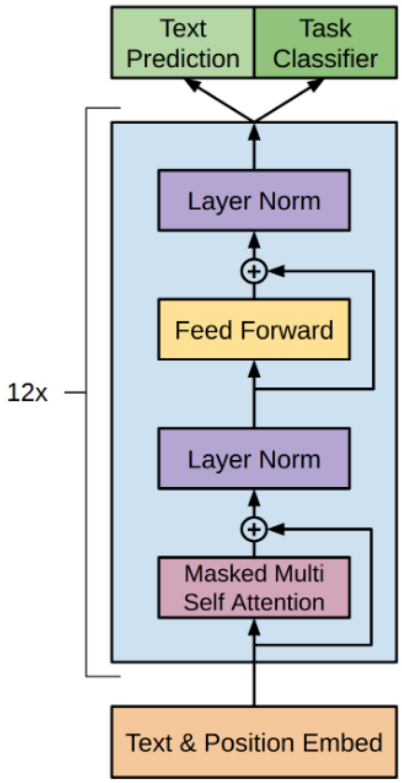
Mamba from scratch (RUST edition)

Audric HARRIS

In this project, we'll build a Mamba model from scratch. I'll share all the websites and videos I found to build my knowledge, formatted as these paper summaries.

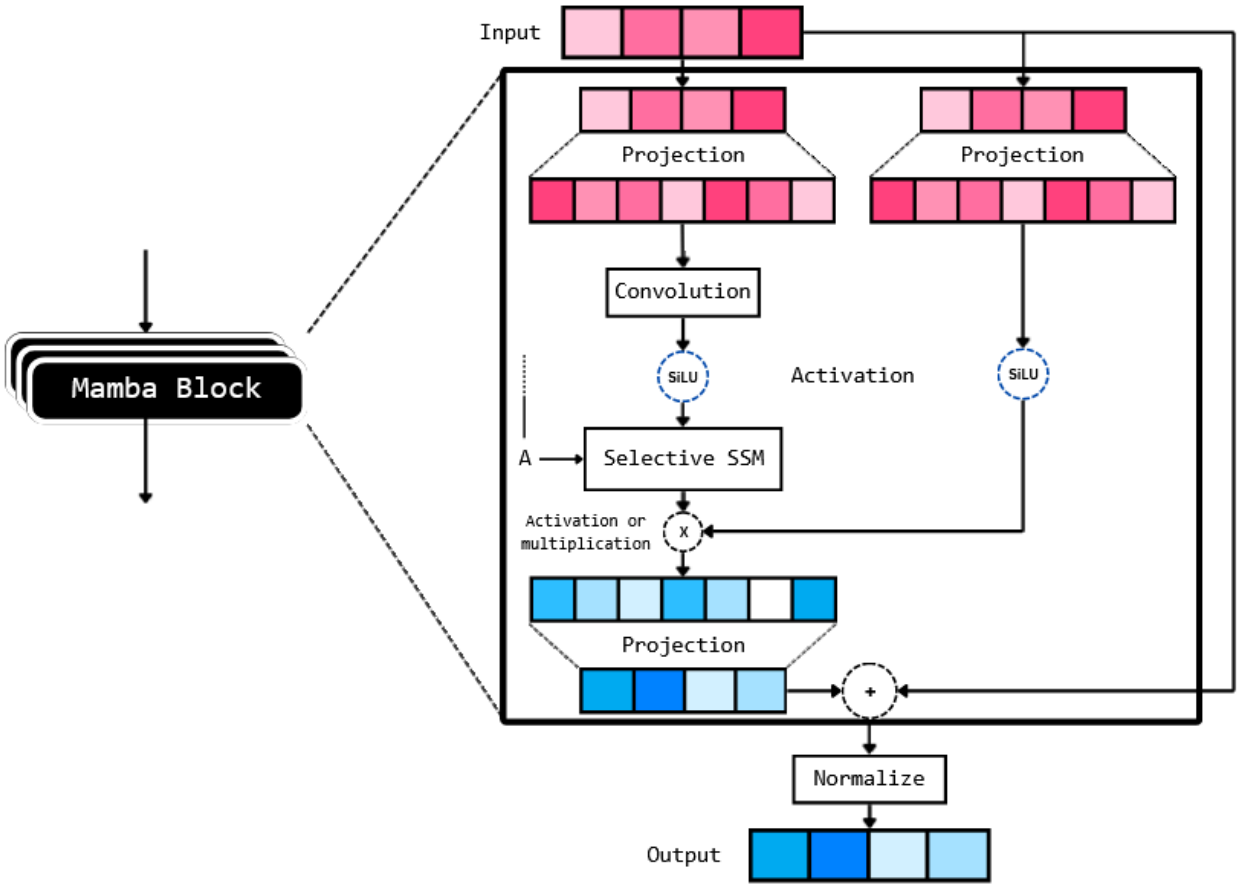
Let’s answer the question: What is Mamba? As you may know, in the past three years, AI has progressed at an impressively rapid pace. All the big AI companies rely on the famous architecture called GPT, famously popularized by OpenAI's ChatGPT model. But a key question is the computational cost of such models; it's high enough to require massive AI data centers that consume more electricity than many big cities.

https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf



The GPT architecture derives most of its computational cost from the self-attention mechanism (specifically, masked multi-head self-attention), which scales quadratically with sequence length and can make it challenging to run large instances on average consumer hardware without optimizations like quantization or distillation.

A team of researchers has proposed an alternative architecture for natural language processing that's far more efficient: Mamba (https://arxiv.org/pdf/2312.00752v2). Mamba is built around a core component called a Selective State Space Model (Selective SSM). This makes key model parameters input-dependent, allowing the model to dynamically propagate or forget information along the sequence based on the current input enabling content-based reasoning without recomputing pairwise interactions.



Compared to the GPT architecture (which is based on Transformers), Mamba processes sequences recurrently, token by token, much like recurrent neural networks, but with a selective mechanism that filters relevant information efficiently. Transformers compute token interactions across all previous tokens via attention, leading to quadratic time complexity in sequence length. In contrast, Mamba's recurrent scan achieves linear time complexity, making it scalable to much longer sequences (e.g., up to 1 million tokens) without the memory explosion of attention. While both use positional information implicitly through their sequential nature, Mamba doesn't require explicit positional embeddings like Transformers do, relying instead on its state evolution to capture order.

I aim to implement a full Mamba model to harness its efficiency. Based on the paper, it delivers performance comparable to Transformers of similar size on language tasks and sometimes outperforms models twice its size while being up to 5× faster in inference. My goal isn't to beat GPT outright; I just need an efficient natural language model for a project featuring an NPC, to enable real-time AI interactions even on lower-end computers.

I made a prototype of this project during August-September, but I will redo everything from scratch using the Burn framework instead of PyTorch. The models should be available on kaggle.com , and I will proceed to try new concepts and ideas to improve AI performance based on papers I have read and my personal experience. I'm still not as proficient in machine learning as I'd like. I might not always write the most accurate math, but I will strive to present the information I share as correctly as possible and avoid stating anything I'm not 100% sure about.