Advanced Human Language Technologies
Exercises on LLMs

Large Language Models

Exercise 1.

Consider a language model trained to predict the next word in a sentence. Given the partial sentence: *The cat sat on the ...*

The model assigns the following probabilities to possible next words:

Word	Probability
mat	0.55
floor	0.20
tree	0.15
chair	0.10

1. What is the most likely next word?
2. If we use *top-k sampling* ($k = 2$), which words are possible?
3. If we apply *temperature scaling* with a high temperature, how might the probabilities change?
4. Why might an LLM assign a low probability to *tree*?

Exercise 2.

LLMs can be trained in two ways:

- *Pretraining* on a large corpus (self-supervised learning).
- *Fine-tuning* on a specific task (e.g., sentiment analysis, medical QA).

1. What types of knowledge does a model learn in *pretraining* ?
2. Why is fine-tuning necessary for domain-specific tasks?
3. What happens if an LLM is not fine-tuned and used for zero-shot tasks?

Exercise 3.

LLMs are sensitive to *how* we phrase a prompt. Consider the following queries given to an LLM:

- *What is the capital of France?*
- *Tell me a fun fact about Paris.*
- *Describe the geography of France.*

1. How do these prompts influence the model's response?
2. Why does a well-structured prompt improve accuracy?
3. How might using few-shot prompting (i.e. providing examples) change the behavior of the model?

Exercise 4.

Many LLMs have a limited context window (e.g. 2048 tokens).

1. Why does this limit long conversations or documents?
2. If an LLM forgets early parts of a conversation, how can techniques like *retrieval-augmented generation (RAG)* help?
3. Why are newer models increasing their context size?

Exercise 5.

LLMs are trained on vast amounts of internet data, which can introduce biases.

1. How might an LLM generate biased responses in a job application review system?
2. What are some ways to mitigate bias in LLMs?
3. Why is explainability important when using LLMs for decision-making?

Exercise 6.

Some LLMs are open-source (e.g., LLaMA, Mistral), while others are closed-source (e.g., GPT-4, Claude).

1. What are the advantages of open models?
2. Why do companies keep some models closed?
3. How does fine-tuning an open model differ from using an API to fine-tune a closed model?

Exercise 7.

A company is developing an AI-powered **customer support chatbot** and needs to decide between:

- **Fine-tuning an open-source LLM** (e.g., LLaMA 3) and hosting it on cloud infrastructure.
- **Using a proprietary LLM API** (e.g., GPT-4o-mini) that allows fine-tuning for a fee per token.

The following costs apply:

- **Proprietary API Costs (GPT-4 example)**:
  - Fine-tuning: $\mathbf{3} \cdot \mathbf{10^{-6}}$ \$/token
  - Inference: $\mathbf{0.5} \cdot \mathbf{10^{-6}}$ \$/token
- **Open-Source LLM Costs (e.g., LLaMA-3 on AWS)**:
  - Cloud machine for fine-tuning (8xA100 GPUs, 80GB): $\mathbf{1,000}$ \$ (one-time cost, 3 days usage)
  - Cloud machine for inference (1xA100 GPU, 80GB): $\mathbf{2,000}$ \$/month.
    LLaMA-3 running in this machine can process $\mathbf{100}$ tokens per second.

The company expects:

- To fine-tune on $\mathbf{10}$ million tokens of training data, using $\mathbf{5}$ epochs.
- Each customer to generate an average of $\mathbf{50,000}$ tokens per month during inference.
- The company to have $\mathbf{1,000}$ customers.

1. Compute the total cost of fine-tuning for both options.
2. Compute the monthly inference cost for both options.
3. What is the number of customers we need to have so the open-source model becomes cheaper ?
4. What other factors (besides cost) should the company consider when choosing between open-source and proprietary models?

Exercise 8.

Large Language Models (LLMs) can be used in different ways depending on the task:

- **Zero-shot learning**: Using the model as-is, without providing examples.
- **Few-shot learning**: Providing a few examples in the prompt to guide the model's response.
- **Fine-tuning**: Training the model on task-specific data to improve performance.

For each of the following tasks and scenarios, explain which approach is most suitable and why.

1. A company wants to deploy an AI-powered customer support chatbot. The chatbot must follow company policies and provide accurate responses to frequently asked questions. The company has access to a large dataset of past customer interactions.
2. A startup is developing an AI assistant that helps developers write Python code. The model should understand coding patterns, generate functions, and complete code snippets. The company does not have its own dataset but can provide a few examples in the prompt.
3. A business needs a model to analyze customer sentiment in *legal documents*, where words have different meanings than in general language. They have a labeled dataset with sentiment annotations.
4. A mobile app needs a lightweight model for *on-device* translation between rare language pairs. There is no labeled data for training, and responses must be fast.
5. A media company wants an AI system to generate *concise summaries* of news articles. The articles vary in topics and writing style. The company has a moderate budget and needs a scalable solution.