Problem 3. PreTraining

Read the paper ”Training Compute-Optimal Large Language Models” Hoffmann et al. [2022] carefully and answer the following questions. For each question, provide specific evidence and citations from the paper to support your answer.

Note: Your answers should be specific and include relevant numerical evidence where appropriate. For each answer, clearly indicate which sections or tables of the paper you are referencing to support your arguments

1. **(5 points)** The paper presents three different approaches to determine the optimal trade-off between

model size and number of training tokens. For Approach 2 (IsoFLOP profiles), what were the exponents a and b found for the relationships N_{opt} $\propto$C_aandD_{opt} ∝C_b$? How do these values differ from Kaplan et al.’s findings, and what is the practical implication of this difference for training large language models?

Answer: According to Table 2, for approach 2, the exponents a and b found for the relationships N_{opt} $\propto$C_aandD_{opt} ∝C_b$ were a = 0.49 and b = 0.51. These values differ from Kaplan et al.'s findings(a = 0.73 and b = 0.27) which are also shown in Table 2. So while the results of approach 2 indicate that the model size and amount of training data provided should scale equally with compute capacity, Kaplan et al.'s findings suggest that model size should increase much more than the amount of training data provided when the compute capacity is increased. Practically, this suggests that many large models have been haven't been trained on enough data to be compute optimal. Given a compute capacity, this paper suggests that it's more optimal(in terms of both compute capacity and overall model accuracy/performance) to train a smaller model with more training data.

2. (5 points) For a given compute budget of 576×10^{23} FLOPs (same as Gopher), what is the optimal model size and number of training tokens according to the paper's analysis? Compare this to Gopher's actual configuration.

Answer: According to Table 3, for a given compute budget of 5.76e23 FLOPs (same as Gopher), the optimal model has 67 billion parameters and 1.5 trillion training tokens. According to Table 1, Gopher's actual configuration was 280 billion parameters and around 300 billion training tokens. As a result, the paper estimates that the computationally optimal model would be 4-5 times as small and contain around 5 times more training tokens then Gopher's setup.

3. (5 points) Did Chinchilla's improvements in performance come at a higher computational cost compared to Gopher? Explain your answer using evidence from the paper about compute budget and model efficiency.

Answer: No, Chinchilla's improvements in performance did not come at a higher computational cost compared to Gopher. Evidence of this can be seen in section 4 of the paper which explictly states that "Both Chinchilla and Gopher have been trained for the same number of FLOPs but differ in the size of the model and the number of training tokens." In addition, Chinchilla’s inference and fine-tuning costs are substantially lower. Evidence of this can again be seen in section 4 of the paper which explicitly states that "Due to being 4×smaller than Gopher, both the memory footprint and inference cost of Chinchilla are also smaller." As a result, these quotes from the paper about compute budget and model efficiency therefore explain how Chinchilla's performance gains did not require more pretraining FLOPs but from reallocating the same compute budget more optimally(using a smaller model and providing more training tokens).

4. (5 points) On the MMLU benchmark, what was Chinchilla's average accuracy and how did it compare to both Gopher and human expert performance? Cite specific numbers from the paper.

Answer: According to Table 6, Chinchilla's average 5-shot accuracy on the MMLU benchmark was 67.6%. This is 7.6% larger than Gopher's average 5-shot accuracy of 60% and around 22.2% smaller than the average human expert performance of 89.8% which are also explicitly stated in Table 6.

5. (5 points) According to the paper's analysis, what are the implications for training a 1 trillion parameter model? Would this be compute-optimal with current practices? Explain using evidence from Table 3 of the paper.

Answer: According to Table 3, a 1 trillion parameter model would require around 1.27e+26 FLOPs(221.3 times the Gopher standardized unit) and around 21.2 trillion training tokens to be provided for training. The paper further states in section 3.4 that "Unless one has a compute budget of 10e26 FLOPs (over 250× the compute used to train Gopher), a 1 trillion parameter model is unlikely to be the optimal model to train." On top of the compute budget required, the paper also states in section 3.4 that "the amount of training data that is projected to be needed is far beyond what is currently used to train large models." Overall, the provided evidence clearly shows how training a 1 trillion parameter model is not compute optimal with current practices due to the extremely high compute budget(number of FLOPs required) and the need for way more data than what's currently used to train large models.

```
In [ ]:
```