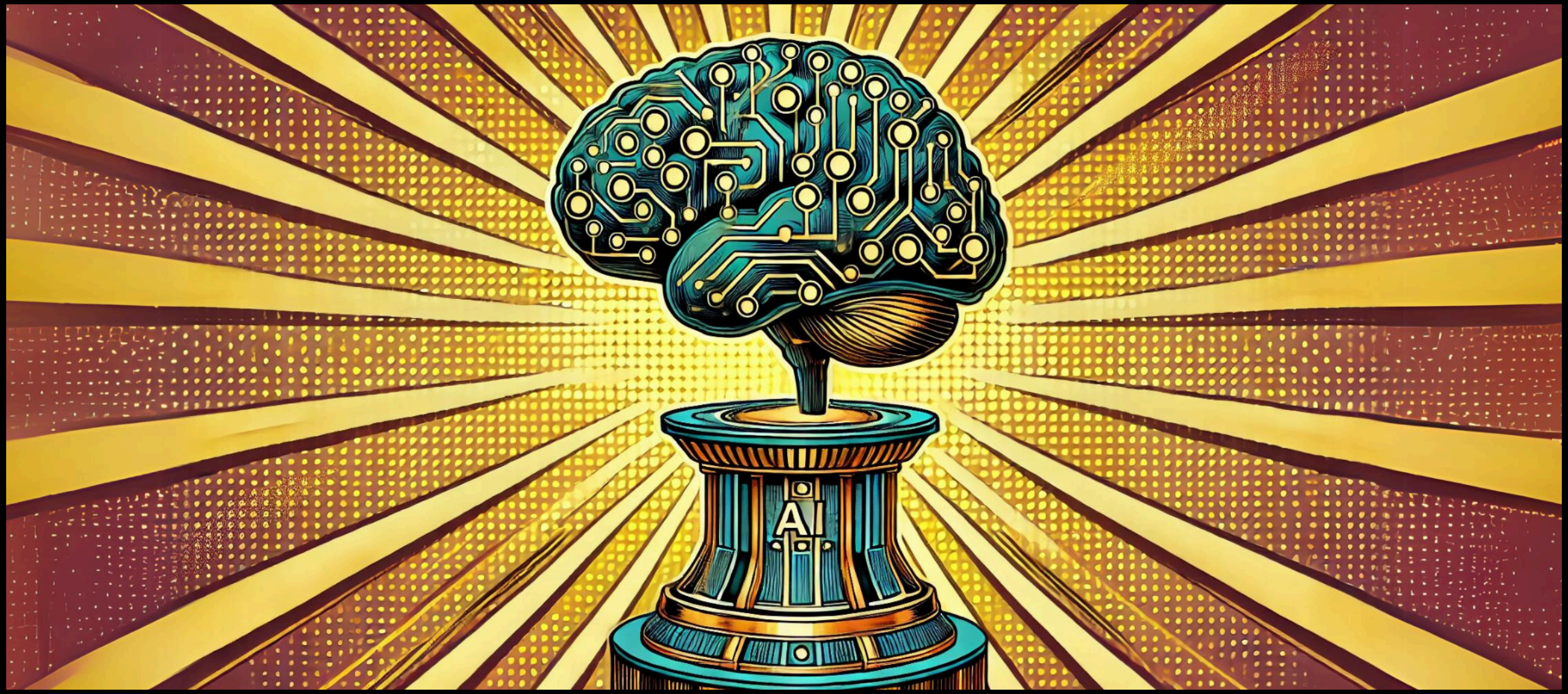
LLM Engineering

MASTER AI & LARGE LANGUAGE MODELS



WEEK 5 DAY 4

Today it all comes together

What you can now do

- Generate text and code with Frontier Models including AI Assistants with Tools, and with open-source models with HuggingFace transformers
- Confidently choose the right LLM for your project, backed by metrics
- Create and populate a Vector Datastore with the contents of a Knowledge Base

---

In a matter of minutes you'll be able to

- Create a Conversation Chain in LangChain for a chat conversation with retrieval
- Ask questions and receive answers demonstrating expert knowledge
- Build a Knowledge Worker assistant with chat UI

Key Abstractions in LangChain

LLM

Retriever

Memory

FINISHING TOUCHES

A Conversation Chain with RAG and Memory

..In 4 lines of code.

```
# create a new Chat with OpenAI
llm = ChatOpenAI(temperature=0.7)

# set up the conversation memory for the chat
memory = ConversationBufferMemory(memory_key='chat_history', return_messages=True)

# create a retriever from the Chroma datastore
retriever = vectorstore.as_retriever()

# putting it together
conversation_chain = ConversationalRetrievalChain.from_llm(llm=llm, retriever=retriever, memory=memory)
```

Now, to try a conversation!

PROGRESS

You have leveled up again!

What you can now do

- Generate text and code with Frontier Models including AI Assistants with Tools, and with open-source models with HuggingFace transformers
- Confidently choose the right LLM for your project, backed by metrics
- Create a RAG Knowledge Worker using LangChain and Chroma

By end of next session you will also be able to:

- Be familiar with LangChain's declarative language LCEL
- Understand how LangChain works behind the scenes
- Debug and fix a common issue with RAG