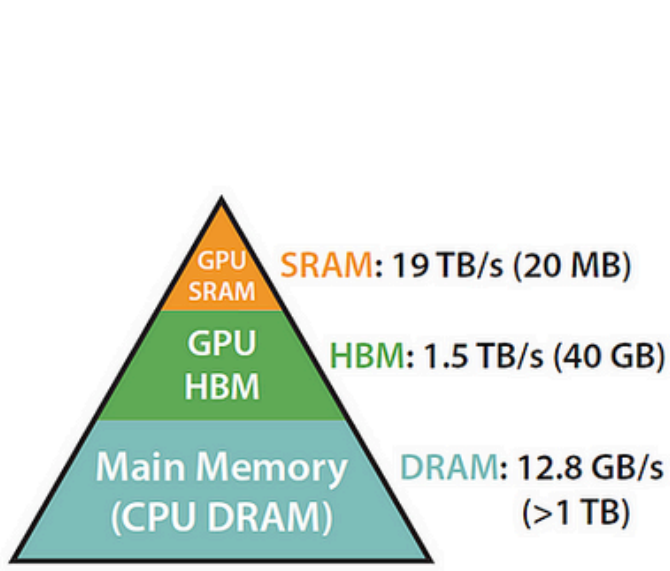
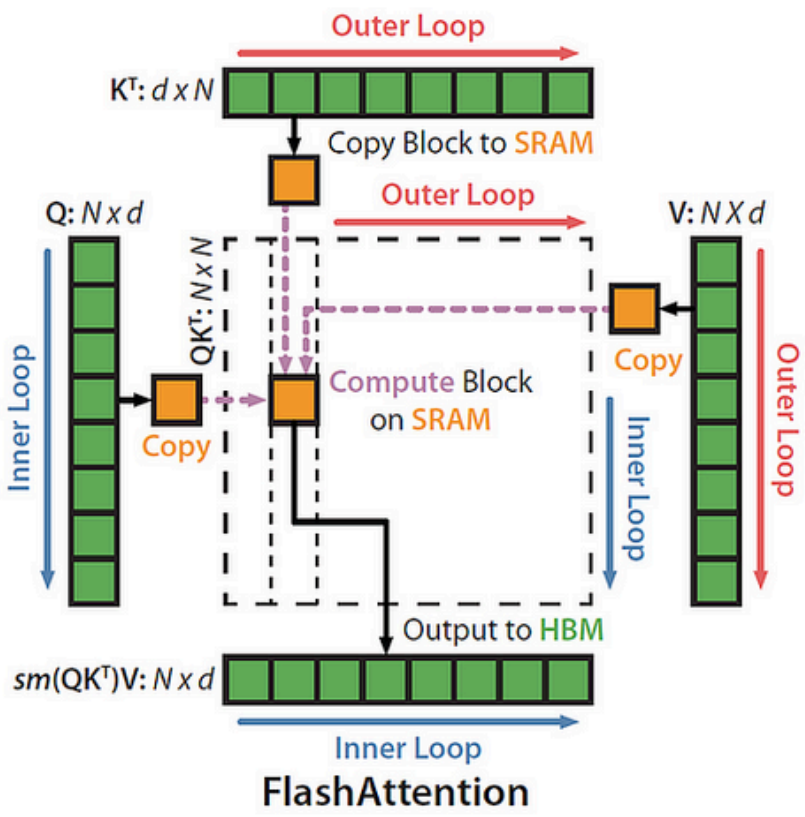
Mastering LLMs

Analytics Vidhya

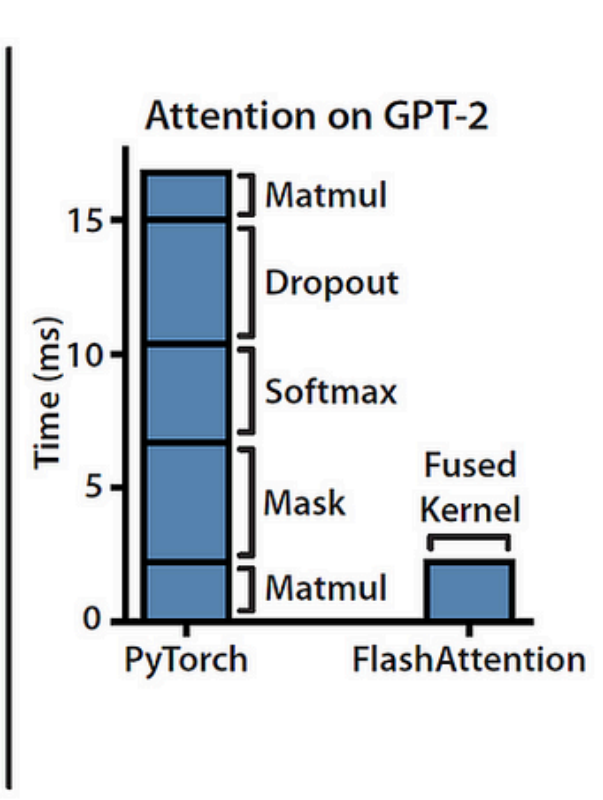
Day 33: Efficient Inference – FlashAttention & Speculative Decoding

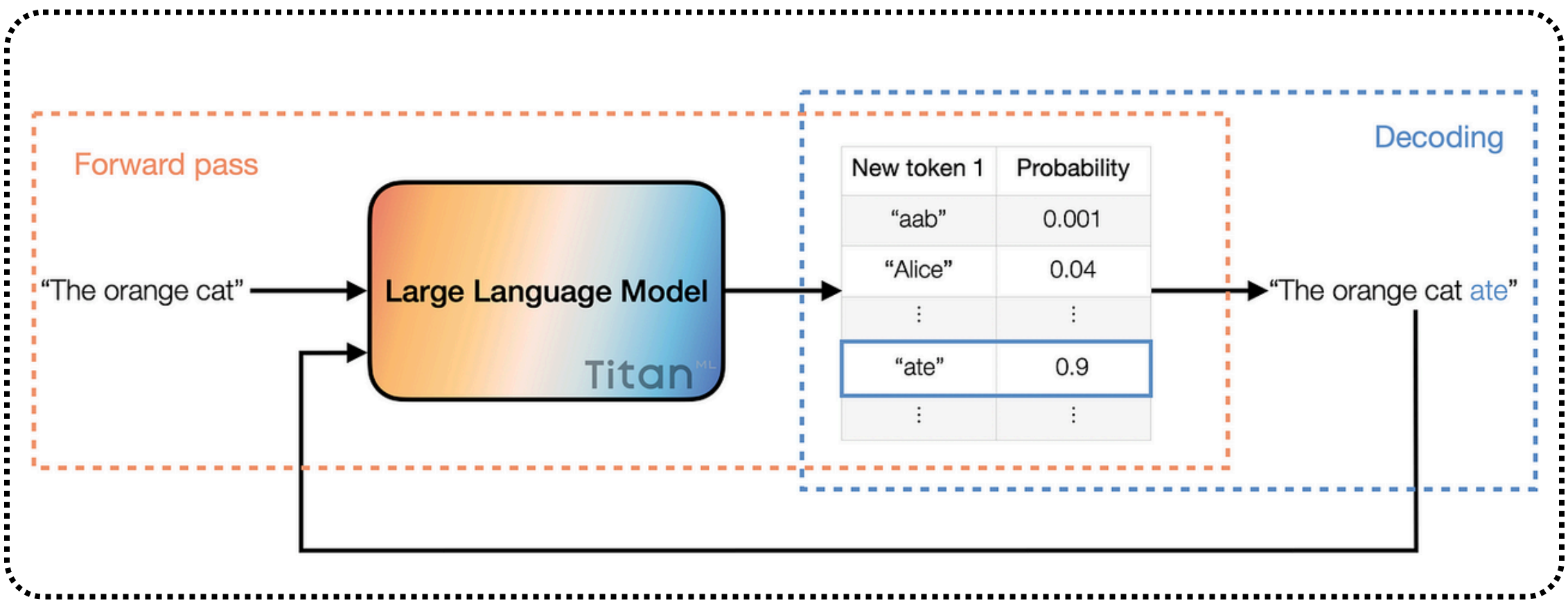


Memory Hierarchy with Bandwidth & Memory Size



FlashAttention



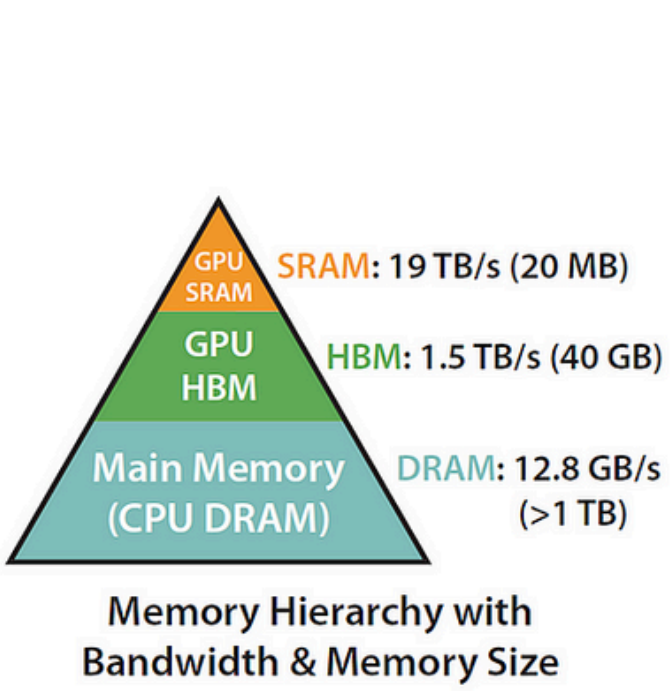


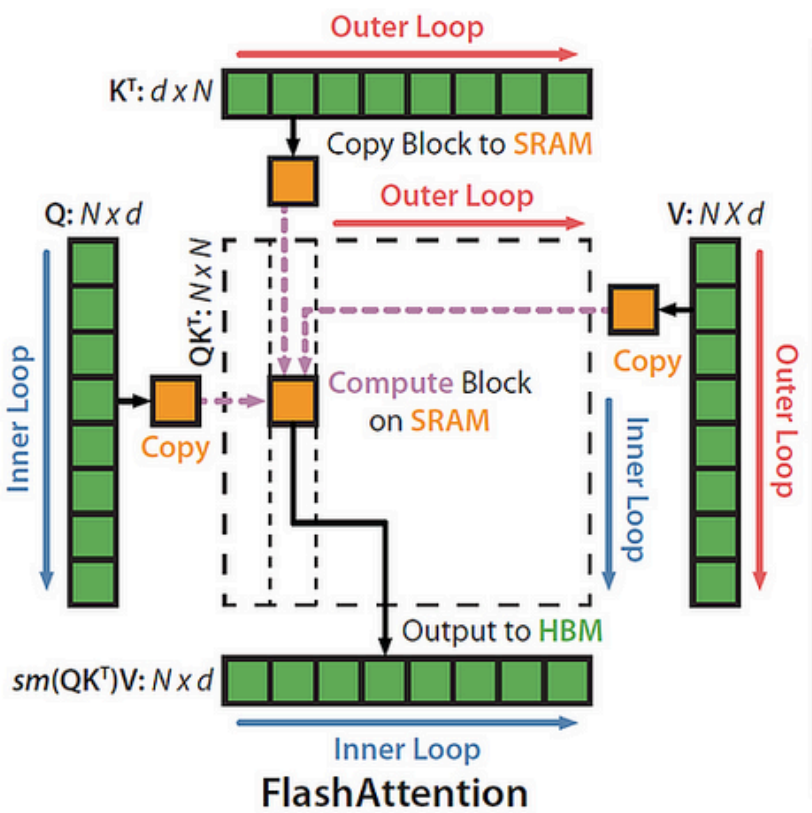
How can we make LLMs faster without sacrificing quality?

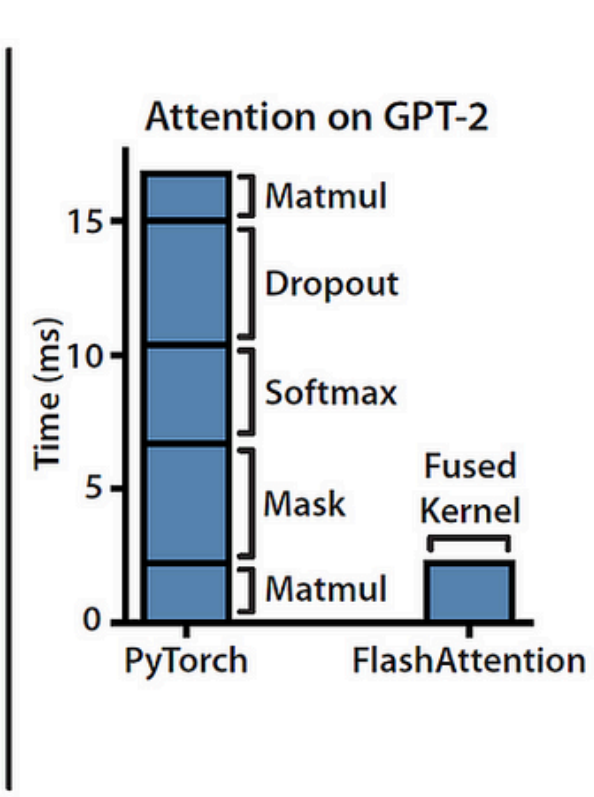
Inference is expensive!

Slow response times + high compute costs make deploying Large Language Models (LLMs) a challenge.

Two cutting-edge techniques, **FlashAttention** & **Speculative Decoding**, are game-changers.







Let’s break them down! 👇

FlashAttention – Making Attention Fast

- LLMs rely on self-attention, but traditional attention is slow and memory-intensive. Enter FlashAttention

How does FlashAttention work?

- Reduces memory reads/writes by keeping computations on-chip (SRAM instead of DRAM).
- Uses tiling techniques to compute attention in smaller, more efficient blocks.
- Optimized GPU kernels reduce latency and improve throughput.

Why FlashAttention is a Game-Changer?

- Speeds up Transformers by 2-4x
- Cuts memory usage in half – allows handling longer context lengths efficiently
- Crucial for long-context models (Llama 3, GPT-4 Turbo, Mistral, etc.)

Speculative Decoding – Faster Text Generation

Why wait for a model to generate one token at a time when we can predict multiple tokens at once?

- Traditional decoding (Greedy, Beam Search) generates one token per step.
- Speculative Decoding uses a smaller, faster model to guess multiple tokens, then verifies them with the main model.

How Speculative Decoding Works?

- Drafting Model (small, lightweight) generates a batch of potential tokens.
- Verification Step – The main model accepts, modifies, or rejects those tokens.
- Final Output – Only verified tokens are kept, speeding up inference.
- 2-3x faster inference without quality loss
- Reduces latency for real-time AI applications
- Works with existing LLMs without retraining

How Do FlashAttention & Speculative Decoding Work Together?

- FlashAttention speeds up the attention computation

- Speculative Decoding speeds up the text generation process

- **Used in**: OpenAI, Meta, Google DeepMind's latest models

Key Takeaways

- **FlashAttention** – Faster attention, optimized memory, better long-context handling

- **Speculative Decoding** – Faster token generation, low latency inference

- Deploying LLMs at scale? Use both to maximize efficiency

Analytics Vidhya

Stay Tuned for **Day 34** of

Mastering LLMs