i3-80M: A Hybrid Architecture Language Model for Efficient Text Generation

Author: FlameF0X, Manus AI

Date: December 2025

Abstract

The **i3-80M** model is an 82.77 million parameter language model designed to explore the efficiency and performance trade-offs of a novel **hybrid architecture** [1] [2]. This model uniquely combines the strengths of recurrent/convolutional sequence modeling, specifically integrating **RWKV-style time-mixing** with **Mamba state-space dynamics** in its initial layers, with the global context capabilities of **Multi-Head Attention** in its deeper layers. Trained on a diverse, multi-domain dataset, i3-80M demonstrates strong convergence and exceptional hardware efficiency, achieving a final perplexity of approximately 6 [2] while maintaining a low VRAM footprint, making it highly suitable for training and inference on consumer-grade hardware [2]. This paper details the architecture, training methodology, and performance metrics of the i3-80M, highlighting its advancements over its predecessor, i3-22M.

1. Introduction

The development of large language models (LLMs) has been characterized by a trend toward increasing scale, often requiring vast computational resources [4]. This has created a need for smaller, more efficient models that can be trained and deployed on readily available hardware. The i3-series of models addresses this challenge by focusing on architectural innovation to maximize performance per parameter and per compute unit.

The **i3-80M** model is the second iteration in this series, scaling up from the original i3-22M. Its core innovation lies in its **hierarchical hybrid architecture**, which leverages the linear complexity of recurrent and state-space models for local context processing and the quadratic complexity of attention for global context and reasoning [1] [2]. This design choice is motivated by the desire to retain the efficient sequence processing of models like RWKV [5] and Mamba [4] while mitigating their known limitations in complex, long-range reasoning by introducing standard attention layers [2].

This paper is structured as follows: Section 2 discusses the architectural design. Section 3 details the tokenization and training methodology. Section 4 presents the results and efficiency analysis. Finally, Section 5 concludes the work.

2. Model Architecture

The i3-80M is a 16-layer model with approximately **82.77 million parameters** [1]. The architecture is divided into two distinct sections, each optimized for different aspects of sequence modeling:

2.1. Hybrid Blocks (Layers 1-10)

The initial ten layers of the model are composed of **RWKV-Mamba Hybrid Blocks** [3]. This block is a conceptual combination of RWKV's time-mixing mechanism and Mamba's Selective State Space Model (SSM) [4].

- **RWKV-Mamba Hybrid Block:** These blocks are designed for efficient, linear-time sequence processing. They are responsible for capturing local dependencies and managing the sequence state efficiently, which is crucial for maintaining low computational overhead [2]. The implementation utilizes a simplified Mamba-like structure, including a convolutional layer for local context and a state-space mechanism for temporal dependencies [3].
- **Dimensions:** The hidden dimension (d_{model}) is 512, the state dimension (d_{state}) is 32, and the convolution kernel size (d_{conv}) is 4 [3].

2.2. Full Attention Blocks (Layers 11-16)

The final six layers of the model transition to **Full Attention Blocks** [1]. This hierarchical approach ensures that while the majority of the sequence is processed efficiently, the model retains the ability to perform complex, global reasoning.

- **Multi-Head Attention:** These layers use standard Multi-Head Attention with 16 heads [1]. By placing the attention layers deeper in the network, the model can leverage the already processed and condensed representations from the hybrid layers to focus on long-range dependencies and complex pattern recognition [2].
- **Feed-Forward Networks (FFN):** All layers are followed by a Feed-Forward Network with a 4x expansion factor [1].

The complete architectural breakdown is summarized in Table 1.

Feature	Value
Total Parameters	~82.77M
Total Layers	16
Hybrid Layers (RWKV-Mamba)	10

Feature	Value
Attention Layers (Full MHA)	6
Hidden Dimension (d_{model})	512
Attention Heads	16
State Dimension (d_{state})	32
Max Sequence Length	256 tokens

Table 1: i3-80M Architecture Specifications [1] [2]

3. Tokenization and Training Methodology

3.1. Tokenization

The i3-80M employs a custom tokenization strategy based on **variable-length chunking** (2-3 characters) with optimization for common trigrams [1]. This approach results in a significantly larger vocabulary compared to its predecessor, with a size of **35,560 tokens** [1]. This larger vocabulary is intended to improve text representation and enhance the model's ability to handle out-of-vocabulary words gracefully [2].

3.2. Training Datasets

The model was pre-trained on a diverse set of three public datasets to ensure better generalization across different text domains [1]:

1. **roneneldan/TinyStories:** Provides simple narratives and storytelling structure.
2. **starhopp3r/TinyChat:** Introduces conversational dynamics and dialogue patterns.
3. **agentlans/high-quality-english-sentences:** Offers linguistic diversity and formal text structure.

The training process involved processing over **3,000,000 tokens** from these combined sources [1].

3.3. Training Configuration

The training was optimized for efficiency on consumer hardware [2] [3].

- **Framework:** PyTorch
- **Hardware:** NVIDIA P100 (16GB VRAM)
- **Training Steps:** 5,000 iterations
- **Batch Size:** 4
- **Optimizer:** AdamW with gradient clipping (max norm: 1.0)
- **Learning Rate Schedule:** 3×10^{-4} with linear warmup and cosine decay [3].
- **Training Time:** Approximately 2–4 hours [1].

4. Results and Discussion

4.1. Training Dynamics

The i3-80M demonstrated stable and rapid convergence during training [2].

Metric	Initial Value	Final Value
Training Loss	~10.0	~1.7
Perplexity	~4000+	~6

Table 2: Training Performance Metrics [2]

The final perplexity of ~6 indicates a strong language modeling capability, especially given the model's small size and short training duration.

4.2. Hardware Efficiency

A key design goal was hardware efficiency. The model achieved high throughput (100–550 tokens/second) with minimal resource consumption [2]:

- **VRAM Usage:** ~2.2GB allocated (peak ~18% of 16GB)
- **Power Draw:** ~40W average
- **Context Window Scaling:** The model's efficiency is further demonstrated by its VRAM usage scaling with context window size, supporting up to 4096 tokens within 16GB of VRAM [2].

4.3. Comparison to i3-22M

The i3-80M represents a significant architectural and performance upgrade over the original i3-22M [2].

Feature	i3-22M	i3-80M	Key Improvement
Parameters	22.6M	82.77M	Increased capacity
Architecture	24 Hybrid layers (Pure Hybrid)	10 Hybrid + 6 Attention	Hierarchical processing for better reasoning
Vocabulary Size	4,466 tokens	35,560 tokens	8x larger for better coverage
Training Data	TinyChat only	Multi-Dataset (3 sources)	Better generalization
Training Time	~17 hours	~2–4 hours	Faster training pipeline
Attention Layers	None	6 Full Attention	Enhanced long-range dependency modeling

Table 3: Comparison of i3-22M and i3-80M [2]

The introduction of full attention layers and multi-dataset training directly addresses the limitations of pure recurrent/state-space models, leading to better general-purpose text generation and stronger attention-based reasoning [2].

5. Conclusion

The **i3-80M** model successfully demonstrates the viability of a **hierarchical hybrid architecture** for creating efficient and capable language models. By strategically combining RWKV-Mamba blocks for local efficiency and Multi-Head Attention for global context, the model achieves strong performance metrics (Perplexity ~6) while remaining accessible for training and deployment on consumer-grade hardware. Future work will focus on scaling the model further and exploring more sophisticated integration methods between the recurrent and attention layers.

References

[1] FlameF0X. i3-80m. *Hugging Face Model Card*. URL: https://huggingface.co/FlameF0X/i3-80m

[2] FlameF0X. i3-80M README. *GitHub Repository*. URL: https://github.com/FlameF0X/open-i3/blob/main/src%2F80m%2FREADME.md

[3] FlameF0X. train.py. *GitHub Repository*. URL: https://github.com/FlameF0X/open-i3/blob/main/src%2F80m%2Ftrain.py

[4] Gu, Albert and Dao, Tri. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. *arXiv preprint arXiv:2312.00752*.

[5] Peng, Bo et al. RWKV: Reinventing RNNs for the Transformer Era.*arXiv preprint arXiv:2305.13048*.