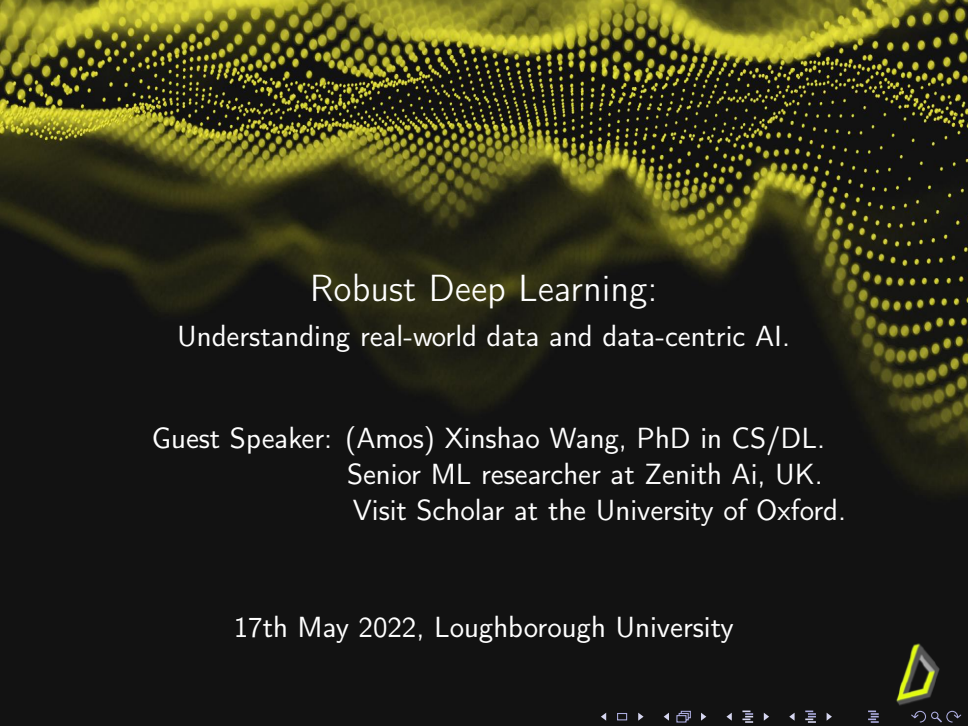




Zenith Ai

Z to A of intelligence



Robust Deep Learning:

Understanding real-world data and data-centric AI.

Guest Speaker: (Amos) Xinshao Wang, PhD in CS/DL.
Senior ML researcher at Zenith Ai, UK.
Visit Scholar at the University of Oxford.

17th May 2022, Loughborough University



Outline

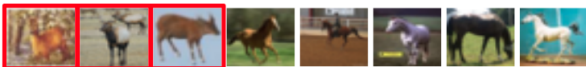
- 1 Robust deep learning
 - Why do we need it
 - Understanding real-world data with adverse cases
 - Learning objectives
- 2 Robust deep learning by example weighting strategies
 - Definition of example weighting
 - Derivative manipulation (DM) for general example weighting
- 3 Robust deep learning by target modification strategies
 - Core research questions
 - Target definition and target modification approaches
 - Defending the entropy minimisation principle
 - Summary
- 4 How is it like doing research in Zenith Ai
 - Industrial research overview
 - Zenith AI research = model-centric AI + data-centric AI

Outline

- 1 Robust deep learning
Why do we need it
Understanding real-world data with adverse cases
Learning objectives
- 2 Robust deep learning by example weighting strategies
Definition of example weighting
Derivative manipulation (DM) for general example weighting
- 3 Robust deep learning by target modification strategies
Core research questions
Target definition and target modification approaches
Defending the entropy minimisation principle
Summary
- 4 How is it like doing research in Zenith Ai
Industrial research overview
Zenith AI research = model-centric AI + data-centric AI

Why do we need robust deep learning?

To learn meaningful patterns under real-world adverse cases.



Horse class: The first three images are deer semantically.



This video is labelled as the person wearing black skirt.



This video is labelled as the person wearing green shirt.

Figure: Display of abnormal training examples highlighted by red boxes.

Outline

1 Robust deep learning

Why do we need it

Understanding real-world data with adverse cases

Learning objectives

2 Robust deep learning by example weighting strategies

Definition of example weighting

Derivative manipulation (DM) for general example weighting

3 Robust deep learning by target modification strategies

Core research questions

Target definition and target modification approaches

Defending the entropy minimisation principle

Summary

4 How is it like doing research in Zenith Ai

Industrial research overview

Zenith AI research = model-centric AI + data-centric AI

Adverse cases in real-world data

Out-of-distribution anomalies: Know the unknown

- Some inputs contain only background and no semantic information at all.
- Some may contain an object that does not belong to any class in the training set.



Adverse cases in real-world data

In-distribution anomalies: Detect => Ignore or Correct

- Label noise arising from:
 - ① Noisy annotations: e.g., some images of deer may be wrongly annotated to horse.
 - ② Missing annotations: we may use some algorithms to predict their labels. If so, the predicted labels are not 100% accurate.
- When an input contains more than one object, it becomes ambiguous without any prior.



Outline

- 1 Robust deep learning
 - Why do we need it
 - Understanding real-world data with adverse cases
 - Learning objectives
- 2 Robust deep learning by example weighting strategies
 - Definition of example weighting
 - Derivative manipulation (DM) for general example weighting
- 3 Robust deep learning by target modification strategies
 - Core research questions
 - Target definition and target modification approaches
 - Defending the entropy minimisation principle
 - Summary
- 4 How is it like doing research in Zenith Ai
 - Industrial research overview
 - Zenith AI research = model-centric AI + data-centric AI

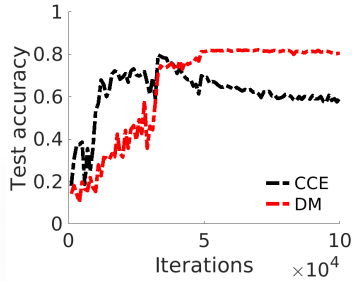
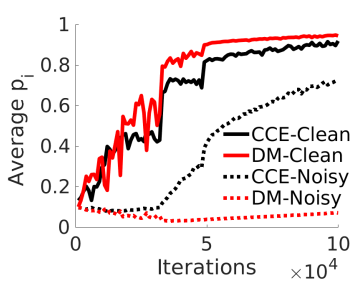
Learning objectives of robust deep learning

What is the meaning of robustness here?

- **To learn meaningful patterns** on semantically clean training data (where noise may exist, however, the semantic matching from observations to annotations is highly correct).
- **Without fitting wrong patterns** on semantically wrong training data, so that the learning process of a model is not contaminated.
- **Generalisation to unseen data.**

Robustness against adverse cases

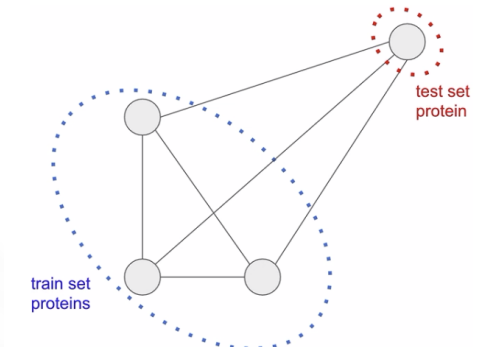
What should a learning process ideally look like?



- $p_i = p(y_i|x_i)$: predicted relevance between an observation x_i and its label y_i .
- We train ResNet-56 on CIFAR-10 with 40% symmetric label noise.

Generalise to unseen data

Build train-validation datasets properly



[2] Bileschi, Maxwell L., et al. "Using deep learning to annotate the protein universe." Nature Biotechnology (2022).

Outline

- 1 Robust deep learning
 - Why do we need it
 - Understanding real-world data with adverse cases
 - Learning objectives
- 2 Robust deep learning by example weighting strategies
 - Definition of example weighting
 - Derivative manipulation (DM) for general example weighting
- 3 Robust deep learning by target modification strategies
 - Core research questions
 - Target definition and target modification approaches
 - Defending the entropy minimisation principle
 - Summary
- 4 How is it like doing research in Zenith Ai
 - Industrial research overview
 - Zenith AI research = model-centric AI + data-centric AI

Robust learning via example weighting

Example weighting is universal in deep learning

We define our interpretation of example weighting [10]:

Definition (Example Weighting). *In gradient-based optimization, the loss's derivative of an example can be interpreted as its effect on the update of a model [3, 1]. Therefore, a derivative's magnitude function can be treated as a weighting scheme from the viewpoint of example weighting.*

Accordingly, one technique that leads to a change of the derivative magnitude function, is equivalent to, modifying an example weighting scheme.



Outline

- 1 Robust deep learning
 - Why do we need it
 - Understanding real-world data with adverse cases
 - Learning objectives
- 2 Robust deep learning by example weighting strategies
 - Definition of example weighting
 - Derivative manipulation (DM) for general example weighting
- 3 Robust deep learning by target modification strategies
 - Core research questions
 - Target definition and target modification approaches
 - Defending the entropy minimisation principle
 - Summary
- 4 How is it like doing research in Zenith Ai
 - Industrial research overview
 - Zenith AI research = model-centric AI + data-centric AI

Derivative manipulation (DM)

Fundamental research questions and conclusions

- 1 What kind of examples to focus on during training? How to weight them properly?
- 2 A loss function is okay to be non-symmetric, unbounded, or even non-differentiable.
- 3 Is a loss function necessary for deriving the gradient used for back-propagation?
- 4 When a training set contains a higher label noise rate, we should focus on easier training examples for better generalisation!



Standard practice versus DM

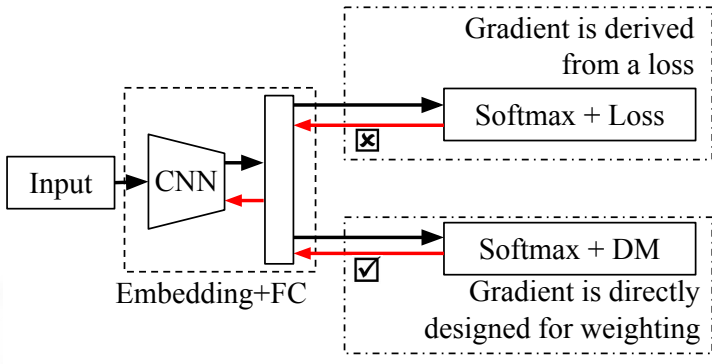


Figure: Black and red arrows denote forward process and gradient back-propagation, respectively.

Direction: the same as common losses

Analysis of common losses

$$\begin{aligned}L_{\text{CCE}}(\mathbf{x}_i, y_i) &= -\log p(y_i|\mathbf{x}_i) \\L_{\text{MAE}}(\mathbf{x}_i, y_i) &= 1 - p(y_i|\mathbf{x}_i) \\L_{\text{MSE}}(\mathbf{x}_i, y_i) &= (1 - p(y_i|\mathbf{x}_i))^2, \\L_{\text{GCE}}(\mathbf{x}_i, y_i) &= \frac{1 - p(y_i|\mathbf{x}_i)^q}{q},\end{aligned}\tag{1}$$

$$\begin{aligned}\frac{\partial L_{\text{CCE}}}{\partial z_{ij}} &= \begin{cases} p(y_i|\mathbf{x}_i) - 1, & j = y_i \\ p(j|\mathbf{x}_i), & j \neq y_i \end{cases} \\ \frac{\partial L_{\text{MAE}}}{\partial z_i} &= p_i \times \frac{\partial L_{\text{CCE}}}{\partial z_i}; \\ \frac{\partial L_{\text{MSE}}}{\partial z_i} &= 2p_i \times (1 - p_i) \times \frac{\partial L_{\text{CCE}}}{\partial z_i}; \\ \frac{\partial L_{\text{GCE}}}{\partial z_i} &= p_i^q \times \frac{\partial L_{\text{CCE}}}{\partial z_i}.\end{aligned}\tag{2}$$

Magnitude: Emphasis Density Function

- Example weighting in common losses

$$\begin{aligned}w_i^{\text{CCE}} &= 2(1 - p_i) \Rightarrow \psi_{\text{CCE}} = 0; \\w_i^{\text{MAE}} &= 2p_i(1 - p_i) \Rightarrow \psi_{\text{MAE}} = 0.5; \\w_i^{\text{MSE}} &= 4p_i(1 - p_i)^2 \Rightarrow \psi_{\text{MSE}} = \frac{1}{3}; \\w_i^{\text{GCE}} &= 2p_i^q(1 - p_i) \Rightarrow \psi_{\text{GCE}} = \frac{q}{q + 1}.\end{aligned}\tag{3}$$

- Our generalised formulation

$$\begin{aligned}\nabla_{\mathbf{z}_i} &= w_i^{\text{DM}} / (2(1 - p_i)) \times \frac{\partial L_{\text{CCE}}}{\partial \mathbf{z}_i}. \\w_i^{\text{DM}} &= \exp(\beta p_i^\lambda (1 - p_i)) \Rightarrow \psi_{\text{DM}} = \frac{\lambda}{\lambda + 1}. \\ \lambda \geq 0 &\Rightarrow \psi_{\text{DM}} \in [0, 1).\end{aligned}\tag{4}$$

Magnitude: Emphasis Density Function

- Emphasis density function:

$$w_i^{\text{DM}} = \exp(\beta p_i^\lambda (1 - p_i)) \Rightarrow \psi_{\text{DM}} = \frac{\lambda}{\lambda + 1}. \quad (5)$$
$$h(w_i^{\text{DM}}) = \frac{w_i^{\text{DM}}}{\int_0^1 w_i^{\text{DM}} d p_i} \Rightarrow \int_0^1 h(w_i^{\text{DM}}) d p_i = 1.$$

Magnitude: Emphasis Density Function

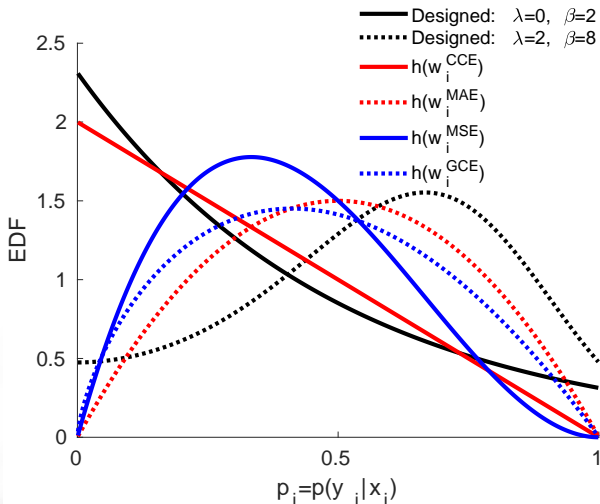


Figure: An EDF is an example-level weighting function normalised by its integral over $[0, 1]$.

Example weighting via EDFs

Summary and definitions

Definition 1 (Emphasis Mode ψ). We define the emphasis mode to be p_i of examples whose weights are the largest, i.e.,
$$\psi = \arg \max_{p_i} w_i, \quad \psi \in [0, 1].$$

For example, by 'emphasis mode is 0 in CCE' we mean those images with $p_i = 0$ own the highest weights.

Definition 2 (Emphasis Variance σ). $\sigma = E((w_i - E(w_i))^2)$, where $E(\cdot)$ denotes the expectation of a variable.



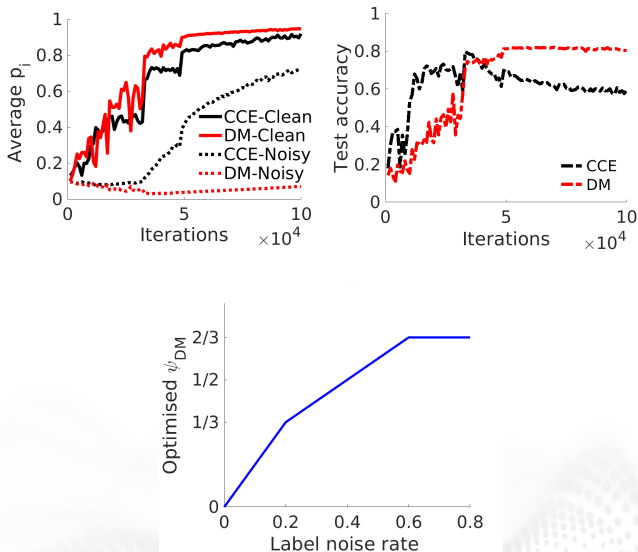


Figure: We observe noisy examples have much less p_i than clean ones, thus being more difficult examples.

Outline

- 1 Robust deep learning
 - Why do we need it
 - Understanding real-world data with adverse cases
 - Learning objectives
- 2 Robust deep learning by example weighting strategies
 - Definition of example weighting
 - Derivative manipulation (DM) for general example weighting
- 3 Robust deep learning by target modification strategies
 - Core research questions**
 - Target definition and target modification approaches
 - Defending the entropy minimisation principle
 - Summary
- 4 How is it like doing research in Zenith Ai
 - Industrial research overview
 - Zenith AI research = model-centric AI + data-centric AI

Core research questions [11]

- 1 In Self LC, how much should we trust a learner to leverage its knowledge?
 - The trust score is fixed or updated stage-by-stage in prior work.
 - ProSelfLC modifies the target progressively, is end-to-end trainable, and requires negligible extra cost.
- 2 Should we penalise a low-entropy status or reward it?
 - OR methods penalise low entropy while LC rewards it.
 - ProSelfLC [11] redirects and promotes entropy minimisation.



Outline

- 1 Robust deep learning
 - Why do we need it
 - Understanding real-world data with adverse cases
 - Learning objectives
- 2 Robust deep learning by example weighting strategies
 - Definition of example weighting
 - Derivative manipulation (DM) for general example weighting
- 3 Robust deep learning by target modification strategies
 - Core research questions
 - Target definition and target modification approaches
 - Defending the entropy minimisation principle
 - Summary
- 4 How is it like doing research in Zenith Ai
 - Industrial research overview
 - Zenith AI research = model-centric AI + data-centric AI

BEYOND SEMANTIC CLASS

THE SIMILARITY STRUCTURE IN A LABEL DISTRIBUTION

A label distribution defines what to learn:

- **Definition 1** (*Semantic Class*). Given a target label distribution $\tilde{q}(x) \in \mathbb{R}^C$, the semantic class is defined by $\arg \max_j \tilde{q}(j|x)$, i.e., the class whose probability is the largest.
- **Definition 2** (*Similarity Structure*). In CCE, LS and CP, a data point has an identical probability of belonging to other classes except for the semantic class. Instead, in LC, a target label distribution captures the probability difference of an example being predicted to every class. We define it to be the similarity structure of one example versus all training classes.

An overview of label (target) modification

OR(LS and CP) + LC(Self LC and Non-self LC)

$\tilde{\mathbf{q}}_{\text{LS}} = (1 - \epsilon)\mathbf{q} + \epsilon\mathbf{u}$		$\tilde{\mathbf{q}}_{\text{CP}} = (1 - \epsilon)\mathbf{q} - \epsilon\mathbf{p}$	
\mathbf{q}	\mathbf{u}	\mathbf{q}	\mathbf{p}
$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1/2 \\ 1/3 \\ 1/6 \end{pmatrix}$
$1 - \epsilon$	ϵ	$1 - \epsilon$	ϵ
\Rightarrow		\Rightarrow	
$\begin{pmatrix} (1 - \epsilon) + \epsilon/3 \\ \epsilon/3 \\ \epsilon/3 \end{pmatrix}$		$\begin{pmatrix} (1 - \epsilon) - \epsilon/2 \\ -\epsilon/3 \\ -\epsilon/6 \end{pmatrix} \Leftrightarrow \begin{pmatrix} (1 - \epsilon) - \epsilon/2 \\ 0 \\ 0 \end{pmatrix}$	
LS		CP	

OR includes LS [8] and CP [6], which smoothes similarity structure:


- LS softens a target by adding a uniform label distribution.
- CP changes the probability 1 to a smaller value $1 - \epsilon$ in the one-hot target.

The double-ended arrow means factual equivalence, because an output is definitely non-negative after a softmax layer.

An overview of label (target) modification

OR(LS and CP) + LC(Self LC and Non-self LC)


$$\bar{q}_{LC} = (1 - \epsilon)q + \epsilon p$$


target learner  p

q	p	
$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$	$+$	$\begin{pmatrix} 1/2 \\ 1/3 \\ 1/6 \end{pmatrix}$
\Rightarrow		$\begin{pmatrix} (1 - \epsilon) + \epsilon/2 \\ \epsilon/3 \\ \epsilon/6 \end{pmatrix}$
$1 - \epsilon$	ϵ	

Self LC: p is the output of a learner itself.

q	p	
$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$	$+$	$\begin{pmatrix} 1/2 \\ 1/6 \\ 1/3 \end{pmatrix}$
\Rightarrow		$\begin{pmatrix} (1 - \epsilon) + \epsilon/2 \\ \epsilon/6 \\ \epsilon/3 \end{pmatrix}$
$1 - \epsilon$	ϵ	

auxiliary learner 

p target learner 

Non-self LC: p is the output of an auxiliary learner.

- LC contains Self LC [5, 7, 9] and Non-self LC [4].
- The convex combination parameter ϵ defines how much a predicted label distribution is trusted.

Drawbacks of existing target modification

Why Self LC to exploit a model's self knowledge?

- 1 OR methods naively penalise confident outputs **without leveraging easily accessible knowledge** from other learners or itself.
- 2 Non-self LC relies on accurate **auxiliary models**.
- 3 Self LC:
 - It exploits its own knowledge;
 - It requires no extra learners;
 - However, **how much should we trust a learner to leverage its knowledge?**



Overview of existing variants of Self LC

Without considering a model's knowledge grows as time goes

- 1 In bootstrapping, ϵ is fixed throughout the training process.
- 2 Joint Optimisation fully trusts a learner by setting $\epsilon = 1$, and uses stage-wise training to gradually train the model.
 - Stage-wise training requires a significant human intervention and is time-consuming in practice.
- 3 Requirements of improving Self LC
 - End-to-end trainable.
 - Negligible extra cost.
 - Modifies the target progressively and adaptively as training goes.



Outline

- 1 Robust deep learning
 - Why do we need it
 - Understanding real-world data with adverse cases
 - Learning objectives
- 2 Robust deep learning by example weighting strategies
 - Definition of example weighting
 - Derivative manipulation (DM) for general example weighting
- 3 Robust deep learning by target modification strategies
 - Core research questions
 - Target definition and target modification approaches
 - Defending the entropy minimisation principle**
 - Summary
- 4 How is it like doing research in Zenith Ai
 - Industrial research overview
 - Zenith AI research = model-centric AI + data-centric AI

To penalise or reward a low-entropy status?

- OR methods penalise low entropy \Rightarrow OR is against entropy minimisation principle.
- LC rewards a low-entropy status \Rightarrow LC defends entropy minimisation principle.
 - LC has the same principle as the widely used expectation-maximization (EM) algorithm.

ProSelfLC

Self Trust according to Training Time and Confidence

ϵ indicates how much a predicted label distribution is trusted.
For any x , we summarise the loss and modified label:

$$\begin{aligned}L(\tilde{q}_{\text{ProSelfLC}}, \mathbf{p}; \epsilon_{\text{ProSelfLC}}) &= H(\tilde{q}_{\text{ProSelfLC}}, \mathbf{p}) = E_{\tilde{q}_{\text{ProSelfLC}}}(-\log \mathbf{p}), \\ \tilde{q}_{\text{ProSelfLC}} &= (1 - \epsilon_{\text{ProSelfLC}})\mathbf{q} + \epsilon_{\text{ProSelfLC}}\mathbf{p}, \\ \epsilon_{\text{ProSelfLC}} &= g(t) \times l(\mathbf{p}), \\ g(t) &= h(t/\Gamma - 0.5, B) \in (0, 1), \Rightarrow \text{Trusting learning time} \\ l(\mathbf{p}) &= 1 - H(\mathbf{p})/H(\mathbf{u}) \in (0, 1). \Rightarrow \text{Trusting sample confidence}\end{aligned}\tag{6}$$

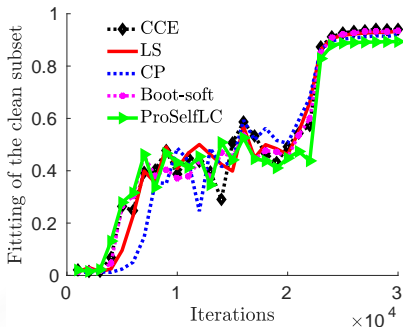
t and Γ are the iteration (time) counter and the number of total iterations, respectively.

$h(\cdot)$ is a logistic function where B controls its smoothness.

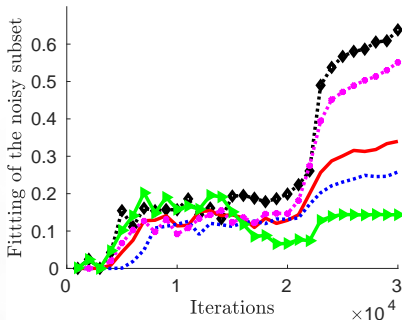
ProSelfLC

Training Dynamics On CIFAR-100 with asymmetric label noise

$r = 0.4$.



(a) Correct fitting.

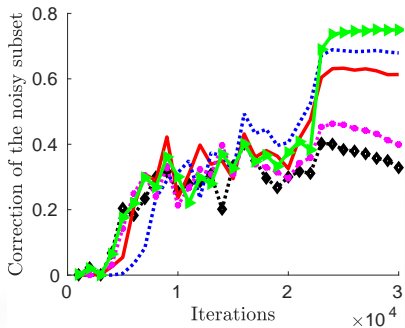


(b) Wrong fitting.

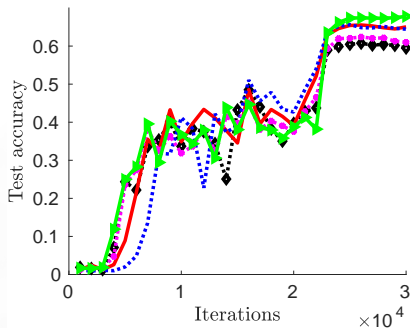
ProSelfLC

Training Dynamics On CIFAR-100 with asymmetric label noise

$r = 0.4$.



(a) Semantic class correction

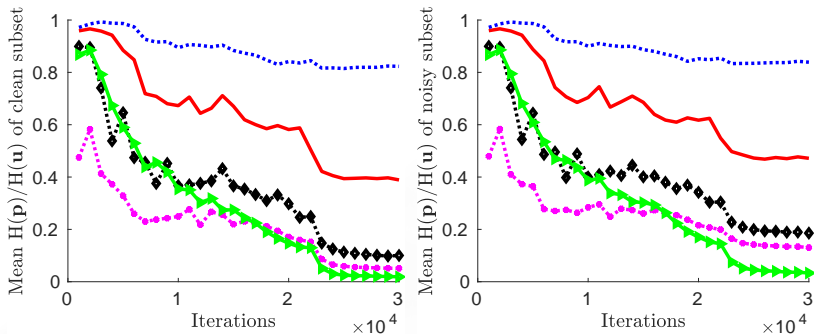


(b) Generalisation.

ProSelfLC

Training Dynamics On CIFAR-100 with asymmetric label noise

$r = 0.4$.



(a) Entropy of clean subset.

(b) Entropy of noisy subset.

Figure: **Should we penalise a low-entropy status or reward it?**

Outline

- 1 Robust deep learning
 - Why do we need it
 - Understanding real-world data with adverse cases
 - Learning objectives
- 2 Robust deep learning by example weighting strategies
 - Definition of example weighting
 - Derivative manipulation (DM) for general example weighting
- 3 Robust deep learning by target modification strategies
 - Core research questions
 - Target definition and target modification approaches
 - Defending the entropy minimisation principle
 - Summary
- 4 How is it like doing research in Zenith Ai
 - Industrial research overview
 - Zenith AI research = model-centric AI + data-centric AI

Summary

① ProSelfLC:

- enhance the similarity structure information over training classes.
- correct the semantic classes of noisy label distributions.
- is the first method to trust self knowledge progressively and adaptively.

② Our extensive experiments:

- defend the entropy minimisation principle.

③ Code:

<https://github.com/XinshaoAmosWang/ProSelfLC-CVPR2021>

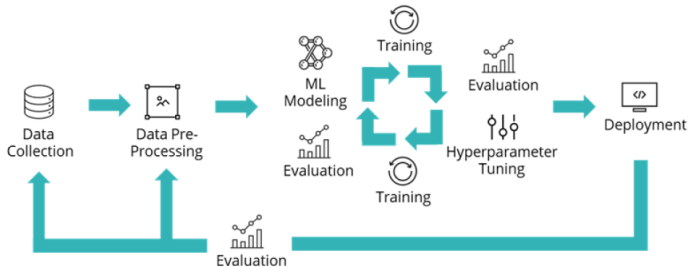


Outline

- 1 Robust deep learning
 - Why do we need it
 - Understanding real-world data with adverse cases
 - Learning objectives
- 2 Robust deep learning by example weighting strategies
 - Definition of example weighting
 - Derivative manipulation (DM) for general example weighting
- 3 Robust deep learning by target modification strategies
 - Core research questions
 - Target definition and target modification approaches
 - Defending the entropy minimisation principle
 - Summary
- 4 How is it like doing research in Zenith Ai
 - Industrial research overview
 - Zenith AI research = model-centric AI + data-centric AI

Industrial research overview

- The focus is to solve real-world impactful applications. Publishing and sharing research is encouraged.
- To build AI toolboxes and a comprehensive AI pipeline to solve diverse applications by creating AI softwares and web applications, etc.

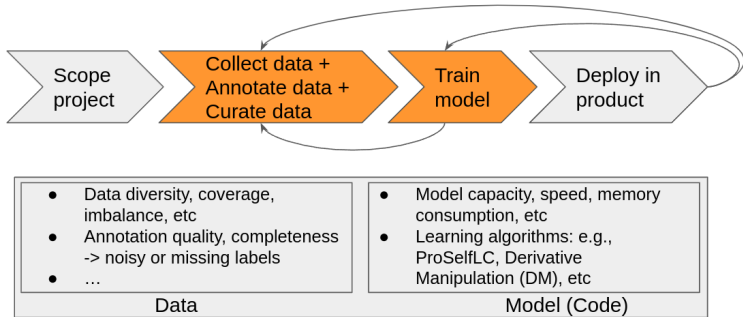


Outline

- 1 Robust deep learning
 - Why do we need it
 - Understanding real-world data with adverse cases
 - Learning objectives
- 2 Robust deep learning by example weighting strategies
 - Definition of example weighting
 - Derivative manipulation (DM) for general example weighting
- 3 Robust deep learning by target modification strategies
 - Core research questions
 - Target definition and target modification approaches
 - Defending the entropy minimisation principle
 - Summary
- 4 How is it like doing research in Zenith Ai
 - Industrial research overview
 - Zenith AI research = model-centric AI + data-centric AI

Zenith AI research

model-centric AI + data-centric AI



AI System = Data + Model (Code)

Thanks for your attention.
Questions and discussions are very welcome.

Zenith Ai:

- Vision to empower life sciences research and industrialization: <https://www.zenify.ai/>
- Partners: <https://www.zenify.ai/partners>
- Science: <https://www.zenify.ai/our-science>
- Press: <https://www.zenify.ai/press>

Personal Info: [Homepage](#) and [Google Scholar](#)

References

- [1] Barron, J. T. A general and adaptive robust loss function. In *CVPR*, 2019.
- [2] Bileschi, M. L., Belanger, D., Bryant, D. H., Sanderson, T., Carter, B., Sculley, D., Bateman, A., DePristo, M. A., and Colwell, L. J. Using deep learning to annotate the protein universe. *Nature Biotechnology*, pp. 1–6, 2022.
- [3] Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. The approach based on influence functions. In *Robust Statistics*. Wiley, 1986.
- [4] Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning and Representation Learning Workshop*, 2015.
- [5] Lee, D.-H. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. 2013.
- [6] Pereyra, G., Tucker, G., Chorowski, J., Kaiser, Ł., and Hinton, G. Regularizing neural networks by penalizing confident output distributions. In *ICLR Workshop*, 2017.
- [7] Reed, S., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., and Rabinovich, A. Training deep neural networks on noisy labels with bootstrapping. In *ICLR Workshop*, 2015.
- [8] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.
- [9] Tanaka, D., Ikami, D., Yamasaki, T., and Aizawa, K. Joint optimization framework for learning with noisy labels. In *CVPR*, 2018.
- [10] Wang, X., Kodirov, E., Hua, Y., and Robertson, N. M. Derivative manipulation for general example weighting. *arXiv preprint arXiv:1905.11233*, 2019.
- [11] Wang, X., Hua, Y., Kodirov, E., Clifton, D. A., and Robertson, N. M. Proselflc: Progressive self label correction for training robust deep neural networks. In *CVPR*, 2021.