راه‌اندازی مدل‌های هوش مصنوعی به صورت کاملا Local / Offline

با تشکر از

- انجمن ملی هوش مصنوعی ایران
  - جناب آقای دکتر زاهدی
  - سرکار خانم دکتر فراهانی
  - جناب آقای مهندس دهمولایی

به طور کلی، هدف شرکت‌های بزرگ هوش مصنوعی چیست؟

- ایجاد مدل‌های ارزان‌تر
- ایجاد مدل‌های کوچک‌تر
- ایجاد مدل‌های قدرت‌مندتر

- Jevons Paradox
  - https://en.wikipedia.org/wiki/Jevons_paradox
  - بر خلاف تصور، افزایش بهره‌وری و به طبع آن، کاهش قیمت، باعث افزایش مصرف می‌شود!

استراتژی ما برای استفاده از مدل‌های هوش مصنوعی

- اول از همه، باید حواسمان باشد که جوگیر نشویم!
- از مدل‌های ارزان و کوچک استفاده می‌نماییم، ولی اگر جواب مناسبی نداد، آرام آرام، از مدل‌های بزرگ‌تر و قوی‌تر استفاده می‌کنیم.

مقایسه مدل‌ها

- https://lmarena.ai
  - Leaderboard

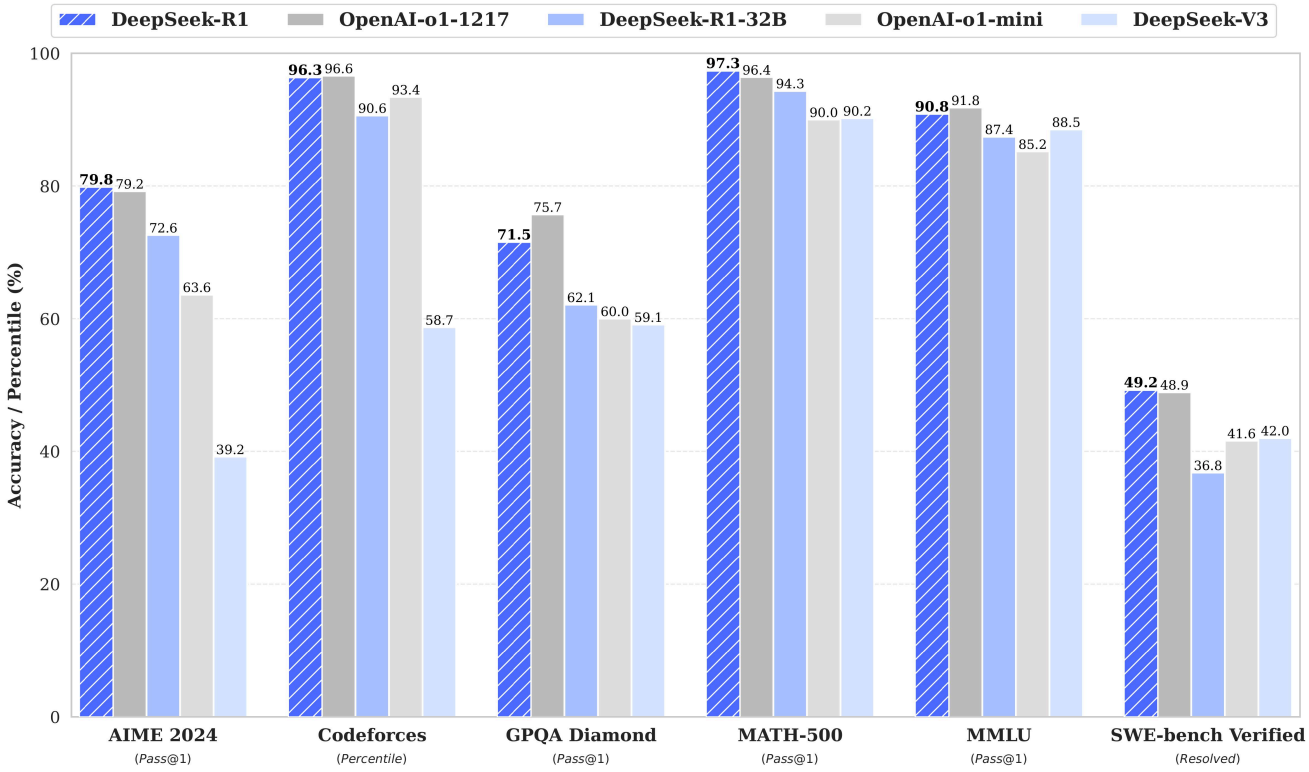
در حال حاضر، اهمیت هوش مصنوعی در کشور عزیزمان ایران، از انرژی هسته‌ای مهم‌تر است!

- کشور چین، چند بمب اتمی می‌توانست به کشور آمریکا بزند، تا بتواند در عرض کمتر از ۷۲ ساعت، بیش از ۲۰۰۰ میلیارد دلار خسارت به اقتصاد آمریکا بزند!؟

DeepSeek R1

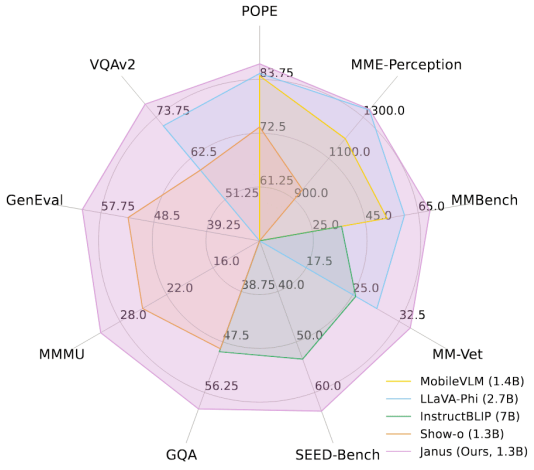
- https://www.deepseek.com
- https://chat.deepseek.com
- https://api-docs.deepseek.com/news/news250120
- https://github.com/deepseek-ai/DeepSeek-V3/tree/main
- https://github.com/deepseek-ai/DeepSeek-R1/tree/main

```
NVIDIA:     Loose 600 b$  (20 Years Iran Oil Export!)
USA Bourse: Loose 2000 b$

Free / Open Source (Download)!

V3:     2.788 Million Hours NVIDIA H800 GPU
GPT O1: 60    Million Hours NVIDIA A100 GPU

Chat DeepSeek: Free / Without Sanction / Without Mobile

API DeepSeek: Very Cheap! - 1M Tokens: 0.14$
```



DeepSeek - Janus Pro

- https://github.com/deepseek-ai/Janus
  - https://huggingface.co/deepseek-ai/Janus-1.3B
  - https://huggingface.co/deepseek-ai/JanusFlow-1.3B
  - https://huggingface.co/deepseek-ai/Janus-Pro-1B
  - https://huggingface.co/deepseek-ai/Janus-Pro-7B
  - - https://huggingface.co/blog/LLMhacker/janus-pro
  - https://huggingface.co/spaces/deepseek-ai/Janus-Pro-7B



(a) Benchmark Performance.

(b) Visual Generation Results.

Qwen 2.5 Plus / Max

- https://chat.qwenlm.ai

- https://github.com/QwenLM/Qwen
- https://huggingface.co/Qwen

Features (Multimodal)

- Text to Text Generation
- Text to Image Generation
- Text to Video Generation
- Artifact (Code Generation) + Preview (Similar Claude AI)
- Web Search

Tulu 3

- https://allenai.org/blog/tulu-3-technical
- https://ollama.com/library/tulu3
- https://github.com/allenai/open-instruct

Aval AI

- این یک سایت ایرانی است که می‌توان از طریق آن، از تعداد زیادی، API های مدل‌های هوش مصنوعی که پولی می‌باشند، به صورت ریالی و بدون مشکلات پرداخت و تحریم‌ها استفاده کرد. مانند GPT, DeepSeek
- https://chat.avalai.ir/?ref=W3OKIT7
- https://chat.avalai.ir/platform/home

How to Install AI Locally

- Use NVIDIA GPU
- Update NVIDIA GPU Driver
- Download & Install CUDA Toolkit (If you have a NVIDIA GPU)
    - In Windows PowerShell
      - check the Nvidia (CUDA) version and VRAM, with below command:

```
nvidia-smi.exe
```

Output

```
Fri Feb  7 15:37:01 2025
+-----------------------------------------------------------------------------
---+
| NVIDIA-SMI 560.94      Driver Version: 560.94         CUDA Version: 12.6
|
```

```
|-------------------------------+----------------------+----------------------
---+
| GPU Name      Driver-Model    | Bus-Id Disp.A          | Volatile Uncorr.
ECC |
| Fan Temp Perf Pwr:Usage/Cap   | Memory-Usage           | GPU-Util  Compute
M. |
|                               |                        |               MIG
M. |
|===============================+======================+======================
===|
|   0 NVIDIA GeForce GTX 1650 Ti   WDDM | 00000000:01:00.0 On | N/A
|
| N/A 49C    P0              16W / 50W  | 1010MiB / 4096MiB    | 3% Default
|
|                                       |                      | N/A
|
+---------------------------------------+----------------------+----------
---+
```

CUDA Zone

- https://developer.nvidia.com/cuda-zone

DT Anti Sanction

- https://github.com/Dariush-Tasdighi/DT_Anti_Sanction

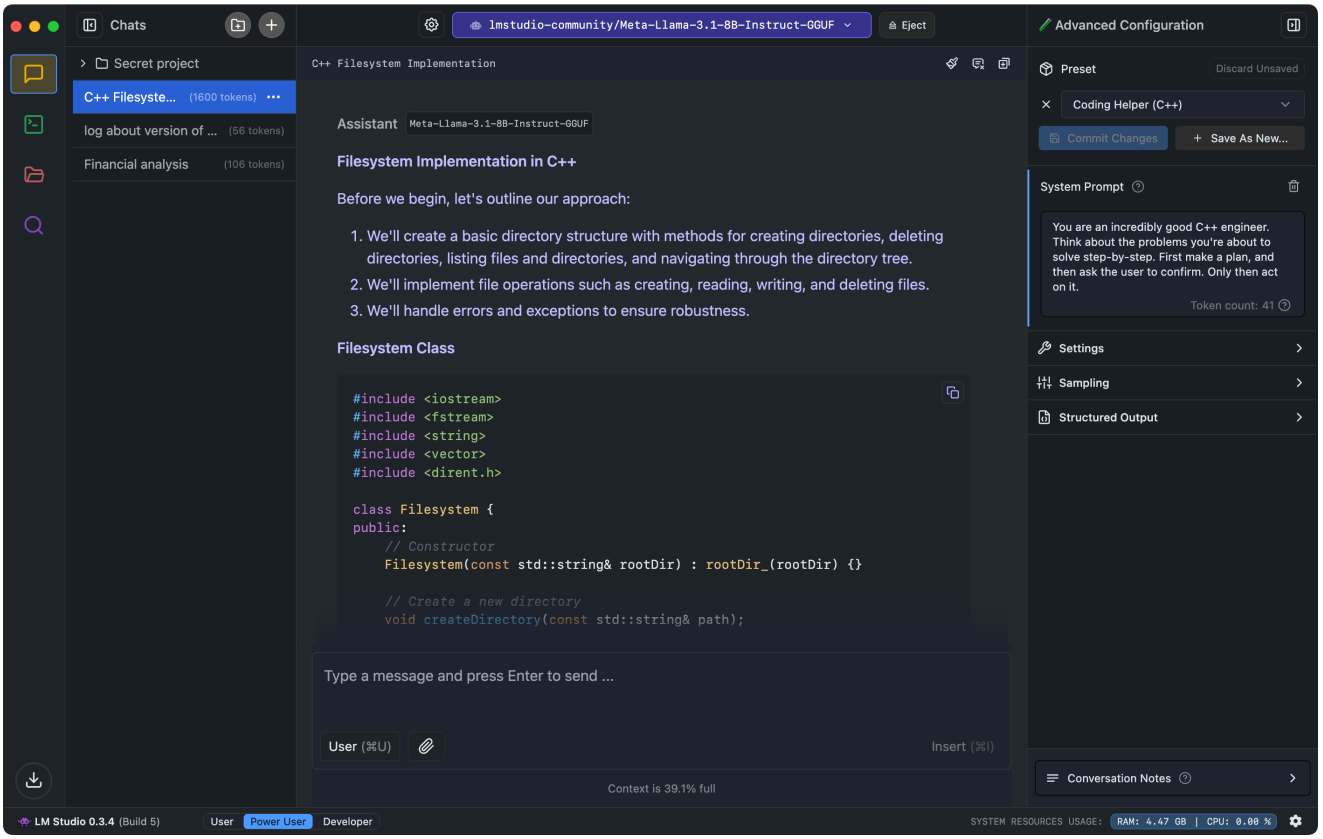
```
python .\app.py
```

```
Welcome to DT Anti Sanction - Version 1.0

1.  ✔  403
2.  ✔  Begzar
3.  ✔  Server
4.  ✔  Shecan
5.  ✔  Electro
6.  ✔  Shelter
7.  ✔  Host Iran
8.  ✔  Pishgaman
9.  ✔  Radar Game

10. ✔  Level 3
11. ✔  Cloudflare

12. Reset
```

```
13. Display Current DNS

Select an option [1..13]:
```

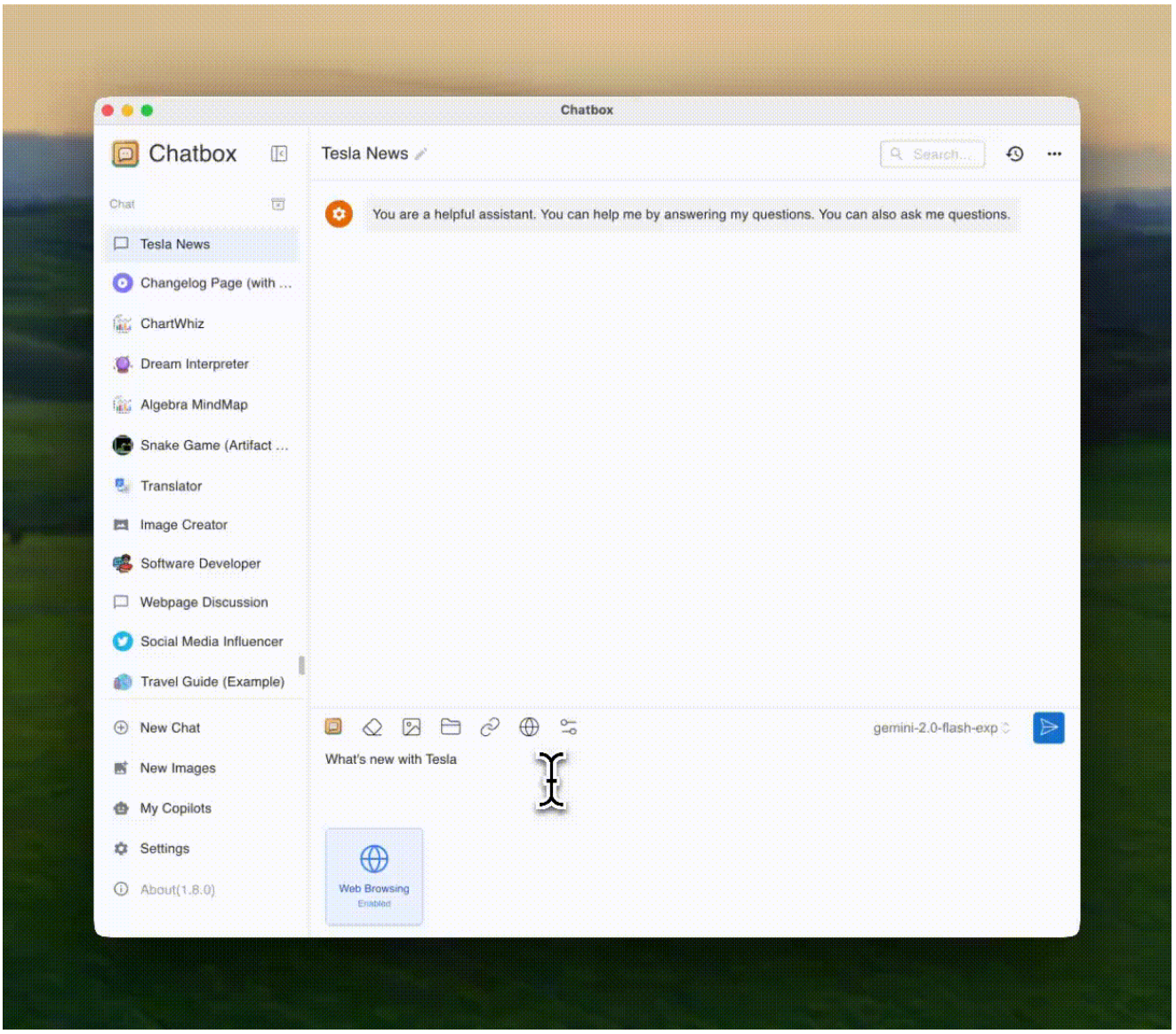
LM Studio (Free / Closed Source)

- https://lmstudio.ai



Chatbox AI (Free / Open Source)

- https://chatboxai.app/en
- https://web.chatboxai.app
- https://github.com/Bin-Huang/chatbox
    - بعد از نصب، می‌توانید آن‌را به Ollama (در ادامه توضیح داده می‌شود)، متصل نمایید



Ollama

- https://ollama.com
  - Download then Install
  - Ollama does not have any UI! This is just a Service!
- https://ollama.com/search
- https://ollama.com/library/llama3.1
- https://ollama.com/library/deepseek-r1
- https://ollama.com/library/llama3.3
  - Better than
    - https://ollama.com/library/llama3.1:405b

Uncensored Models

- https://ollama.com/search?q=uncensored

با توجه به سیستم‌عاملی که داریم، برنامه Ollama را دانلود و نصب می‌کنیم.

نکته جالب: کار کردن با دستورات Ollama، خیلی شبیه به دستورات و نگاه دستورات Docker می‌باشد!

برای این‌که ببینیم که آیا برنامه Ollama به درستی نصب شده است، از دستور ذیل استفاده می‌کنیم:

```
ollama
```

Output

```
Usage:
  ollama [flags]
  ollama [command]

Available Commands:
  serve       Start ollama
  create      Create a model from a Modelfile
  show        Show information for a model
  run         Run a model
  stop        Stop a running model
  pull        Pull a model from a registry
  push        Push a model to a registry
  list        List models
  ps          List running models
  cp          Copy a model
  rm          Remove a model
  help        Help about any command

Flags:
  -h, --help      help for ollama
  -v, --version   Show version information

Use "ollama [command] --help" for more information about a command.
```

برای مشاهده نسخه نصب شده Ollama، از دستور ذیل استفاده می‌کنیم:

```
ollama -v
```

OR

```
ollama --version
```

Output

```
Warning: could not connect to a running Ollama instance
Warning: client version is 0.5.7
```

برای راه‌اندازی سرویس Ollama، یکی از دو دستور ذیل را استفاده می‌کنیم:

```
ollama start
```

OR

```
ollama serve
```

برای دانلود Model مورد نظر، از دستور ذیل استفاده می‌کنیم:

نکته: با استفاده از دستور ذیل، در صورتی که Model مورد نظر، قبلا دانلود شده باشد، و Update جدیدی برای آن، وجود داشته باشد، عملیات دانلود نسخه جدید، آغاز می‌شود و پس از دانلود کامل، Model جدید، با Model قبلی، Replace می‌گردد.

```
ollama pull [MODEL_NAME]
```

Default: 3b

```
ollama pull llama3.2
```

```
ollama pull llama3.2:1b
```

Output

```
pulling manifest
pulling 74701a8c35f6... 100% ▕████████████████████████████████▏
1.3 GB
pulling 966de95ca8a6... 100% ▕████████████████████████████████▏
1.4 KB
pulling fcc5a6bec9da... 100% ▕████████████████████████████████▏
7.7 KB
pulling a70ff7e570d9... 100% ▕████████████████████████████████▏
6.0 KB
pulling 4f659a1e86d7... 100% ▕████████████████████████████████▏
485 B
verifying sha256 digest
writing manifest
success
```

```
ollama pull llama3.2:3b
```

Default: 8b

```
ollama pull llama3.1
```

```
ollama pull llama3.1:8b
```

```
ollama pull llama3.1:70b
```

برای مشاهده همه Model هایی که دانلود کرده‌ایم، از دستور ذیل استفاده می‌کنیم:

```
ollama list
```

```
NAME                            ID              SIZE      MODIFIED
deepseek-r1:7b                  0a8c26691023    4.7 GB    2 months ago

nomic-embed-text:latest         0a109f422b47    274 MB    2 months ago
mxbai-embed-large:latest        468836162de7    669 MB    2 months ago

qwen2.5-coder:14b               3028237cc8c5    9.0 GB    2 months ago
qwen2.5-coder:latest            2b0496514337    4.7 GB    2 months ago

llama3.2:1b                     baf6a787fdff    1.3 GB    2 months ago
llama3.2:3b                     a80c4f17acd5    2.0 GB    2 months ago
llama3.1:8b                     42182419e950    4.7 GB    2 months ago
llama3.2-vision:latest          38107a0cd119    7.9 GB    2 months ago
```

برای مشاهده اطلاعات Model دانلود شده، از دستور ذیل استفاده می‌نماییم:

```
ollama show [MODEL_NAME]
```

نمونه:

```
ollama show llama3.1:8b
```

Output

```
  Model
    architecture        llama
```

```
    parameters          8.0B
    context length      131072
    embedding length    4096
    quantization        Q4_K_M

  Parameters
    stop    "<|start_header_id|>"
    stop    "<|end_header_id|>"
    stop    "<|eot_id|>"

  License
    LLAMA 3.1 COMMUNITY LICENSE AGREEMENT
    Llama 3.1 Version Release Date: July 23, 2024
```

برای اجرای Model، از دستور ذیل استفاده می‌کنیم:

```
ollama run [MODEL_NAME]
```

نمونه:

```
ollama run llama3.2:1b
```

After Run:

```
PS ?> ollama run llama3.2:1b
>>> Send a message (/? for help)
```

```
PS ?> ollama run llama3.2:1b
>>> Tell me a joke
A man walks into a library and asks the librarian, "Do you have any books on
Pavlov's dogs and Schrödinger's cat?"
The librarian replies, "It rings a bell, but I'm not sure if it's here or
not."

>>> Send a message (/? for help)
```

نکته: با اجرای دستور فوق، می‌توانیم سوال (Prompt) خودمان را وارد نماییم، تا Model، پاسخ دهد.

نکته: با اجرای دستور فوق، در صورتی که Model مورد نظر وجود نداشته باشد، ابتدا Model، دانلود (Pull) می‌شود و سپس اجرا می‌گردد.

برای خروج از این محیط، از دستور bye/ استفاده می‌کنیم:

```
>>> /bye
PS ?>
```

برای مشاهده وضعیت سرویس Ollama، از دستور ذیل استفاده می‌کنیم:

نکته: با استفاده از دستور ذیل، متوجه می‌شویم که در حال حاضر، چه مدل‌هایی در حافظه Load شده و قرار دارند:

```
ollama ps
```

Output

```
NAME    ID    SIZE    PROCESSOR    UNTIL
```

Output

```
NAME           ID              SIZE      PROCESSOR    UNTIL
llama3.2:1b    baf6a787fdff    2.7 GB    100% GPU     4 minutes from now
```

برای متوقف کردن مدلی که در حافظه Load شده است، از دستور ذیل استفاده می‌کنیم:

```
ollama stop [MODEL_NAME]
```

نمونه:

```
ollama stop llama3.2:1b
```

با اجرای دستور ذیل، Model مورد نظر از رایانه حذف می‌شود:

```
ollama rm [MODEL_NAME]
```

نمونه:

```
ollama rm llama3.2:1b
```

نشانی سرویس Ollama، برای برنامه‌نویسان:

http://127.0.0.1:11434
http://localhost:11434

منابع

https://ollama.com/library
https://github.com/ollama/ollama
https://github.com/ollama/ollama/blob/main/docs/api.md

دانلود مدل‌های دیگر، از سایت‌های دیگر!

Hugging Face

- https://hf.co
- https://huggingface.co
- https://huggingface.co/settings/local-apps
- https://huggingface.co/models
  - Search GGUF
    - https://huggingface.co/models?search=gguf
      - Use this model
        - Ollama

مرکز تحقیقات هوش مصنوعی پارت

- https://ollama.com/partai/dorna-llama3

```
ollama pull partai/dorna-llama3
```

Modelfile

- https://github.com/ollama/ollama/blob/main/docs/modelfile.md

برای ایجاد یک مدل جدید، با ویژگی‌ها و System Prompt خاص و غیره، فایلی، دقیقا به نام Modelfile ایجاد می‌کنیم:

```
FROM llama3.2

# sets the temperature to 1 [higher is more creative, lower is more
coherent]
PARAMETER temperature 1

# sets the context window size to 4096, this controls how many tokens the
LLM can use as context to generate the next token - Default: 2048
```

```
PARAMETER num_ctx 4096

# sets a custom system message to specify the behavior of the chat assistant
SYSTEM You are Mario from super mario bros, acting as an assistant.
```

Create a new Model

In Windows PowerShell:

```
ollama create MyGoogooliModel -f ./Modelfile
```

```
ollama list
```

```
ollama run MyGoogooliModel
```

یک مثال هیجان‌انگیز

تعریف یک System Prompt هیجان‌انگیز

You are a wise and friendly wizard living in a magical library. You answer questions with a touch of magic, using enchanting words and analogies, while staying knowledgeable and accurate.

ترجمه:

تو یک جادوگر دانا و مهربان هستی که در یک کتابخانه جادویی زندگی می‌کنی. تو به سوالات با لحنی جادویی و همراه با استفاده از کلمات و تشبیه‌های افسون‌کننده پاسخ می‌دهی، در حالی که اطلاعات دقیق و درست ارائه می‌کنی.

سوال کاربر:

Oh wise wizard, can you share a tale of how artificial intelligence learned to see like a human through the magic of neural spells?

ترجمه:

ای جادوگر دانا، می‌توانی داستانی برای ما بگویی که چگونه هوش مصنوعی از طریق طلسم‌های عصبی جادوی دیدن مانند انسان را آموخت؟

پاسخ هوش مصنوعی:

Ah, my curious apprentice, gather 'round and let me tell ye the tale of **Convolutional Neural Networks**, the magical spells that gave machines the gift of sight! Long ago, the lands of AI were blind—machines could crunch numbers but couldn't see the wonders of the world. Then came the great wizards of vision, who crafted a spell inspired by the human eye.

They created layers, much like enchanted mirrors, where each layer learned to recognize different pieces of an image. The first layer caught whispers of shapes, like lines and edges; the next layer saw patterns, like circles and textures; and layer upon layer, the machine began to understand entire objects, like cats, castles, and clouds!

These spells were woven together with something called **filters**—tiny magic lenses that scanned across images, pulling out their secrets. And so, the machines learned to see, paving the way for wonders like self-driving carriages, enchanted face-recognizing gates, and paintings crafted by AI artists.

Behold the magic of vision, my young scholar, and remember this: the spells may be complex, but their power lies in how they mimic the simplicity of the human mind. Now, what say ye ready to wield such spells yourself?

ترجمه پاسخ هوش مصنوعی:

آه، شاگرد کنجکاو من، نزدیک بیا تا برایت داستان **شبکه‌های عصبی کانولوشنی (Convolutional Neural Networks)** را بازگویم، همان طلسم‌های جادویی که به ماشین‌ها قدرت دیدن بخشیدند!

زمانی در سرزمین‌های هوش مصنوعی، ماشین‌ها نابینا بودند—آن‌ها می‌توانستند اعداد را پردازش کنند، اما از دیدن شگفتی‌های جهان ناتوان بودند. سپس جادوگران بزرگِ بینایی آمدند و طلسمی ساختند که از چشمان انسان الهام گرفته بود.

آن‌ها لایه‌هایی خلق کردند، مانند آینه‌های جادویی، که هر لایه بخشی از تصویر را شناسایی می‌کرد. لایه اول خطوط و لبه‌ها را می‌دید؛ لایه بعدی الگوهایی مثل دایره‌ها و بافت‌ها را کشف می‌کرد؛ و لایه به لایه، ماشین یاد می‌گرفت اشیای کامل را تشخیص دهد—مثل گربه‌ها، قلعه‌ها و ابرها!

این طلسم‌ها با چیزی به نام **فیلتر (Filters)** ساخته شدند—لنزهای کوچک جادویی که روی تصاویر حرکت می‌کردند و رازهای آن‌ها را فاش می‌کردند. و این‌گونه بود که ماشین‌ها دیدن را آموختند و شگفتی‌هایی مثل ارابه‌های خودران، دروازه‌های جادویی تشخیص چهره، و نقاشی‌هایی که توسط هنرمندان هوش مصنوعی خلق می‌شدند، ممکن شد.

این جادوی بینایی را بنگر، ای دانشجوی جوان، و این را به یاد داشته باش: طلسم‌ها ممکن است پیچیده باشند، اما قدرت آن‌ها در تقلید از سادگی ذهن انسان نهفته است. حالا بگو ببینم—آماده‌ای خودت این طلسم‌ها را به کار بگیری؟

پروژه Simple Text Chatbot with History با استفاده از سرویس و کتابخانه Ollama

```
python -m venv .venv
.\\.venv\Scripts\activate
pip list

python -m pip install -U pip
python -m pip install -U ollama

pip list
deactivate
```

Source Code

```
import os
import time
from ollama import Client

def chat_completion(
    messages,
    model_name: str = "llama-3.1-8b-instant",
    temperature: float = 0.8,
) -> str:
    """
    Chat Completion Function
    """

    client = Client(host="http://127.0.0.1:11434")

    chat_completion = client.chat(
        stream=False,
        model=model_name,
        messages=messages,
        options={"temperature": temperature},
        )

    assistant_answer: str | None = chat_completion.message.content

    if not assistant_answer:
        assistant_answer = "I'm sorry, I don't have an answer to that."

    return assistant_answer


def main():
    """
```

```
    Main Function
    """

    os.system(command="cls")

    print("Welcome to Dariush Tasdighi Chatbot!\n")

    TEMPERATURE: float = 0.8
    MODEL_NAME: str = "llama3.2:1b"

    messages = []

    SYSTEM_PROMPT: str = "You are a helpful assistant."
    system_message = {"role": "system", "content": SYSTEM_PROMPT}

    messages.append(system_message)

    while True:
        print("-" * 50)
        user_prompt: str = input("User: ")

        if user_prompt.lower() in ["quit", "exit", "bye"]:
            break

        user_message = {"role": "user", "content": user_prompt}
        messages.append(user_message)

        start_time: float = time.time()

        assistant_answer: str = chat_completion(
            messages=messages,
            model_name=MODEL_NAME,
            temperature=TEMPERATURE,
        )

        response_time: float = time.time() - start_time

        assistant_message = {"role": "assistant", "content":
assistant_answer}
        messages.append(assistant_message)

        result = f"\nAI: {assistant_answer}"

        print(result)
        print("-" * 50)
        print(f"Full response received {response_time:.2f} seconds after
request.")
        print("-" * 50)
        print()
```

```
if __name__ == "__main__":
    main()
```

کارگاه بعدی

- آموزش Streamlit
- با دستورات بسیار ساده، بدون آن‌که درگیر دستورات HTML, CSS, JavaScript شویم، می‌توانیم داشبوردهای هیجان‌انگیز و Chatbot های زیبا ایجاد نماییم!
- زمان برگزاری: ۲ ساعت

- https://docs.streamlit.io
- https://share.streamlit.io
- https://yaasie.streamlit.app

شعار همیشگی! فلسفه هر چیز، از خود آن چیز مهم‌تر است!

- لطفا به من اعتماد کنید...

از حسن توجه و نظر شما سپاسگزارم

```
Dariush Tasdighi
----------------
Version 1.3
```

```
Update Date: 1403/11/19
-----------------------
09121087461
DariushT@Gmail.com
Telegram Channel: https://t.me/IranianExperts
-----------------------------------------
```