Self-speculative decoding Stern et al. (2018) are, to the best of our knowledge, the first to suggest a speculative decoding scheme for faster inference. Our architecture replaces their linear prediction heads by transformer layers, but is otherwise similar. By reorganizing the order of the forward/backward, we can use all loss terms instead of stochastically picking one head for loss computation. Cai et al. (2024) present a more elaborate self-speculative decoding scheme that uses the top-k predictions of each head instead of the best one only. It can be used with the multi-token prediction models we train.

Multi-target prediction Multi-task learning is the paradigm of training neural networks jointly on several tasks to improve performance on the tasks of interest (Caruana, 1997). Learning with such auxiliary tasks allows models to exploit dependencies between target variables and can even be preferable in the case of independent targets (Waegeman et al., 2019). While more specifically tailored architectures for multi-target prediction are conceivable (Spyromitros-Xioufis et al., 2016; Read et al., 2021), modern deep learning approaches usually rely on large shared model trunks with separate prediction heads for the respective tasks (Caruana, 1997; Silver et al., 2016; Lample et al., 2022) like we do. Multi-target prediction has been shown to be a successful strategy in various domains, e.g. for learning time series prediction with more distant time steps in the future as auxiliary targets (Vapnik and Vashist, 2009) or for learning from videos with several future frames (Mathieu et al., 2016; Srivastava et al., 2016) or representations of future frames (Vondrick et al., 2016) as auxiliary targets.

7. Conclusion

We have proposed multi-token prediction as an improvement over next-token prediction in training language models for generative or reasoning tasks. Our experiments (up to 7B parameters and 1T tokens) show that this is increasingly useful for larger models and in particular show strong improvements for code tasks. We posit that our method reduces distribution mismatch between teacher-forced training and autoregressive generation. When used with speculative decoding, exact inference gets 3 times faster.

In future work we would like to better understand how to automatically choose n in multi-token prediction losses. One possibility to do so is to use loss scales and loss balancing (Défossez et al., 2022). Also, optimal vocabulary sizes for multi-token prediction are likely different from those for next-token prediction, and tuning them could lead to better results, as well as improved trade-offs between compressed sequence length and compute-per-byte expenses. Finally, we would like to develop improved auxiliary prediction losses that operate in embedding spaces (LeCun, 2022).

Impact statement

The goal of this paper is to make language models more compute and data efficient. While this may in principle reduce the ecological impact of training LLMs, we shall be careful about *rebound effects*. All societal advantages, as well as risks, of LLMs should be considered while using this work.

Environmental impact

In aggregate, training all models reported in the paper required around 500K GPU hours of computation on hardware of type A100-80GB and H100. Estimated total emissions were around 50 tCO2eq, 100% of which were offset by Meta's sustainability program.

Acknowledgements

We thank Jianyu Zhang, Léon Bottou, Emmanuel Dupoux, Pierre-Emmanuel Mazaré, Yann LeCun, Quentin Garrido, Megi Dervishi, Mathurin Videau and Timothée Darcet and other FAIR PhD students and CodeGen team members for helpful discussions. We thank Jonas Gehring for his technical expertise and the original Llama team and xFormers team for enabling this kind of research.

AlphaZero was something similar?