SUPER TINY LANGUAGE MODELS RESEARCH SERIES - DROPOUT

Dylan Hillier
A*STAR and Singapore Management University
das.hillier.2023@phdcs.smu.edu.sg

Leon Guertler
A*STAR and Nanyang Technological University
leon002@e.ntu.edu.sg

Bobby Cheng
A*STAR, Institute of Infocomm Research (I2R)
bobby_cheng@i2r.a-star.edu.sg

Cheston Tan
A*STAR, Institute of High Performance Computing (IHPC)
cheston-tan@i2r.a-star.edu.sg

ABSTRACT

In this work we explore the relevance of dropout for modern language models, particularly in the context of models on the scale of <100M parameters. We explore it's relevance firstly in the regime of improving the sample efficiency of models given small, high quality datasets, and secondly in the regime of improving the quality of its fit on larger datasets where models may underfit.

***K*eywords** dropout · language models · overfitting

1 Motivation

In our earlier work, titled 'Super Tiny Language Models'[1], we outlined our ambitions and approach for developing Super Tiny Language Models (STLMs) that are high performing in spite of having significantly smaller parameter counts. One of our listed strategies is experimentation over the effects of dropout.

First proposed by Srivastasa et al (2014) [2], dropout involves the probabilistic disabling of neurons – implemented by zeroing out the activations of a given layer. This is broadly equivalent to subsampling a smaller network, and then only performing gradient updates on this smaller network. As such a network trained with dropout is thought to be regularized through it's approximation of a larger ensemble of networks. While early uses of the technique had high dropout ratios of 0.5 [2] – dropping out half of the neurons on any given training pass, as data has been increasingly scaled, the dropout ratio has been scaled down, with models either using a ratio of 0.1, or for most modern large language models omitting it entirely [3]. For instance while early models like GPT-2 [4] used dropout throughout the model (at a rate of 0.1, in addition to L2-Normalisation), recent works like Llama-2 [5] don't use it at all.

1.1 Sample Efficiency

A large advantage of smaller language models, is that they can be trained with smaller amounts of data – in principle this should allow models to be trained with higher quality data due to the increased ease of collecting such data at a scale sufficient for optimal performance in the sense of scaling laws [6]. Indeed highly-performant small models such as ORCA [7] and Phi [8] have relied on high quality synthetic data to outperform other models at their scale.

Data efficient LLMs may also play a key role in empowering the modelling of under-resourced languages [9]. One potential motivation for using dropout on these models then, is to increase this sample efficiency by enabling samples to be used over multiple epochs. Repeated training of LLMs across the data has been show to cause the phenomenon of a 'Token-Crisis' - where multi-epoch trainings causes the performance of language models to degrade [3]. The promise of dropout then lies in its use as a regularization technique, and indeed it has been shown to be uniquely effective in this role for alleviating this token-crisis [3].

1.2 Reducing Underfitting

In addition to being useful for combating overfitting and the ability to train with smaller datasets, Liu et al. (2024) [10] report that dropout can also play a role in *combatting underfitting*. The authors report that during the beginning of training the variance of mini-batch gradient directions and error in the gradient with respect to the full-batch gradient are respectively far higher. Dropout can function to reduce this gradient variance and error during training. They argue that this has the effect of preventing it from overfitting to specific, high-norm batches. Accordingly the authors propose using dropout in early stages to reduce the effect of this gradient variance. They demonstrate that across a variety of models and optimizers over imagenet, the performance of a model can be increased for underfitting models.

2 Method

As described in Section 1, dropout with ratio p consists in setting the activations of a layer to 0 with chance p. In particular for a given sample in the training data we define the dropout layer as $\text{Dropout}_p(\mathbf{x})$, given $\mathbf{x} : \mathbb{R}^d$, $\mathbf{m} : \mathbb{R}^d \sim \text{Bernoulli}(p)$ independently generated for each sample, and layer

$$\text{Dropout}_p(\mathbf{x}) = \mathbf{m} * \mathbf{x}$$

, where $*$ is the elementwise multiplication operator.

2.1 Dropout Schedulers

While dropout itself is a simple enough concept, there may be some benefit to using a scheduler to mitigate any instability caused by switching the dropout on/off during training. Indeed in Liu et al.[10] the authors experiment with a linear scheduled dropout.

For the case of underfitting, we propose to investigate the following:

- Constant dropout ratio of 0.1
- Early dropout with a cutout after 5000 iterations
- Linear scheduled (decreasing) from 0.1 to 0
- Linear early schedule (decreasing) from 0.1 to 0 over 5000 iterations
- Triangular Dropout with 3 cycles

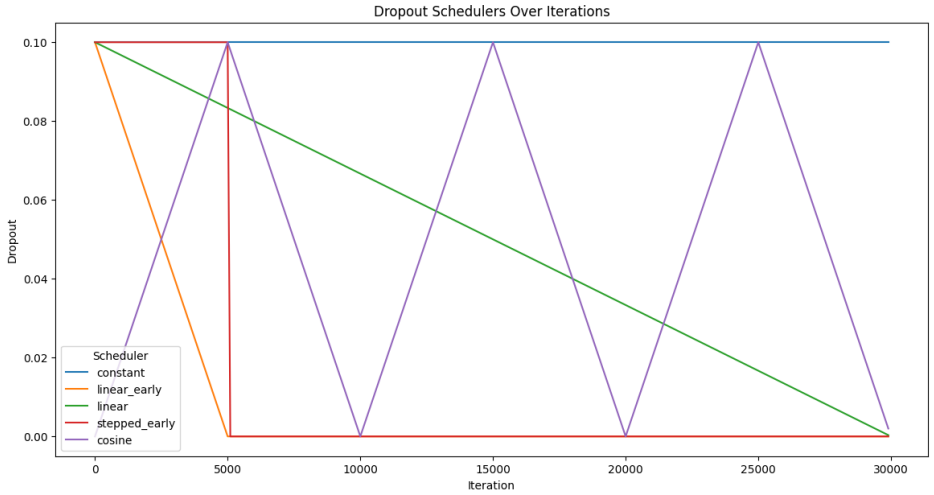

Figure 1: The dropout schedulers we plan to experiment over for combatting underfitting on openwebtext

For the case of overfitting, we will use a smaller training data set and the following regimes:

- Linear schedule (increasing) from 0 to 0.1
- Late dropout starting at 5000 iterations
- Triangular Dropout with 3 cycles

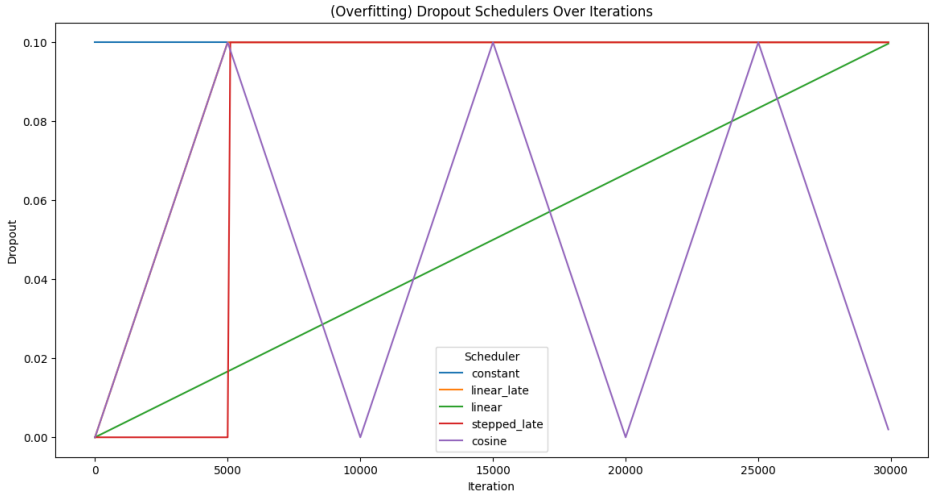

Figure 2: The scheduled dropout rates we plan to experiment over for combatting overfitting on simple english wikipedia

3 Pre-registration Disclosure

This paper is intended to act as a preregistered copy of the eventual research report. Ideally we would enter such a research report blind to partial results, however we have previously run several relevant experiments while developing our framework. Additionally this particular paper is largely recreating the results of Liu et al. [10] rather than attempting to show particularly new hypotheses, as such this rigour is somewhat less important. In particular we found dropout to improve modelling performance across the board for simple-wikipedia scale datasets and reduce performance for openwebtext scale datasets (given the 50 million parameter model size), motivating somewhat the experimental design.

References

[1] Dylan Hillier, Leon Guertler, Cheston Tan, Palaash Agrawal, Chen Ruirui, and Bobby Cheng. Super tiny language models, 2024.

[2] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[3] Fuzhao Xue, Yao Fu, Wangchunshu Zhou, Zangwei Zheng, and Yang You. To repeat or not to repeat: Insights from scaling llm under token-crisis. *Advances in Neural Information Processing Systems*, 36, 2024.

[4] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[5] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[6] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[7] Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*, 2023.

[8] Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

[9] Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*, 2022.

[10] Zhuang Liu, Zhiqiu Xu, Joseph Jin, Zhiqiang Shen, and Trevor Darrell. Dropout reduces underfitting. In *International Conference on Machine Learning*, pages 22233–22248. PMLR, 2023.