ChatXGB

Seth Chatterton and Jason Jia
setcha@mit.edu, jasonjia@mit.edu

Problem Summary

Can tree-based methods perform autoregressive language modeling (next token prediction) as well as neural networks?

Motivation:

- Transformer models based on neural networks such as GPT have gained immense popularity as the model of choice for autoregressive language modeling (predicting the next word from a set of words in a given context).
- However, they are computationally expensive to train. On the other hand, tree-based models such as XGBoost, RF, CART and OCT can be easier to train and may be able to generate the same predictions. This is also in light of the proof in class that neural networks are equivalent to trees under certain assumptions.
- We would thus like to experiment with tree-based methods and compare their performance with established neural network-based methods for autoregressive language modeling.

Dataset we plan to use

We plan to use the Cleaned Alpaca Dataset[1], which is a slightly modified version of the dataset used to train the Alpaca large language model[2]. Alpaca is a fine-tuned version of the LLaMA[3] large language model. The data consists of instructions for the model, inputs for those instructions, and outputs that the model should try to recreate.

Methods

- Methods explicitly covered in class which we plan to use: XGBoost (CatBoost), Random Forest, CART, OCT
- Methods which are not explicitly covered in class which we plan to use: Neural Networks (Transformer and/or LSTM/GRU/RNN)
- Metrics we plan to evaluate: token error rate, perplexity loss, training time, model size, inference time, interpretability
- Potentially novel method for generating token embeddings from autoregressive tree-based models:
    - Initialize embeddings for every token randomly
    - Train an initial model to predict the next embedding from the previous embeddings in the context window.
    - For every training example, predict the next token embedding and adjust the current token embedding in the direction of the predicted token embedding
    - After every epoch of going through all training data, retrain the model on the new token embeddings
    - This procedure jointly learns models and dense representations of words, potentially leading to greater generalization

Challenges, and ideas to overcome them

- Size of dataset required to get good results; training time
- Implementation of autoregressive language modeling for tree-based methods

[1] Ruebsamen, G. 2023. Cleaned Alpaca Dataset. https://github.com/gururise/AlpacaDataCleaned/tree/main
[2] Taori R et al. 2023. Stanford Alpaca: An Instruction-following LLaMA model. Github Repository https://github.com/tatsu-lab/stanford_alpaca
[3] Touvron H et al. 2023. LLaMA: Open and Efficient Foundation Language Model. arXiv preprint arXiv:2302.13971