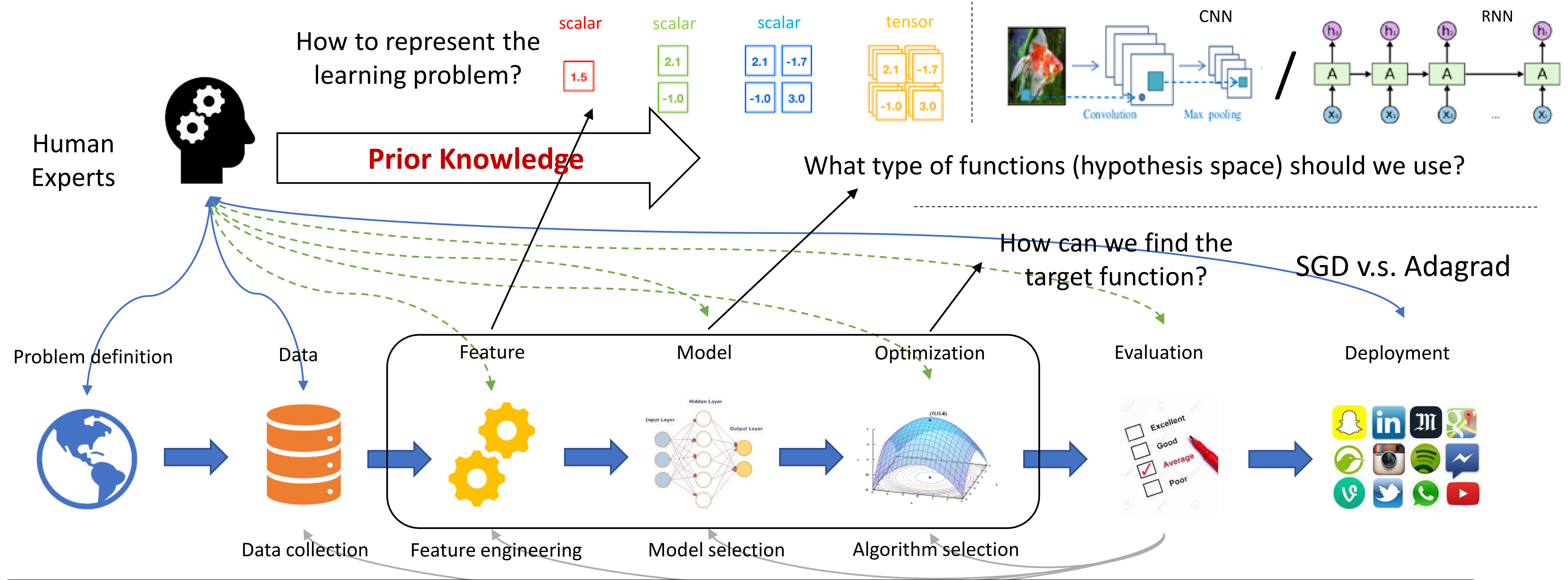


Figure is from Q. Yao et.al. Taking Human out of Learning Applications: A Survey on Automated Machine Learning. arXiv 2018

What is AutoML – Practical Viewpoint



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)

清华大学电子工程系
Department of Electronic Engineering, Tsinghua University

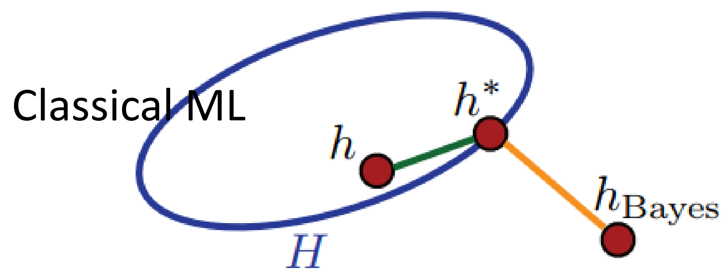
What is AutoML – Generalization Viewpoint

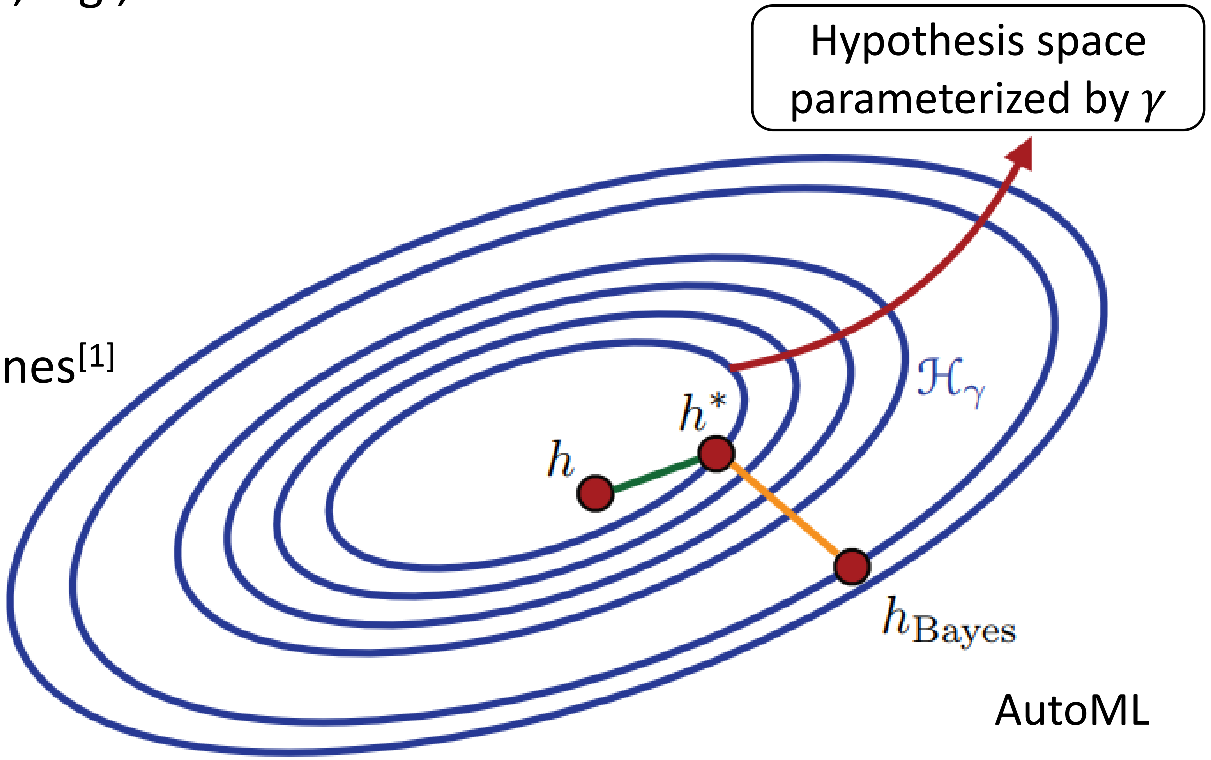
Parameterized the prior knowledge of learning methods, e.g.,

- minimize the total error
- reduce parameter numbers

Perform efficient search in the designed (new) space

- combinatorial generalize new models from existing ones[1]

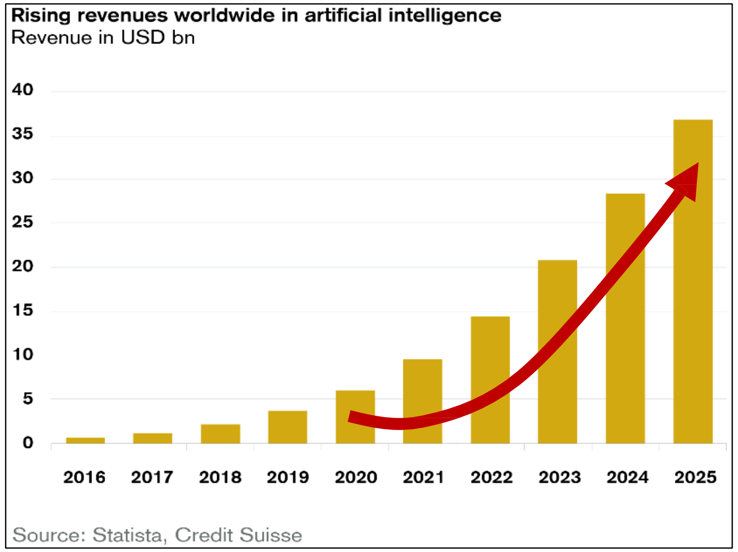




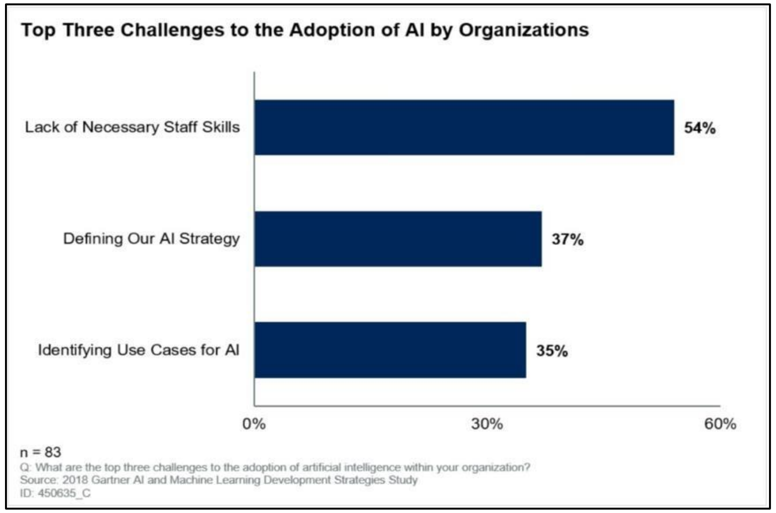
Parameterize (low-level) prior knowledge in the usage and design of machine learning

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

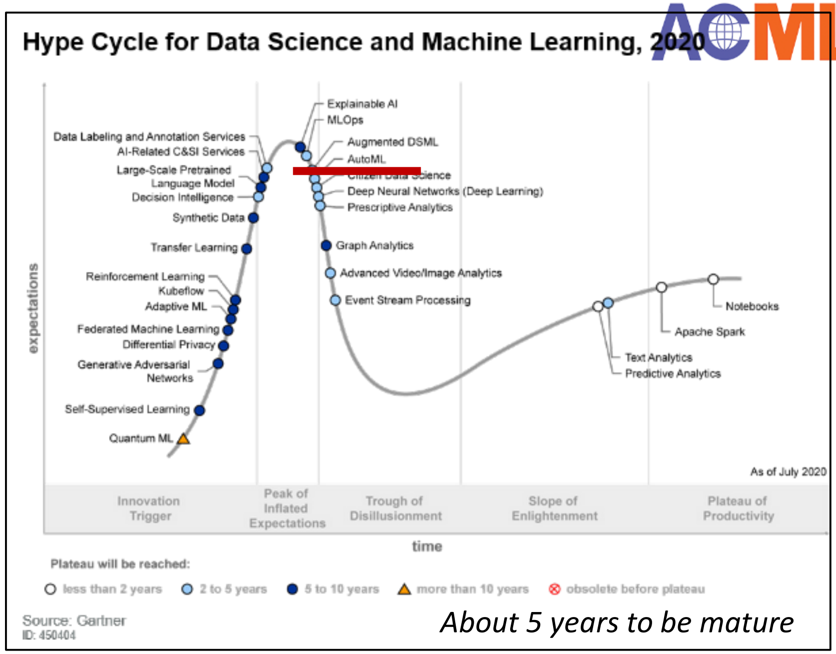
Why We need AutoML?



Investment in AI industry



Practical needs



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. 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

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

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

Search Space → $\lambda \in \mathcal{S}$

Search Objective → $M(F(w^*;\lambda), D_{\text{val}})$

Training Objective → $\min_{w} L(F(w;\lambda), D_{\text{tra}})$

Search Constraints → $G(\lambda) \le C$

2. Design or select proper search algorithm

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

What is AutoML – Short Summary

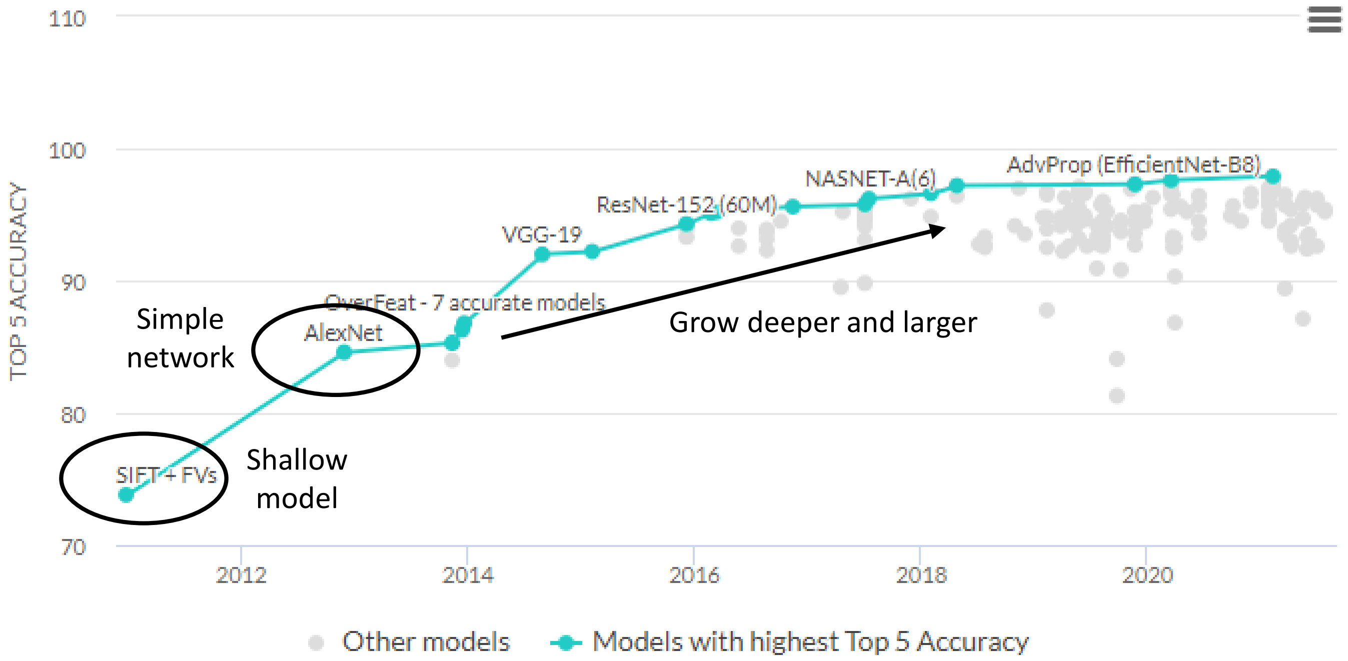
- Exploring prior knowledge is important in machine learning
  - Cost time and critical to generalization performance
- AutoML attempts to parameterize low-level prior knowledge
  - Human participations can be naturally replaced by computation power
  - total error can be reduced (generalization can 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



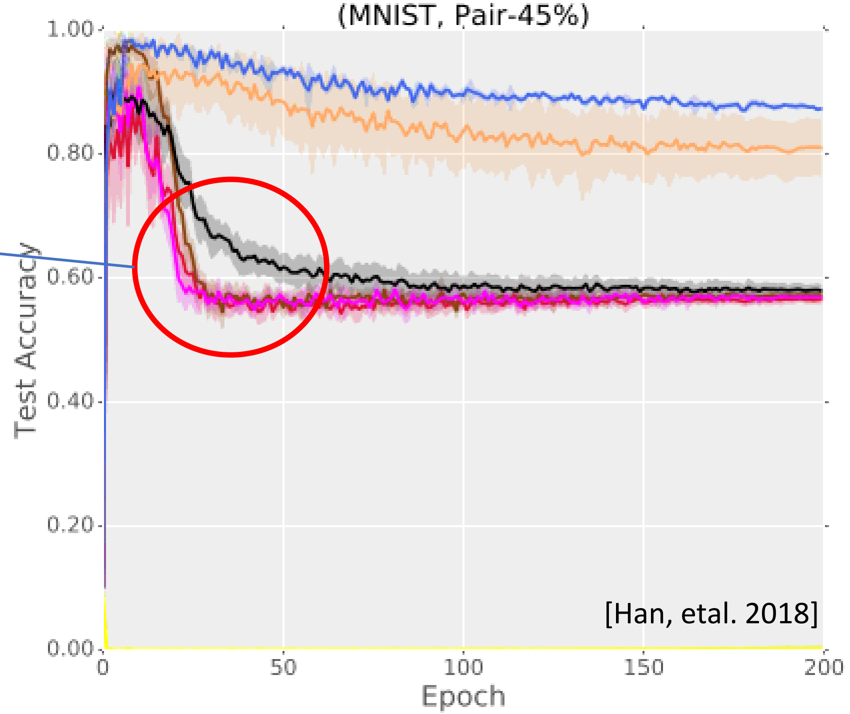
Big & High-quality data is the fuel

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

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.

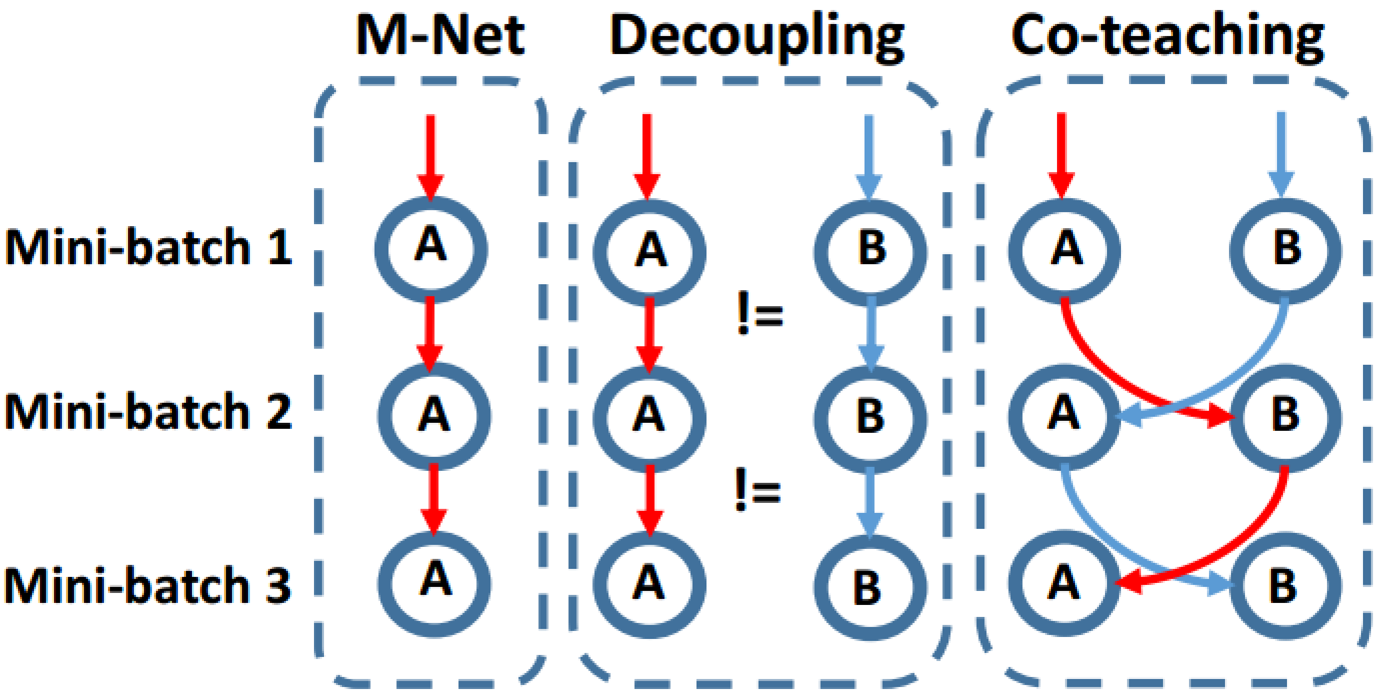
C. Zhang et.al. Understanding deep learning requires rethinking generalization. ICLR 2017
D Arpit et.al. A closer look at memorization in deep networks. NIPS 2017

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

Co-teaching – Core idea

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 η, fixed τ, 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

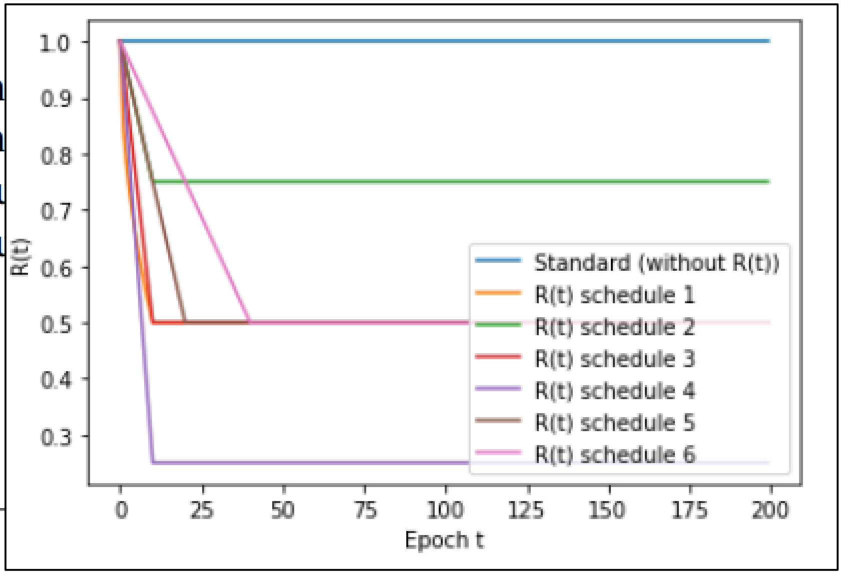
Co-teaching – Selection rule

Algorithm 1 Co-teaching Paradigm.

1: **Input** w_f and w_g, learning rate η, fixed τ, 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** $R(T) = 1 - \min\left\{\frac{T}{T_k}\tau, \tau\right\}$;

How many samples to be kept

end
9: **Output** w_f **and** w_g

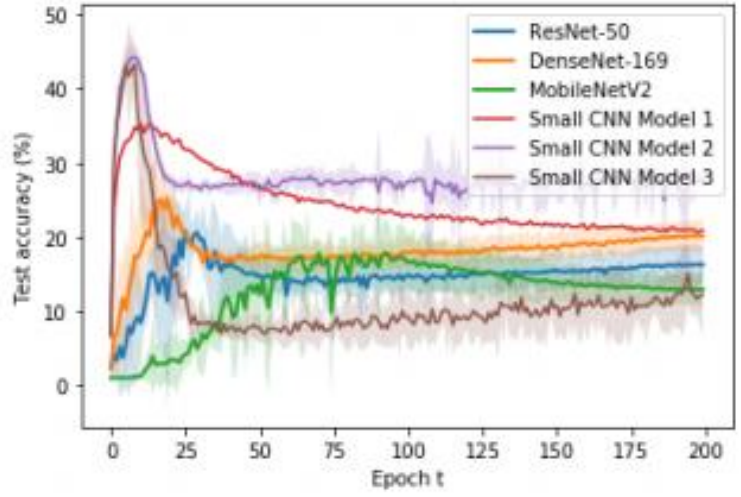


$$R(t) = 1 - \tau \cdot \min\left((t/t_k)^c, 1\right),$$

Co-teaching – Selection rule

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.



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

清华大学电子工程系
Department of Electronic Engineering, Tsinghua University

ACML

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

Some materials are still under construction of the journal version.

https://github.com/AutoML-Research/S2E

Message on using AutoML

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

Search Space

~~Search Objective~~

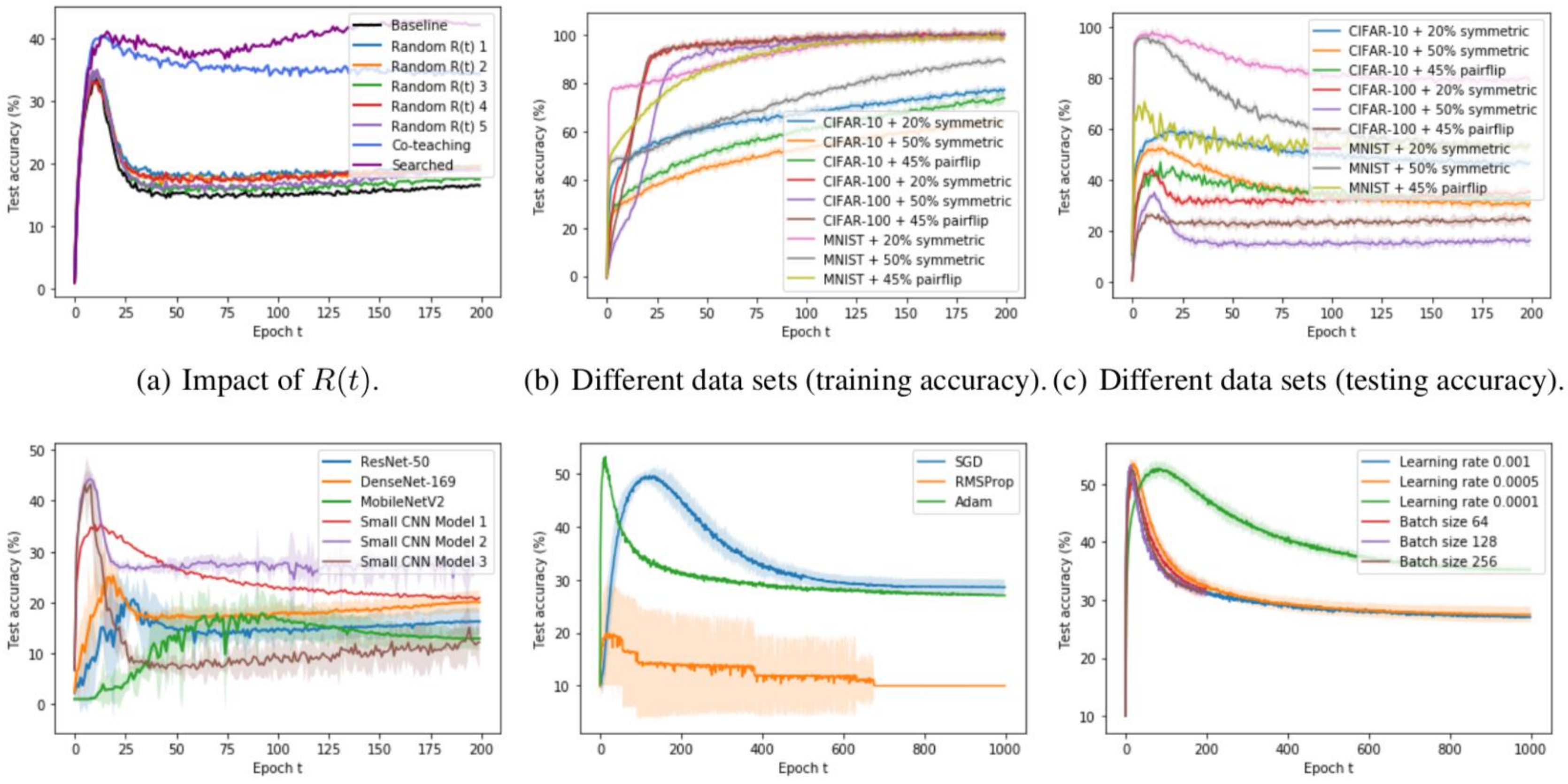
$$\min_{\boxed{\lambda\in\mathcal{S}}} \boxed{M(F(w^*;\lambda), D_{\mathrm{val}})}$$

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

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.

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

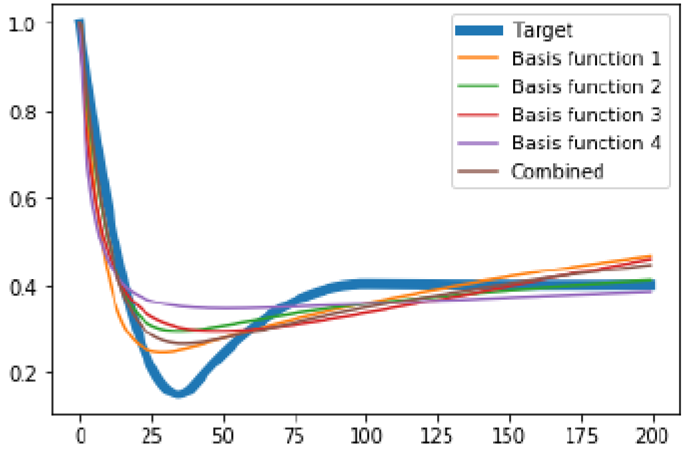


Figure 4: Plots of the basis functions in Table 1. An example $R(\cdot)$ to be learned is shown in blue.

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

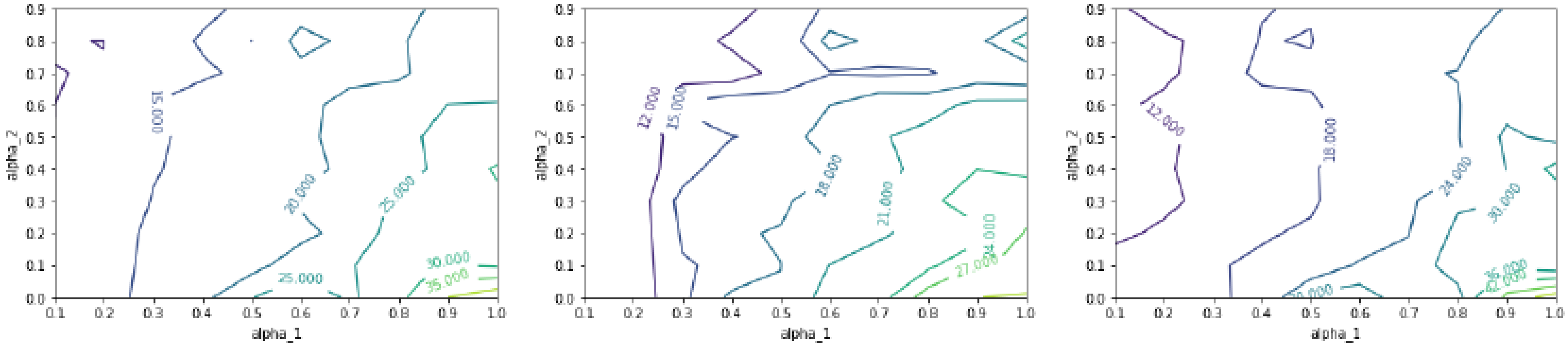
Search objective: $$\mathcal{J}(\boldsymbol{\theta}) \equiv \mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{\theta}}(\boldsymbol{x})}[\mathcal{L}_{\text{val}}(\bar{\boldsymbol{w}}(R_{\boldsymbol{x}}))] = \int_{\boldsymbol{x}\in\mathcal{S}} \mathcal{L}_{\text{val}}(\bar{\boldsymbol{w}}(R_{\boldsymbol{x}}))p_{\boldsymbol{\theta}}(\boldsymbol{x})\,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

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

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

Visualization of Validation Surface



(a) CIFAR-10 symmetric 50%. (b) CIFAR-10 pair flipping 45%. (c) CIFAR-100 pair flipping 45%.

- under different datasets, noise ratios and noise types, the landscapes of validation accuracy of these different models are all very complex.
- it contains bad local optimums (in the middle of figure), which has much worse performance than the actual optimal (in the right-down corner)

Derive a Search Algorithm

The general idea is to introduce Hessian matrix / cubic regularization 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** $m = 1, \ldots, M$ **do**
3:     **for** $k = 1, \ldots, K$ **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

Table 4: Testing accuracy (in %) on CIFAR-10. The term “early” means highest testing accuracy, and “average” means the averaged performance over the last ten epochs.

noise	symmetric 20% early	symmetric 20% average	symmetric 35% early	symmetric 35% average	symmetric 50% early	symmetric 50% average
Standard	59.18±0.58	47.12±0.05	55.55±0.85	37.86±0.03	52.23±1.32	32.75±0.07
MentorNet	59.74±0.88	54.36±0.05	55.13±0.47	49.47±0.05	51.08±1.06	46.98±0.07
Co-teaching	60.88±1.01	55.06±0.03	56.86±0.87	50.95±0.02	53.48±0.86	50.24±0.14
Co-teaching+	59.59±1.03	57.08±0.06	52.68±1.21	50.43±0.08	52.49±1.52	50.74±0.11
JoCoR	56.67±1.25	56.02±0.05	53.92±1.96	53.86±0.04	50.04±2.29	49.53±0.03
PRL	60.01±0.70	54.30±0.14	**57.55±0.79**	52.34±0.15	53.41±0.56	48.48±0.13
S2E	59.70±1.04	59.36±0.04	54.64±0.81	51.22±0.04	53.46±1.11	53.06±0.08
S2E (Cubic)	**61.27±1.07**	**61.09±0.08**	57.11±0.74	**54.75±0.05**	**54.30±1.21**	**54.05±0.12**
noise	**pairflip 25% early**	**pairflip 25% average**	**pairflip 35% early**	**pairflip 35% average**	**pairflip 45% early**	**pairflip 45% average**
Standard	57.44±1.22	43.11±0.03	53.28±1.07	37.86±0.03	44.01±1.49	33.74±0.06
MentorNet	54.23±1.27	47.13±0.07	48.23±1.55	41.63±0.05	37.45±2.45	34.49±0.07
Co-teaching	56.44±0.95	49.84±0.05	51.11±0.77	44.66±0.03	41.26±0.74	38.11±0.04
Co-teaching+	53.51±0.99	51.46±0.10	47.27±0.29	44.20±0.11	43.66±1.28	37.89±0.25
JoCoR	57.39±1.04	56.93±0.05	51.21±1.28	49.52±0.06	40.68±1.41	38.10±0.16
PRL	**59.63±0.89**	53.56±0.16	**56.69±0.79**	50.89±0.11	48.43±1.01	43.50±0.15
S2E	57.22±0.64	57.19±0.02	50.58±0.88	50.42±0.05	46.35±1.03	46.21±0.05
S2E (Cubic)	57.86±0.52	**57.66±0.05**	54.79±0.31	**54.71±0.05**	**49.62±1.14**	**49.39±0.11**

Compared methods

(i) MentorNet (Jiang et al., 2018)

(ii) Co-teaching (Han et al., 2018)

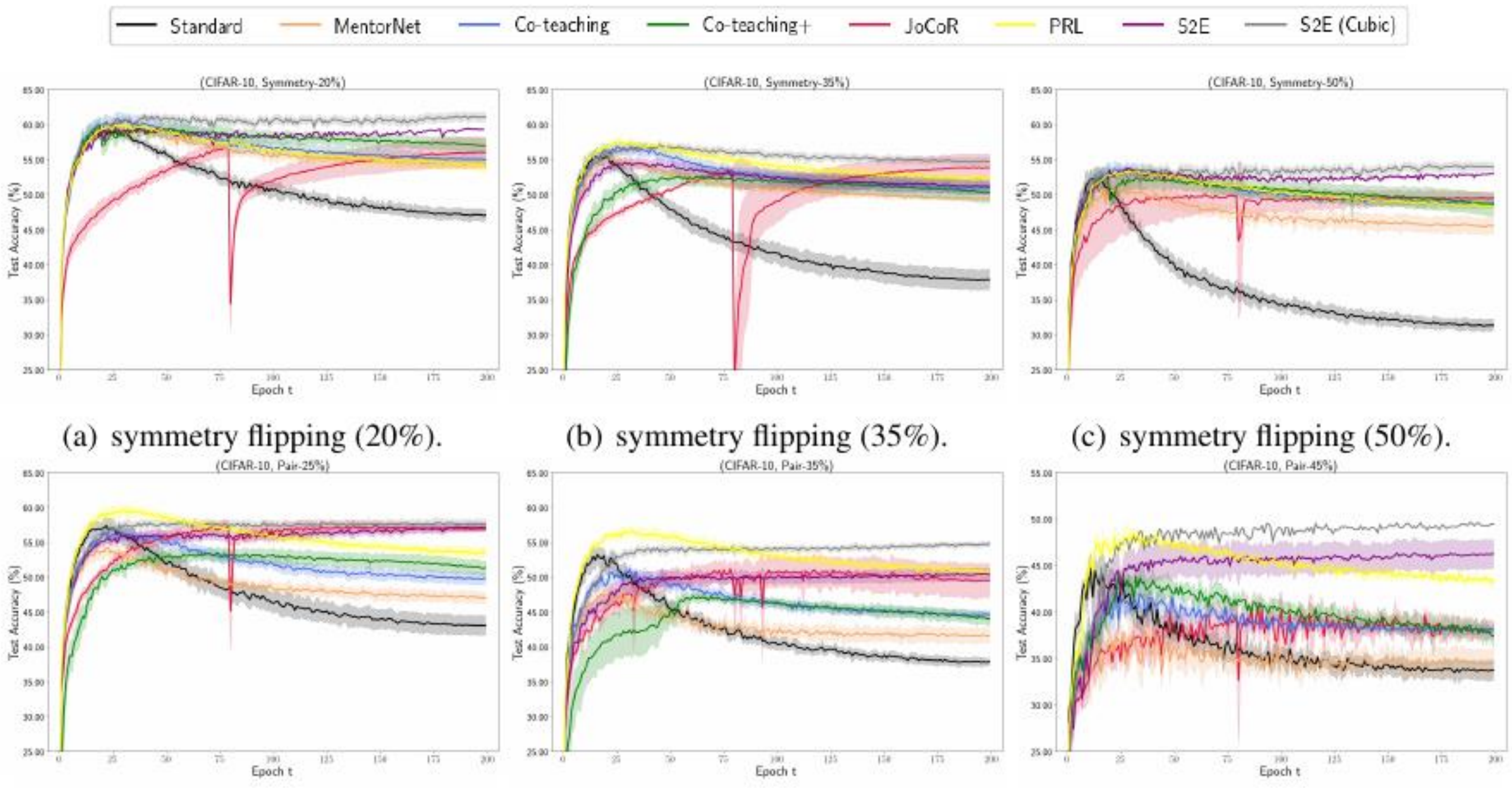
(iii) Co-teaching+ (Yu et al., 2019)

(iv) JoCoR (Wei et al., 2020); and

(v) PRL (Liu et al., 2021).

Combine other techniques with sample selection.

Experiments – Overall performance



(a) symmetry flipping (20%). (b) symmetry flipping (35%). (c) symmetry flipping (50%).

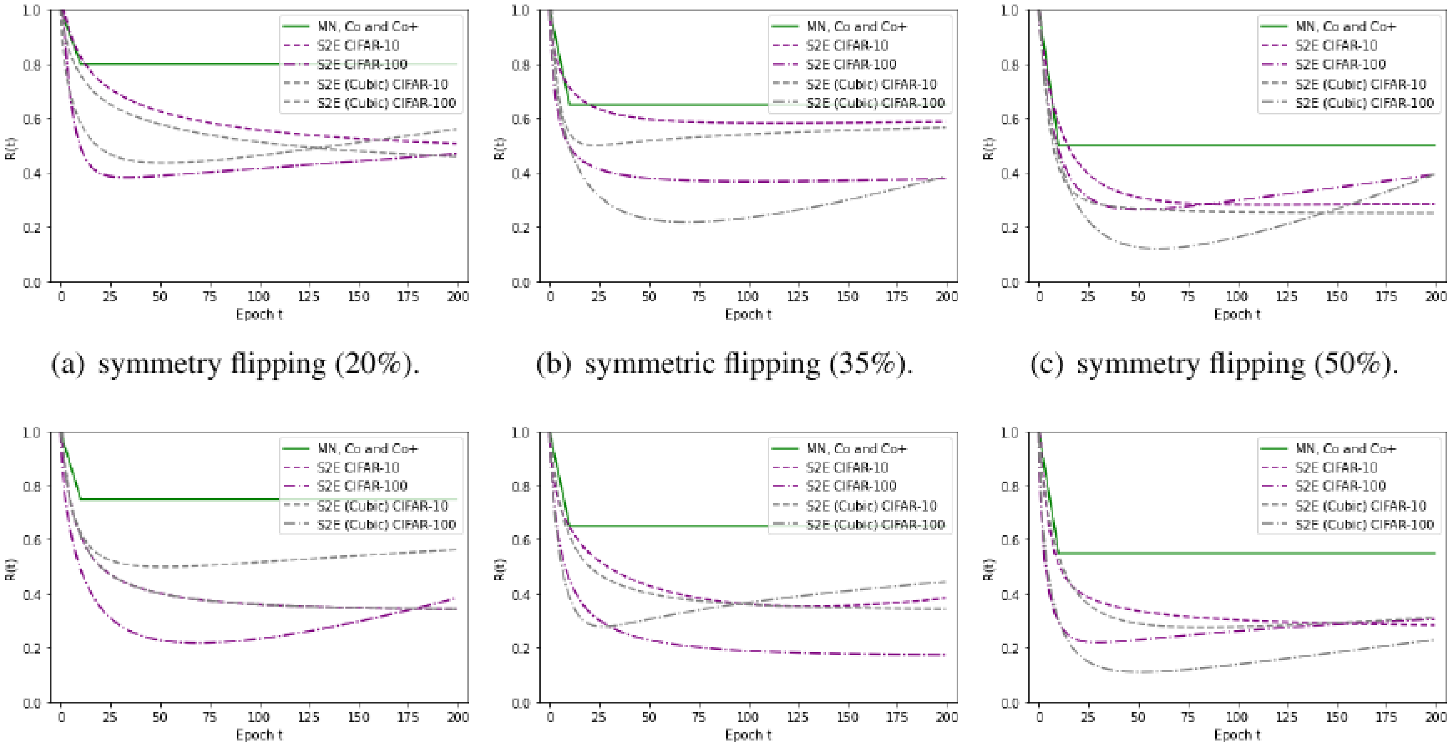
(d) pair flipping (25%). (e) pair flipping (35%). (f) pair flipping (45%).

Demonstrate the huge potential of the small loss criteria that may be overlooked by simply using predefined schedules.

Experiments – Searched R(t)

Our searched R(t)

- more flexible



(a) symmetry flipping (20%). (b) symmetric flipping (35%). (c) symmetry flipping (50%).

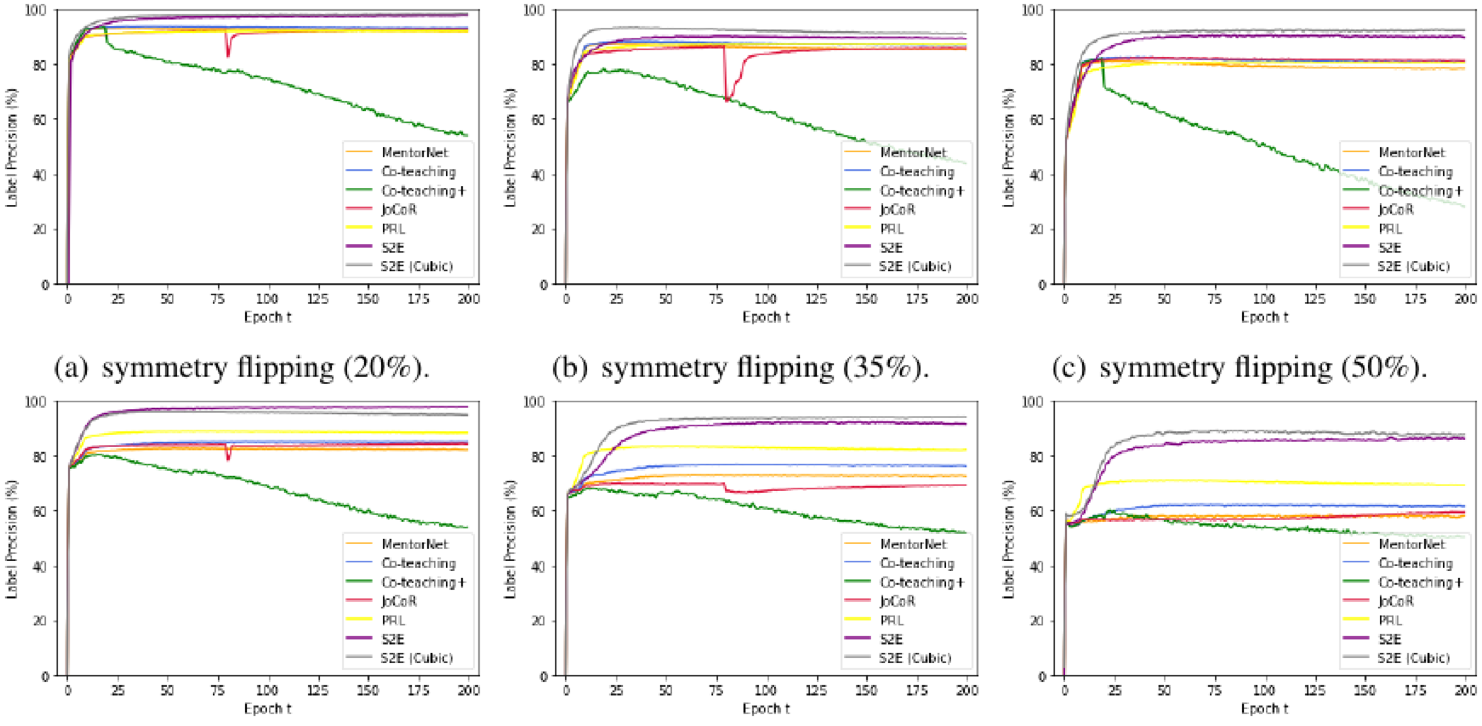
(d) pair flipping (25%). (e) pair flipping (35%). (f) pair flipping (45%).

Figure 12: $R(\cdot)$ obtained by S2E and S2E (Cubic). We also include the $R(t)$ used in*MentorNet* (*MN*), *Co-teaching* (*Co*) and *Co-teaching+* (*Co+*) for comparison.

Experiments – Label precision

Our searched R(t)

- cleaner training set



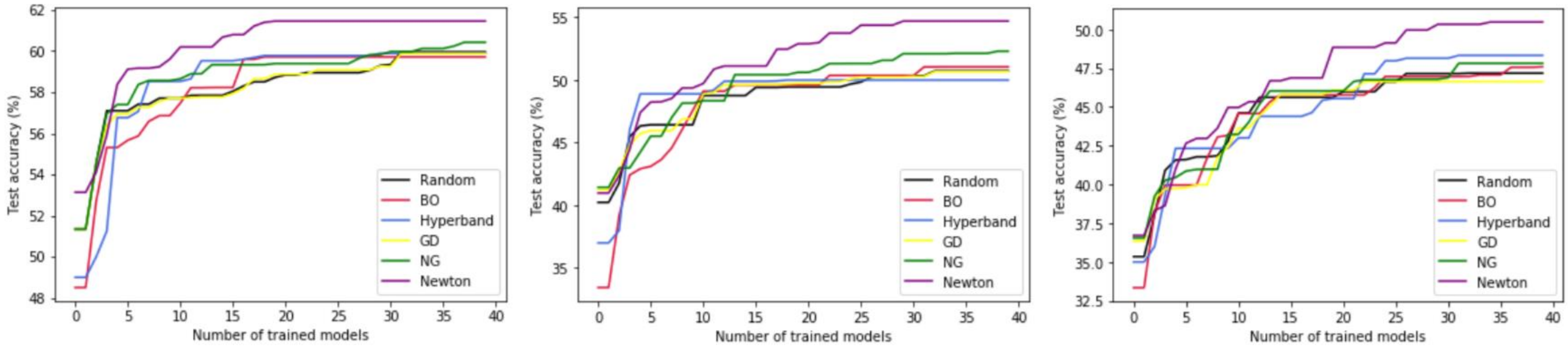
(a) symmetry flipping (20%). (b) symmetry flipping (35%). (c) symmetry flipping (50%).

(d) pair flipping (25%). (e) pair flipping (35%). (f) pair flipping (45%).

Figure 10: Label precision of *MentorNet*, *Co-teaching*, *Co-teaching+* and *S2E* on CIFAR-10.

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.

Experiments – Overall performance (semi)

Table 7: Testing accuracy (in %) on CIFAR-100. The term "early" means highest testing accuracy, and "average" means the averaged performance over the last ten epochs.

noise	symmetric 20%		symmetric 35%		symmetric 50%	
	early	average	early	average	early	average
Meta-Weight-Net	58.92±0.25	57.67±0.13	50.77±0.37	39.36±0.13	42.54±0.45	29.83±0.09
DivideMix	63.04±0.48	62.76±0.32	61.69±0.69	61.32±0.14	58.17±0.43	57.99±0.30
ELR+	61.48±0.35	61.05±0.15	58.71±0.35	58.05±0.11	53.68±0.43	53.27±0.26
CDR	51.69±0.23	42.51±0.15	47.29±0.35	35.57±0.16	41.71±0.79	29.61±0.11
Class2Simi	53.59±1.22	51.04±0.31	50.48±1.03	47.03±0.23	45.87±1.15	43.49±0.75
S2E (Semi)	64.08±0.18	63.96±0.12	**62.64±0.26**	**62.25±0.20**	59.23±0.45	59.08±0.21
S2E (Cubic, semi)	**64.32±0.22**	**64.17±0.09**	**62.69±0.14**	**62.38±0.11**	**59.94±0.33**	**59.75±0.17**
noise	pairflip 25%		pairflip 35%		pairflip 45%	
	early	average	early	average	early	average
Meta-Weight-Net	48.75±0.69	44.12±0.16	42.00±0.48	38.76±0.12	32.80±0.41	31.10±0.14
DivideMix	61.55±0.54	61.16±0.20	53.18±0.33	52.72±0.31	38.51±0.37	38.22±0.14
ELR+	59.15±0.77	58.83±0.19	54.07±0.37	53.80±0.14	**42.98±0.51**	**42.14±0.12**
CDR	45.76±0.39	41.39±0.20	38.94±0.55	35.45±0.21	30.66±0.63	28.98±0.20
Class2Simi	46.40±0.88	42.82±0.70	39.38±1.29	36.31±0.63	30.64±1.32	29.74±0.57
S2E (Semi)	61.79±0.32	61.38±0.15	53.29±0.15	52.89±0.20	39.37±0.27	39.19±0.13
S2E (Cubic, semi)	**62.24±0.30**	**61.77±0.16**	**54.51±0.19**	**54.15±0.21**	39.78±0.25	39.66±0.13

S2E (Semi) and S2E (Cubic, semi) with the

(i) Meta-Weight-Net (Shu et al., 2019);

(ii) DivideMix (Li et al., 2020);

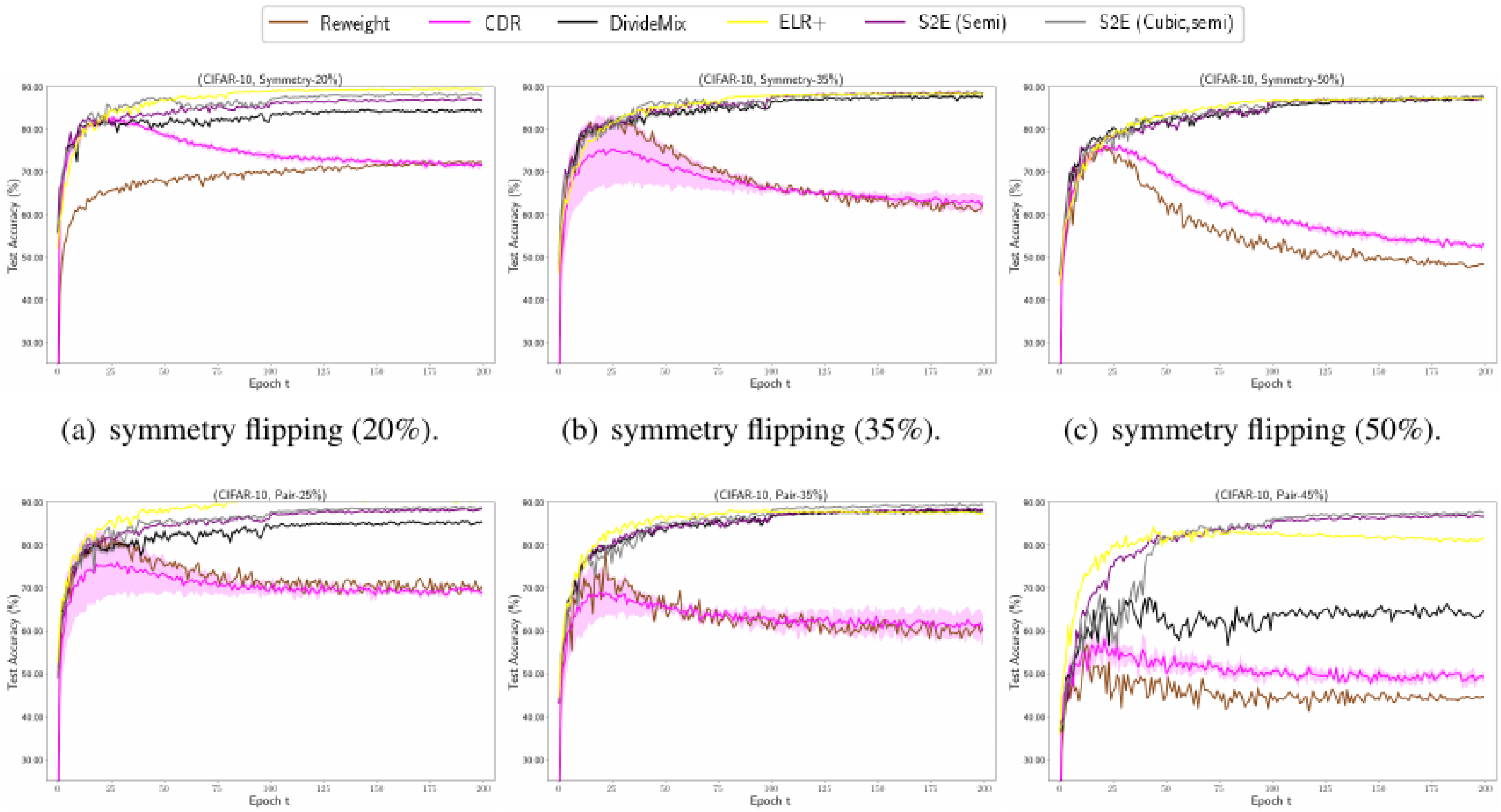
(iii) ELR+ (Liu et al., 2020);

(iv) CDR (Xia et al., 2021); and

(v) Class2Simi (Wu et al., 2021).

Take noisy instance as semi-supervised samples.

Experiments – Overall performance (semi)



(a) symmetry flipping (20%). (b) symmetry flipping (35%). (c) symmetry flipping (50%).

(d) pair flipping (25%). (e) pair flipping (35%). (f) pair flipping (45%).

Figure 13: Testing accuracies (mean and standard deviation) on CIFAR-10.

S2E (Semi) and S2E (Cubic, semi) with the

(i) Meta-Weight-Net (Shu et al., 2019);

(ii) DivideMix (Li et al., 2020);

(iii) ELR+ (Liu et al., 2020);

(iv) CDR (Xia et al., 2021); and

(v) Class2Simi (Wu et al., 2021).

Take noisy instance as semi-supervised samples.

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!