Using DeepSeek-R1 Locally

Link: https://www.kdnuggets.com/using-deepseek-r1-locally

Run powerful reasoning models locally, matching the performance of OpenAI's o1 capabilities, completely free, and avoid paying $200 a month for a pro subscription.

By **Abid Ali Awan**, KDnuggets Assistant Editor on January 27, 2025 in Language Models

Using DeepSeek-R1 Locally
Image by Author

Many professionals are buzzing about the new DeepSeek model, claiming it could be an "OpenAI killer," and the hype surrounding it seems justified. Recently, DeepSeek launched the DeepSeek-R1-Zero and DeepSeek-R1 models in various versions. These models deliver performance comparable to OpenAI's o1 on benchmarks like MMLU, Math-500, Codeforces, and more.

In this short tutorial, we will explore the DeepSeek-R1 model and walk through how to run its Distill version locally using Ollama, Docker, and Open WebUI. This means you will be able to use a reasoning model with a user interface similar to ChatGPT—completely free and without needing an internet connection.

What is DeepSeek-R1?

DeepSeek has introduced first-generation reasoning models, DeepSeek-R1-Zero and DeepSeek-R1. The DeepSeek-R1 was built upon its predecessor, DeepSeek-R1-Zero, which was trained exclusively with large-scale reinforcement learning (RL) without supervised fine-tuning (SFT). While DeepSeek-R1-Zero showcased advanced reasoning behaviors such as self-verification and generating long chain-of-thoughts (CoTs), it faced challenges like repetitive responses, poor readability, and language mixing. To address these limitations, DeepSeek-R1 incorporates cold-start data before RL, enhancing reasoning performance across math, code, and logic tasks. It achieves results similar to OpenAI-o1 and has led to developing smaller, high-performing distilled models, such as DeepSeek-R1-Distill-Qwen-32B, which achieves state-of-the-art results on reasoning benchmarks.

Setting up Open WebUI

Before we install Open WebUI, an open-source chat user interface similar to ChatGPT, we have to download and install Docker desktop by going to the official website: https://www.docker.com/.

After that, you can pull the Open WebUI image from the GitHub container repository by typing the following command in the terminal.

```
docker pull ghcr.io/open-webui/open-webui:main
```

After successfully pulling the Docker image, we need to run the Docker container using the Open WebUI image. We will map the volume for persistent data storage with the option `-v open-webui:/app/backend/data`. Additionally, we will map the port using `-p 9783:8080`, which exposes the WebUI on port 9783 of your local machine.

```
docker run -d -p 9783:8080 -v open-webui:/app/backend/data --name open-webui ghcr.io/open-webui/open-webui:main
```

Using DeepSeek-R1 Locally

Wait a few seconds, then access the web app by copying and pasting the URL http://localhost:9783/ into your browser. It will prompt you to create an account, and after that, you will be redirected to the main chat menu. As you can see, there are no models available for selection. To resolve this, we will set up Ollama next.

Using DeepSeek-R1 Locally

Setting up Ollama

Go to the official website, https://ollama.com/, to download and install Ollama. Afterward, go to the “Models” menu and select the deepseek-r1 option. This page will contain a run command to download and run various versions of the DeepSeek R1 models.

Using DeepSeek-R1 Locally

In our case, we will be downloading the 8B Llama DeepSeek R1 model by typing the following command in the terminal.

```
ollama run deepseek-r1:8b
```

Using DeepSeek-R1 Locally

Using DeepSeek-R1 Locally

Refresh the Open WebUI page, and you will see the `deespseek-r1:8b` model. Select the model and start using it.

Using DeepSeek-R1 Locally

After typing the default prompt, it took the model 18 seconds to think before responding, which is great and similar to the o1 model.

Using DeepSeek-R1 Locally

You can see the thought process by clicking on the “Thought for 18 Second” drop-down menu.

Using DeepSeek-R1 Locally

The model response generation was fast, close to 54 tokens per second. This is the best performance you can achieve from the 8B parameter quantized model.

To test the full version of the DeepSeek-R1 model, please visit https://chat.deepseek.com/ and select the `DeepThink (R1)` option.

Conclusion

Open-source AI is the future, and even big tech giants recognize this. With companies from China entering the scene, we, as users and everyday people, have a great opportunity to take advantage of advanced AI models privately using local resources.

All you need to do is install Ollama and Docker, then pull the Docker image of the Open WebUI application using a simple Docker command. Trust me, it’s that straightforward! This setup

requires limited computing resources, so even a laptop with 8GB of RAM and no GPU can run these models effectively.

So, what are you waiting for? Start building and integrating these tools into your workspace.

Abid Ali Awan (@1abidaliawan) is a certified data scientist professional who loves building machine learning models. Currently, he is focusing on content creation and writing technical blogs on machine learning and data science technologies. Abid holds a Master's degree in technology management and a bachelor's degree in telecommunication engineering. His vision is to build an AI product using a graph neural network for students struggling with mental illness.

More On This Topic

- Using Groq Llama 3 70B Locally: Step by Step Guide
- Using FLUX.1 Locally
- Using Llama 3.2 Locally
- Run an LLM Locally with LM Studio
- Use (Almost) Any Language Model Locally with Ollama and Hugging Face Hub
- Ollama Tutorial: Running LLMs Locally Made Super Simple

Revision #1
Created 2025-01-28 16:19:30 UTC by Administrador
Updated 2025-01-28 16:25:38 UTC by Administrador