# Challenges in MNLI

Haoming Jiang

# Content

- Label Noise, Label Uncertainty, and Overfitting
- Conditional Language Model

# Label Noise, Label Uncertainty, and Overfitting

- Label Noise:
  - Training Set: 1 labeler
  - Dev Set: 5 labeler → well calibrated
  - Label Noise (% of first label ≠ calibrated label): **10%**

- Label Uncertainty:

Certain Label ~58%

Uncertain Label ~42%

| Contradict | Neutral | Entail | Match | Mismatch |
|---|---|---|---|---|
| F | F | T | 0.221 | 0.229 |
| F | T | F | 0.143 | 0.149 |
| T | F | F | 0.223 | 0.231 |
| F | T | T | 0.189 | 0.183 |
| T | F | T | 0.043 | 0.037 |
| T | T | F | 0.150 | 0.144 |
| T | T | T | 0.030 | 0.028 |

# Example

- Look, there's a legend here.
- See, there is a well-known hero here.

- Labels: Entailment Neutral Entailment Entailment Neutral

- Entailment or Neutral?
- Legend = Well-known hero?

# Motivating Experiment: Exchanging premise and hypothesis

- Original Idea: Augment data by cycle consistency

$$G(x,y) = \text{contradict} \Leftrightarrow G(y,x) = \text{contradict}$$
$$G(x,y) = \text{neutral} \Leftrightarrow G(y,x) = \text{neutral}$$
$$G(x,y) = \text{entail} \Leftrightarrow G(y,x) = \text{entail by}$$

- Results:
  - Augmented with the above rule: -1.3% (Baseline 84.6%)

# Motivating Experiment: Exchanging premise and hypothesis

- The cycle consistency does not hold in general

$$G(x,y) = \text{contradict} \Rightarrow G(y,x) = \text{contradict?}$$
$$G(x,y) = \text{neutral} \Rightarrow G(y,x) = \text{neutral?}$$
$$G(x,y) = \text{entail} \Rightarrow G(y,x) = ?$$

- Example:
- The document is signed. Director signed the document. (neural)
- Director signed the document. The document is signed. (entail)

# Motivating Experiment: Exchanging premise and hypothesis

- Tried more:
  - contradict ⇒ contradict; neutral ⇒ neutral by; entail ⇒ entrail by: -0.7%
  - And some other tricks trying to use the augmented data, all hurt the performance at different level.
- Why? Augmented data induce more noise
- We need to handle the noise in the original data & augmented data.

# Handling Label noise and label uncertainty

- 1. Prompout: (84.6%)
- 2. MC Dropout:  (85.07%)
- 3. Mean Teacher: (85.06%)
- 4. SWA : (85.06%)
- 5. Mean Teacher + MC Dropout: (85.24%, matched 85.49%, mis 84.98%)
- …
- (Baseline 84.6%)

# Handling Label noise and label uncertainty

- Questions:

- How these method works?
- How can we do better?

# A unified view of calibrating the certainty

- Let's consider cross entropy loss:

$$l(x_i, y_i; \theta) = -\log P_{y_i}(x_i; \theta)$$

- Gradient Descent Direction :

$$-\frac{\partial l}{\partial \theta} = \underbrace{\boxed{\frac{1}{P_{y_i}}}}_{\text{Reward}} \frac{\partial P_{y_i}}{\partial \theta} + \sum_{j \neq y_i} \underbrace{\boxed{0}}_{\text{Reward}} * \frac{\partial P_j}{\partial \theta}$$

# A unified view of calibrating the certainty

- Mean Teacher as an example:

$$\theta_t\text{: moving average of } \theta$$

Ensemble of trajectory

- Loss:

$$l(x_i, y_i; \theta) = -\log P_{y_i}(x_i; \theta) + \frac{1}{K}\left|\left|P_j(x_i; \theta) - P_j(x_i; \theta_t)\right|\right|_2^2$$

- Gradient Descent Direction:

$$-\frac{\partial l}{\partial \theta} = \boxed{[\frac{1}{P_{y_i}} - \frac{1}{K}(P_{y_i} - \tilde{P}_{y_i})]}\frac{\partial P_{y_i}}{\partial \theta} + \sum_{j \neq y_i} \boxed{[0 - \frac{1}{K}(P_j - \tilde{P}_j)]} * \frac{\partial P_j}{\partial \theta}$$

Reward

# A unified view of calibrating the certainty

- Reward Analysis:

$$\begin{array}{lcc} \text{Training Label:} & \boxed{\begin{array}{c} \dfrac{1}{P_{y_i}} \\ 0 \end{array}} & \boxed{\begin{array}{c} -\dfrac{1}{K}\left(P_{y_i} - \tilde{P}_{y_i}\right) \\ -\dfrac{1}{K}\left(P_j - \tilde{P}_j\right) \end{array}} \\ \text{Other Label:} & & \end{array}$$

| Reward From training data $R_{train}(\mathrm{P})$ | Calibration From Teacher $R_{teach}(P, \tilde{P})$ |
|---|---|

# A unified view of calibrating the certainty

| Method | $R_{train}(\mathrm{P})$ | | $R_{teach}(P,\tilde{P})$ | Source of $\tilde{P}$ | Inference $\boldsymbol{\theta}$ or $\boldsymbol{\theta_t}$? |
|---|---|---|---|---|---|
| | labels | Others | | | |
| Naive | $1/\mathrm{P}$ | $0$ | $0$ | None | Student |
| Mean Teacher | $1/\mathrm{P}$ | $0$ | MSE: $-\ (P-\tilde{P})/K$ | Mean/Ensemble of trajectory | Student |
| MC Dropout | $1/\mathrm{P}$ | $0$ | $0$ | Dropout Ensemble | Teacher |
| Virtual Adversarial | $1/\mathrm{P}$ | $0$ | Cross Entropy: $\tilde{P}/P$ | Adversarial Input | Student |
| Data Aug.: mixup | $1/\mathrm{P}$ | $0$ | $\tilde{P}/P$ | Augmented Input | Student |
| Stoch Weight Avg. | $1/\mathrm{P}$ | $0$ | $0$ | Mean/Ensemble of trajectory | Teacher |
| Knowledge Distill. | $0$ | $0$ | $\tilde{P}/P$ | Ensemble | Student/Teacher |
| Label Smoothing | $\dfrac{1-\frac{\epsilon(\mathrm{K}-1)}{K}}{\mathrm{P}}$ | $\epsilon/\mathrm{P}$ | $0$ | None | Student |
| Prior Based: Promptout | Adjusted based on prior (updated) | | $0$ | None | Student |

# A unified view of calibrating the certainty

- These methods are designed for label noise, label uncertainty, adversarial attack, semi-supervised learning, domain adaptation, generalization, … , and have internal connection.
- What do we learn from this view?
  - Directly adjust $R_{train}(\mathrm{P})$ based on prior is not working well, we need to calibrate based on each instance
    - prompout, label smoothing, …
  - Ensemble Teacher is useful
    - Knowledge distillation, mean teacher, MC dropout, adversarial training, data augmentation…

# How can we do better?

- 1. Better/More ensemble/self-ensemble ($\tilde{P}$)
  - Borrow ideas from population learning: diverse self-ensemble
- 2. Better $R_{train}(\mathrm{P})$ $R_{teach}(P, \tilde{P})$ :
  - Dynamically adjust weight of each instance
- 3. Sampling instead of adjusting weight to accelerate training and achieving better performance.

# How can we do better?

- Useful Observations:
  - Overconfidence in Deep Neural Network (C Guo et al., 2017)

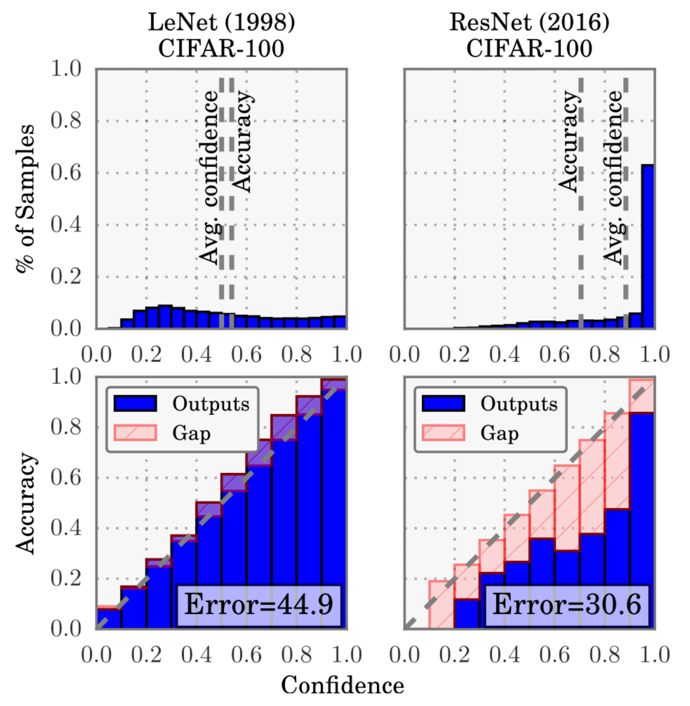


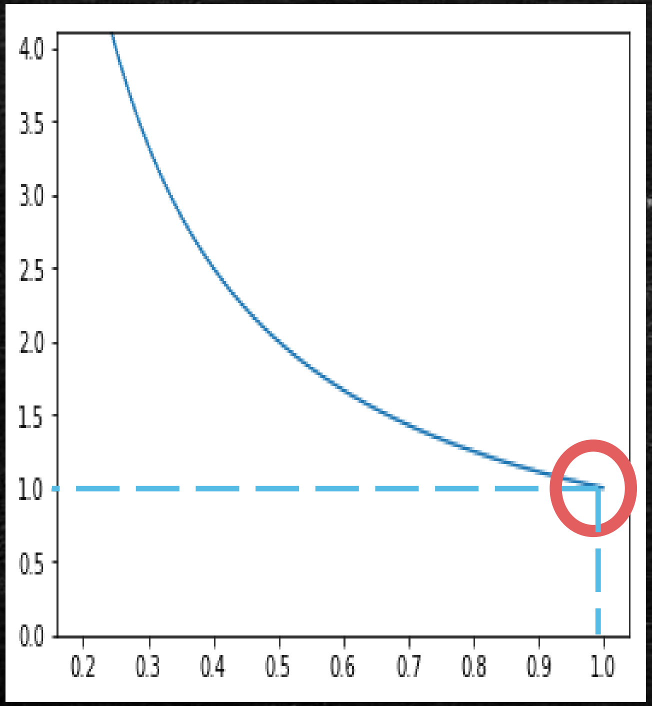

*Figure 1.* Confidence histograms (top) and reliability diagrams (bottom) for a 5-layer LeNet (left) and a 110-layer ResNet (right) on CIFAR-100. Refer to the text below for detailed illustration.

# How can we do better?

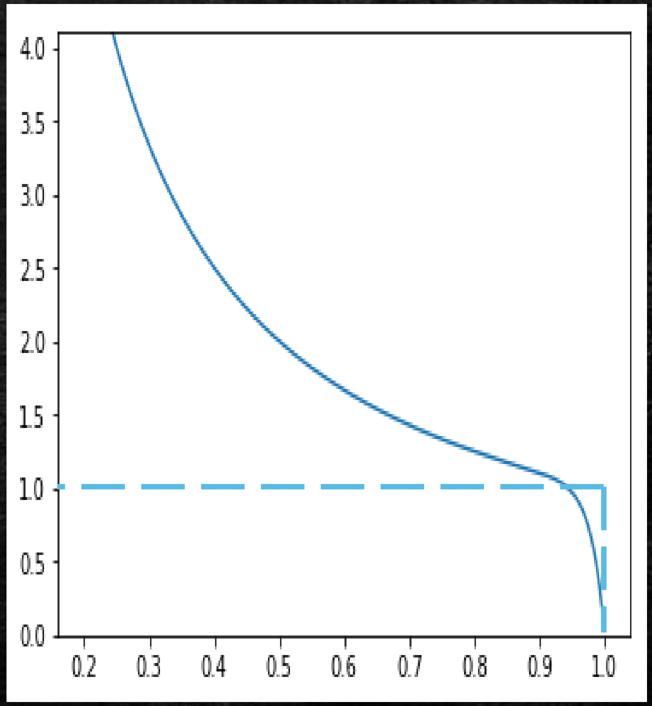
- Useful Observations:
  - Overconfidence in Deep Neural Network (C Guo et al., 2017)

$$R_{train}(\mathrm{P}) = 1/P$$

Reason:
Keep pushing
Even already
confident

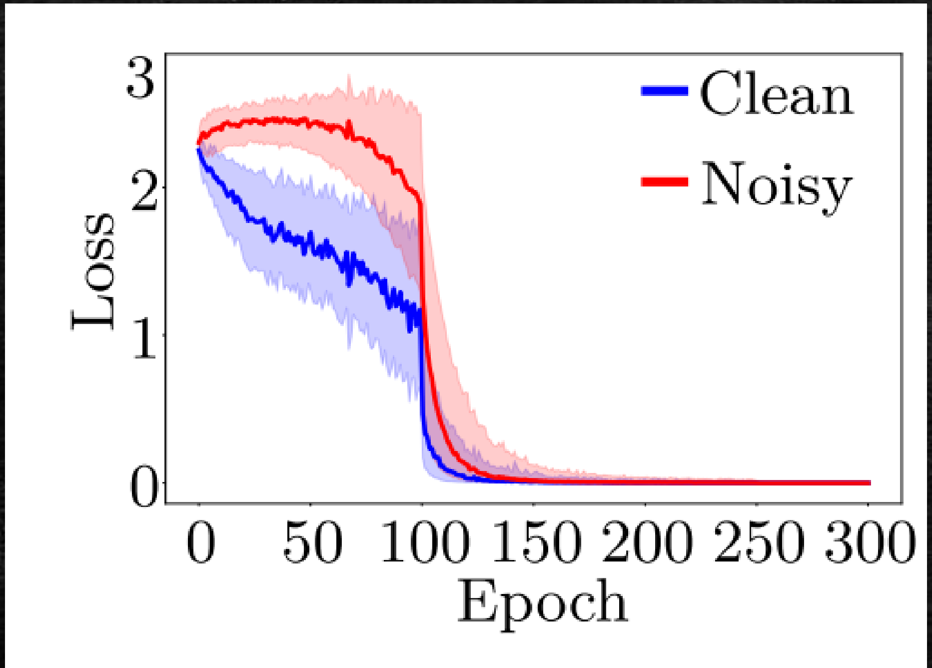



$$R_{train}(\mathrm{P}) = \frac{1}{P} - \exp(50(P-1))$$



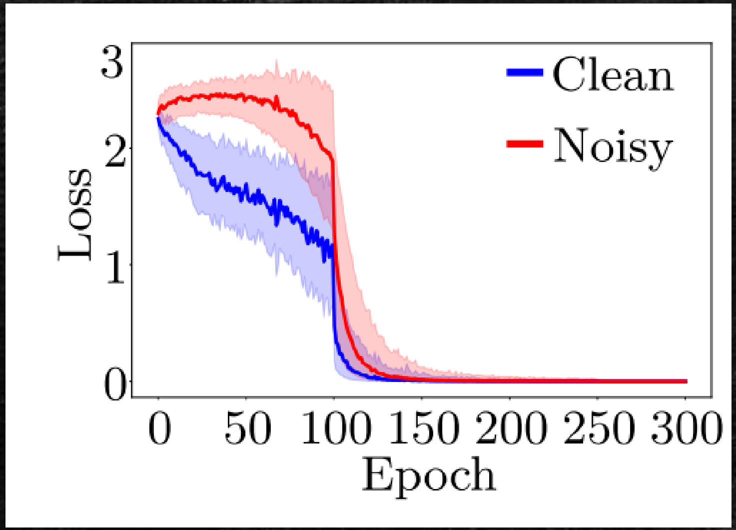
# How can we do better?

- Useful Observations:
  - Different speed of fitting easy and (hard/noise/uncertain) data (E Arazo et al., 2019)

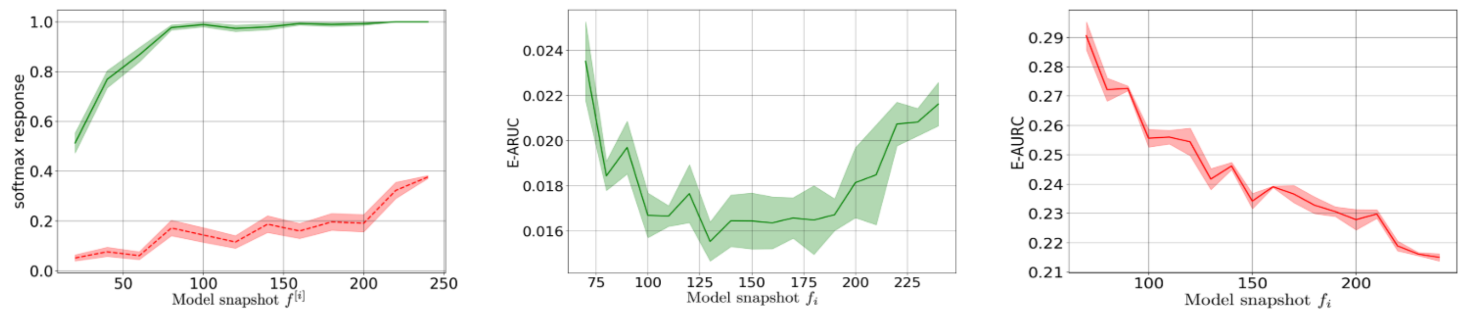


# How can we do better?

- Useful Observations:
  - Different speed of fitting easy and (hard/noise/uncertain) data




(E Arazo et al., 2019)




(a) (b) (c)

Figure 2: (a): Average confidence score based on softmax values along the training process. Green (solid): 100 points with the highest confidence; red (dashed): 100 points with the lowest confidence. (b, c): The E-AURC of softmax response on CIFAR-100 along training for 5000 points with highest confidence (b), and 5000 points with lowest confidence (c).

(Y Geifman et al., 2019)

# How can we do better?

- Useful Observations:
  - Different speed of fitting easy and (hard/noise/uncertain) data
- How to use this observation?
  - Related to the trajectory ensemble: mean teacher, …
  - Related to curriculum learning: learn the easy sample first and hard sample latter, and maybe come back to the mis-understood samples.

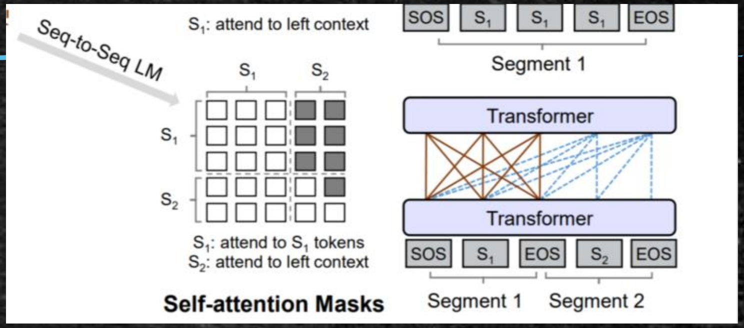
# Content

- Label Noise, Label Uncertainty, and Overfitting
- Conditional Language Model

# Conditional Language Model

- Recap:
- Given a sentence $x$, using conditional language model generate reliable paired sentence $x'$ and label $l$ to augment training data of MNLI with $(x, x', l)$

# Conditional Language Model

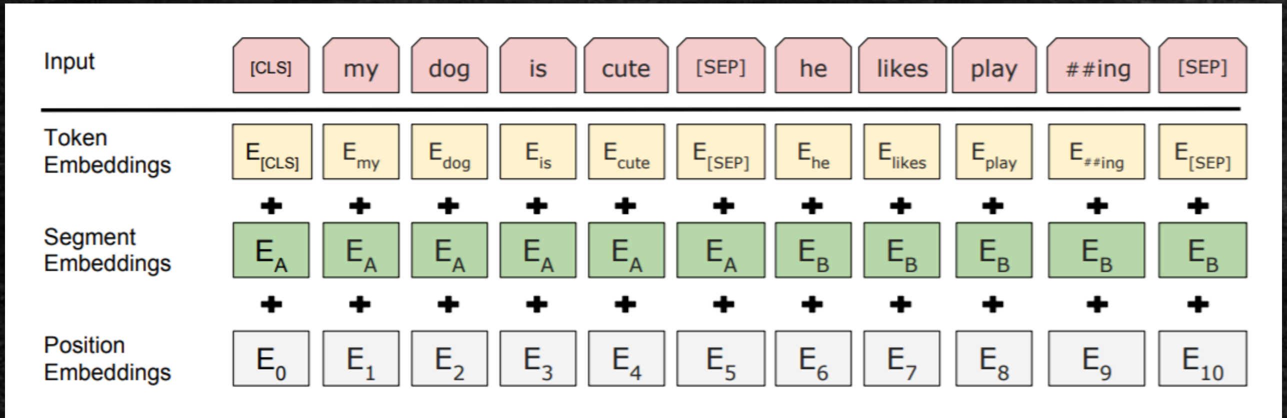



- Conditional language by masking the attention.
- Model 1:

[CLS] sentence1 [SEP] sentence2 [SEP]

[Entail/Neutral/Contradict] sentence1 [SEP] sentence2 [SEP]

- Model 2:




+ NLI embedding

# Conditional Language Model

- Result:
- None of them attend to the condition, i.e.
  $$P(y|x,c=contradict) \simeq P(y|x,c=neutral) \simeq P(y|x,c=entail)$$
- Possible reason: NLI relation is too hard to be capture by simple Language Model.

# Conditional Language Model

- Possible Directions:
  - Integrate more supervision to help the learning of conditional language model.
  - Generate $x'$ from $x$ just by replacing some word in $x$
    - Apply random mask to $x$
      $$x_1, x_2, x_3, x_4, x_5, \ldots \Rightarrow x_1, [MASK], x_3, [MASK], [MASK], x_5, \ldots$$
    - Use top k word from BERT to fill in $[MASK]$ to get $x'$
    - Apply some filtering
    - Augment original data with $(x, x')$
    - Use semi-supervised learning to learn from both labeled data and unlabeled data.