Knowledge Distillation in Wide Neural Networks: Risk Bound, Data Efficiency and Imperfect Teacher

Guangda Ji

jiguangda@pku.edu.cn

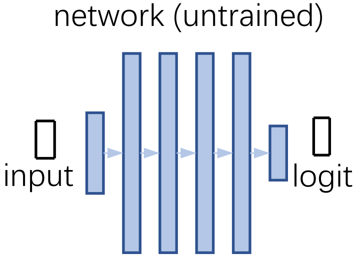
Zhanxing Zhu

zhanxing.zhu@pku.edu.cn

Peking University

Introduction to Knowledge distillation(KD)

- Vanilla training: only hard (one-hot) labels.
  - example:



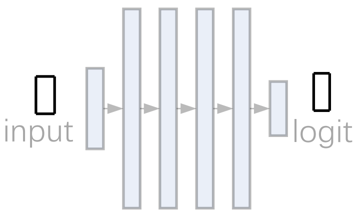
hard label

$$\mathcal{L}_{\text{vanilla}} = \mathbf{CrossEntropy}(\mathbf{softmax}(\mathbf{z}), \begin{bmatrix}0\\1\\0\\0\end{bmatrix})$$

Introduction to Knowledge distillation(KD)

- Vanilla training: only hard (one-hot) labels.
  - example:

network (untrained)

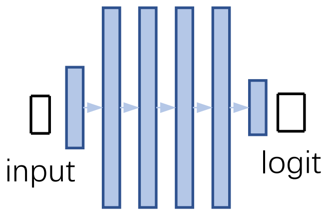


hard label

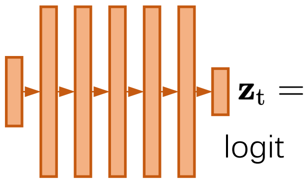
$$\mathcal{L}_{\text{vanilla}} = \text{CrossEntropy}(\text{softmax}(\mathbf{z}), \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix})$$

- Knowledge distillation[1]: combinations of soft and hard labels. example:

student (untrained)



teacher (well trained)



$$\mathbf{x} \quad \mathbf{z}_{\text{t}} = \begin{bmatrix} -0.1 \\ 1.5 \\ -0.4 \\ 0.1 \end{bmatrix}$$

$$\mathbf{y}_{\text{t}} = \text{softmax}(\mathbf{z}_{\text{t}}) = \begin{bmatrix} 0.13 \\ 0.63 \\ 0.09 \\ 0.15 \end{bmatrix}$$

label

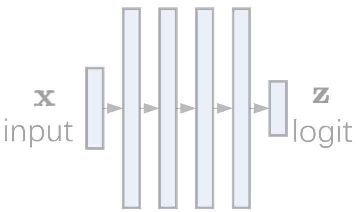
hard label, soft label

$$\mathcal{L}_{\text{KD}} = \underbrace{(1-\rho)}_{\text{hard ratio}}\text{CrossEntropy}(\text{softmax}(\mathbf{z}_{\text{s}}), \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}) + \underbrace{\rho}_{\text{soft ratio}}\text{CrossEntropy}(\mathbf{z}_{\text{s}}), \begin{bmatrix} 0.13 \\ 0.63 \\ 0.09 \\ 0.15 \end{bmatrix})$$

[1] Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network.

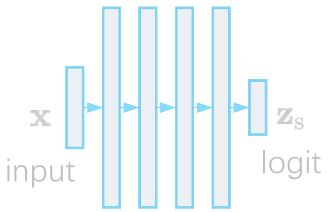
Introduction to Knowledge distillation(KD)

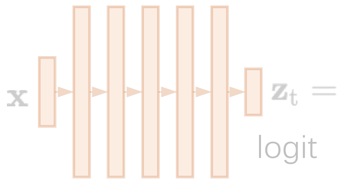
- Vanilla training: only hard (one-hot) labels.
  - example:



$$\mathcal{L}_{\text{vanilla}} = \text{CrossEntropy}(\text{softmax}(\mathbf{z}), \overset{\text{hard label}}{\begin{bmatrix}0\\1\\0\\0\end{bmatrix}})$$

- Knowledge distillation: combinations of soft and hard labels. example:





$$\mathbf{z}_t = \begin{bmatrix}-0.1\\1.5\\-0.4\\0.1\end{bmatrix} \qquad \underset{\text{label}}{\mathbf{y}_t} = \text{softmax}(\mathbf{z}_t) = \begin{bmatrix}0.13\\0.63\\0.09\\0.15\end{bmatrix}$$

$$\mathcal{L}_{\text{KD}} = \underset{\text{hard ratio}}{(1-\rho)}\text{CrossEntropy}(\text{softmax}(\mathbf{z}_s), \overset{\text{hard label}}{\begin{bmatrix}0\\1\\0\\0\end{bmatrix}}) + \underset{\text{soft ratio}}{\rho}\text{CrossEntropy}(\mathbf{z}_s), \overset{\text{soft label}}{\begin{bmatrix}0.13\\0.63\\0.09\\0.15\end{bmatrix}})$$

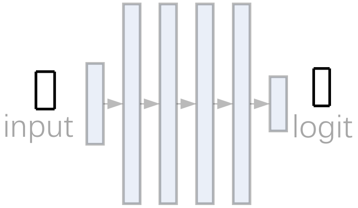
- KD is widely used in industry but lacks a satisfying explanation.

- Our work is to
  - establish a theoretical understanding on KD.
  - give instructions on the optimal choice of parameters.

Introduction to Knowledge distillation(KD)

- Vanilla training: only hard (one-hot) labels.
  - example:

network (untrained)

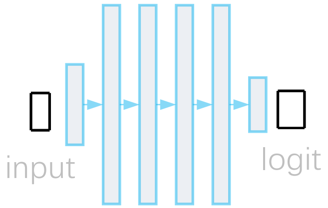


hard label

$$\mathcal{L}_{\text{vanilla}} = \text{CrossEntropy}(\text{softmax}(\mathbf{z}), \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix})$$

- Knowledge distillation: combinations of soft and hard labels. example:

student (untrained)



$$\mathbf{z}_{\text{t}} = \begin{bmatrix} -0.1 \\ 1.5 \\ -0.4 \\ 0.1 \end{bmatrix} \qquad \mathbf{y}_{\text{t}} = \text{softmax}(\mathbf{z}_{\text{t}}) = \begin{bmatrix} 0.13 \\ 0.63 \\ 0.09 \\ 0.15 \end{bmatrix}$$

label

hard label, soft label

$$\mathcal{L}_{\text{KD}} = \underbrace{(1-\rho)}_{\text{hard ratio}}\text{CrossEntropy}(\text{softmax}(\mathbf{z}_{\text{s}}), \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}) + \underbrace{\rho}_{\text{soft ratio}}\text{CrossEntropy}(\mathbf{z}_{\text{s}}), \begin{bmatrix} 0.13 \\ 0.63 \\ 0.09 \\ 0.15 \end{bmatrix})$$

- KD is widely used in industry but lacks a satisfying explanation.

- Our contribution in this work:
  - Transfer risk bound.
  - Metric of *data inefficiency* for perfect teacher distillation
  - Hard labels in imperfect distillation.

[1] Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network

Problem Setup

- Binary classification,
  - logit $z \in R$, hard label $y \in \{0, 1\}$, soft label $y \in [0, 1]$.
  - distillation loss:

$$\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N} \ell_n = \frac{1}{N}\sum_{n=1}^{N} \rho \underbrace{H(\sigma(\frac{z_{\mathrm{t},n}}{T}), \sigma(\frac{z_{\mathrm{s},n}}{T}))}_{\text{soft loss}} + (1-\rho)\underbrace{H(y_{\mathrm{g},n}, \sigma(z_{\mathrm{s},n}))}_{\text{hard loss}}$$

ρ: soft ratio

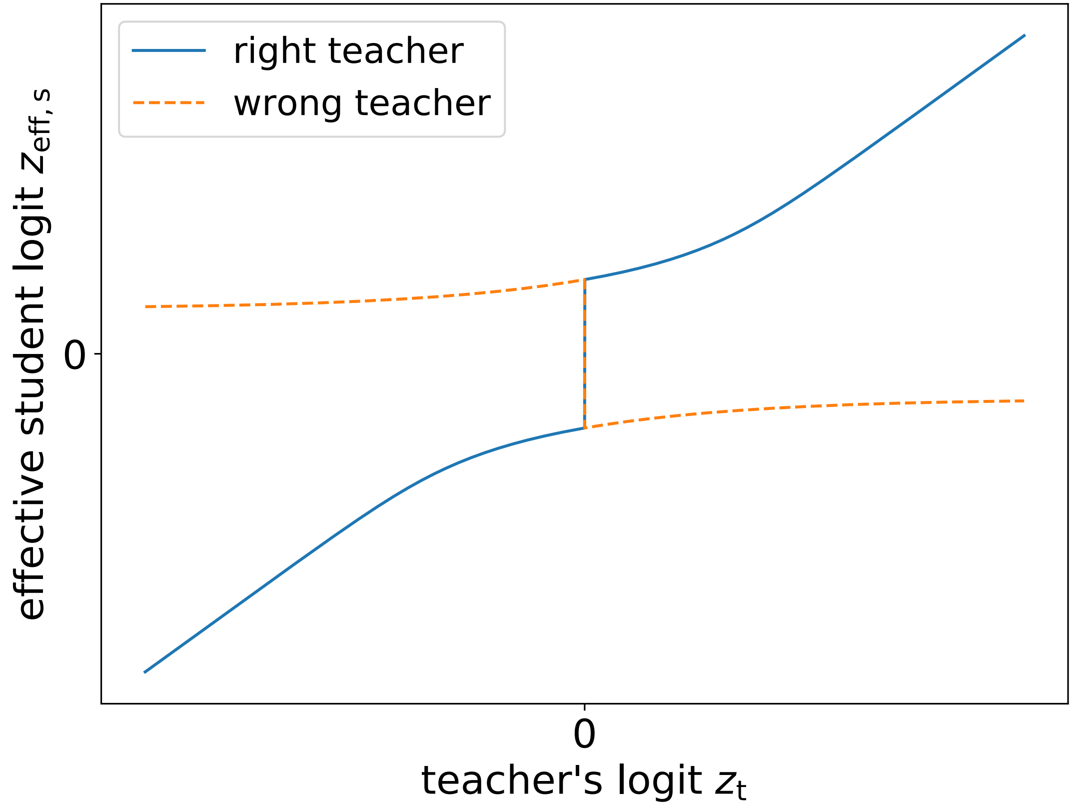
- Assumption: the student network is wide and over-parameterized.
  - convergence to global minima[2].
  - minimization at each sample point,

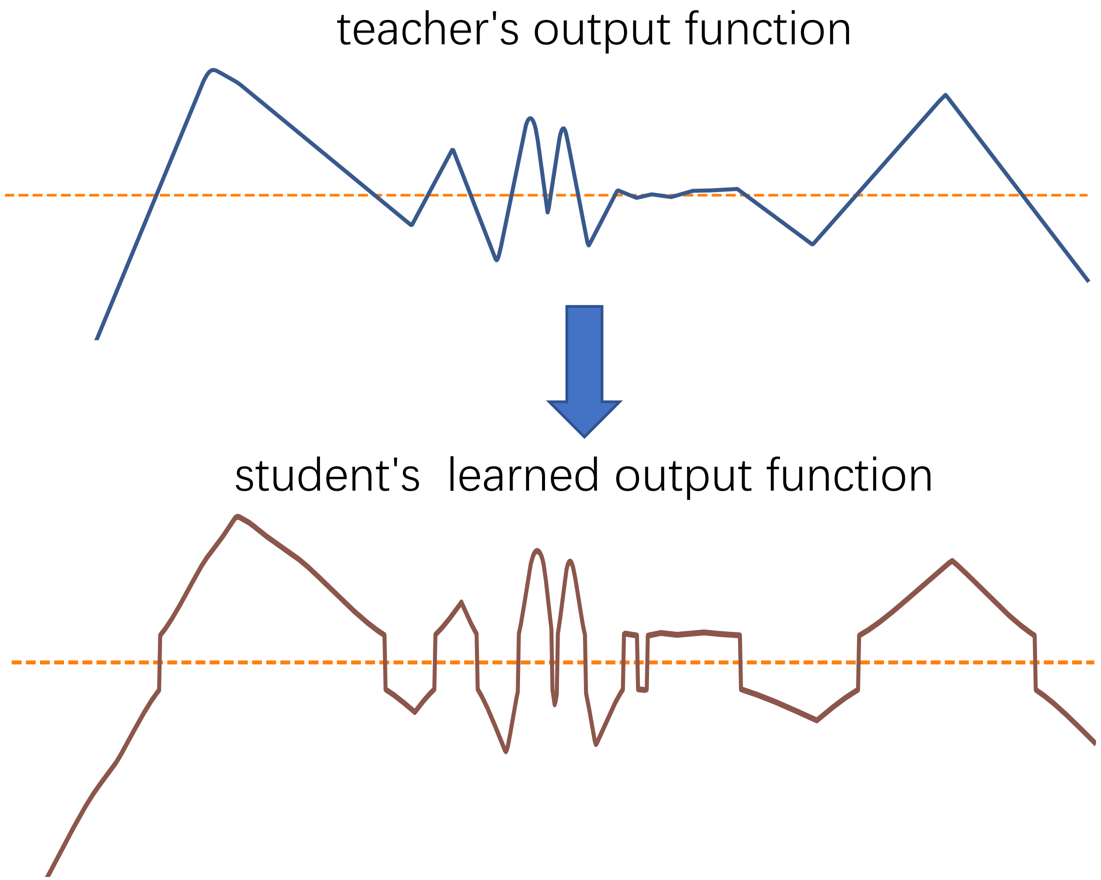
$$\lim_{\tau \to \infty} z_{\mathrm{s}}(\tau) = \hat{z}_{\mathrm{s}}, \quad \frac{d\ell}{d\hat{z}_{\mathrm{s}}} = \frac{\rho}{T}(\sigma(\hat{z}_{\mathrm{s}}/T) - \sigma(z_{\mathrm{t}}/T)) + (1-\rho)(\sigma(\hat{z}_{\mathrm{s}}) - y_{\mathrm{g}}) = 0.$$

  - Through this equation, student learns an effective logit. $z_{\mathrm{s,eff}}(z_{\mathrm{t}}, y_{\mathrm{g}}) = \hat{z}_{\mathrm{s}}$.
  - Result in thresholds and discontinuities in the output function.

[2] Du, Simon, et al. "Gradient descent finds global minima of deep neural networks." ICML 2019.

Problem Setup





- Through this equation, student learns an effective logit $z_{\mathrm{s,eff}}(z_{\mathrm{t}}, y_{\mathrm{g}}) = \hat{z}_{\mathrm{s}}$.
- Result in thresholds and discontinuities in the output function.

Problem Setup

- Assumption: the student network is wide and over-parameterized.
  - The use of neural tangent kernel (NTK) technique[3].
    - Approximate student network's output with its linearized version,

$$f(x; w_{\mathrm{nlin}}) \approx f(x; w_0) + \mathbf{\Delta}_w^\top \phi(x)$$

    - where
      - $\mathbf{\Delta}_w = w - w_0 \in \mathbb{R}^p$: weight change,
      - $\phi(x) = \partial_w f(x; w_0) \in \mathbb{R}^p$: random feature.

    - Also enable us to establish a direct link between network's weight change and its logits,

$$\mathbf{\Delta}_{\hat{w}} = \phi(\mathbf{X})(\hat{\Theta}(\mathbf{X}, \mathbf{X}))^{-1} \mathbf{\Delta}_{\mathbf{z}},$$

    - where
      - $\hat{\Theta}(\mathbf{X}, \mathbf{X}) = \phi(\mathbf{X})^\top \phi(\mathbf{X})$: tangent kernel
      - $\Delta_{\mathbf{z}} = \mathbf{z} - f(\mathbf{X}; w_0)$.

[3] Lee, Jaehoon, et al. "Wide neural networks of any depth evolve as linear models under gradient descent." Neurips 2019

Result 1: Transfer Risk Bound

- Transfer risk $\mathcal{R}$: probability of different prediction w.r.t. teacher.
- **Theorem 1 (Risk bound):**

$$\mathcal{R}_n \leq p(\frac{\pi}{2} - \bar{\alpha}_n),$$

  - n: sample size,
  - $\bar{\alpha}_n = \bar{\alpha}(\Delta_{w_*}, \Delta_{\hat{w}})$: angle between oracle weight change Δ_{w_*} and student's weight change $\Delta_{\hat{w}}$,
  - $p(\beta)$: pdf of angle between random feature $\phi(x)$ and oracle Δ_{w_*}.
  - Tighter than the bound in [4].

[4] Phuong, Mary, and Christoph Lampert. "Towards understanding knowledge distillation." ICML 2019.

Result 1: Transfer Risk Bound

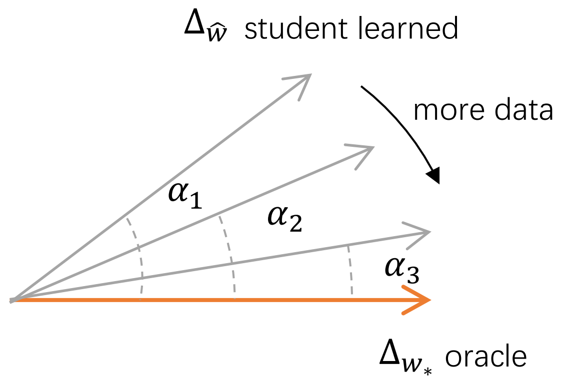
- Transfer risk $\mathcal{R}$: probability of different prediction w.r.t. teacher.
- **Theorem 1 (Risk bound):**

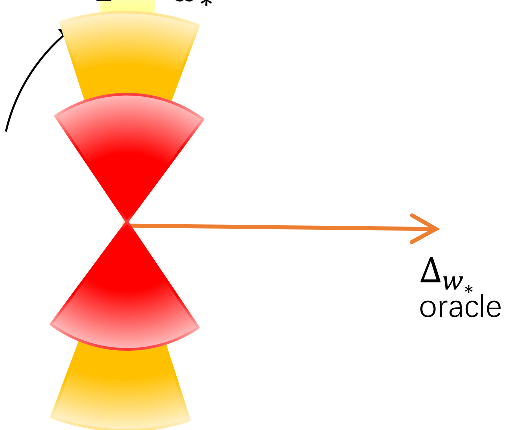
$$\mathcal{R}_n \leq p(\frac{\pi}{2} - \bar{\alpha}_n),$$

  - n: sample size,
  - $\bar{\alpha}_n = \bar{\alpha}(\Delta_{w_*}, \Delta_{\hat{w}})$: angle between oracle weight change Δ_{w_*} and student's weight change $\Delta_{\hat{w}}$,
  - $p(\beta)$: pdf of angle between random feature $\phi(x)$ and oracle Δ_{w_*} .
  - Tighter than the bound in [4].

- **Key idea of this theorem:**
  - student learns a projection of oracle weight change

$$\boldsymbol{\Delta}_{\hat{w}} = \phi(\mathbf{X})\Theta_n^{-1}\phi(\mathbf{X})^\top \boldsymbol{\Delta}_{w_*} = \mathbf{P}_\Phi \boldsymbol{\Delta}_{w_*}$$





- $\bar{\alpha}_n$ decreases when more data is used.
- Error region ( ) in random feature space decreases

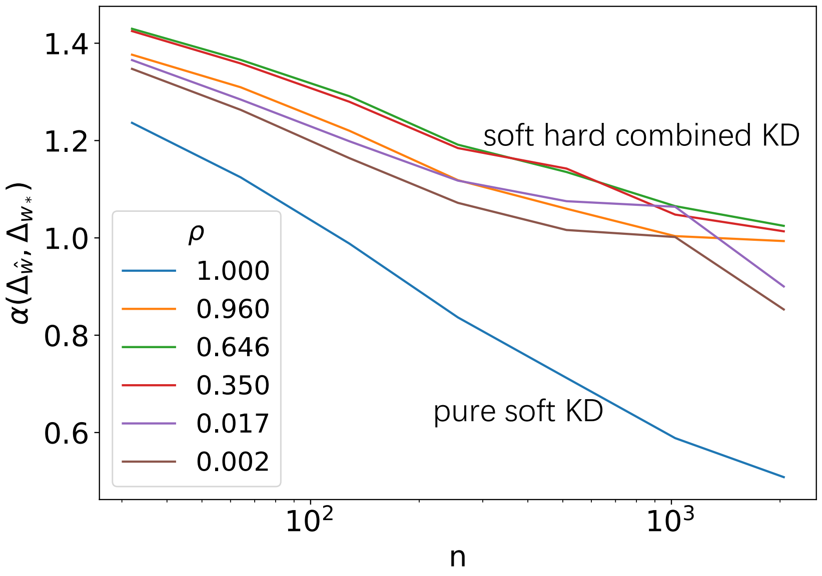
[4] Phuong, Mary, and Christoph Lampert. "Towards understanding knowledge distillation." ICML 2019.

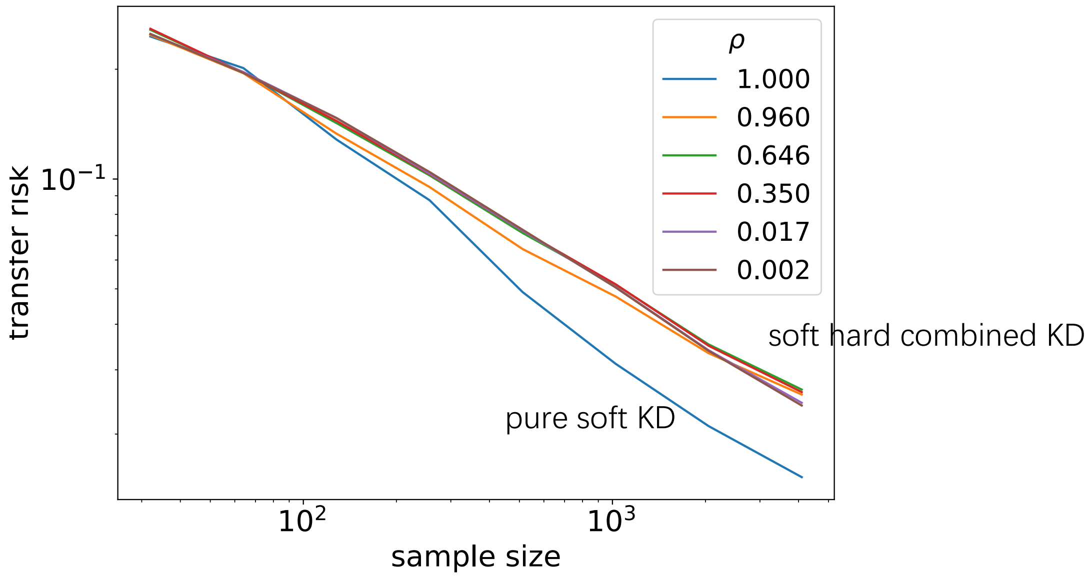
Result 1: Transfer Risk Bound

- **Theorem 1 (Risk bound):**

$$\mathcal{R}_n \leq p(\frac{\pi}{2} - \bar{\alpha}_n),$$

  - n: sample size,
  - $\bar{\alpha}_n = \bar{\alpha}(\Delta_{w_*}, \Delta_{\hat{w}})$: angle between oracle weight change Δ_{w_*} and student's weight change $\Delta_{\hat{w}}$,
  - $p(\beta)$: pdf of angle between random feature $\phi(x)$ and oracle Δ_{w_*}.
- **Good generalization happens when smaller angle is achieved with same amount of data.**
  - faster angle converging speed for pure soft distillation.
  - explains the fast converging error for pure soft distillation.





Result 2 & 3: Data Inefficiency and Imperfect KD

- **Definition (Data inefficiency):** the increasing speed of the norm of weight change,

$$\mathcal{I}(n) = n\big[\ln \mathbb{E}||\Delta_{\hat{w},n+1}||_2 - \ln \mathbb{E}||\Delta_{\hat{w},n}||_2\big] \approx \frac{\partial \ln ||\Delta_{\hat{w},n}||_2}{\partial \ln n}$$

  - characterizes the converging speed of angle $\bar{\alpha}_n = \bar{\alpha}(\Delta_{w_*}, \Delta_{\hat{w}})$.
  - more inefficient ➪ task is harder to train.

Result 2 & 3: Data Inefficiency and Imperfect KD

- **Definition (Data inefficiency):** the increasing speed of the norm of weight change,

$$\mathcal{I}(n) = n\left[\ln \mathbb{E}||\Delta_{\hat{w},n+1}||_2 - \ln \mathbb{E}||\Delta_{\hat{w},n}||_2\right] \approx \frac{\partial \ln ||\Delta_{\hat{w},n}||_2}{\partial \ln n}$$

  - characterizes the converging speed of angle $\bar{\alpha}_n = \bar{\alpha}(\Delta_{w_*}, \Delta_{\hat{w}})$.
  - more inefficient ➪ task is harder to train.

- Two factors that reduces data inefficiency
  1. early stopping epoch of teacher network,
  2. higher soft ratio ρ.
  - both may have a smoothing effect on student output function.

Result 2 & 3: Data Inefficiency and Imperfect KD

- **Definition (Data inefficiency):** the increasing speed of the norm of weight change,

$$\mathcal{I}(n) = n\big[\ln \mathbb{E}||\Delta_{\hat{w},n+1}||_2 - \ln \mathbb{E}||\Delta_{\hat{w},n}||_2\big] \approx \frac{\partial \ln ||\Delta_{\hat{w},n}||_2}{\partial \ln n}$$

  - characterizes the converging speed of angle $\bar{\alpha}_n = \bar{\alpha}(\Delta_{w_*}, \Delta_{\hat{w}})$.
  - more inefficient ➪ task is harder to train.

- Two factors that reduces data inefficiency
  1. early stopping epoch of teacher network,
  2. higher soft ratio ρ.
  - both may have a smoothing effect on student output function.

- **Imperfect teacher**
  - Our risk bound and data inefficiency assumes teacher is 100% accurate (perfect)
    - results in a favor in pure soft distillation
  - If teacher have a chance of mistake,
    - hard labels can partially correct the sign of student logits
    - hard labels can reduce $\bar{\alpha}_n = \bar{\alpha}(\Delta_{w_*}, \Delta_{\hat{w}})$

Conclusion

- Transfer risk bound under NTK settings.
- Data inefficiency
  - early stopping and higher soft ratio are beneficial for perfect distillation.
- Hard labels are need in imperfect distillation as a trade-off against teacher's mistake.

See our paper at https://arxiv.org/abs/2010.10090