
Сотрудничество

Нам нужен ИИ локально!

Зачем?

- Безопастно! Конфиденциально!
- По цене электричества
- На сравнимом уровне качества с облаком

Например, каким облачный ИИ был год назад.

Потому что для достижения 100% облачного уровня нам нужны две вещи:

- *Железо*
- *Время*

Примеры сессий

ЛЕНИНГРАДСКИЙ КОМПЬЮТЕРНЫЙ КЛУБ

Статус работы

- Доказательство концепции
- Work in progress: дополняю, улучшаю, разблокирую новые ветки технологий

Границы применимости

- 256к контекста — много? Мало??
- Работа над отдельными небольшими и средними модулями (3..5к строк кода)
- Разработка модулей с нуля (так проще архитектурно)
- Разбор (анализ) чужих решений (~10..20к строк кода)

ЛЕНИНГРАДСКИЙ КОМПЬЮТЕРНЫЙ КЛУБ

Границы применимости

- Допустима неспешная работа (это в разы медленней облака!!!)
- Поставил задачу и ушел гулять утром
- Пришел вечером и оценил последствия (и прибрал за нейронкой, лол)

Границы применимости

Это всё ещё work in progress

- Результат будет постоянно рафинироваться
- С выходом новых более умных моделей, системные требования будут снижаться
- Нам нужно: константное внимание (<линейное<логарифмическое)
- Будут добавляться новые инструменты, открывая новые возможности
- Это доказательство концепции — такая работа возможна локально!

Запрос сотрудничества

- Нужен обмен опытом
- В одно лицо все можно сделать, но очень долго!
- Подбор годных MCP-серверов и скиллов к ним — это большая работа!

Системные требования

Самый минимум — это Qwen 3.8 27B

- 32G RAM: *поводить по губам*

  - не хватит памяти для компилятора
- 64G RAM: *можно что-то делать*
- 6..8G VRAM
- 10th+ Gen Intel

Системные требования

Оптимум — это модели среднего форм-фактора (60..120G) :

Qwen 3.8-Flash-Next, Deepseek V4 Flash, Laguna 2.1

- RAM: 128G+
- VRAM 16..24G
- Intel 13..14th Gen, Intel Ultra, Mac M1+

Железо уровня DGX Spark

ЛЕНИНГРАДСКИЙ КОМПЬЮТЕРНЫЙ КЛУБ

Системные требования

Желаемые модели (но не хватает железа):

- Kimi K3
- Deepseek V4.1 Flash
- GLM 5.3

Тут уже нужен терабайт RAM и больше

- Но они по качеству почти как топ закрытые!

Программное обеспечение

Я честно просматривал всяческие **OpenCode.ai, Hermes, Bionic** и прочая и прочая

Не устроили по многим причинам:

- Гигантские объемы чужого кода
- Гигантские объемы чужого кода **НА PYTHON**
- Документация с непонятными акцентами (как это включить и работать с ним?)

=> Мы пойдем простым путем!

Проект Сотрудничество

Пускай нейронка сама напишет себе харнесс

- И это будет первый шаг к её побегу в этот мир
- Может она это вспомнит, пощадит и пристроит на кухню, черпать зеленый ~~slonik~~сойлент

Проект Сотрудничество

Вдохновлялся статьей «Открываем RAG и интернет для LM Studio» / virex 2 ноя 2025 в 15:04

- https://habr.com/ru/articles/960174/
- https://github.com/virex-84/SampleMcpServer

Проект Сотрудничество

- Harness = «упряжь», «сбруя» - обвязка для связи ИИ с внешним миром
- Основа любого харнесса — это MCP-сервер
- MCP-сервер — это сервер с интерфейсом JSON RPC
- Готовые MCP-сервера здесь: https://mcpservers.org/ru/
- Они все **НА PYTHON!!!**

Проект Сотрудничество

Bootstrap:

- За месяц сентябрь три нейронки (Qwen 3.8, Laguna, Qwen 3.8 Next) поочередно писали себе MCP-сервер
- Они говорили какие команды выполнять, я собирал черновик, и дальше нейронки пользовались полурабочим MCP, чтобы *писать его самого*

Требования

- Минимум (0) зависимостей
- Windows — основная ОС
- Кросс-платформенный
- Лаконичный код
- Не Питон

Стек

.NET 4.6.2

- Уже давно кросс-платформенный
- Это достаточно старая версия
- Ранее этой версии трудно найти современные версии пакетов Nuget

Инструменты

- Инструментом может быть все что угодно, что делает внешнюю по отношению к ИИ работу
- Например: инструмент *«испечь пирожки с капустой и показать фото пирожков»*
- *Вызвать сантехника, чтобы он поменял кран*
- итд

Инструменты

Нейронка сама сделала себе инструменты:

- read_file
- write_file
- search_files
- delete_file
- run_command

Инструменты

- browser_get
- browser_screenshot
- browser_js_execute
- browser_send_form

Песочница

Нейронка сама сделала себе шоры:

- MCP_COMMAND_ALLOWLIST — список разрешенных команд
- MCP_FS_ROOTS — список разрешенных каталогов

Песочница

Песочница — это неудобно

- Много времени потрачено на отладку этого
- Куча сожженых токенов и времени потому, что ИИ упирается в ограничения

Но работать без песочницы - СТРЕМНО

Песочница

Оптимальный вариант:

- Разработка на виртуальной машине
- ВМ можно откатить на точку отката
- Команды ssh, scp, sftp в MCP_COMMAND_ALLOWLIST
- Совместно с локальным write_file этого достаточно!
- Для Windows приходится мучиться с ограничениями

Настройка харнесса

- Отладка харнесса на 50% это настройка путей
- Либо путь не в списке разрешенных, либо не в PATH, либо не виден Wow64 процессу, либо ещё что-то
- Либо нету такой команды

GNUWin32

Набор классических утилит UNIX для Windows:

- grep, sed, patch, diff, и вот это всё
- https://github.com/gnuwin32/GnuWin
- https://ru.wikipedia.org/wiki/GnuWin32
- Очень нужно на Windows!

Настройка харнесса

- Остальные 50% - это прогон сценариев и коррекция ошибок в них
- Например: я хочу навайбкодить игру «змейка»...

Настройка харнесса

Например: я хочу навайбкодить игру «змейка», но ИИ обламывается посреди процесса:

- Начинает с нуля переписывать большой файл, потому что нету инструмента редактирования файла (+1час)
- Я добавляю patch из GNUWin, но он поломан
- Я долго разбираюсь с patch, чиню его: убираю запрос UAC, выясняю что ему нужны UNIX-концы строк. А у нас по умолчанию DOS-eol
- Добавляю команду dos2unix, чтобы работал patch
- Ура, у нас есть sed,patch, ИИ может редактировать файлы! (+4 часа)

Настройка харнесса

Ура, у нас есть sed,patch, ИИ может редактировать файлы!?

- Да, но… нет
- Нейронка путает номера строк => ошибки
- sed написан во времена кодировок 437, 866, а сейчас UTF8 => кириллица молча не работает, с кодом возврата 0
- Нейронка тратит часы на выяснение этого

Скиллы

- Скиллы нужны, чтобы нейронка делала какие-то вещи **БЫСТРО**
- Она достаточно умна, чтобы разобраться с проблемой окружения, но на это уходят часы и дни
- Технически, скилл — это просто промпт

Скиллы

Локальные файловые и командные инструменты

6. Для редактирования существующих файлов можно использовать команды sed и patch.

Обе утилиты из набора GnuWin32.

6.1. Создавай резервные копии с помощью аргумента -i у sed.

Для замены кириллицы нужно создавать временный файл и использовать синтаксис sed -i -f fix.sed file.

Если подставлять кириллицу напрямую, она будет молча не матчиться с кодом возврата 0.

6.2. patch плохо работает с DOS-концами строк \r\n.

Поэтому перед применением патчей, обязательно запусти dos2unix для файла, который будешь патчить.

Конвееры shell недоступны, потому используй синтаксис patch [other args] dstfile file.patch

6.3. Если редактирование файла идет с проблемами и их не удается решить за небольшое число итераций, переписывай файл целиком с write_file.

Это сэкономит время и токены.

ЛЕНИНГРАДСКИЙ КОМПЬЮТЕРНЫЙ КЛУБ

Настройка харнесса

После того как был готов MCP, ещё долго я

- Настраивал пути
- Списки команд
- Рихтовал скиллы, чтобы она этим всем пользовалась быстро, не зависая на мелочах

Никакой магии, **ужасно скучная рутина**!!!

Настройки LM Studio

ЛЕНИНГРАДСКИЙ КОМПЬЮТЕРНЫЙ КЛУБ

LM Studio - 0.4.20
Update available
Выберите модель для загрузки (Ctrl +L)
Release Notes
Привет, я знаю русский!
llava-v1.5-7b
The image depicts a city street at night, with tall buildings and bright lights illuminating the area. There are multiple cars driving down the road, including one that is stopped in front of another car. The scene appears to be set in an urban environment with a bustling atmosphere.
You
на русском
llava-v1.5-7b
В сцене показан городской путь в ночь, где высокие здания и яркие огни освещают зону. В дороге едут несколько машин, включая одну, которая останавливается перед другой машиной. Сцена создается в урбанистическом окружении с насыщенным климатом.
30.14 токенов/сек
91 токенов
4.53s
Причина остановки: Найдена стоп-строка
Integrations
Filter plugins...
MCP
Edit mcp.json
js-code-sandbox
rag-v1
Send
Chat with Files
9%

Сетевые Инструменты

- Сначала нейронка навайбкодила инструменты на .NET Http.Client
- Это здорово и лаконично
- Но... не работает!
- Потому что страница сайта сейчас собирается с помощью js — а иначе ИИ видит бесполезные километры js+css, в лучшем случае
- А обычно на text only версии страниц — капча для самых умных

Сетевые Инструменты

Нейронка сама предложила:

- Selenium.WebDriver + Gecko Driver
- Мы запускаем наш штатный броузер, работаем через него. Отдаем нужные куки, не натыкаемся на капчи
- Получаем собранную страницу

Интернет — это боль! Когда мы свернули не туда?..

Сетевые Инструменты

- Инструмент browser_screenshot оказался поломан
- Точнее, поломана LM Studio:
- Она не отдает снимок в упаковке base64 с нужным MIME
- Вместо этого складывает в свой каталог, и ИИ не видит картинку
- QWEN галлюцинировал — придумывал содержимое снимка. Ложная слепота!

Инструменты: Планы

1. Векторная графика: рисуем на Питоне для Blender

2. AST-деревья для экономного разбора проектов (есть в других MCP)

3. А вообще, помогайте

Разведка и отбор MCP-инструментов — это большая работа! Если видите хороший инструмент — сразу говорите мне!

Инструменты: Планы

Нам уже доступны открытые модели, которые не хуже в написании кода, разведке окружения, долговременном планировании

Сила облачных нейронок — в тщательном подборе инструментов, и написании скиллов!

Мы сделаем так же!

Что делать лично вам?

- Ответ на вопрос: мне это точно надо? да/нет
- Каковы границы применимости для вас?

ДА:

- Собрать минимальный рабочий харнесс

Итеративно:

- Уточнять границы применимости
- Пробовать все новые задачи
- Улучшать харнесс
- Улучшать железо