**Xiomi Redmi 14C.**

**Какие модели и уровни квантизации нейросете пойдут на нём в следующих интерфейсах.**

Интерфейсы:

* • PocketPal AI
* • ChatterUI
* • LLM Farm
* • Private LLM / Open LLM

**Какие из них ещё придётся настраивать?**

Меня интересовали следующие модели с huggingface и ollama:

Mykes/medicus, gemma3, codegemma, codellama, dolphin-mistral, llava, falcon, mistral, llama2-uncensored, reefer/erplegend, gdisney/neural-chat-uncensored, reefer/erphermesl3, jimscard/adult-film-screenwriter-nsfw, Llama-2, Llama-3.2, nidum/Nidum-Llama-3.2-3B-Uncensored-GGUF, pygmalion (PygmalionAI).

Из них на Redmi 14C с **PocketPal AI** и **ChatterUI** лучше запускать модели с размером до 3-4 млрд параметров и с квантизацией 4-bit или 8-bit в формате GGUF/GGML.

Например:

* • nidum/Nidum-Llama-3.2-3B-Uncensored-GGUF — 3B модель с GGUF
* • pygmalion (PygmalionAI) — часто доступна в облегчённых версиях.
* • falcon, mistral — есть версии с квантизацией, но лучше выбирать меньшие варианты (7B и ниже).
* • llama2-uncensored, Llama-2, Llama-3.2 — для Redmi 14C лучше использовать 3B или 7B версии с квантизацией.
* • codellama, codegemma — кодовые модели, обычно тяжелее, но с квантизацией могут работать на 7B и ниже.

**Какие модели и интерфейсы придётся настраивать?**

* • PocketPal AI и ChatterUI — требуют минимальной настройки, если модель уже в GGUF с квантизацией. Для моделей без готовой квантизации придётся конвертировать и оптимизировать.
* • LLM Farm — требует больше ручной настройки, особенно для мобильных устройств.
* • Private LLM / Open LLM — часто требуют настройки под конкретное железо, оптимизации квантизации и формата.

**Если потребуется настройка.**

**1) Общие параметры для всех интерфейсов**

• Модель (Model path)

Укажите путь к модели в формате GGUF/GGML с квантизацией 4-bit или 8-bit, например:

models/nidum-llama-3.2-3b.gguf

Это позволит загрузить оптимизированную модель, подходящую для ограниченных ресурсов Redmi 14C.

• Квантизация (Quantization)

Укажите тип квантизации, например:

--quantize 4bit или --quantize 8bit

Это уменьшит размер модели и нагрузку на CPU/GPU.

• Размер контекста (Context size)

Например:

--ctx\_size 2048 или --ctx\_size 4096

Чем больше контекст, тем больше памяти требуется. Для Redmi 14C лучше ограничиться 2048 токенами, чтобы избежать тормозов.

• Потоков CPU (Threads)

Например:

--threads 4

Redmi 14C обычно имеет 4-8 ядер, но для стабильности лучше ограничить количество потоков до 4.

• Использование GPU (если поддерживается)

Если интерфейс поддерживает GPU-ускорение (например, через Vulkan или OpenCL), включите его:

--use\_gpu true

Но на Redmi 14C GPU слабый, поэтому часто лучше использовать CPU.

**2) Специфичные параметры для PocketPal AI и ChatterUI**

• Формат модели

PocketPal AI и ChatterUI лучше работают с GGUF/GGML. Убедитесь, что модель конвертирована в этот формат.

• Оптимизация памяти

В PocketPal AI можно включить опцию:

--low\_vram true

Это уменьшит использование оперативной памяти.

• Темп генерации (Temperature)

Например:

--temperature 0.7

Регулирует креативность ответов. Для более стабильных ответов ставьте 0.6-0.8.

• Максимальная длина ответа (Max tokens)

Например:

--max\_tokens 256

Ограничивает длину генерируемого текста, чтобы не перегружать устройство.

**3) Параметры для LLM Farm и Private/Open LLM**

• Путь к модели и квантизация — как в общих параметрах.

• Параметры запуска

В LLM Farm часто нужно указывать дополнительные параметры запуска модели, например:

--use\_mlock true — блокирует модель в памяти, чтобы избежать свопа (если хватает RAM).

--batch\_size 8 — размер батча для генерации, уменьшайте для экономии ресурсов.

• Настройка логирования

Для отладки полезно включить подробный лог:

--verbose true

• Оптимизация под мобильные устройства

В некоторых случаях нужно вручную конвертировать модель с помощью инструментов типа llama.cpp с параметрами:

./quantize model.bin model.gguf 4 — для 4-битной квантизации.

**4) Пояснения к параметрам**

* --quantize Снижает размер модели и нагрузку на устройство
* --ctx\_size Размер окна контекста, влияет на память и качество
* --threads Количество CPU потоков для обработки
* --use\_gpu Включение GPU-ускорения (если поддерживается)
* --low\_vram Оптимизация использования памяти для слабых устройств
* --temperature Контролирует креативность и разнообразие ответов
* --max\_tokens Максимальная длина генерируемого текста
* --use\_mlock Блокирует модель в RAM, чтобы избежать свопа
* --batch\_size Размер батча для генерации, влияет на скорость и память
* --verbose Включает подробный лог для отладки

**Итог**

Для Xiaomi Redmi 14C при настройке интерфейсов LLM стоит ориентироваться на:

* • Использование моделей с квантизацией 4-bit или 8-bit в формате GGUF/GGML.
* • Ограничение контекста до 2048 токенов.
* • Установка количества потоков CPU в 4.
* • Включение оптимизаций памяти (--low\_vram).
* • Контроль температуры и максимальной длины ответа для баланса качества и производительности.