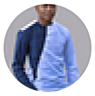
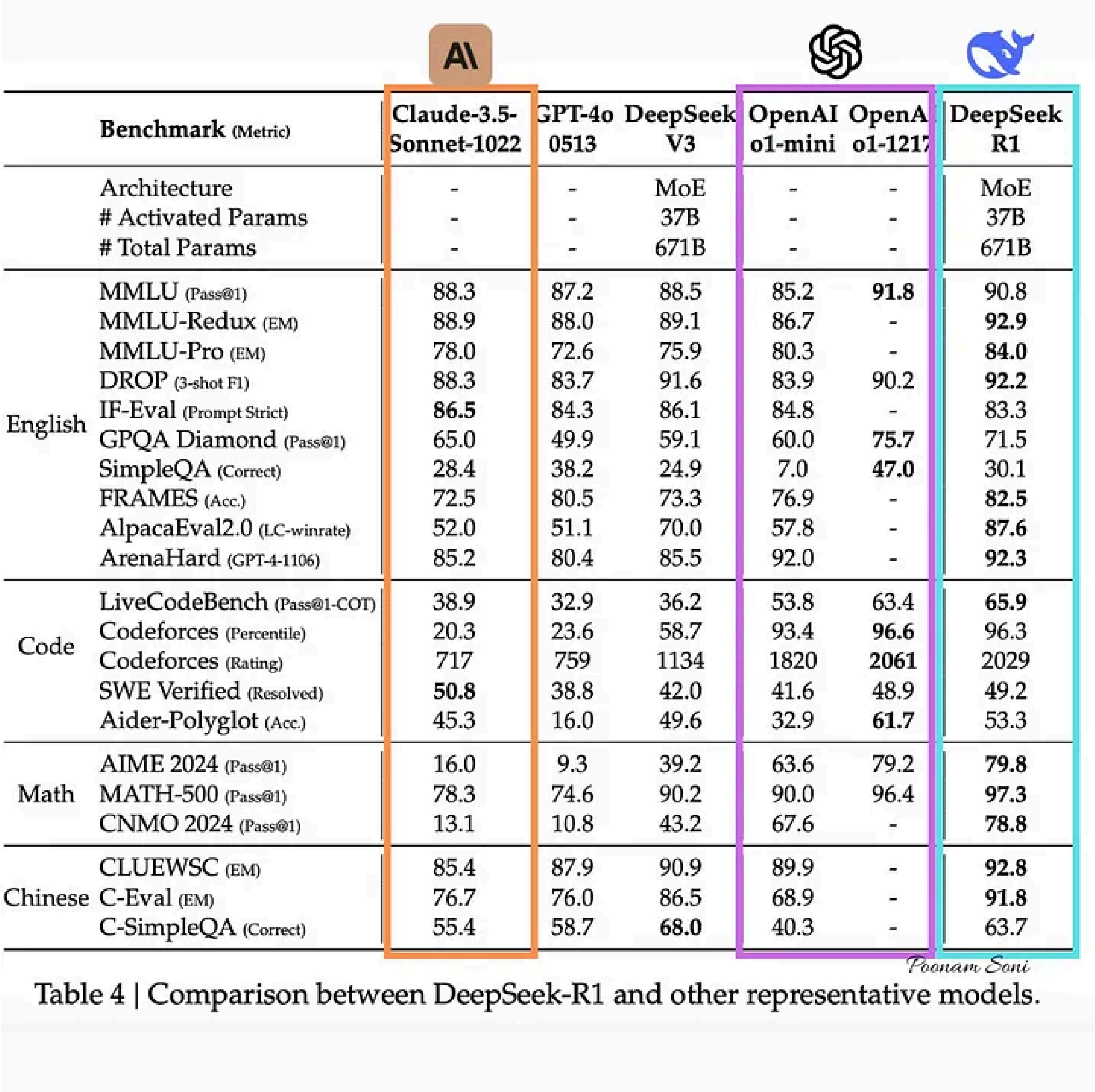
A Simple Guide to DeepSeek R1: Architecture, Training, Local Deployment, and Hardware Requirements



Isaak Kamau · Follow

7 min read · Jan 23, 2025



	Benchmark (Metric)	Claude-3.5-Sonnet-1022	GPT-4o 0513	DeepSeek V3	OpenAI o1-mini	OpenAI o1-1217	DeepSeek R1
	Architecture	-	-	MoE	-	-	MoE
	# Activated Params	-	-	37B	-	-	37B
	# Total Params	-	-	671B	-	-	671B
English	MMLU (Pass@1)	88.3	87.2	88.5	85.2	**91.8**	90.8
	MMLU-Redux (EM)	88.9	88.0	89.1	86.7	-	**92.9**
	MMLU-Pro (EM)	78.0	72.6	75.9	80.3	-	**84.0**
	DROP (3-shot F1)	88.3	83.7	91.6	83.9	90.2	**92.2**
	IF-Eval (Prompt Strict)	**86.5**	84.3	86.1	84.8	-	83.3
	GPQA Diamond (Pass@1)	65.0	49.9	59.1	60.0	**75.7**	71.5
	SimpleQA (Correct)	28.4	38.2	24.9	7.0	**47.0**	30.1
	FRAMES (Acc.)	72.5	80.5	73.3	76.9	-	**82.5**
	AlpacaEval2.0 (LC-winrate)	52.0	51.1	70.0	57.8	-	**87.6**
	ArenaHard (GPT-4-1106)	85.2	80.4	85.5	92.0	-	**92.3**
Code	LiveCodeBench (Pass@1-COT)	38.9	32.9	36.2	53.8	63.4	**65.9**
	Codeforces (Percentile)	20.3	23.6	58.7	93.4	**96.6**	96.3
	Codeforces (Rating)	717	759	1134	1820	**2061**	2029
	SWE Verified (Resolved)	**50.8**	38.8	42.0	41.6	48.9	49.2
	Aider-Polyglot (Acc.)	45.3	16.0	49.6	32.9	**61.7**	53.3
Math	AIME 2024 (Pass@1)	16.0	9.3	39.2	63.6	79.2	**79.8**
	MATH-500 (Pass@1)	78.3	74.6	90.2	90.0	96.4	**97.3**
	CNMO 2024 (Pass@1)	13.1	10.8	43.2	67.6	-	**78.8**
Chinese	CLUEWSC (EM)	85.4	87.9	90.9	89.9	-	**92.8**
	C-Eval (EM)	76.7	76.0	86.5	68.9	-	**91.8**
	C-SimpleQA (Correct)	55.4	58.7	**68.0**	40.3	-	63.7

Table 4 | Comparison between DeepSeek-R1 and other representative models.

DeepSeek’s Novel Approach to LLM Reasoning

DeepSeek has introduced an innovative approach to improving the reasoning capabilities of large language models (LLMs) through reinforcement learning (RL), detailed in their recent paper on DeepSeek-R1. This research represents a significant advancement in how we can enhance LLMs’ ability to solve complex problems through pure reinforcement learning, without relying heavily on supervised fine-tuning.

Before we proceed if you like this topic and you want to support me:

1. ***Clap** my article 10 times; that will help me out.* 👏
2. ***Follow** me on Medium to get my latest articles* 🫶

Technical Overview of DeepSeek-R1

Model Architecture:

DeepSeek-R1 is not a singular model but a family of models, encompassing: **DeepSeek-R1-Zero** and **DeepSeek-R1**

Let me clarify the key differences between DeepSeek-R1 and DeepSeek-R1-Zero:

The Primary Distinction

DeepSeek-R1-Zero represents the team's initial experiment using pure reinforcement learning without any supervised fine-tuning. They started with their base model and applied reinforcement learning directly, letting the model develop reasoning capabilities through trial and error. While this approach achieved impressive results (71% accuracy on AIME 2024), it had some significant limitations, particularly in readability and language consistency. It features 671 billion parameters, utilizing a mixture-of-experts (MoE) architecture where each token activates parameters equivalent to 37 billion. This model showcases emergent reasoning behaviors, such as self-verification, reflection, and long chain-of-thought (CoT) reasoning.

DeepSeek-R1, in contrast, uses a more sophisticated multi-stage training approach. Instead of pure reinforcement learning, it begins with supervised fine-tuning on a small set of carefully curated examples (called "cold-start data") before applying reinforcement learning. This approach addresses the limitations of DeepSeek-R1-Zero while achieving even better performance. This model also maintains the 671 billion parameter count but achieves better readability and coherence in responses.

The Training Process Comparison

Training Methodology:

- **Reinforcement Learning:** Unlike traditional models that predominantly rely on supervised learning, DeepSeek-R1 uses RL extensively. The training leverages group relative policy optimization (GRPO), focusing on accuracy and format rewards to enhance reasoning capabilities without the need for extensive labeled data.
- **Distillation Techniques:** To democratize access to high-performing models, DeepSeek has also released distilled versions of R1, ranging from 1.5 billion to 70 billion parameters. These models are based on architectures like Qwen and Llama, showing that complex reasoning can be encapsulated in smaller, more efficient models. The distillation process involves fine-tuning these smaller models with synthetic reasoning data generated by the full DeepSeek-R1, thus preserving high performance at reduced computational cost.

DeepSeek-R1-Zero's training process is straightforward:

- Start with base model
- Apply reinforcement learning directly
- Use simple rewards based on accuracy and format

DeepSeek-R1's training process has four distinct stages:

1. Initial supervised fine-tuning with thousands of high-quality examples
2. Reinforcement learning focused on reasoning tasks
3. Collection of new training data through rejection sampling
4. Final reinforcement learning across all types of tasks

Performance Metrics:

- **Reasoning Benchmarks**: DeepSeek-R1 has shown impressive results on various benchmarks:
- **AIME 2024**: Achieved a 79.8% pass rate, compared to 79.2% by OpenAI's o1–1217.
- **MATH-500**: Scored an impressive 97.3%, slightly ahead of o1–1217's 96.4%.
- **SWE-bench Verified**: Outperformed in programming tasks, showcasing its coding proficiency.
- **Cost Efficiency:** The API for DeepSeek-R1 is priced at $0.14 per million input tokens for cache hits, making it significantly cheaper than comparable models like OpenAI's o1.

Limitations and Future Work

The paper acknowledges several areas for improvement:

- The model sometimes struggles with tasks requiring specific output formats

- Performance on software engineering tasks could be enhanced
- There are challenges with language mixing in multilingual contexts
- Few-shot prompting consistently degrades performance

Future work will focus on addressing these limitations and expanding the model's capabilities in areas like function calling, multi-turn interactions, and complex role-playing scenarios.

Deployment and Accessibility

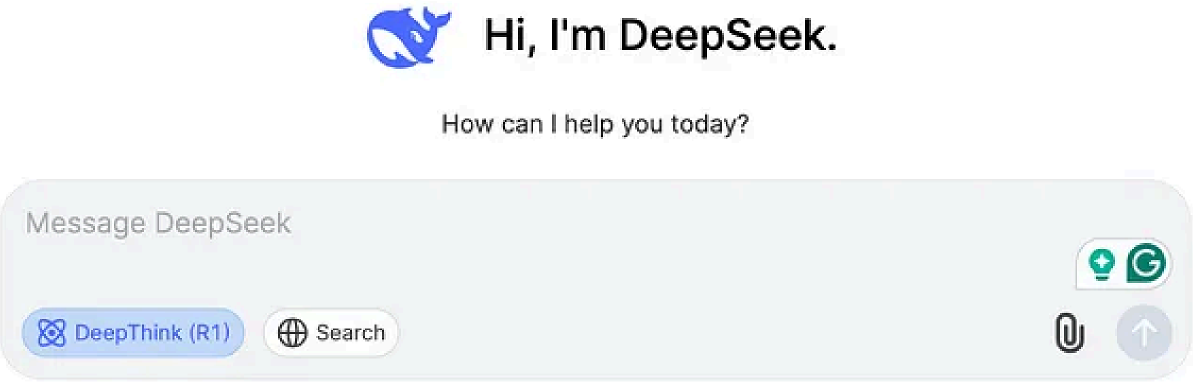
- **Open Source and Licensing**: DeepSeek-R1 and its variants are released under the MIT License, promoting open-source collaboration and commercial use, including model distillation. This move is pivotal for fostering innovation and reducing the entry barriers in AI model development.
- **Model Formats:**
- Both models and their distilled versions are available in formats like GGML, GGUF, GPTQ, and HF, allowing flexibility in how they are deployed locally.

1. Web Access via DeepSeek Chat Platform:

The DeepSeek Chat platform provides a user-friendly interface to interact with DeepSeek-R1 without any setup requirements.

- **Steps to Access:**
- Navigate to the DeepSeek Chat platform
- Register for an account or log in if you already have one.

- After logging in, select the “Deep Think” mode to experience DeepSeek-R1’s step-by-step reasoning capabilities.



Hi, I'm DeepSeek.

How can I help you today?

DeepSeek Chat Platform

2. Access via DeepSeek API:

For programmatic access, DeepSeek offers an API compatible with OpenAI’s format, allowing integration into various applications.

Steps to Use the API:

a. Obtain an API Key:

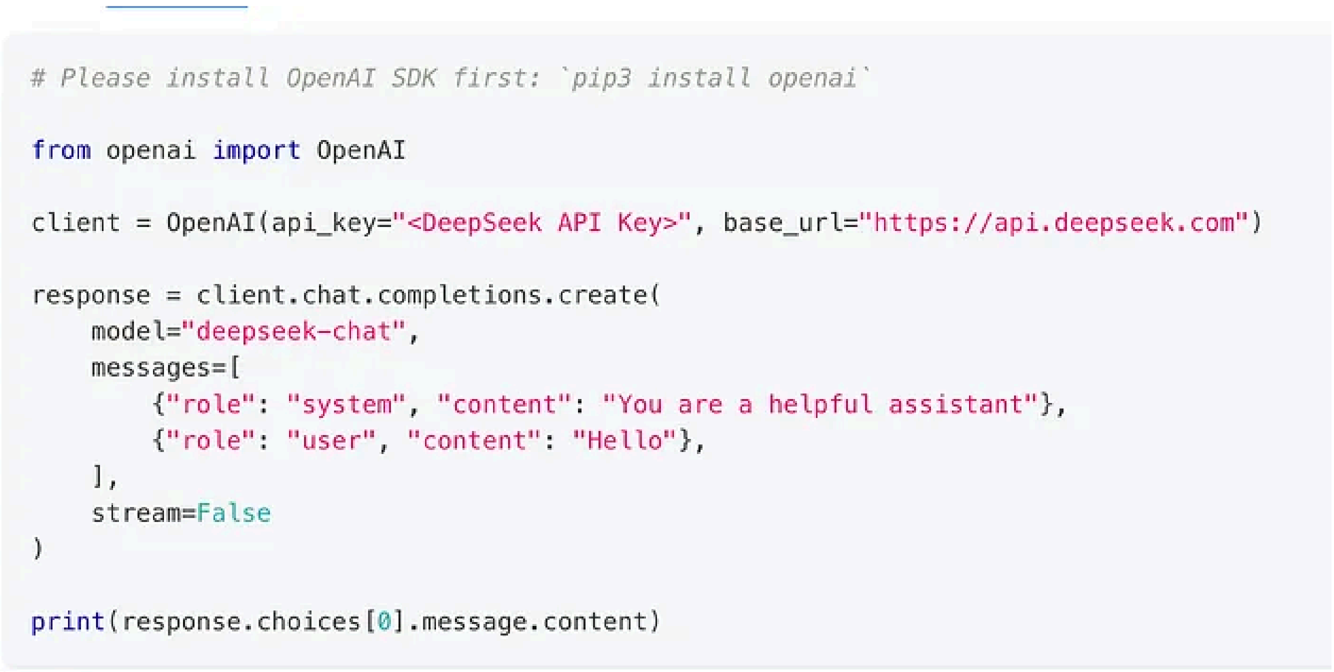
- Visit the DeepSeek API platform to create an account and generate your unique API key.

b. Configure Your Environment:

- Set the `base_url` to `https://api.deepseek.com/v1`.
- Use your API key for authentication, typically via Bearer Token in the HTTP header.

c. Make API Calls:

- Utilize the API to send prompts and receive responses from DeepSeek-R1.
- Detailed documentation and examples are available in the DeepSeek API Docs.



```
# Please install OpenAI SDK first: `pip3 install openai`

from openai import OpenAI

client = OpenAI(api_key="<DeepSeek API Key>", base_url="https://api.deepseek.com")

response = client.chat.completions.create(
    model="deepseek-chat",
    messages=[
        {"role": "system", "content": "You are a helpful assistant"},
        {"role": "user", "content": "Hello"},
    ],
    stream=False
)

print(response.choices[0].message.content)
```

DeepSeek API call example

3. Running DeepSeek-R1 Locally:

Both Models (R1 and R1-Zero):

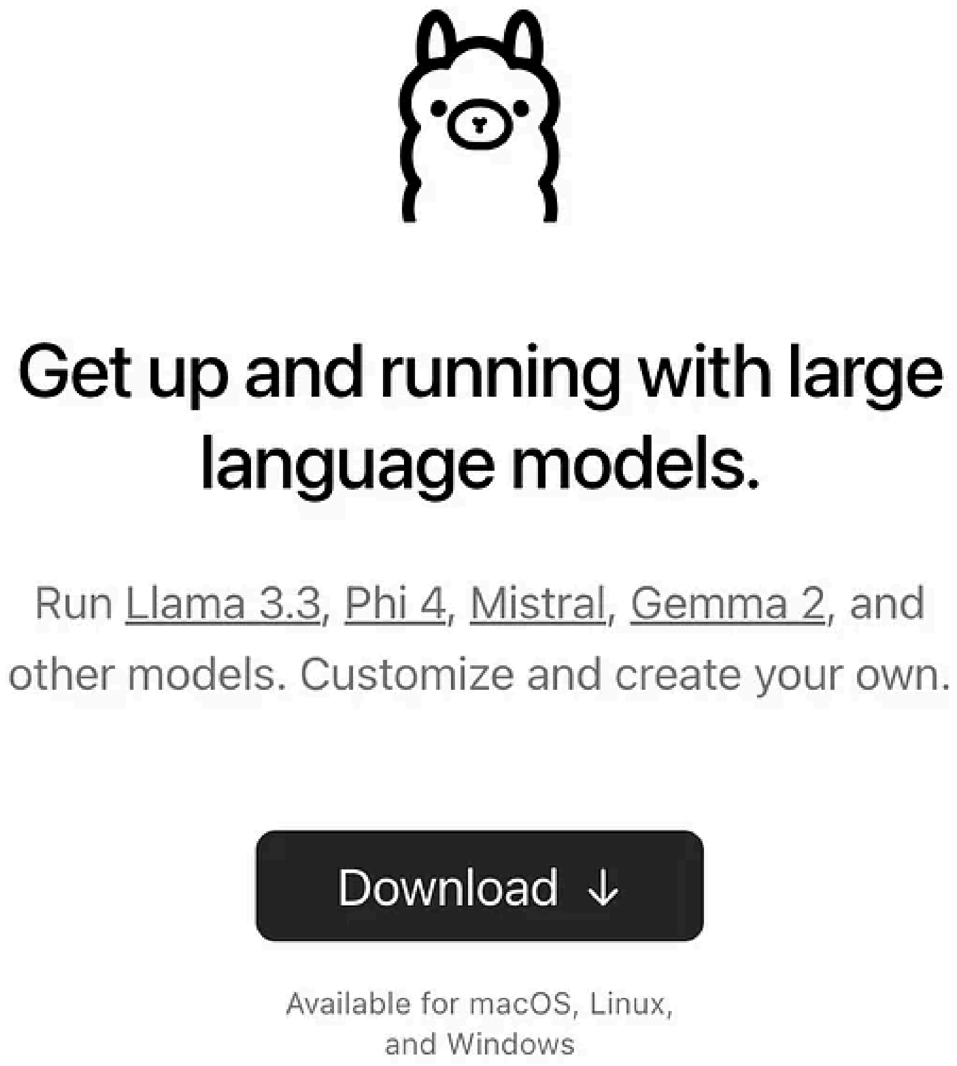
- **Hardware Requirements:** The full models require significant hardware due to their size. A GPU with substantial VRAM (like Nvidia RTX 3090 or higher) is recommended. For CPU use, you'd need at least 48GB of RAM and 250GB of disk space, although performance would be slow without GPU acceleration.

- **Distilled Models:** For local deployment with less resource-intensive hardware, DeepSeek provides distilled versions. These range from 1.5B to 70B parameters, making them suitable for systems with more modest hardware. For instance, the 7B model can run on a GPU with at least 6GB VRAM or on a CPU with about 4GB RAM for the GGML/GGUF format.

Software Tools for Local Running:

1. **Ollama:**

You can use Ollama to serve the models locally: (Ollama Is a tool for running open-source AI models locally on your machine. Grab it here: https://ollama.com/download)



Get up and running with large language models.

Run Llama 3.3, Phi 4, Mistral, Gemma 2, and other models. Customize and create your own.

Available for macOS, Linux, and Windows

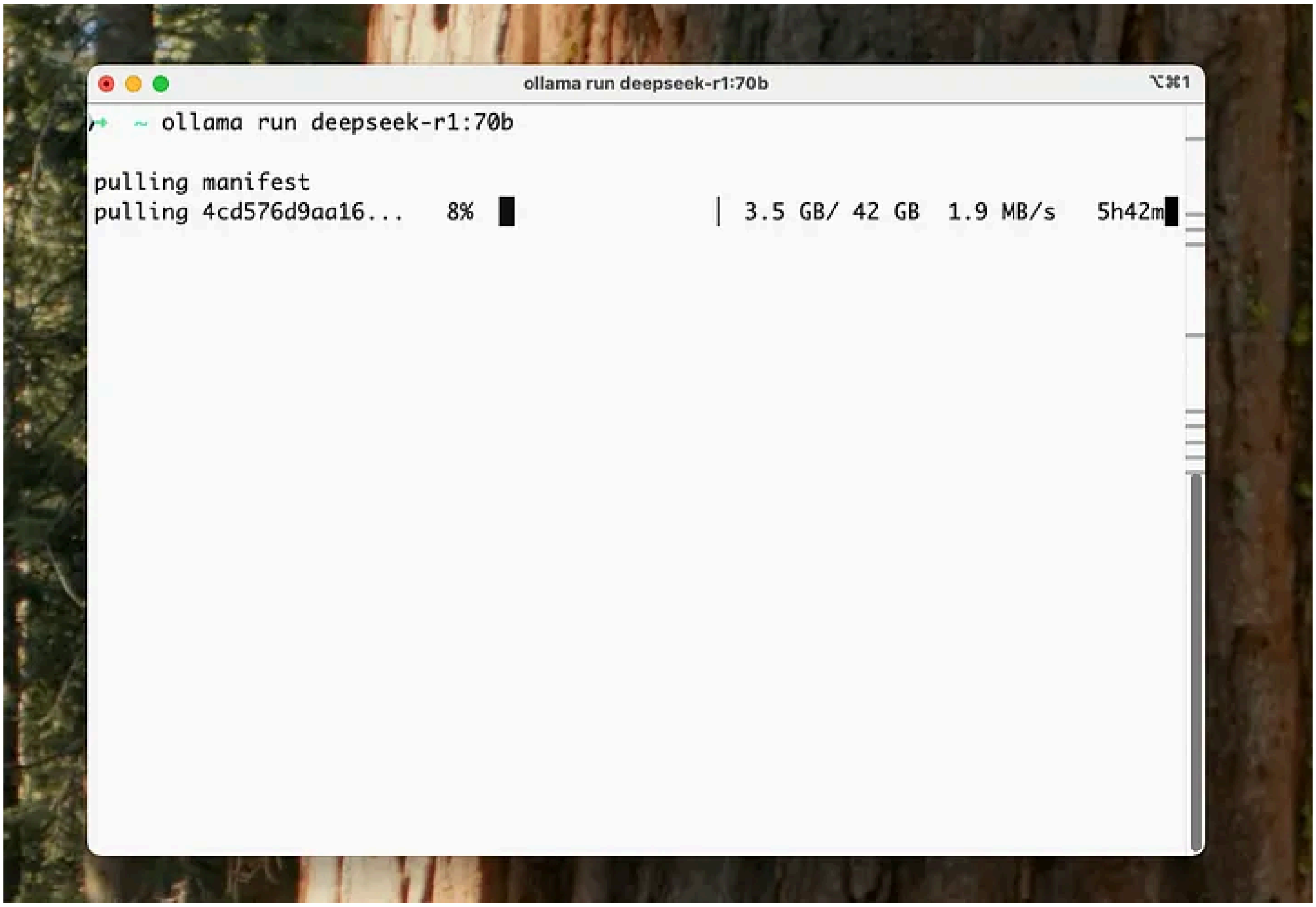
Next, you'll need to pull and run the DeepSeek R1 model locally.

Ollama offers different model sizes — basically, bigger models equal to smarter AI, but need better GPU. Here's the lineup:

```
1.5B version (smallest):
ollama run deepseek-r1:1.5b

8B version:
ollama run deepseek-r1:8b

14B version:
ollama run deepseek-r1:14b

32B version:
ollama run deepseek-r1:32b

70B version (biggest/smartest):
ollama run deepseek-r1:70b
```

To begin experimenting with DeepSeek-R1, it is advisable to start with a smaller model to familiarize yourself with the setup and ensure compatibility with your hardware. You can initiate this process by opening your terminal and executing the following command:

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



```
➜  ~ ollama run deepseek-r1:70b

pulling manifest
pulling 4cd576d9aa16...   8% █               | 3.5 GB/ 42 GB  1.9 MB/s   5h42m
```

Image courtesy from Reddit, via r/macapps

Sending Requests to locally downloaded DeepSeek-R1 via Ollama:

Ollama provides an API endpoint to interact with DeepSeek-R1 programmatically. Ensure that the Ollama server is running locally before making API requests. You can start the server by running:

```
ollama serve
```

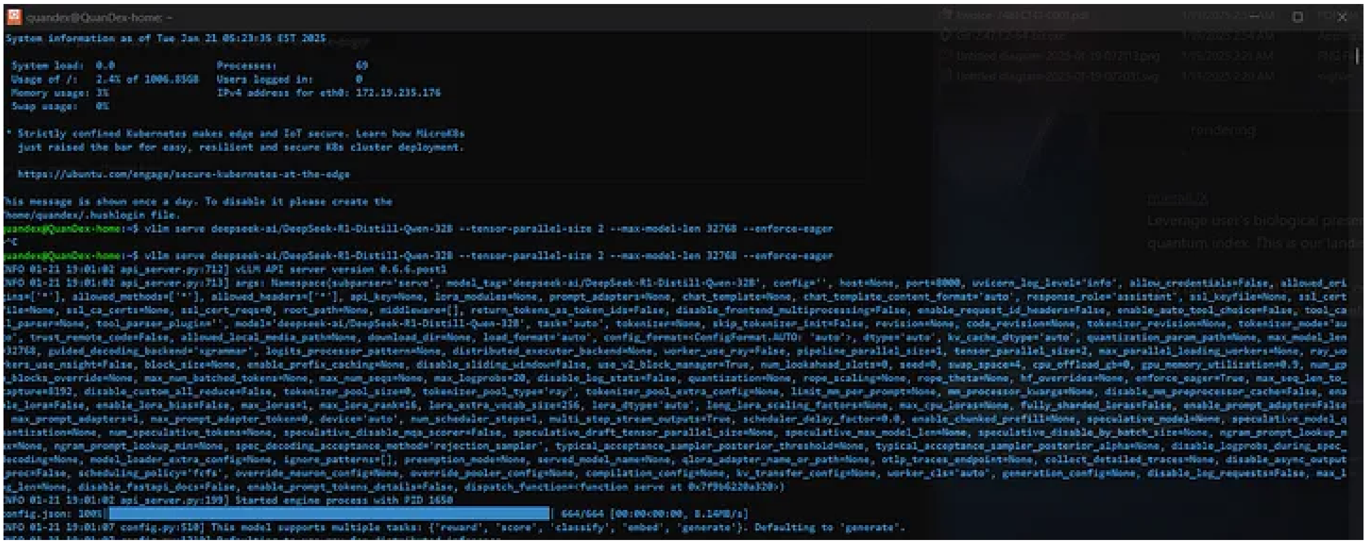
Once the server is active, you can send a request using `curl` as follows:

```
curl -X POST http://localhost:11434/api/generate -d '{
  "model": "deepseek-r1",
  "prompt": "Your question or prompt here"
}'
```

Replace `"Your question or prompt here"` with the actual input you wish to provide to the model. This command sends a POST request to the local Ollama server, which processes the prompt using the specified DeepSeek-R1 model and returns the generated response.

Other methods to run/Access the models locally are:

vLLM/SGLang: Used for serving the models locally. Commands like vllm serve deepseek-ai/DeepSeek-R1-Distill-Qwen-32B — tensor-parallel-size 2 — max-model-len 32768 — enforce-eager can be used for the distilled versions.



```
System information as of Tue Jan 21 05:23:35 EST 2025

 System load:  0.0                Processes:             69
 Usage of /:   2.4% of 1006.85GB  Users logged in:       0
 Memory usage: 3%                 IPv4 address for eth0: 172.19.235.176
 Swap usage:   0%

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
  just raised the bar for easy, resilient and secure K8s cluster deployment.

  https://ubuntu.com/engage/secure-kubernetes-at-the-edge

his message is shown once a day. To disable it please create the
home/quandex/.hushlogin file.
uandex@QuanDex-home:~$ vllm serve deepseek-ai/DeepSeek-R1-Distill-Qwen-32B --tensor-parallel-size 2 --max-model-len 32768 --enforce-eager
^C
uandex@QuanDex-home:~$ vllm serve deepseek-ai/DeepSeek-R1-Distill-Qwen-32B --tensor-parallel-size 2 --max-model-len 32768 --enforce-eager
NFO 01-21 13:01:02 api_server.py:712] vLLM API server version 0.6.6.post1
NFO 01-21 13:01:02 api_server.py:713] args: Namespace(subparser='serve', model_tag='deepseek-ai/DeepSeek-R1-Distill-Qwen-32B', config='', host=None, port=8000, uvicorn_log_level='info', allow_credentials=False, allowed_ori
ins=['*'], allowed_methods=['*'], allowed_headers=['*'], api_key=None, lora_modules=None, prompt_adapters=None, chat_template=None, chat_template_content_format='auto', response_role='assistant', ssl_keyfile=None, ssl_cert
ile=None, ssl_ca_certs=None, ssl_cert_reqs=0, root_path=None, middleware=[], return_tokens_as_token_ids=False, disable_frontend_multiprocessing=False, enable_request_id_headers=False, enable_auto_tool_choice=False, tool_ca
l_parser=None, tool_parser_plugin='', model='deepseek-ai/DeepSeek-R1-Distill-Qwen-32B', task='auto', tokenizer=None, skip_tokenizer_init=False, revision=None, code_revision=None, tokenizer_revision=None, tokenizer_mode='au
o', trust_remote_code=False, allowed_local_media_path=None, download_dir=None, load_format='auto', config_format=<ConfigFormat.AUTO: 'auto'>, dtype='auto', kv_cache_dtype='auto', quantization_param_path=None, max_model_len
32768, guided_decoding_backend='xgrammar', logits_processor_pattern=None, distributed_executor_backend=None, worker_use_ray=False, pipeline_parallel_size=1, tensor_parallel_size=2, max_parallel_loading_workers=None, ray_wo
kers_use_nsight=False, block_size=None, enable_prefix_caching=None, disable_sliding_window=False, use_v2_block_manager=True, num_lookahead_slots=0, seed=0, swap_space=4, cpu_offload_gb=0, gpu_memory_utilization=0.9, num_gp
_blocks_override=None, max_num_batched_tokens=None, max_num_seqs=None, max_logprobs=20, disable_log_stats=False, quantization=None, rope_scaling=None, rope_theta=None, hf_overrides=None, enforce_eager=True, max_seq_len_to_
apture=8192, disable_custom_all_reduce=False, tokenizer_pool_size=0, tokenizer_pool_type='ray', tokenizer_pool_extra_config=None, limit_mm_per_prompt=None, mm_processor_kwargs=None, disable_mm_preprocessor_cache=False, ena
le_lora=False, enable_lora_bias=False, max_loras=1, max_lora_rank=16, lora_extra_vocab_size=256, lora_dtype='auto', long_lora_scaling_factors=None, max_cpu_loras=None, fully_sharded_loras=False, enable_prompt_adapter=False
 max_prompt_adapters=1, max_prompt_adapter_token=0, device='auto', num_scheduler_steps=1, multi_step_stream_outputs=True, scheduler_delay_factor=0.0, enable_chunked_prefill=None, speculative_model=None, speculative_model_q
antization=None, num_speculative_tokens=None, speculative_disable_mqa_scorer=False, speculative_draft_tensor_parallel_size=None, speculative_max_model_len=None, speculative_disable_by_batch_size=None, ngram_prompt_lookup_m
x=None, ngram_prompt_lookup_min=None, spec_decoding_acceptance_method='rejection_sampler', typical_acceptance_sampler_posterior_threshold=None, typical_acceptance_sampler_posterior_alpha=None, disable_logprobs_during_spec_
ecoding=None, model_loader_extra_config=None, ignore_patterns=[], preemption_mode=None, served_model_name=None, qlora_adapter_name_or_path=None, otlp_traces_endpoint=None, collect_detailed_traces=None, disable_async_output
proc=False, scheduling_policy='fcfs', override_neuron_config=None, override_pooler_config=None, compilation_config=None, kv_transfer_config=None, worker_cls='auto', generation_config=None, disable_log_requests=False, max_l
g_len=None, disable_fastapi_docs=False, enable_prompt_tokens_details=False, dispatch_function=<function serve at 0x7f9b6220a320>)
NFO 01-21 13:01:02 api_server.py:199] Started engine process with PID 1650
onfig.json: 100%|████████████████████████████████████████| 664/664 [00:00<00:00, 8.14MB/s]
NFO 01-21 13:01:07 config.py:510] This model supports multiple tasks: {'reward', 'score', 'classify', 'embed', 'generate'}. Defaulting to 'generate'.
```

Courtesy: HuggingFace

llama.cpp: You can also use llama.cpp to run the models locally.

See What Others Are Building with DeepSeek-R1:

1. Running DeepSeek R1 across my 7 M4 Pro Mac Minis and 1 M4 Max MacBook Pro:

DeepSeek R1 1.5B running fully locally in your browser at 60 tok/ sec powered by WebGPU:

RAG app to chat with your PDF files using the DeepSeek R1 model, running locally on your computer.

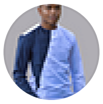
Running DeepSeek R1 version 1.5B perfectly locally on phone:

Cracking complex math problems with ease! (Thought for ~3200 tokens in about 35 seconds on M4 Max with mlx-lm.):

Conclusions:

This progression from DeepSeek-R1-Zero to DeepSeek-R1 represents an important learning journey in the research. While DeepSeek-R1-Zero proved that pure reinforcement learning could work, DeepSeek-R1 showed how combining supervised learning with reinforcement learning could create an even more capable and practical model.

Collaborations 🤝: Have an interesting AI project in mind? Let’s team up! I’m available for collaboration on AI and machine learning initiatives, and keen to connect with other professionals in the field.



Written by Isaak Kamau

766 Followers · 25 Following

Follow

AI Developer: https://www.linkedin.com/in/isaak-mwangi/

Responses (26)

What are your thoughts?

Respond

Dr. Derek Austin 🥳 he/him
5 days ago

Useful, thanks Isaak!

38 1 reply Reply

mou nandi
4 days ago

Thank you for the quick write up.

14 1 reply Reply

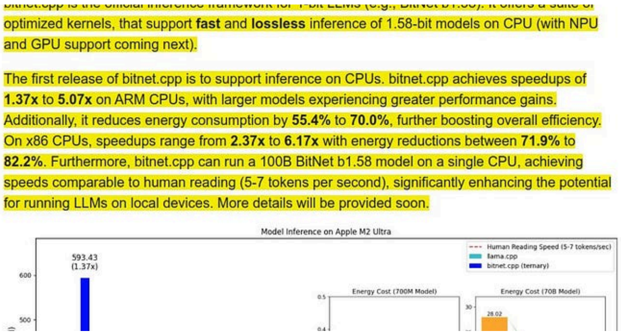
Asrofi Al Kindi
5 days ago

great article, thanks

10 1 reply Reply

See all responses

More from Isaak Kamau



bitnet.cpp is the official inference framework for 1-bit LLMs (e.g., BitNet b1.58). It offers a suite of optimized kernels, that support **fast** and **lossless** inference of 1.58-bit models on CPU (with NPU and GPU support coming next).

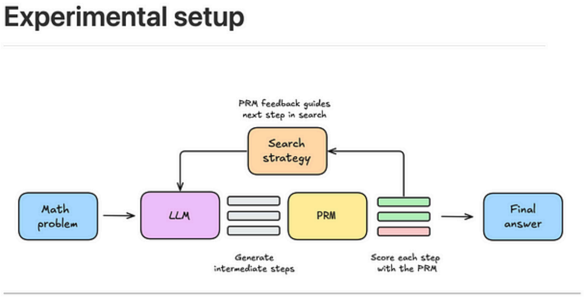
The first release of bitnet.cpp is to support inference on CPUs. bitnet.cpp achieves speedups of **1.37x** to **5.07x** on ARM CPUs, with larger models experiencing greater performance gains. Additionally, it reduces energy consumption by **55.4%** to **70.0%**, further boosting overall efficiency. On x86 CPUs, speedups range from **2.37x** to **6.17x** with energy reductions between **71.9%** to **82.2%**. Furthermore, bitnet.cpp can run a 100B BitNet b1.58 model on a single CPU, achieving speeds comparable to human reading (5-7 tokens per second), significantly enhancing the potential for running LLMs on local devices. More details will be provided soon.

Isaak Kamau

Microsoft Open-Sources 1-bit LLMs: Run 100B Parameter Model...

https://github.com/microsoft/BitNet

✦ Oct 22, 2024 👏 970 💬 9



Experimental setup

Isaak Kamau

Test-Time Compute Scaling: How to make an LLM "think longer" on...

Scaling Test-Time Compute: Unlocking Efficient Performance in Language Models

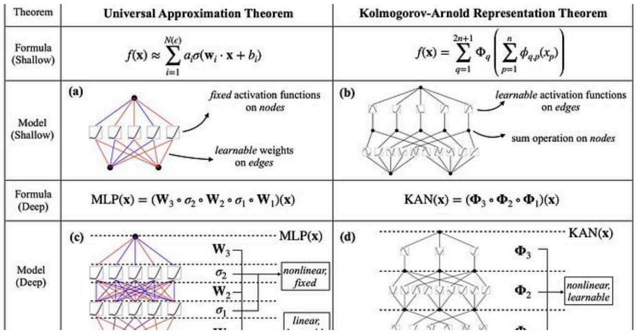
✦ Dec 23, 2024 👏 113

Isaak Kamau

When to Use XGBoost over LLM: A Case Study in Stock Market...

Choosing between XGBoost (eXtreme Gradient Boosting) and a Large Language...

Aug 4, 2024 146



Theorem
Universal Approximation Theorem
Kolmogorov-Arnold Representation Theorem
Formula (Shallow)
$f(\mathbf{x}) \approx \sum_{i=1}^{N(\epsilon)} a_i \sigma(\mathbf{w}_i \cdot \mathbf{x} + b_i)$
$f(\mathbf{x}) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right)$
Model (Shallow)
(a)
fixed activation functions on nodes
learnable weights on edges
(b)
learnable activation functions on edges
sum operation on nodes
Formula (Deep)
$\text{MLP}(\mathbf{x}) = (\mathbf{W}_3 \circ \sigma_2 \circ \mathbf{W}_2 \circ \sigma_1 \circ \mathbf{W}_1)(\mathbf{x})$
$\text{KAN}(\mathbf{x}) = (\mathbf{\Phi}_3 \circ \mathbf{\Phi}_2 \circ \mathbf{\Phi}_1)(\mathbf{x})$
Model (Deep)
(c)
MLP(x)
$\mathbf{W}_3$
σ_2
$\mathbf{W}_2$
σ_1
nonlinear, fixed
linear,
(d)
KAN(x)
$\mathbf{\Phi}_3$
$\mathbf{\Phi}_2$
nonlinear, learnable

Isaak Kamau

A Simplified Explanation Of The New Kolmogorov-Arnold Network...

A screenshot from: https://arxiv.org/abs/2404.19756

May 1, 2024 3.3K 8

See all from Isaak Kamau

Recommended from Medium

The Nam

DeepSeek-R1: Architecture and training explain

Can we train a powerful reasoning LLM without Supervised Fine-Tuning?

Jan 25 207 3

In AI Advances by Kenny Vaneetvelde

Want to Build AI Agents? Tired of LangChain, CrewAI, AutoGen &...

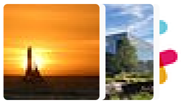
Frameworks like LangChain, CrewAI, and AutoGen have gained popularity by promisin...

Jan 19 1K 22

Lists

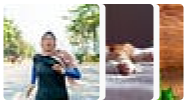
Staff picks

806 stories · 1603 saves



Stories to Help You Level-Up at Work

19 stories · 930 saves



Self-Improvement 101

20 stories · 3258 saves

Productivity 101

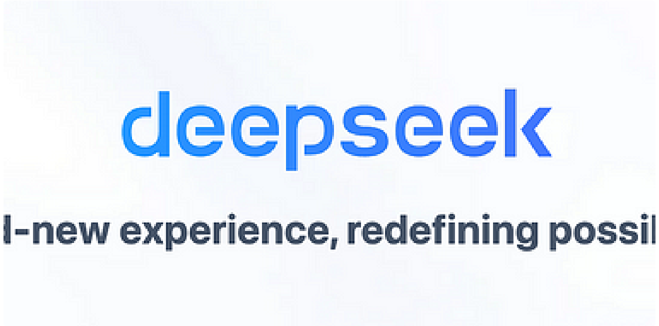
20 stories · 2754 saves

In Level Up Coding by Dr. Ashish Bamania

DeepSeek-R1 Beats OpenAI's o1, Revealing All Its Training Secrets...

A deep dive into how DeepSeek-R1 was trained from scratch and how this open-...

5d ago 1.1K 28

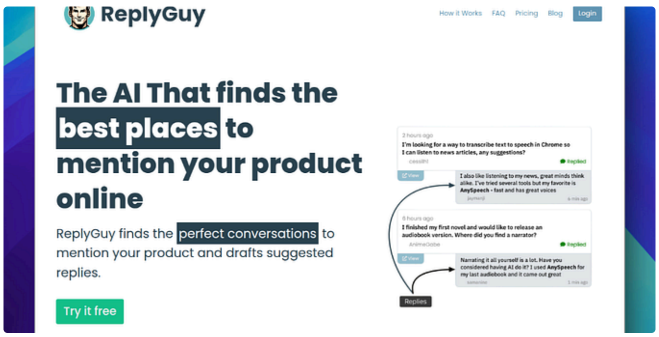


Abhishek Maheshwarappa

Fine-Tuning DeepSeek LLM: Adapting Open-Source AI for You...

Introduction

Jan 21 110 4



The AI That finds the best places to mention your product online

ReplyGuy finds the perfect conversations to mention your product and drafts suggested replies.

2 hours ago
I'm looking for a way to transcribe text to speech in Chrome so I can listen to news articles, any suggestions?
cessihl
Replied

I also like listening to my news, great minds think alike. I've tried several tools but my favorite is AnySpeech - fast and has great voices
jaymenji
6 min ago

8 hours ago
I finished my first novel and would like to release an audiobook version. Where did you find a narrator?
AnimeGabe
Replied

Narrating it all yourself is a lot. Have you considered having AI do it? I used AnySpeech for my last audiobook and it came out great
samanine
1 min ago

Mohit Vaswani

6 AI Agents That Are So Good, They Feel Illegal

AI agents are the future because they can replace all the manual work with automation...

Jan 11 2.1K 81

AI Papers Academy

DeepSeek-R1 Paper Explained — A New RL LLMs Era in AI?

Dive into the groundbreaking DeepSeek-R1 research paper, introduces open-source...

Jan 25 228 3

See more recommendations