CodeGPT on XTC

Compressing a Code-Generation Model Using Knowledge Distillation for Layer Reduction and Extreme Quantisation.

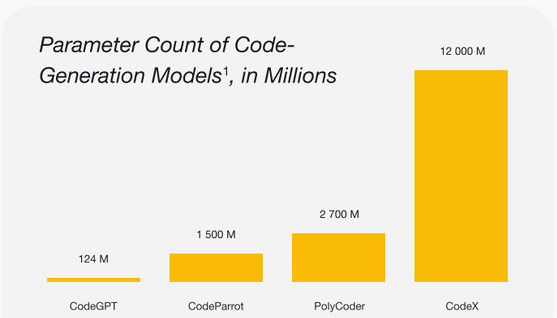
Aral de Moor
a.d.demoor@student.tudelft.nl

Ali Al-Kaswan & Maliheh Izadi
Supervisors

Arie van Deursen
Professor

1 Introduction

Large Language Models are increasingly prominent due to their human-like language capabilities. This led to the emergence of 'pair-programmer' tools, which assist developers by providing code auto-completion. However, their growing size makes them prohibitively expensive to run locally.



Compressing language models for resource-constrained devices is thus an active area of research. Notably, the XTC (extreme compression) pipeline achieves a 50x size reduction while maintaining 97.3% of the original model's accuracy[2]. We adapt XTC for a GPT-style code-generation model.

2 Compression Method

XTC consists of the following two steps facilitated by *knowledge distillation*: training a smaller student model from the outputs of a larger teacher model.

Lightweight Layer Reduction
Remove every other layer from the student's decoder stack.

Extreme Quantisation
Convert FP32 parameters in the student model to 1-bit weights and 8-bit activations.

To serve as a compression baseline, we fine-tune CodeGPT[3] for code-completion on 150K Python files[4]. We then create three compressed models: a quantised model; a 6-layer reduced model; and a hybrid model combining both techniques.

3 Evaluation

We evaluate the baseline and compressed models on the CodeXGLUE[3] benchmark test for line-level *code completion*, consisting of two metrics.

Exact Match
Percentage of perfect responses.

Edit Similarity
Measure based on how many single-character edits are required to fix the output.

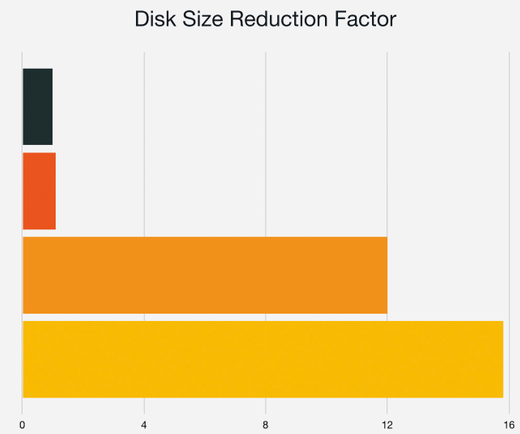
4 Results

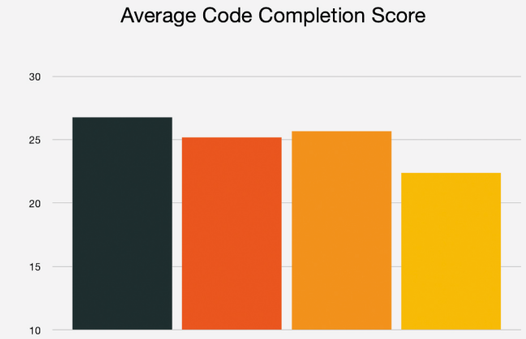
Hybrid layer reduction and quantisation can compress the 510 MB baseline model into 32 MB, while retaining 84% of the original accuracy.

Layer reduction nets a 2x inference speedup, and quantisation allows for a 12x compression alone.

15x
Size Reduction

84%
Accuracy Retention



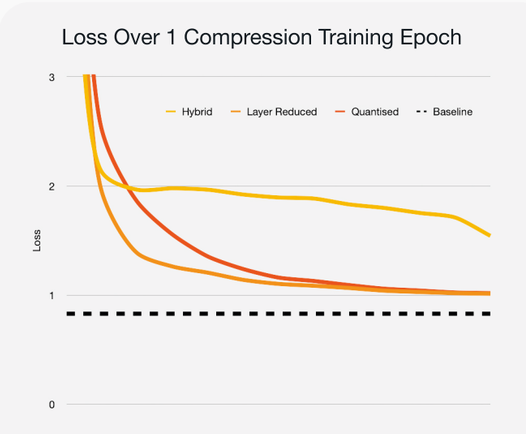


5 Discussion & Future Work

Novel compression techniques like XTC can be adapted to GPT-style models. This shows promise for compression on powerful GPUs, after which a model can be deployed locally. However, we identify key limitations in our findings:

Limited computing power restricts us to only 1 compression epoch, with a ten-fold faster learning rate. The below loss curves indicate volatile training, especially for the hybrid model. It would be worthwhile for subsequent studies to use the intended 18-epoch knowledge distillation.

The in-training nature of XTC means that the student model has to be retrained, which is computationally expensive. Especially for quantisation, future research could investigate hybrid post-training approaches.



1. F. F. Xu, U. Alon, G. Neubig, and V. J. Hellendoorn, 'A Systematic Evaluation of Large Language Models of Code'. 2022.
2. X. Wu, Z. Yao, M. Zhang, C. Li, and Y. He, 'Extreme Compression for Pre-trained Transformers Made Simple and Efficient'. arXiv, Jun. 03, 2022. doi: 10.48550/arXiv.2206.01859.
3. S. Lu et al., 'CodeXGLUE: A Machine Learning Benchmark Dataset for Code Understanding and Generation'. arXiv, Mar. 16, 2021. Accessed: May 16, 2023. [Online]. Available: http://arxiv.org/abs/2102.04664
4. https://huggingface.co/datasets/0n1xus/codexglue

TU Delft