Fast Inference from Transformers via Speculative Decoding

Nipun Tulsian
2021101055

Vyom Goyal
2021101099

Rhythm Aggarwal
2021101081

***Abstract*—Deep autoregressive sequence-to-sequence models have demonstrated impressive performance across a wide variety of tasks in recent years. But inference still remains an inherently sequential and slow process - decoding K tokens takes K serial runs of the model. To overcome this limitation, we introduce speculative decoding - an algorithm to sample from autoregressive models faster without any changes to the outputs, by computing several tokens in parallel. We verify our approach empirically through a series of experiments using state-of-the-art self-attention models for machine translation and summarisation tasks without retraining or architectural changes. We demonstrate it on T5-Large and show a 1.5X-3X acceleration compared to the standard T5X implementation, with identical outputs.**

I. Introduction

Large autoregressive models, notably large Transformers are much more capable than smaller model. These models, such as GPT-3 and LaMDA, exhibit remarkable capabilities but face significant challenges in inference speed. The project aims to address the challenge of slow inference times in transformer-based models, which are widely used in natural language processing (NLP) tasks such as text generation, translation, and sentiment analysis. Despite their high accuracy and performance, a single decode step from these larger models is significantly slower than a step from their smaller counterparts, and making things worse, these steps are done serially - decoding K tokens takes K serial runs of the model.

The niche problem identified is the need for a method that accelerates inference without sacrificing the quality of the generated outputs. The proposed solution is speculative decoding, which allows the model to generate multiple candidate sequences during inference and select the most promising ones efficiently. This approach aims to enhance the responsiveness of transformer models, making them more suitable for applications requiring real-time interaction.

II. Speculative Decoding

A. Overview

In this paper the authors introduce speculative decoding - an algorithm to sample from autoregressive models faster without any changes to the outputs, by computing several tokens in parallel.

Their approach is based on the observations that:

1) Hard language-modeling tasks often include easier subtasks that can be approximated well by more efficient models
2) Using speculative execution and a novel sampling method, we can make exact decoding from the large models faster, by running them in parallel on the outputs of the approximation models, potentially generating several tokens concurrently, and without changing the distribution.

The authors leverage speculative execution, a concept borrowed from computer architecture that allows parallel task execution while verifying their necessity. By employing an efficient approximation model alongside the target model, we can generate multiple tokens concurrently which act as speculative prefixes for slower target model and by deploying a novel sampling method they are maximizing the probability for these to be accepted, significantly reducing the number of required serial runs.

So compared to previous approaches used to accelerate inferencing, this method is easy to deploy in actual production settings as it doesn't require training new models and doesn't change the outputs.

B. Speculative Sampling

Let M_p be the target model, and $p(x_t|x_{<t})$ the distribution we get from the model for a prefix $x_{<t}$. Let M_q be a more efficient approximation model for the same task, and denote by $q(x_t|x_{<t})$ the distribution we get from the model for a prefix $x_{<t}$.

To sample $x \sim p(x)$, we instead sample $x \sim q(x)$, keeping it if $q(x) \leq p(x)$, and in case $q(x) > p(x)$ we reject the sample with probability $1 - \frac{p(x)}{q(x)}$ and sample x again from an adjusted distribution $p'(x) = \text{norm}(\max(0, p(x) - q(x)))$ instead.

The proposed speculative decoding method operates by:

1) Utilizing a smaller approximation model to generate γ speculative token completions.
2) M_p is then run in parallel to evaluate these guesses and their respective probabilities from M_q in parallel, accepting all those that can lead to an identical distribution.
3) Sampling an additional token from an adjusted distribution to fix the first one that was rejected, or to add an additional one if they are all accepted.

Algorithm 1 Speculative Decoding Step

```
Inputs: $M_p$, $M_q$, prefix
Sample $\gamma$ guesses $x_1, \ldots, x_\gamma$ from $M_q$ autoregressively.
for $i = 1$ to $\gamma$ do
    $q_i(x) \leftarrow M_q(\text{prefix} + [x_1, \ldots, x_{i-1}])$
    $x_i \sim q_i(x)$
end for
Run $M_p$ in parallel.
$p(x_1), \ldots, p(x_{\gamma+1}) \leftarrow M_p(\text{prefix}), \ldots, M_p(\text{prefix} + [x_1, \ldots, x_\gamma])$
Determine the number of accepted guesses $n$.
$r_1 \sim U(0,1), \ldots, r_\gamma \sim U(0,1)$
$n \leftarrow \min(\{i - 1 \mid 1 \leq i \leq \gamma, r_i > \frac{p_i(x)}{q_i(x)}\} \cup \{\gamma\})$
Adjust the distribution from $M_p$ if needed.
$p'(x) \leftarrow p_{n+1}(x)$
if $n < \gamma$ then
    $q'(x) \leftarrow \text{norm}(\max(0, p_{n+1}(x) - q_{n+1}(x)))$
end if
Return one token from $M_p$, and $n$ tokens from $M_q$.
$t \sim p'(x)$
return  $\text{prefix} + [x_1, \ldots, x_n, t]$
```

C. Experiments

We test a standard encoder-decoder T5 model on two tasks:

1) English to German translation
2) Text summarization

For both the tasks we use T5-Large for M_p. For the approximation model M_q we use T5-small.

Table 1. Empirical results for base models

Task	Model	Ag. Inference Time	Avg. Bleu Score
EnDe	T5-Large	6.39 sec	0.717
Ende	T5-Small	0.75 sec	0.554
CNNDM	T5-Large	216.62 sec	0.38
CNNDM	T5-Small	12.02 sec	0.35

Table 2. Empirical results for speeding up inference from a T5-Large model

Task	M_q	Avg. Bleu Score	γ	α	Speed
EnDe	T5-Small	0.73	3	0.79	2.68
EnDe	T5-Small	0.68	7	0.55	2.56
CNNDM	T5-Small	0.37	3	0.48	2.11
CNNDM	T5-Small	0.36	7	0.33	2.17

Speculative decoding significantly improves inference speed compared to traditional decoding while maintaining near-parity in output quality. For example, T5-Large achieves BLEU scores of 0.717 for EnDe translation and 0.38 for CNNDM summarization, but with inference times of 6.39s and 216.62s, respectively. In contrast, speculative decoding with T5-Small achieves comparable BLEU scores of 0.73 (EnDe) and 0.37 (CNNDM) at much faster speeds, with speed-up factors of up to 2.68×. By leveraging a smaller approximation model M_q to accelerate inference, speculative decoding balances efficiency and quality, making it a practical solution for real-time applications. Additionally, tunable parameters like γ (speculative steps) and α (confidence threshold) allow further optimization between speed and accuracy.

D. Novelty

*1) **Beam Search:*** In the paper the authors didn't investigate the compatibility of speculative decoding with beam search. We have tried to implement speculative decoding with beam search.

To incorporate beam search, we use the beam width k and max length as parameters. At each generation step, we maintain the top k beams, scoring them based on their running log probabilities. From all candidates generated at a step, the top k beams are selected. The process stops when an EOS token is generated or the max length is reached.

Algorithm 2 Speculative Decoding with Beam Search

```
Inputs: $M_p$, $M_q$, prefix, beam width $k$, max length $L$
Initialize beams $\leftarrow \{prefix\}$, scores $\leftarrow \{0\}$
while not all beams end with EOS and length of beams $< L$ do
    Initialize candidate beams $\leftarrow \emptyset$
    for each beam in beams do
        Sample $\gamma$ guesses $x_1, \ldots, x_\gamma$ from $M_q$ autoregressively.
        for $i = 1$ to $\gamma$ do
            $q_i(x) \leftarrow M_q(\text{beam} + [x_1, \ldots, x_{i-1}])$
            $x_i \sim q_i(x)$
        end for
        Run $M_p$ in parallel.
        $p(x_1), \ldots, p(x_{\gamma+1}) \leftarrow M_p(\text{beam}), \ldots, M_p(\text{beam} + [x_1, \ldots, x_\gamma])$
        Determine the number of accepted guesses $n$.
        $r_1 \sim U(0,1), \ldots, r_\gamma \sim U(0,1)$
        $n \leftarrow \min(\{i - 1 \mid 1 \leq i \leq \gamma, r_i > \frac{p_i(x)}{q_i(x)}\} \cup \{\gamma\})$
        Adjust the distribution from $M_p$ if needed.
        $p'(x) \leftarrow p_{n+1}(x)$
        if $n < \gamma$ then
            $q'(x) \leftarrow \text{norm}(\max(0, p_{n+1}(x) - q_{n+1}(x)))$
        end if
        Sample one token $t \sim p'(x)$
        Add candidate beam and score:
        new beam $\leftarrow \text{beam} + [x_1, \ldots, x_n, t]$
        new score $\leftarrow \text{score(beam)} + \log(p'(t))$
        Add new beam and new score to candidates
    end for
    Select top $k$ beams by score:
    beams, scores $\leftarrow$ top $k$ candidates by score
end while
return  Best beam by score
```

Table 3. Empirical results for speeding up inference from a T5-Large model with Beam Search

Task	M_q	**Bleu Score**	**No. of beams**	α	**Speed**
EnDe	T5-Small	0.73	3	0.79	0.79
EnDe	T5-Small	0.72	5	0.75	0.45

Incorporating speculative decoding with beam search improves prediction quality by enabling the use of beam search's inherent strength in exploring multiple hypotheses while maintaining efficiency. The empirical results demonstrate that speculative beam decoding achieves BLEU scores of 0.73 and 0.72 for beam widths of 3 and 5, respectively, which are competitive with traditional beam search setups. The use of speculative decoding allows these improvements with only a modest reduction in inference speed. For instance, with a beam width of 3, the speed is $0.79\times$, and for a beam width of 5, it is $0.45\times$. This indicates that speculative beam decoding enables better predictions by incorporating the beam search algorithm while mitigating its computational cost. By maintaining multiple candidate sequences and leveraging the smaller approximation model M_q for accelerated generation, this method achieves a balance between quality and speed, making it a practical approach for scenarios requiring both accurate and efficient inference.

*2) **Varying Gamma**:* The optimal gamma for the algorithm is the one maximizing the wall-time improvement equation:

$$\frac{1-\alpha^{\gamma+1}}{(1-\alpha)(\gamma c+1)}$$

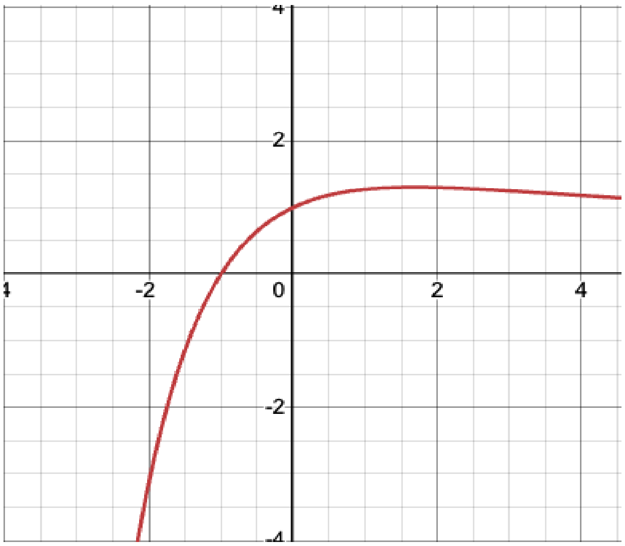

Fig. 1. Plot of Wall Time Improvement Equation

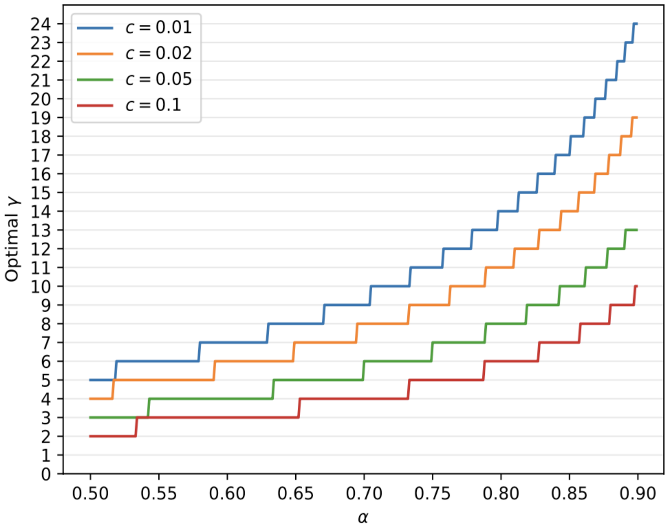

Fig. 2. The optimal γ as a function of α for various values of c

In the paper, a fixed value of γ was used throughout the run. We propose an improvement by dynamically predicting the value of β and adjusting γ accordingly during the execution of the algorithm. To predict β, which represents the expected acceptance rate, we compute the running average of the acceptance rate during the run. Using this prediction, we apply gradient ascent on the wall-time improvement equation to maximize it, thereby determining the optimal value of γ dynamically.

Table 4. Empirical results for speeding up inference from a T5-Large with varying gamma

Task	M_q	**Avg. Bleu Score**	α	**Speed**
EnDe	T5-Small	0.71	0.68	2.61
CNNDM	T5-Small	0.35	0.47	2.33

Dynamically varying γ during speculative decoding optimizes the trade-off between speedup and output quality by adjusting to the model's real-time performance, unlike fixed-γ approaches that rely on static assumptions. By predicting β, the acceptance rate, using a running average and applying gradient ascent on the wall-time improvement equation, the algorithm determines the optimal γ adaptively. This approach results in consistent speed improvements (e.g., $2.61\times$ for EnDe and $2.33\times$ for CNNDM) while maintaining high BLEU scores (0.71 for EnDe and 0.35 for CNNDM). Compared to fixed-γ setups, dynamic adjustment ensures the system adapts to variability in acceptance rates, achieving both better quality retention and computational efficiency without requiring manual tuning of γ. This flexibility makes the approach robust for diverse tasks and real-world deployment scenarios.

*3) **Hierarchical Version**:* We tried to explore a hierarchical version of the algorithm, where the approximation model is itself accelerated by an even faster model, which could allow for more capable approximation models.

This approach employs speculative decoding across three models. First, γ tokens are generated sequentially using the small model. The medium model then processes the resulting $\gamma + 1$ prefixes in parallel, applying an acceptance-rejection mechanism as given in paper. If n tokens are accepted, the medium model sequentially generates the remaining $\gamma - n$ tokens. These prefixes are then evaluated in parallel by the large model and the tokens are accepted according to acceptance-rejection mechanism as given in paper.

Algorithm 3 Hierarchical Speculative Decoding

```
**Inputs:** $M_s$, $M_m$, $M_l$, *prefix*
**Initialize:** new tokens $\leftarrow$ *prefix*
**Step 1: Decode using Small Model**
**Sample** $\gamma$ guesses $x_1, \dots, x_\gamma$ sequentially from $M_s$.
**for** $i = 1$ to $\gamma$ **do**
    $q_s(x) \leftarrow M_s(\text{prefix} + [x_1, \dots, x_{i-1}])$
    $x_i \sim q_s(x)$
**end for**
**Step 2: Parallel Decoding with Medium Model**
**Run** $M_m$ in parallel
$p_m(x_1), \dots, p_m(x_{\gamma+1}) \leftarrow M_m(\text{prefix}), \dots, M_m(\text{prefix} + [x_1, \dots, x_\gamma])$
**Determine accepted tokens:**
$r_1 \sim U(0,1), \dots, r_\gamma \sim U(0,1)$
$n_m \leftarrow \min(\{i - 1 \mid 1 \leq i \leq \gamma, r_i > \frac{p_m(x)}{q_s(x)}\} \cup \{\gamma\})$
new tokens $\leftarrow [x_1, \dots, x_{n_m}]$
**Step 3: Sequential Decoding with Medium Model**
**Sample** $\gamma - n_m$ tokens $y_1, \dots, y_{\gamma - n_m}$ sequentially from $M_m$.
**for** $j = 1$ to $\gamma - n_m$ **do**
    $q_m(x) \leftarrow M_m(\text{prefix} + \text{new tokens})$
    $y_j \sim q_m(x)$
    new tokens $\leftarrow$ new tokens $+ [y_j]$
**end for**
**Step 4: Parallel Decoding with Large Model**
**Run** $M_l$ in parallel for $\gamma + 1$ prefixes.
$p_l(x_1), \dots, p_l(x_{\gamma+1}) \leftarrow M_l(\text{prefix}), \dots, M_l(\text{prefix} + \text{new tokens})$
**Determine accepted tokens:**
$r_1 \sim U(0,1), \dots, r_\gamma \sim U(0,1)$
$n_l \leftarrow \min(\{i - 1 \mid 1 \leq i \leq \gamma, r_i > \frac{p_l(x)}{q(x)}\} \cup \{\gamma\})$
**Adjust the distribution from** $M_l$ **if needed**.
$p'(x) \leftarrow p_{n_l+1}(x)$
**if** $n_l < \gamma$ **then**
    $p'(x) \leftarrow \text{norm}(\max(0, p_{n_l+1}(x) - q_{n+1}(x)))$
**end if**
**Return one token from Mp and** $n_l$ **tokens from new tokens**
$t \sim p'(x)$
**return** prefix $+$ new tokens$[1, \dots, n_l] + t$
```

Table 5. Empirical results for speeding up inference from a T5-Large Heirarchical version

Task	Combination	Avg. Bleu Score	Speed
EnDe	T5-Large + T5-Small	0.723	2.68
EnDe	T5-Large + T5-Base	0.730	1.13
EnDe	Hierarchy	0.728	1.46

The hierarchical speculative decoding approach offers a balanced solution by maintaining a high BLEU score while improving inference speed compared to a direct combination of the large and base models. Specifically, the hierarchical setup achieves a BLEU score of 0.728, which is slightly lower than the large-base pair (0.730) but significantly higher than the large-small pair (0.723). In terms of speed, the hierarchical model achieves a $1.46\times$ speedup, outperforming the large-base pair ($1.13\times$) while being slower than the large-small pair ($2.68\times$). This trade-off is attributable to the intermediate layer (T5-Base) enabling more informed approximations while reducing the burden on the large model. By accelerating the small model's predictions through the base model before involving the large model, the hierarchy allows for a more capable approximation process, effectively striking a balance between quality and efficiency. This makes it a compelling choice for tasks requiring both high-quality outputs and moderate speed improvements.

References

[1] Leviathan, Yaniv, Matan Kalman, and Yossi Matias. ”Fast inference from transformers via speculative decoding.” International Conference on Machine Learning. PMLR, 2023

[2] Stern, Mitchell, Noam Shazeer, and Jakob Uszkoreit. ”Blockwise parallel decoding for deep autoregressive models.” Advances in Neural Information Processing Systems 31 (2018).

[3] Schuster, Tal, et al. ”Confident adaptive language modeling.” Advances in Neural Information Processing Systems 35 (2022): 17456-17472.