TinyLlama: An Open-Source Small Language Model

Peiyuan Zhang* **Guangtao Zeng*** **Tianduo Wang** **Wei Lu**
StatNLP Research Group
Singapore University of Technology and Design
{peiyuan_zhang, tianduo_wang, luwei}@sutd.edu.sg
guangtao_zeng@mymail.sutd.edu.sg

Abstract

We present TinyLlama, a compact 1.1B language model pretrained on around 1 trillion tokens for approximately 3 epochs. Building on the architecture and tokenizer of Llama 2 (Touvron et al., 2023b), TinyLlama leverages various advances contributed by the open-source community (e.g., FlashAttention (Dao, 2023)), achieving better computational efficiency. Despite its relatively small size, TinyLlama demonstrates remarkable performance in a series of downstream tasks. It significantly outperforms existing open-source language models with comparable sizes. Our model checkpoints and code are publicly available on GitHub at https://github.com/jzhang38/TinyLlama.

1 Introduction

Recent progress in natural language processing (NLP) has been largely propelled by scaling up language model sizes (Brown et al., 2020; Chowdhery et al., 2022; Touvron et al., 2023a,b). Large Language Models (LLMs) pre-trained on extensive text corpora have demonstrated their effectiveness on a wide range of tasks (OpenAI, 2023; Touvron et al., 2023b). Some empirical studies demonstrated emergent abilities in LLMs, abilities that may only manifest in models with a sufficiently large number of parameters, such as few-shot prompting (Brown et al., 2020) and chain-of-thought reasoning (Wei et al., 2022). Other studies focus on modeling the scaling behavior of LLMs (Kaplan et al., 2020; Hoffmann et al., 2022). Hoffmann et al. (2022) suggest that, to train a compute-optimal model, the size of the model and the amount of training data should be increased at the same rate. This provides a guideline on how to optimally select the model size and allocate the amount of training data when the compute budget is fixed.

*The first two authors contributed equally.