Efficient Estimation of Influence of a Training Instance

Sosuke Kobayashi
Tohoku University
Preferred Networks, Inc.

Sho Yokoi   Jun Suzuki   Kentaro Inui
Tohoku University
RIKEN AIP

SustaiNLP 2020

Introduction

- **Goal**: estimate the influence of a training instance z_i
  = "how a model prediction or loss would be changed
  IF z_i was NOT used for training"

$$= L(f_{D \setminus \{z_i\}}, z_{\text{target}}) - L(f_D, z_{\text{target}})$$

- **What for?**
  - **Interpretability**: analyze bad training instances
    contributing to error prediction
  - **Data filtering**: ignore bad training instances
    contributing to high validation error

Introduction

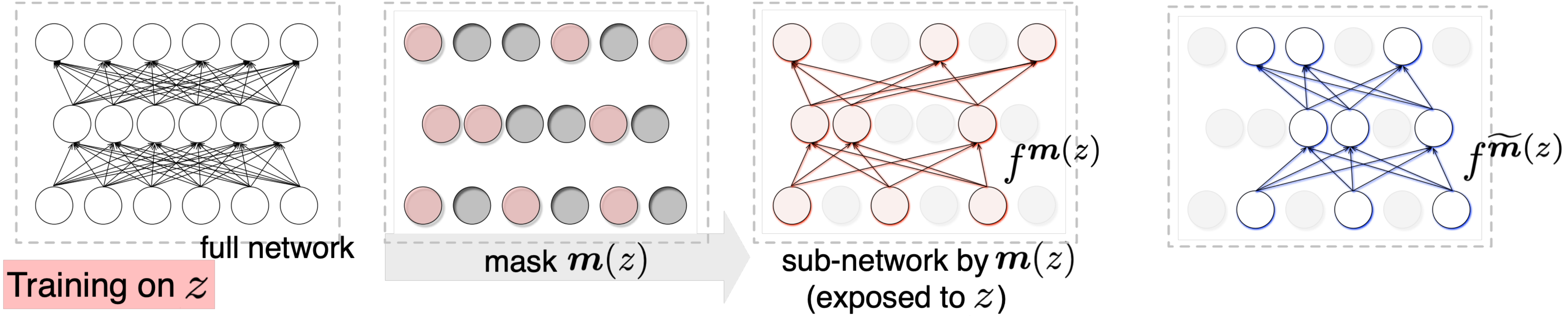
- **Goal**: estimate the influence of a training instance z_i
  = "how a model prediction or loss would be changed
  IF z_i was NOT used for training"

- **Naïve approach**: leave-one-out retraining: train two
  models on two dataset with or without the instance z_i
  → However, computation cost is high…

- **Our Approach**: estimate the influence based on
  two dropout sub-networks,
  which learned z_i or not

Idea: Dropout Makes Ignorant Sub-networks

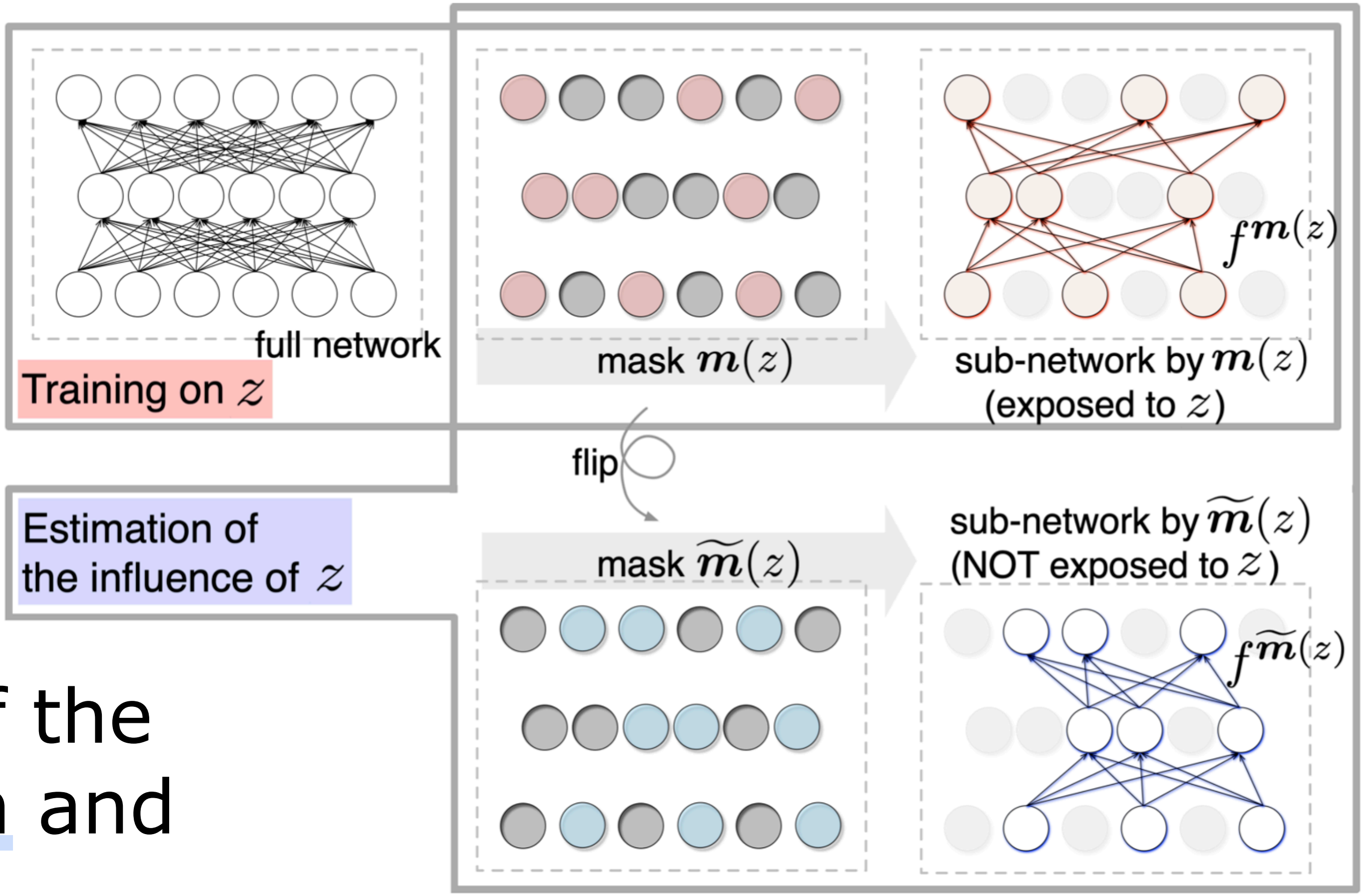
- A random set of parameters is zero-masked at each update



- Insight: the hidden (pruned) sub-network with the zero-masked parameters does NOT learn the instance
- If we *deterministically* use the same mask for an instance, its hidden sub-network NEVER learn the instance

Proposed: Turn-Over Dropout

- Make instance-specific dropout masks
for each instance
- Train a model with the instance-specific masks and dropout
- Compare the outputs of the instance-specific hidden and exposed sub-networks



$L(f_D^{\widetilde{\boldsymbol{m}}(z_i)}, z_{\text{target}}) - L(f_D^{\boldsymbol{m}(z_i)}, z_{\text{target}})$ instead of $L(f_{D\backslash\{z_i\}}, z_{\text{target}}) - L(f_D, z_{\text{target}})$

Challenge: Instance-specific Dropout Masks

- Random mask for each instance $z \in D$
  → naively, it costs $O(|D||\theta|)$ to store…

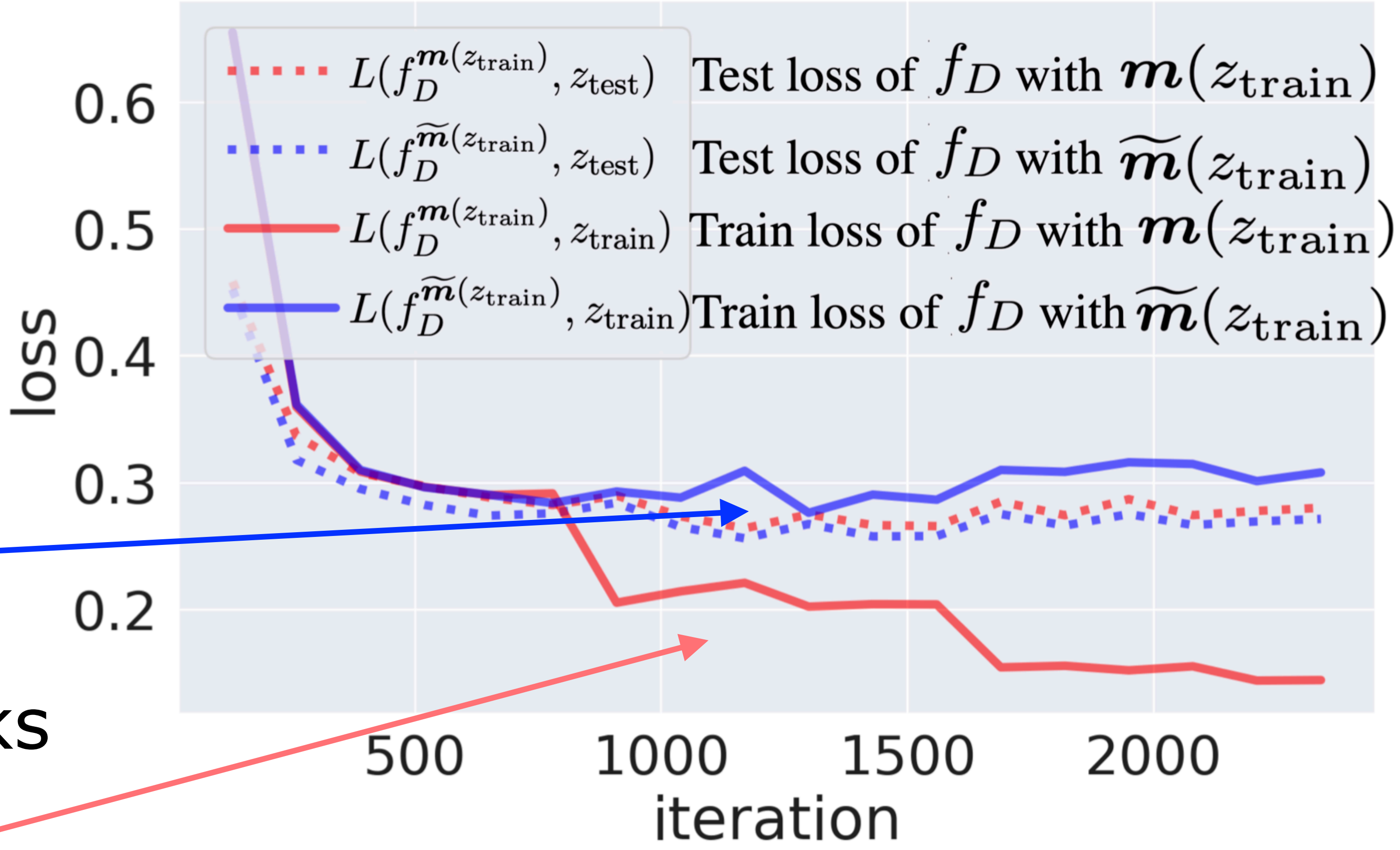
- However, instead of storing it,
  we can deterministically generate it from a random seed
  (e.g., we can set instance indices as the seeds)
  → reduced to $O(1)$

Thus,
we can use turn-over dropout with minimum additional costs
on top of the usual training of a single model

Experiment: Hidden Sub-networks Didn't Learn the Corresponding Instances? → Actually, Yes!

Dotted: test loss
Solid: training loss

- Hidden sub-networks did NOT overfit to training dataset (blue solid)
- Exposed sub-networks did overfit as usual (red solid)



Loss curve when finetuning BERT on SST-2 with turn-over dropout

Experiment: Interpretation of Error Prediction

- Analyze training instances with the largest influences for the misclassified label → we can guess the reason

e.g.

- The badly-influential training instances share the phrase "ch ##rist" with the test instance

Test input:

- why do some people assume they know who the ask ##er is , based on the question he asks ? from reading back the answers that i get on my questions , i am now officially a m ##us ##lim - ch ##rist ##ian - b ##udd ##his ##t , a straight … it ok with you that i still don ' t have an identity - crisis ?

[Society & Culture → Business & Finance]

Influentials:

- is ch ##rist ##mas eve in the evening or is it just the day before ch ##rist ##mas ? my sister thinks that ch ##rist ##mas eve …
- how i can be ch ##rist ##ain ? i want to be real ch ##rist ##ain , how i can be ch ##rist ##ain ask ch ##rist into …

Yahoo! Answers question topic classification

Experiment: Interpretation of Error Prediction

- Analyze training instances with the largest influences for the misclassified label → we can guess the reason

  e.g.

- The badly-influential training instances share the similar visual appearances (shape, color, layout) individually

CIFAR-10 object recognition

Experiment: Data Filtering for Domain Adaptation

- Remove training instances with the largest negative influences on validation set
  → retraining on the filtered set is improved

	Accuracy (%)	Loss
1% Random Removal	76.8 ± 1.1	0.521 ± 0.030
No Cleansing	77.0 ± 0.9	0.536 ± 0.063
1% Cleansing	$\mathbf{78.3 \pm 0.2}$	$\mathbf{0.484 \pm 0.008}$

Training set: Movie review (to be filtered)
→ Validation/Test set: Electronics product review

Conclusion

- **Goal**: estimate the influence of a training instance
  = "how a model prediction or loss would be changed
  IF it was not used for training"

- **Our Approach**: estimate it using dropout as
  generator of instance-specific ignorant sub-networks
  - worked for data filtering and model interpretation
  - is the most efficient ever (see the paper in detail)