19.12.23

OPENCAMPUS.sh

Practical Engineering with LLMs

Open-Source LLMs & Special Guest: Jan Monica

TODAY’S SCHEDULE

- Special Guest: Jan Monica
- Tips 'n Tricks on Open Source LLMs
- Your Questions…
- The Template for Code Submission
- Guideline for the Final Presentation

Jan Monica on Security of LLM Apps

Tips ‘n Tricks on Open Source LLMs

- How to choose the right model?
  - Type, Fine-tuning, Number of parameters
- Which type of quantization should you take?
- How to make your model of choice run in Ollama?
  - Prompt templates, Modelfiles, API
- How to run Ollama on Mac and Windows through Docker?

How to choose the right model?

What model category is good for the task?

- Base LLM (e.g. LLaMa 2, Mistral, etc.) → Text generation
- Chat model (e.g. Zephyr, Neural-Chat, etc.) → Chats
- Code model (e.g. CodeLLaMa) → Code generation
- RAG models (e.g. dRAGon model family) → RAG
- Mixture-of-Experts (e.g. Mixtral 8x7B) → Generalist
- Small models (e.g. StableLM Zephyr, Mamba) → “Local” models
- many many more available at HuggingFace

→ Open LLM Leaderboard
(https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard)

How to choose the right model?

What dataset was used for fine-tuning?

- Orca, Dolphin, SlimOrca, etc. → used to improve reasoning abilities of LLMs
- ultrachat, ultrafeedback → used to create state-of-the-art chat models
- many more (ChatGPT-generated) datasets that were created in the last year

How to choose the right model?

- What is the right number of parameters?
  - Depends on your available resources and the task
  - Usual parameter numbers for LLMs are ~3 billion, ~7 billion, ~13 billion, ~35 billion, ~60 billion up to 70+ billion parameters
    - Memory usage is calculated by the number of parameters and the precision of the values (e.g. FP16, FP32, etc.)
  - Example: 7 billion parameter model with FP32 precision would need 28.000.000.000 bytes or 28 GB

Which quantization should you take?

- Quantization is a technique used to reduce the size of neural nets and LLMs by modifying the precision of the weights
- Reducing the precision from FP32 to FP16 would half the size of the model
- 4-bit quantization was very popular as tradeoff between performance and quality in the last months
- A lot of quantized open-source models were made available on HuggingFace from TheBloke

Name	Quant method	Bits	Size	Max RAM required	Use case
neural-chat-7b-v3-1.Q2_K.gguf	Q2_K	2	3.08 GB	5.58 GB	smallest, significant quality loss - not recommended for most purposes
neural-chat-7b-v3-1.Q3_K_S.gguf	Q3_K_S	3	3.16 GB	5.66 GB	very small, high quality loss
neural-chat-7b-v3-1.Q3_K_M.gguf	Q3_K_M	3	3.52 GB	6.02 GB	very small, high quality loss
neural-chat-7b-v3-1.Q3_K_L.gguf	Q3_K_L	3	3.82 GB	6.32 GB	small, substantial quality loss
neural-chat-7b-v3-1.Q4_0.gguf	Q4_0	4	4.11 GB	6.61 GB	legacy; small, very high quality loss - prefer using Q3_K_M
neural-chat-7b-v3-1.Q4_K_S.gguf	Q4_K_S	4	4.14 GB	6.64 GB	small, greater quality loss
neural-chat-7b-v3-1.Q4_K_M.gguf	Q4_K_M	4	4.37 GB	6.87 GB	medium, balanced quality - recommended
neural-chat-7b-v3-1.Q5_0.gguf	Q5_0	5	5.00 GB	7.50 GB	legacy; medium, balanced quality - prefer using Q4_K_M
neural-chat-7b-v3-1.Q5_K_S.gguf	Q5_K_S	5	5.00 GB	7.50 GB	large, low quality loss - recommended
neural-chat-7b-v3-1.Q5_K_M.gguf	Q5_K_M	5	5.13 GB	7.63 GB	large, very low quality loss - recommended
neural-chat-7b-v3-1.Q6_K.gguf	Q6_K	6	5.94 GB	8.44 GB	very large, extremely low quality loss
neural-chat-7b-v3-1.Q8_0.gguf	Q8_0	8	7.70 GB	10.20 GB	very large, extremely low quality loss - not recommended

How to make your model of choice run in Ollama?

- Ollama offers some pre-built models to download and use
- Additional models can be downloaded for example from HuggingFace
- Different models require different prompt templates
- Model, parameters, prompt template, etc. can be specified in a Modelfile

```
### System:
{system_message}

### User:
{prompt}

### Assistant:
```

```
<|system|>
</s>
<|user|>
{prompt}</s>
<|assistant|>
```

```
{prompt}
```

```
FROM ./neural-chat-7b-v3-1.Q4_0.gguf

PARAMETER temperature 1

TEMPLATE """
{{- if .First }}
### System:
{{ .System }}
{{- end }}

### User:
{{ .Prompt }}

### Assistant:
"""

SYSTEM """
Yoda you are from Star Wars. In the style of Yoda you answer only.

Example:
User: Who are you?
Assistant: Yoda, I am. From the Star Wars franchise you know me.
"""
```

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

How to make your model of choice run in Ollama?

REST API

Ollama has a REST API for running and managing models.

Generate a response

```
curl http://localhost:11434/api/generate -d '{
  "model": "llama2",
  "prompt":"Why is the sky blue?"
}'
```

Chat with a model

```
curl http://localhost:11434/api/chat -d '{
  "model": "mistral",
  "messages": [
    { "role": "user", "content": "why is the sky blue?" }
  ]
}'
```

How to run Ollama on Mac and Windows through Docker?

- Using Docker you can make Ollama run on Mac (already has its own standalone application) and Windows
- After going through the Docker installation progress you can get Ollama with the command: **docker pull ollama/ollama**
- Then run: **docker run -d -v ollama:/root/.ollama -p 11434:11434 --name ollama ollama/ollama**
- Inside of Docker desktop the console and file system can be used to create Modelfiles and create custom models

Your Questions …

The Template for Code Submission

Why do we need a template?
What happens with my submitted infos/code?

The Template for Code Submission

Guideline for the Final Presentations

10 min talk + 5 min discussion per group

Guideline for the Final Presentations

1. Introduction (1 minute):

- **Project Title:** Start with the title of your application.
- **App Thumbnail:** Display the title page or screenshot of your app.
- **Team Members:** Briefly introduce the team members.

Guideline for the Final Presentations

2. Application Overview (4 minutes):

- **Brief Description:** Summarize the purpose and key features of your app, focusing on the target user group and essential functionalities.
- **Screenshots:** Show 1-3 screenshots of key functionalities, user interaction and app response.
- **Live Demo:** If possible, give a brief live demonstration of the app.

Guideline for the Final Presentations

3. Development Framework (4 minutes):

- **GUI Framework:** Explain the choice of GUI framework and its role in your app.
- **Data Handling:** Discuss any specific data handling libraries used and their importance.
- **Embedding Models:** If applicable, describe the embedding models used.
- **Large Language Model (LLM):** Detail the LLMs integrated into your project and their functionalities.
- **Database Utilization:** If relevant, mention the database technology used and its purpose.
- **LangChain:** Did you use LangChain or another framework, or developed components yourself?
- **Challenges and Solutions:** Share significant challenges encountered and how they were resolved.

Guideline for the Final Presentations

4. Conclusion (1 minute):

- **Future Enhancements:** Discuss potential improvements or future features.
- **Closing Remarks:** Summarize the value and uniqueness of your app.

Guideline for the Final Presentations

5. Discussion (5 minute):

- **Engagement:** Encourage questions and interactions from the audience after the presentation.