1 Lower Bound of Cross-Entropy Loss Estimation

- date: 2025-06-13 14:22
- tags: NLP

2 Introduction

Usually, we use the cross-entropy loss to fine-tune Large Language Models (LLMs). However, we often find that there could be a sharp drop of the loss at the second epoch of tuning as shown in Figure 1. And we all know that is mainly because the LLM may have "memorized" the training data in the second epoch, which is also called "overfitting". But how can we verify this hypothesis? In other words, how can we understand the "overfitting" in NLP tasks?

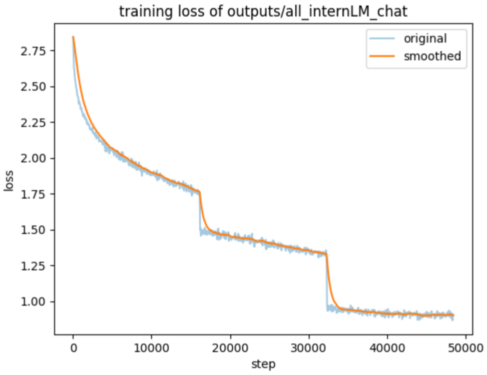

Figure 1: The loss curve of fine-tuning a full-parameter LLM with cross-entropy loss.

3 What is Overfitting in Natural Languages?

The overfitting phenomenon in some other tasks like classification, recommendation, recogonition, etc., is that the model have memorized the features, users behaviors, or shapes of data in training set. But most of the time, we don't know whether the model is overfitting or not. The only thing we can do is to use the validation set to check the performance of the model.

But in natural languages, the overfitting could be observed. Because we human could understand the text and we are naturally general sentence generators. With a given input, if the uncertainty of a fine-tuned LLM to generate the next token is lower than that of a general sentence generator, which could be one of "us", we could easily tell that the model is overfitting. It is very intuitive and easy to understand. For example, when we have recited a poem in our high school class, the uncertainty of us to speak out the poem could be very low. And that's the result of memorizing the poem. So as to the LLM.

And the cross-entropy loss is exactly the uncertainty of the LLM to generate the next token. So not like the loss in other tasks, the cross-entropy loss can also be used to detect the overfitting phenomenon in natural languages.

4 Which Kind of Cross-Entropy Loss is Overfitting?

First, we have the following theorem:

Theorem 1. *Given a ground truth data distribution Q and a model predicted distribution P, where $P(i)$ and $Q(i)$ represent the probability of a data point in P and Q respectively, the cross-entropy loss is defined as:*

$$\mathcal{L}(P, Q) = -\sum_{i=1}^{n} Q(i) \log P(i), \tag{1}$$

and can be rewritten as:

$$\mathcal{L}(P, Q) = \mathcal{H}(Q) + \mathcal{D}_{KL}(Q||P), \tag{2}$$

where $\mathcal{H}(Q)$ is the entropy of Q and $\mathcal{D}_{KL}(Q||P)$ is the KL-divergence between distribution Q and P.

Here is the proof.

$$\mathcal{H}(Q) = -\sum_{i=1}^{n} Q(i) \log Q(i), \tag{3}$$

$$\mathcal{D}_{KL}(Q||P) = \sum_{i=1}^{n} Q(i) \log \frac{Q(i)}{P(i)}. \tag{4}$$

Then, we have:

$$\begin{aligned}\mathcal{H}(Q) + \mathcal{D}_{KL}(Q||P) &= -\sum_{i=1}^{n} Q(i) \log Q(i) + \sum_{i=1}^{n} Q(i) \log \frac{Q(i)}{P(i)} \\ &= -\sum_{i=1}^{n} Q(i) \log P(i) \\ &= \mathcal{L}(P, Q).\end{aligned} \tag{5}$$

So, according to the Theorem 4, the lower bound of the cross-entropy loss is the approximately the entropy of the ground truth data distribution $\mathcal{H}(Q)$, which is the case when the KL-divergence is 0. And if the cross-entropy loss is lower than the entropy of the ground truth data distribution, which is the lower bound of the cross-entropy loss, it means that the model is overfitting. In other words, the model is memorizing the training data and that leads to the uncertainty of the model to generate the next token is lower than that of a general sentence generator to generate the ground truth data. Note that here we assume the ground truth data comes from the a general sentence generator, usually the blogs from the websites or scientific papers from human. And that's why we can consider $\mathcal{H}(Q)$ as the uncertainty of a general sentence generator to generate the ground truth data.

However, what is the exact value of $\mathcal{H}(Q)$? Generally, it is very hard to compute, but according to the assumption, the uncertainty to generate a sentence from a general sentence generator should be the same as that of us human. Because we are naturally general sentence generators. An interesting finding in Shannon [1951] demonstrates that the entropy of an English character is approximately 2.62 bit, which is equal to 1.82 nat. However, this value is calculated by gathering results from real human with given the word of each English character. So the conditions are very short, compared with the sentence used in next token prediction, and the English characters are usually shorter than tokens in LLMs, both of which lead to a larger value of general entropy. Maybe in the futher, based on a large corpus of human-generated web-data, we can estimate a more precise value of a general token-level entropy to verify the overfitting problem in NL generation tasks.

References

Claude E Shannon. Prediction and entropy of printed english. *Bell system technical journal*, 30(1): 50–64, 1951.