Mixtral by Mistral

Primarily based on a course by DeepLearning.AI

Yogesh Haribhau Kulkarni (PhD)

Published in Technology Hits

4 min read · Just now

Photo by Duy Pham on Unsplash

A startup from Paris, France, has been making waves with their innovative approach. Mistral, a pioneering company, has introduced Mixtral, a cutting-edge model that deviates from the traditional architectures employed by other Large Language Models (LLMs) like GPT from OpenAI and Gemini from Google. At the core of Mixtral lies the concept of Mixture of Experts (MoE), a unique blend of specialized neural networks that promises to reshape the future of Generative AI.

This story goes into more details about Mistral, which is based on a DeepLearning.ai course by Sophia Yang and Andrew Ng.

TL;DR

The course covers:

- Use and prompt Mistral models via API calls for tasks
- Do Native function calling
- Build a basic RAG system from scratch
- Build a chat interface on your doc

What is a Mixture of Experts (MoE)?

In the context of transformer models, a MoE comprises two essential components:

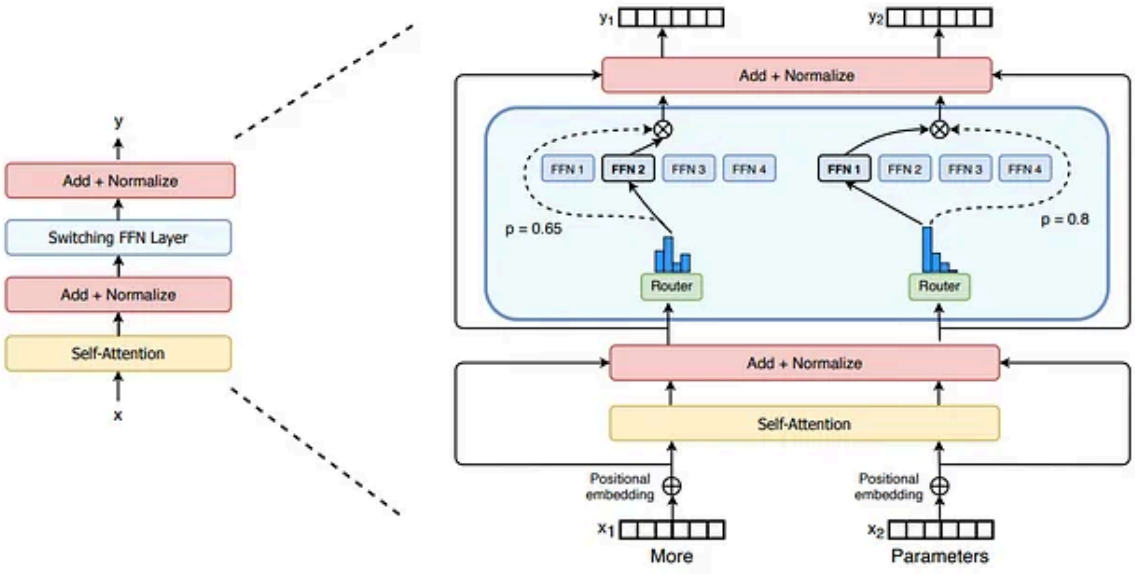

Figure 2: Illustration of a Switch Transformer encoder block. We replace the dense feed forward network (FFN) layer present in the Transformer with a sparse Switch FFN layer (light blue). The layer operates independently on the tokens in the sequence. We diagram two tokens (x_1 = "More" and x_2 = "Parameters" below) being routed (solid lines) across four FFN experts, where the router independently routes each token. The switch FFN layer returns the output of the selected FFN multiplied by the router gate value (dotted-line).

MoE layer from the [Switch Transformers paper](https://arxiv.org/abs/2101.03961)

- **Sparse MoE Layers:** Unlike dense feed-forward network (FFN) layers, MoE layers incorporate a predetermined number of "experts," each of which is a neural

network in its own right. Typically implemented as FFNs, these experts can also take on more complex forms, including nested MoEs, giving rise to hierarchical architectures.

- **Gate Network or Router:** This crucial component determines which tokens are routed to which expert. For instance, in the image below, the token “More” is sent to the second expert, while the token “Parameters” is directed to the first network. Notably, a single token can be sent to multiple experts, adding flexibility to the routing process. The router itself comprises learned parameters that are pre-trained concurrently with the rest of the network, making the routing decision a critical aspect of the MoE design.

Model Selection

To leverage the power of Mixtral in your code, you can obtain an API key from `console.mistral.ai` and set it as the `MISTRAL_API_KEY` environment variable. The `mistralai` Python package provides a convenient interface for interacting with Mistral’s models:

```
!pip install mistralai

from mistralai.client import MistralClient
from mistralai.models.chat_completion import ChatMessage

def mistral(user_message,
            model="mistral-small-latest",
            is_json=False):
    client = MistralClient(api_key=os.getenv("MISTRAL_API_KEY"))
    messages = [ChatMessage(role="user", content=user_message)]

    if is_json:
        chat_response = client.chat(
            model=model,
            messages=messages,
            response_format={"type": "json_object"})
    else:
        chat_response = client.chat(
            model=model,
            messages=messages)

    return chat_response.choices[0].message.content

prompt = "   " # ToDO
```

```
response = mistral(prompt) # (prompt, is_json=True) for structured output
print(response)
```

Mistral offers a diverse range of models, both open-source and closed-source, to cater to various use cases:

Open-source Models:

- Mistral 7B (‘*open-mistral-7b*’)
- Mixtral 8x7B (‘*open-mixtral-8x7b’*)

Closed-source Models:

- Mistral Small (‘*mistral-small-latest’*): Suitable for simple tasks, fast inference, and lower cost.
- Mistral Medium (‘*mistral-medium-latest*’): Ideal for intermediate tasks such as language transformation.
- Mistral Large (‘*mistral-large-latest*’): Designed for complex tasks that require advanced reasoning.
- Mistral Embedding: Tailored for Retrieval-Augmented Generation (RAG), similarity search, and related applications.

For local runs, you can leverage the transformers library from Hugging Face, llama.cpp, ollama, or LM Studio.

Function Calling

One of the standout features of Mistral models is their ability to integrate with external tools through function calling. This process involves the following steps:

- **User-defined Tools:** Users define functions or external APIs that they wish to incorporate.
- **Function Argument Generation:** The Mistral model generates appropriate function arguments based on the query and function signature.
- **Function Execution:** The user executes the defined function to obtain the desired response.

- **Final Answer Generation**: The Mistral model generates the final answer by leveraging the function's output.

Retrieval-Augmented Generation (RAG)

Mistral models can be leveraged to build a Retrieval-Augmented Generation (RAG) system from the ground up. This approach combines the power of retrieval systems with language models, allowing for more informed and context-aware responses. Here's a high-level overview of the process:

1. **Obtaining Text Embeddings**: Use Mistral's embedding model to generate embeddings for your text corpus:

```
import os
from mistralai.client import MistralClient
def get_text_embedding(txt):
 client = MistralClient(api_key=api_key, endpoint=dlai_endpoint)
 embeddings_batch_response = client.embeddings(model="mistral-embed", input=txt
 return embeddings_batch_response.data[0].embedding
```

2. **Similarity Search**: Search for text chunks similar to the query by comparing their embeddings.
3. **Prompt Construction**: Create a prompt by combining the retrieved chunks and the original query:

```
prompt = f"""
Context information is below.
 - - - - - - - - - - -
{retrieved_chunk}
 - - - - - - - - - - -
Given the context information and not prior knowledge, answer the query.
Query: {question}
Answer:
```

4. **Final Response Generation**: Feed the constructed prompt to a Mistral model to obtain the final, context-aware response.

Conclusion

Mistral's Mixtral models, powered by the innovative Mixture of Experts architecture, are ushering in a new era of Generative AI. With their unique approach to token routing, specialized expert networks, and seamless integration with external tools, Mixtral models offer a powerful and flexible solution for a wide range of tasks. Whether you're seeking to build a cutting-edge RAG system, leverage function calling, or simply harness the capabilities of state-of-the-art language models, Mistral's offerings are poised to revolutionize the way we interact with and leverage Generative AI.

References

Getting Started With Mistral

Learn to use Mistral AI's LLMs effectively. Leverage features like JSON mode, RAG, function calling, and more.

www.deeplearning.ai

Mixtral of experts

A high quality Sparse Mixture-of-Experts.

mistral.ai

Mixture of Experts Explained

We're on a journey to advance and democratize artificial intelligence through open source and open science.

huggingface.co

Mixtral of Experts

We introduce Mixtral 8×7B, a Sparse Mixture of Experts (SMoE) language model. Mixtral has the same architecture as...

arxiv.org

Click pic below or visit LinkedIn to know more about the author

Artificial Intelligence | Generative Ai Tools | Deep Learning | Large Language Models | Summary

Edit profile

Written by Yogesh Haribhau Kulkarni (PhD)

1.4K Followers · Editor for Technology Hits

PhD in Geometric Modeling | Google Developer Expert (Machine Learning) | Top Writer 3x (Medium) | More at https://www.linkedin.com/in/yogeshkulkarni/

Recommended from Medium

Aldric Chen in Change Your Mind Change Your Life

I [Just] Conducted an Exit Interview for an Unhappy Gen Z. She Confused Me Big Time.

Am I out of touch? Maybe.

· 5 min read · Apr 8, 2024

7.7K 277

Jake Page

The guide to Git I never had.

Doctors have stethoscopes.

13 min read · Apr 11, 2024

2.6K 22

Lists

AI Regulation

6 stories · 421 saves

Natural Language Processing

1390 stories · 890 saves

ChatGPT

21 stories · 588 saves

Generative AI Recommended Reading

52 stories · 949 saves

Somnath Singh in Level Up Coding

The Era of High-Paying Tech Jobs is Over

The Death of Tech Jobs.

✨ · 14 min read · Apr 1, 2024

8.4K 242

Hazel Paradise

How I Create Passive Income With No Money

many ways to start a passive income today

5 min read · Mar 27, 2024

7.1K 168

Paddy Murphy in ILLUMINATION-Curated

Bitcoin Founder, Satoshi Nakamoto's Identity might have been Revealed

It looks like a Redditor has figured out who is behind the mysterious name.

· 5 min read · Apr 10, 2024

1.8K 26

Dylan Cooper in Stackademic

Mojo, 90,000 Times Faster Than Python, Finally Open Sourced!

On March 29, 2024, Modular Inc. announced the open sourcing of the core components of Mojo.

· 10 min read · Apr 8, 2024

3.4K 23