Data Science & AI for Economists

Understanding Large Language Models: From Theory to Practice

Zhaopeng Qu
Business School, Nanjing University
October 23 2025

Roadmap

1. Introduction to LLMs(very brief)
2. Practical Guidelines for Applications of LLMs in China
3. AI tools for Coding in Practice: GitHub Copilot in VS Code

Understanding LLMs

The Fundamentals of LLMs

- Most AI tools today we are discussing are **Generative AI Models**,or **Large Language Models** for short, which are one type of AI systems that generate human-like text.
- Generally speaking, LLMs **predict human-like text** based on previous input,just like a Chatbot(聊天机器人).

The Fundamentals of LLMs

- Actually, you can thinks of LLMs models as a **regression machine to predict the outcome** we learned in the econometrics course.
- Review our regression model in the econometrics course:

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \cdots + \beta_n X_{ni} + \epsilon_i \quad (1)$$

- Y_i is the dependent variable like test score, $X_{1i}, X_{2i}, \ldots, X_{ni}$ are the independent variables all factors that affect the test score like class size, hours of study, family income etc. and ϵ_i is the error term.
- Suppose we have a dataset with n observations, we could obtain all values of the parameters $\beta_0, \beta_1, \beta_2, \ldots, \beta_n$ by minimizing the sum of squared errors.
- Then we could write down the estimated model as follows:

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_{1i} + \hat{\beta}_2 X_{2i} + \cdots + \hat{\beta}_n X_{ni} \quad (2)$$

- In other words, if we have a new observation $X_{1j}, X_{2j}, \ldots, X_{nj}$, we can use the estimated model to predict the outcome $\hat{Y}_j$.

The Fundamentals of LLMs

- Now let's turn the linear regression model into a categorical dependent variable model.
  - the Y here is the categorical dependent variable like A/B/C/D or pass or fail etc.
- Then we could use the multinomial logit model or other statistical models to estimate the model and obtain all values of the parameters as linear regression.
  - the **training process** of the machine learning models.
- Then we could use the test data to evaluate the performance of the models and choose the best model or the best parameters
  - the **testing process** of the machine learning models.
- In the end, we could use the best model to predict the outcome for a new observation.
  - given the new observation $X_{1j}, X_{2j}, \ldots, X_{nj}$ students characteristics like class size, hours of study, family income etc., we can calculate the probability of each outcome like A/B/C/D or pass or fail etc. for the new student using the best model.

The Fundamentals of LLMs

- Now let's turn the machine learning model into a LLM context.
  - the data for training the model and testing the model is from all the text data in the internet like books, articles, websites, all texts AI companies collected.
  - the estimate method is complex neural networks(like transformer model) which related with huge amount of parameters.
- Input to the LLMs is as the **new observation** $X_{1j}, X_{2j}, \ldots, X_{nj}$, students characteristics like class size, hours of study, family income etc in our example but in text form.
- Output from the LLMs is as the **predicted outcome** $\hat{Y}_j$, the probability of each outcome like A/B/C/D or pass or fail etc. for the new student using the best model in text form.

> **LLMs are just a prediction model of texts based on the input text, they lack the ability to understand the meaning of the text.**

Tokenization

- **Tokenization**: breaking down text into smaller units called **tokens (词元)** — the fundamental unit LLMs process.

Example: Next-Token Prediction

Input: "中国经济增长面临"

Tokenized: ["中国", "经济", "增长", "面", "临"]

Predicted outputs with probabilities:

- "下行压力"(45%)
- "挑战"(30%)
- "机遇"(15%)
- "不确定性"(10%)

- In the end, the predicted outcome in the output is "下行压力"。The input and output are all in the form of tokens to form a sentence.

> "中国经济增长面临下行压力"

Training and Usage Costs of LLMs

- Since LLMs are just a text prediction model,it is similar to other predictive models based on numerical data, LLM accuracy depends on:
  - **Training data quality and quantity**
  - **Model complexity** (measured by number of **parameters**)
  - **Input data quality and quantity**
- For example, GPT-4 utilized 45TB of data, over 300 trillion tokens, and involved 175 billion parameters.
- Such training typically requires **months** and costs **tens of millions of dollars**.
  - It does not update frequently.
- Therefore, how accurate the LLMs is depends largely on your input.

Extended Thinking and Research Mode

- Standard LLMs give instant answers based on training data. But some tasks need deeper reasoning.

1. Extended Thinking Mode

- LLM "thinks" longer before responding
- Step by step reasoning process (like inner monologue)
- Takes minutes instead of seconds for complex tasks

Examples:

- OpenAI O1, O3 models
- Claude Extended mode
- DeepSeek R1

2. Deep Research Mode

- Combines extended thinking + multiple searches
- Takes 10+ minutes for comprehensive analysis
- Generates cited research reports
- Useful for literature reviews, policy analysis

Use when:

- Complex economic problems
- Mathematical proofs
- Multi-step causal reasoning
- Comprehensive research synthesis

Trade-off: More tokens (higher cost) vs. Higher accuracy

Context Window

- Your input to the LLMs is not unlimited, you need to be careful about what you put in it.
- **Context Window**: The sequence of all tokens in the current conversation.
  - Like your working memory in your brain(or your computer), the context window is limited in size, you can only remember a limited amount of information at a time.
- Both the user and the model contribute tokens sequentially into this context window.

> "中国经济增长面临下行压力"

- If the context window is full, you have to open a `new chat` to start a new conversation.
  - then the model will forget the oldest tokens(information), then the performance of the prediction will degrade.
- Practical Note: **Start new chat when switching topics!**

Context Management Best Practices

Treat Context as a Precious Resource

DO:

✓ Start **new chat** for new topics

✓ Keep prompts focused and concise

✓ Remove irrelevant uploaded files

✓ Use clear, structured messages

✓ Break long tasks into steps

DON'T:

✗ Dump entire documents unnecessarily

✗ Keep rambling conversations going

✗ Mix unrelated topics in one chat

✗ Repeat information already in context

✗ Use vague references ("it", "that")

The Knowledge Cutoff Problem

- The training data of the LLMs is from all the text data in the internet like books, articles, websites, all texts AI companies collected.
  - This large knowledge base is very **knowledgeable** and **language-based**.
  - Processing diverse **text**-based tasks,like writing, summarizing, translating, even coding and math problems solving.
- However, the training typically requires **months** and costs **tens of millions of dollars**. Then the LLMs models are not updated frequently.
- **Knowledge cutoff**: Out of date!
- **Solution**: Give LLMs tools!
  - like internet search, RAG, MCP and other tools.
- **Solution**: Give LLMs tools!

Extending LLM Capabilities

Using Tools for LLMs

The Idea: LLM emits special tokens(commands) External system executes $\Rightarrow$ Results fed back

Major Tool Categories:

1. **Internet Search**
   - Fetch current information
   - Find recent papers
2. **Code Execution**
   - Run Python/R code
   - Perform calculations
1. **File Access**
   - Read uploaded documents
   - Query databases
2. **Model Context Protocol**
   - Access external tools and data sources
   - like FRED, World Bank, etc. or custom tools.

RAG: Retrieval-Augmented Generation

What is RAG?

Definition: **Fetch relevant information first, then generate response**

Traditional LLM:

```
Your question
      ⇓
LLM (static knowledge)
      ⇓
Answer
```

- **Limited to training data**

RAG-enabled LLM:

```
Your question
      ⇓
*Search and Retrieve relevant docs**
      ⇓
LLM + Retrieved context
      ⇓
Answer (grounded in sources)
```

- **Can access external knowledge**

How RAG Works: Step by Step

The RAG Pipeline

Step 1: Indexing (Done beforehand)

```
Your documents ⇒ Split into chunks ⇒
Create embeddings ⇒ Store in vector database
```

Step 2: Retrieval (When you ask a question)

```
Your question ⇒ Create embedding ⇒
Search vector DB ⇒ Retrieve top-K similar chunks
```

Step 3: Generation

```
Original question + Retrieved chunks ⇒
Insert into context window ⇒
LLM generates answer
```

Key Advantage: **Answers are grounded in actual documents, with citations!**

RAG for Economists: Research Paper Analysis

Setup: Upload 50 economics papers on minimum wage

Your question:

- "What identification strategies do papers use to study minimum wage effects?"

RAG Process:

1. Search papers for "identification", "causal", "minimum wage"
2. Retrieve top 10 relevant sections
3. Feed to LLM with question
4. Generate synthesis

LLM Response (with RAG):

- "Based on the papers in your collection..."

1. Difference-in-Differences (Card & Krueger 1994, Dube et al. 2010)
    - Exploit state-level variation
    - Border discontinuity designs
2. Event Studies (Clemens & Wither 2019)
    - Staggered timing
3. RDD (Cengiz et al. 2019)
    - Bunching around minimum

RAG Applications in Data Science

Practical Use Cases

1. Dataset Documentation

- Upload codebooks, readme files
- Ask: "How is income measured?"
- Get precise answer from docs

2. Code Documentation

- Upload project code + docs
- Some special syntax based on the programming language will be needed like Stata/R/Python

Tool Use: Internet Search

- Keeping LLMs Current: How it works?

1. Input the question to the LLM.
2. LLM detects need for current info
3. LLM emits "search" signal
4. External system performs web search
5. Extracts text from top results
6. Inserts into context window
7. LLM generates answer with those newly retrieved information, former question and other information in the context window.

- Knowledge Cutoff Problem is partially solved by this method.
  - If the search is not successful, then no difference from the traditional LLM.
  - if the search is successful, however, not all information searched are use to retrieved but only limited number of chunks are used.

Tool Use: Access File System

Capability: LLM can read, analyze, and manipulate files in your workspace.

What It Can Do:

- Read datasets (.csv, .dta, .xlsx)
- Analyze code files
- Parse documentation
- Search through project files
- Extract information from PDFs

Example Workflow:

```
You: "Analyze mydata.csv"
      ⇓
LLM reads file
      ⇓
"I see 5 variables, 1000 obs.
 Missing data in income (12%).
 Should I clean and summarize?"
```

No manual upload needed!

MCP: Model Context Protocol

MCP: Model Context Protocol: A standardized protocol for LLMs to access external tools and data sources

Without MCP:

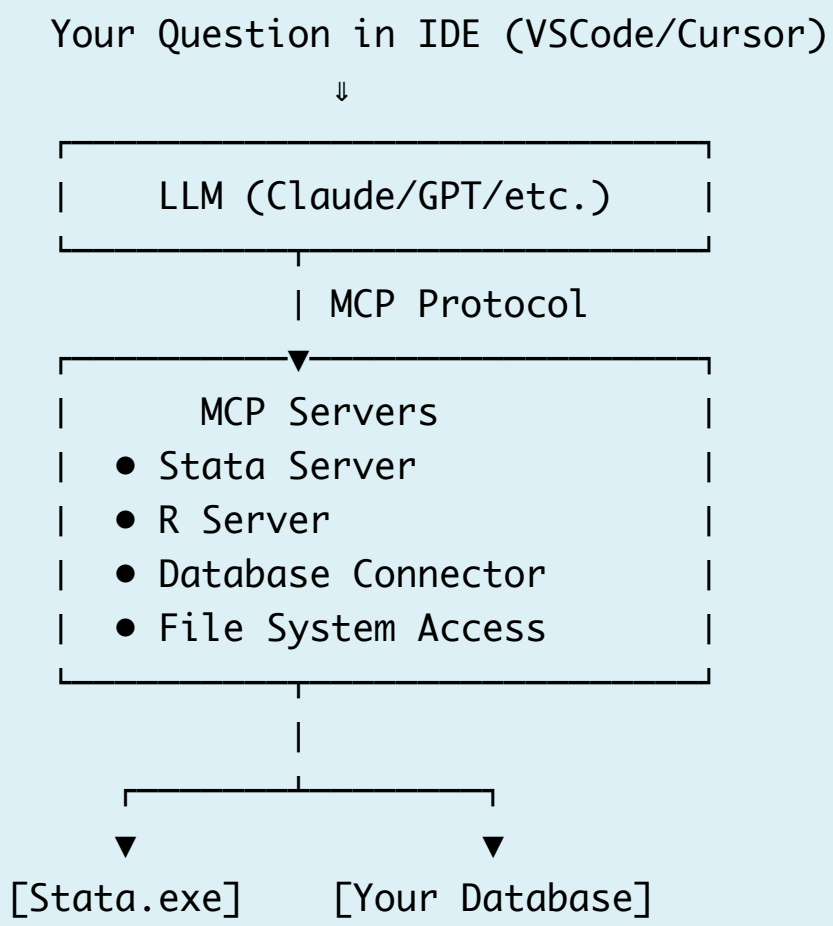
- Each LLM needs custom tool integration
- GPT tools ≠ Claude tools ≠ Gemini tools
- Fragmented ecosystem

With MCP:

- **One protocol, any LLM**
- Write once, works everywhere
- Plug-and-play tool ecosystem

Think USB-C for AI

How MCP Works



Key Benefit: Configure once ⇒ All MCP-compatible LLMs can use your tools.

Tool Use: Code Interpreter

Running Programs

Capability: LLM writes code $\Rightarrow$ Code executes $\Rightarrow$ Results return to LLM

Upload dataset $\Rightarrow$ Ask questions

LLM:

1. Writes Python/R code
2. Processes your data
3. Creates visualizations
4. Returns plots and insights

Use cases:

- Exploratory data analysis
- Quick statistical tests
- Data cleaning pipelines
- Custom visualizations

Multimodality: Beyond Text

LLMs Can Now See and Hear

Vision (Image Input):

- Screenshot analysis
- Chart interpretation
- PDF/document reading
- Handwriting recognition

Example for economists:

Upload graph ⇒ "What's the trend?" Upload table ⇒ "Extract into CSV" Photo of whiteboard ⇒ "Transcribe equations"

Two types of audio input:

1. Fake audio: Text-to-Speech (TTS)
    - Speech ⇒ Text ⇒ LLM ⇒ Text ⇒ Speech
2. True audio (Advanced Voice): Audio tokens processed directly
    - Natural conversation
    - Emotional tone
    - Audio tokens processed directly

Engineering Better Interactions

What is Prompt Engineering?

The Art of Communicating with AI

Definition: **Writing instructions that get LLMs to do exactly what you want**

Why it matters:

- Same LLM, different prompts $\Rightarrow$ vastly different results
- Good prompts save time and frustration
- Critical skill for AI-assisted work

Analogy:

Bad prompt = Unclear homework $\Rightarrow$ Confused students $\Rightarrow$ Poor work

Good prompt = Clear instructions $\Rightarrow$ Focused students $\Rightarrow$ Excellent work

Example:

Bad prompt: "Analyze data"

Good prompt: "Calculate year-over-year GDP growth rates for G7 countries from 2000-2020. Handle missing values with linear interpolation. Create a line plot with different colors for each country, including a legend and proper axis labels."

Anatomy of a Good Prompt

Five Key Components

1. Context: What is the situation?

```
"I'm analyzing minimum wage effects on employment using state-level panel data"
```

2. Task: What do you want done?

```
"Estimate a difference-in-differences model"
```

3. Requirements: Specific features/constraints

```
"Include state and year fixed effects, cluster standard errors at state level"
```

4. Format: How should output look?

```
"Create a regression table in stargazer format with significance stars"
```

5. Examples: Show what you mean (optional but powerful)

```
"Similar to Table 3 in Card & Krueger (1994)"
```

Prompt Engineering Techniques

Technique 1: Chain-of-Thought

Idea: Ask LLM to think step-by-step

Without CoT:

```
"Calculate real GDP growth"
```

LLM might skip steps or make assumptions

With CoT:

```
"Calculate real GDP growth:
Step 1: Deflate nominal GDP using CPI
Step 2: Calculate YoY % change
Step 3: Handle first year (no prior) with NA
Step 4: Create new column 'real_gdp_growth'"
```

LLM follows your logic explicitly

Result: More accurate, easier to debug, transparent reasoning

Prompt Engineering Techniques

Technique 2: Few-Shot Prompting

Idea: Show examples of desired input/output

Example:

```
# Create summary table similar to academic papers
# Example format:
# Variable     | Mean   | SD    | Min   | Max   | N
# GDP          | 5.2    | 1.3   | 2.1   | 8.7   | 150
# Inflation    | 2.3    | 0.8   | 0.5   | 4.2   | 150
#
# Use my dataset 'macro_data' with variables:
# gdp, inflation, unemployment, interest_rate
```

Why it works:

- Shows exact format you want
- Clarifies ambiguous requests
- Reduces back-and-forth

Prompt Engineering Techniques

Technique 3: Role Prompting

Idea: Tell LLM what expertise to adopt

Generic:

```
"Help me with regression analysis"
```

Result: Generic statistical advice

Role-based:

```
"As an econometrician, help me
specify a DiD model with:
- Staggered treatment timing
- Heterogeneous treatment effects
- Potential parallel trends violations

Suggest appropriate estimator and
R implementation."
```

Result: Econometrics-specific guidance

Other useful roles: "As a labor economist...", "As a time series expert...", "As a data scientist..."

Best Practices Summary

DO's

✓ Start with clear context about your research

✓ Use structured prompts (context, task, requirements, format)

✓ Iterate: Refine prompts based on initial outputs

✓ Validate: Always check LLM outputs for correctness

✓ Document: Save prompts that worked well

✓ Combine techniques: CoT + Few-shot + Role prompting

DON'Ts

✗ Don't blindly trust outputs (validate statistically)

✗ Don't overload context with irrelevant information

✗ Don't use vague references ("it", "that", "the thing")

✗ Don't forget citations when using AI in research

Practical Guidelines for Applications of LLMs

LLMs in the real world

- As of May 2025, The most popular LLMs in the world market:

Company	Name	Models	APP	API	Open Sources	Service in China
OpenAI	ChatGPT	GPT 4.5	Yes	Yes	No	No
Google	Gemini	Gemini 2.5 Pro	Yes	Yes	No	No
Microsoft	Copilot	ChatGPT o3 and others	Yes	Yes	No	**Yes**
Anthropic	Claude	Claude 3.7 sonnet	Yes	Yes	No	No
Meta	Llama	Llama 4-MoE	Yes	Yes	Yes	Yes
DeepSeek	DeepSeek	DeepSeek R2	Yes	Yes	Yes	Yes
Alibaba	Qwen	Qwen 3	Yes	Yes	Yes	Yes

- **Copilot is the only foreign company who provides partially service in China.**
- DeepSeek and Alibaba: Chinese companies provide world-class open source LLMs in China.

Popular Chinese LLMs

- As of August 2024, there are over 200 players in China LLMs market:
- Besides Deepseek and Alibaba, Other popular general models providers:

Company	Name	APP	API	Open Sources
百度	文心一言 4.0	Yes	Yes	No
腾讯	混元	Yes	Yes	No
科大讯飞	星火3.5	Yes	Yes	No
月之暗面	Kimi	No	Yes	No
字节跳动	豆包	No	Yes	No

Ways to use LLMs

- Web interface
- Apps on devices (e.g., mobile phones, tablets, even PCs)
- Command-line interface (CLI)
- Application Programming Interface (API)
- Integrated into some applications (e.g., Microsoft Offices, Google Docs, etc.)
- Local installation (e.g., on a server, a PC, etc.) for Open source LLMs
- Main ways to use LLMs for social science research:
  - ChatBots(APP or Web interface)
  - In IDE/Editor(Cursor, VSCode, etc.)
  - In API(e.g., Claude API, OpenAI API, etc.)

Which model(s) should I use?

- Each product on the previous slide has strengths and weaknesses.
- As a social science researcher, I strongly recommend you to use at least two LLMs :
  - One from outside China: ChatGPT, Claude, Gemini or Copilot.
  - The other from China: Deepseek or 通义千问.

Which model(s) should I use?

- Pricing and discount for different models(API is not included):

Model	Pricing	Student Discount
ChatGPT	Plus: $20/month Prefessional: $200/month	Yes, only students in US and Canada from May 31, 2025
Claude	Pro: $20/month	No. only institutional users
Google AI	Advanced: $19.99/month	Yes, only for students in US
Copilot	Pro: $20/month	Yes, Free for students all over the world
Deepseek	Pro: ¥299/month	Yes, 50% off Pro (¥149/month) or institutional users
通义千问	Free basic version	No. only institutional users

Practice in VS Code with Copilot