Summer research project report:

Statistical aspects of Large Language Models

Michelle Ching Ioana Popescu Nico Smith

Supervisors: Tianyi Ma, William G. Underwood, Richard J. Samworth

Large language models (LLMs) such as ChatGPT have recently become a central topic of discussion, both in research and in everyday life. Their underlying machine learning mechanism, called the **transformer**, was first introduced by researchers at Google DeepMind in 2017 and has since revolutionized the way we process language. Despite their remarkable practical success, the reasons behind why these models work as well as they do remain only partially understood. Most insights are still **empirical** rather than theoretical, which fuels ongoing skepticism and motivates deeper study. Motivated by these questions, I joined my peers Michelle Ching and Nico Smith in a summer project aimed at developing a rigorous statistical understanding of various aspects of LLMs.

1 Next-token prediction

Large language models (LLMs) generate text by predicting the next **token** (i.e. word or word-piece) in a sequence, given the context of the preceding ones. Mathematically, the model assigns a probability distribution over its vocabulary (of size in the tens of thousands), and the next token is then chosen from this distribution.

A simple strategy is to always select the token of highest probability (greedy decoding), while more sophisticated methods, such as top-k sampling, restrict attention to the k most likely candidates to encourage diversity. However, high probability (reflecting high model confidence) does not always correspond to factual accuracy, and in open-ended tasks (such as story generation) diversity can be desirable.

In the first stage of our project, we explored whether hypothesis testing methods, applied to the tail of the logit distribution (the pre-softmax scores that get turned to probabilities), could provide a principled alternative for token selection. Due to computational constraints and unpromising empirical results, we ultimately set this direction aside and turned to a different line of investigation.

2 In-context learning

A phenomenon that empirically emerges in LLMs is that they can solve tasks without being explicitly trained on them. This is called **in-context learning**: if you provide a few input–output pairs in the prompt, for example, English words and their French translations, the model can then translate a new English word, even if it has never seen this exact task of translation before. People have been trying to answer: mathematically, why can a trained model display this kind of generalization?

The core components of LLMs give some clues. Sentences are broken up into words, and then words are represented as **embeddings**, meaning high-dimensional vectors (often thousands of dimensions). A transformer processes these embeddings through a mechanism called **attention**, which lets tokens interact and thus build context. Unlike traditional neural networks, this architecture is expressive enough to capture complex relationships between tokens.

A leading hypothesis is that in-context learning performs a form of **regression** on embeddings. In the translation example, the model may implicitly map English embeddings to French embeddings using a kind of linear regression learned from the prompt examples. This suggests that transformers are not just memorizing patterns, but actually performing a statistical algorithm over the examples provided.

To probe this idea, we can study transformers in a simpler setting: can they perform regression on tabular data, not just text embeddings? If they succeed, it would strengthen the view that transformers can implement general regression-like algorithms inside their attention layers.

Our key finding is that transformers are not limited to memorizing tasks, they can in fact implement general-purpose algorithms such as gradient descent or linear regression. In our research, we focused specifically on **local polynomial regression**. Given a few examples, transformers can identify a suitable strategy and then apply it to entirely new inputs. At a theoretical level, we now understand that transformers are capable of reproducing classic non-parametric regression techniques, which means that in practice a trained transformer should perform at least as well as these established methods. This helps explain why large language models show such remarkable adaptability when faced with unfamiliar challenges.