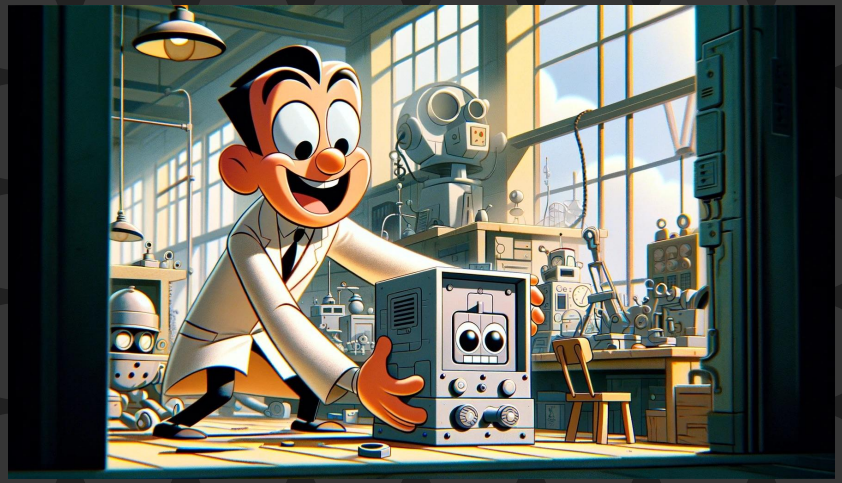
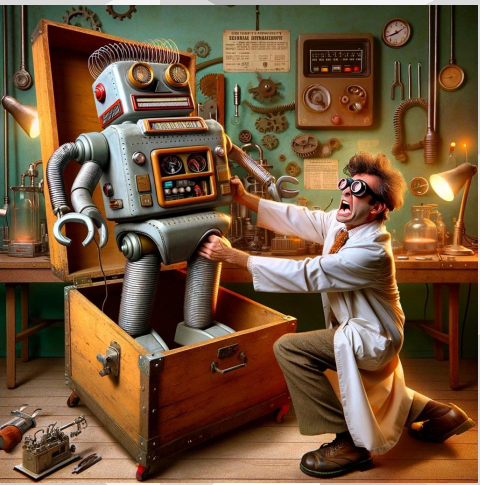


Running your LLMs locally

(on your own Mac and Linux hardware)

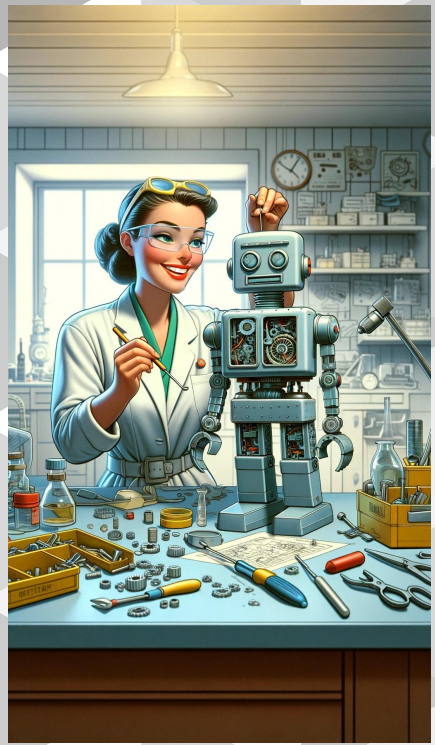
Bill McIntyre - Thinkiac.ai
RMAIIG Workshop - 2/13/2024

© 2024 - Free to distribute, use, and copy under GNU GPL3*



Why Run an LLM locally?

- Data Privacy
  - You have proprietary data
  - You have proprietary code
  - You have proprietary IP/methods and prompts
- Security
  - Custom Network and Physical Security
  - Compliance and Regulatory Requirements
  - Reduced Attack Footprint
- Connectivity
  - Low Latency / Direct Integration
  - Consistent Performance
  - You can have a live co-pilot on a plane
  - No outages



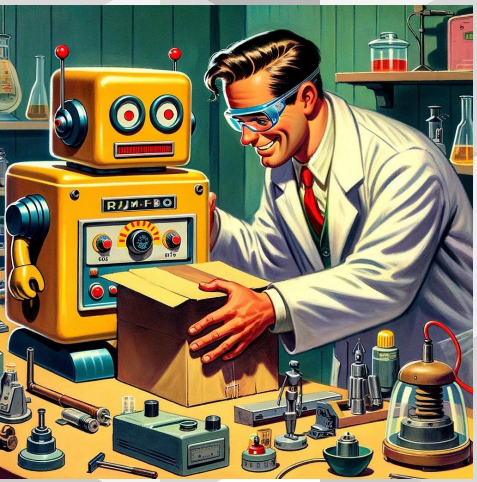
You Own It

- Hands-on experience. Local ownership means you learn more.
- Lower barrier to customization and experimentation.
- Gain a Better Understanding of Performance
  - And the factors that go into it
  - Token budgeting becomes visceral
- Cost
  - You are not paying rent
  - Open Source is magic and virtuous

Installing Ollama on MacOS and Linux

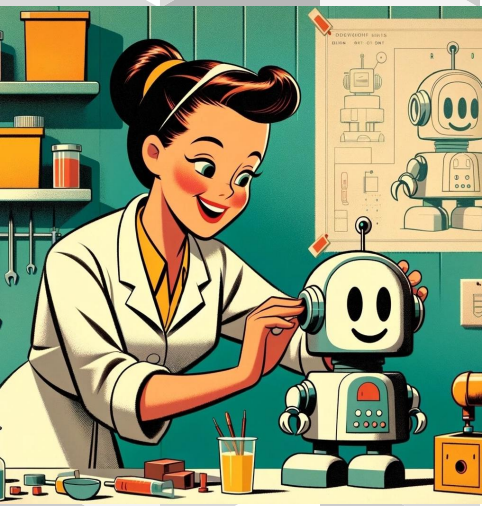
- GUI / Web:
  - Go to https://ollama.com/ and click on “Download”
    - For OSX and Linux.
    - Has full instructions including configuring GPU Drivers for Linux.

- Command line installation
  - MacOS:
    - brew install ollama
  - Linux:
    - curl -fsSL https://ollama.com/install.sh | sh



Ollama CLI Commands

- serve Start ollama
- create Create a model from a Modelfile
- show Show information for a model
- run Run a model
- pull Pull a model from a registry
- list List models
- cp Copy a model
- rm Remove a model
- help Help about any command



Example Runs:

LLama 2 CLI

- ollama pull llama2:latest
- ollama run llama2:latest
- Use Ctrl + d or /bye to exit.

CodeLLama CLI

- ollama pull codellama:latest
- ollama run codellama:latest

Using Ollama as CoPilot

- In VSCode (or Cursor)
  - Install the “Continue” extension
  - Exit your editor (eg: VSCode)
  - Edit the .continue config.json file
  - (in your home dir. Eg: nano ~/.continue/config.json) to include codellama in your models.

```
{
 "models": [
   {
     "title": "CodeLlama",
     "provider": "ollama",
     "model": "codellama"
   },
```

  - Start / restart your editor
  - Select CodeLlama in your “Continue” selector.

Some fun Continue functions

- Cmd/Ctrl + M = Select code
  - Bring the selection into the context
- Cmd/Ctrl + Shift + M = Select code for follow-up
- Cmd/Ctrl + Shift + L = Quick edit
- Cmd/Ctrl + Shift + R = Automatically debug terminal

Notes from the Workshop

- Models and Quantization
  - On Ollama.com, under the “Models” link at the top of the page, you’ll find a list of available models. (eg: CodeLlama, Mixtral, Dolphin-Mistral) for various use cases.
  - Finding the proper sized model to run on your local environment can depend on the size, and quantization (compression) of the model, with some performance/accuracy tradeoffs. Testing is important.
  - If you look at the “Tags” subsection of a particular model, you’ll find a collection of builds for each model in a variety of sizes and quantization (compression) levels.
    - Eg: https://ollama.com/library/llama2/tags brings up available llama2 builds
  - The filename of the model will usually tell you it’s quantization level, denoted by a q#, which can be helpful in predicting its performance on your system.
    - Eg: ollama run llama2:7b-text-q4_1 is encoded with 4-bit precision/quantization. (-q4)
  - RMAIIG members with M1 macbooks described models in the 4GB range with 4-bit precision/quantization as a reasonable sweet-spot (as a generalization. Your mileage may vary)

More Notes

- Ollama Web UI is a nice local web-frontend for Ollama, that gives you familiar GPT-style chat conversation features.
  - (https://github.com/ollama-webui/ollama-webui)
- Several RMAIIG members preferred Cursor (https://cursor.sh/) as their IDE of choice. It's a fork of VSCode with some nice LLM-centric features for bringing code into the context window, and applying LLM suggestions to both the code and terminal windows.
- LM Studio (https://lmstudio.ai/) is an interesting alternative to Ollama that some members used, with its own model builds and a more gui-centric approach to model management.

*CopyLeft Statement

These materials are free to use, modify, copy and distribute under the GNU GPL v3.: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

These materials are distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.