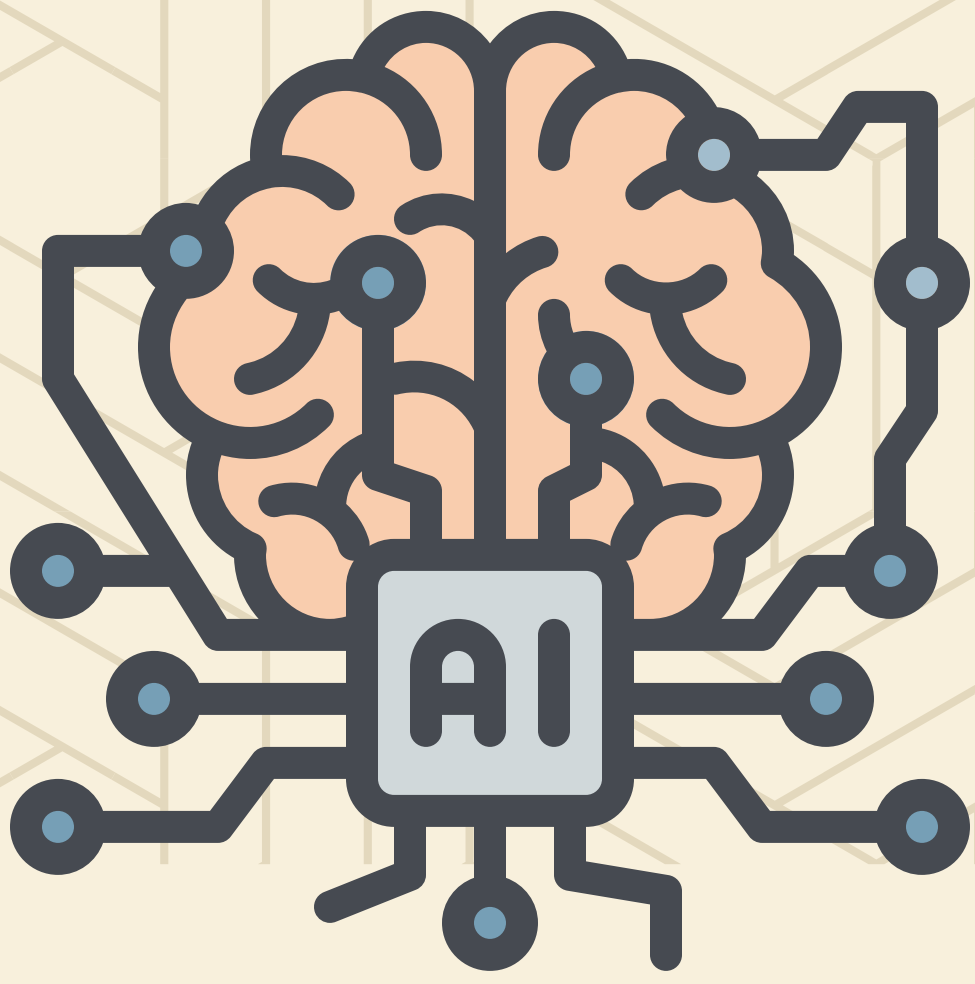
TOP 10 LLM INTERVIEW QUESTIONS

Part I



Bhavishya Pandit

LLM

Q1. What are Large Language Models?

Ans -

A Large Language Model (LLM) is an AI system trained on vast amounts of text to understand, generate, and predict human-like language. It learns patterns, context, and relationships in the data to produce relevant, coherent responses.

LLMs can handle a wide range of tasks, from answering questions and summarizing text to performing translations and even creative writing. Their ability to generalize across different language tasks comes from training on diverse datasets, allowing them to generate contextually appropriate and meaningful content based on the input they receive.

Q2. How do you measure the performance of an LLM?

Ans - To measure the performance of a Large Language Model (LLM), various evaluation metrics are used:

- **Perplexity:** Assesses how well the model predicts text, often used in language modeling.

- **Accuracy:** Evaluates the proportion of correct predictions, commonly used in text classification tasks.

- **F1 Score:** Combines precision and recall into a single metric, useful for tasks like named entity recognition.

- **BLEU Score:** Measures the quality of machine-generated text compared to reference translations, typically used in machine translation.

- **ROUGE:** A set of metrics that check how well the generated text overlaps with reference text, often applied in text summarization.

Q3. Explain the concept of "few-shot learning" in LLMs and its advantages.

Ans - Few-shot learning in LLMs is the ability of the model to understand and tackle new tasks with just a few examples. This is made possible by the model's extensive pre-training, which allows it to generalize from limited data.

The main benefits of few-shot learning include:

- **Reduced Data Needs**: It requires fewer examples to perform well, minimizing the need for large, task-specific datasets.

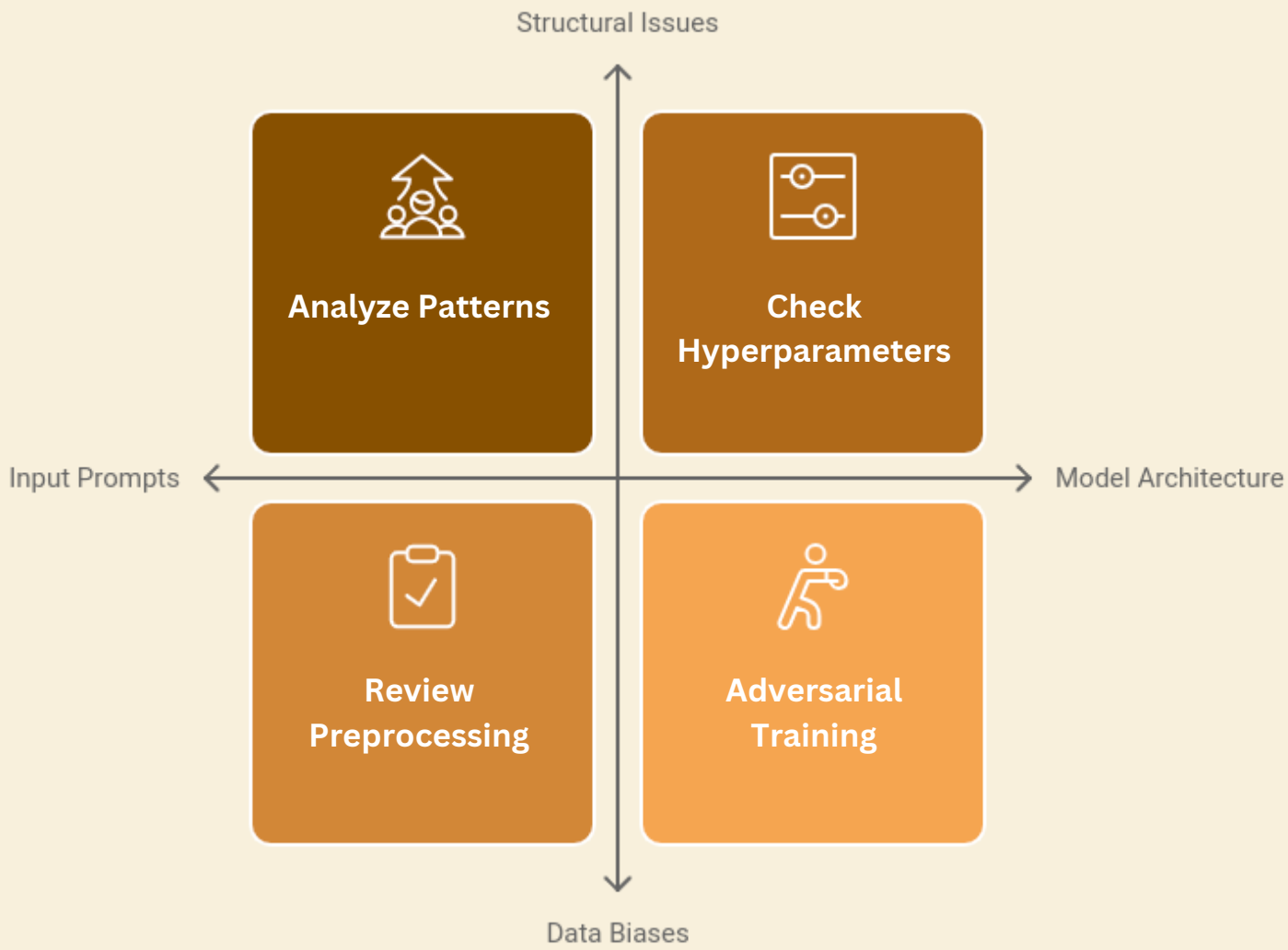
- **Increased Flexibility:** The model can easily adapt to various tasks with minimal additional training.

- **Cost Efficiency:** With less need for extensive data and reduced training times, it lowers the costs associated with data collection and computational resources.
- **Interpretability:** It can be challenging to understand and explain how LLMs make their decisions due to their complex and often opaque nature.

- **Data Privacy:** Training on large datasets can raise concerns about data privacy and security.
- **Cost:** Developing, training, and deploying LLMs can be costly, which may limit their use for smaller organizations.

Q4. You're working on an LLM, and it starts generating offensive or factually incorrect outputs. How would you diagnose and address this issue?

Ans - If an LLM produces offensive or inaccurate outputs, I would first analyze the patterns, check input prompts, and assess if the issue stems from biases or gaps in the training data. I'd review the preprocessing pipeline for errors or biases and examine the dataset for imbalances.

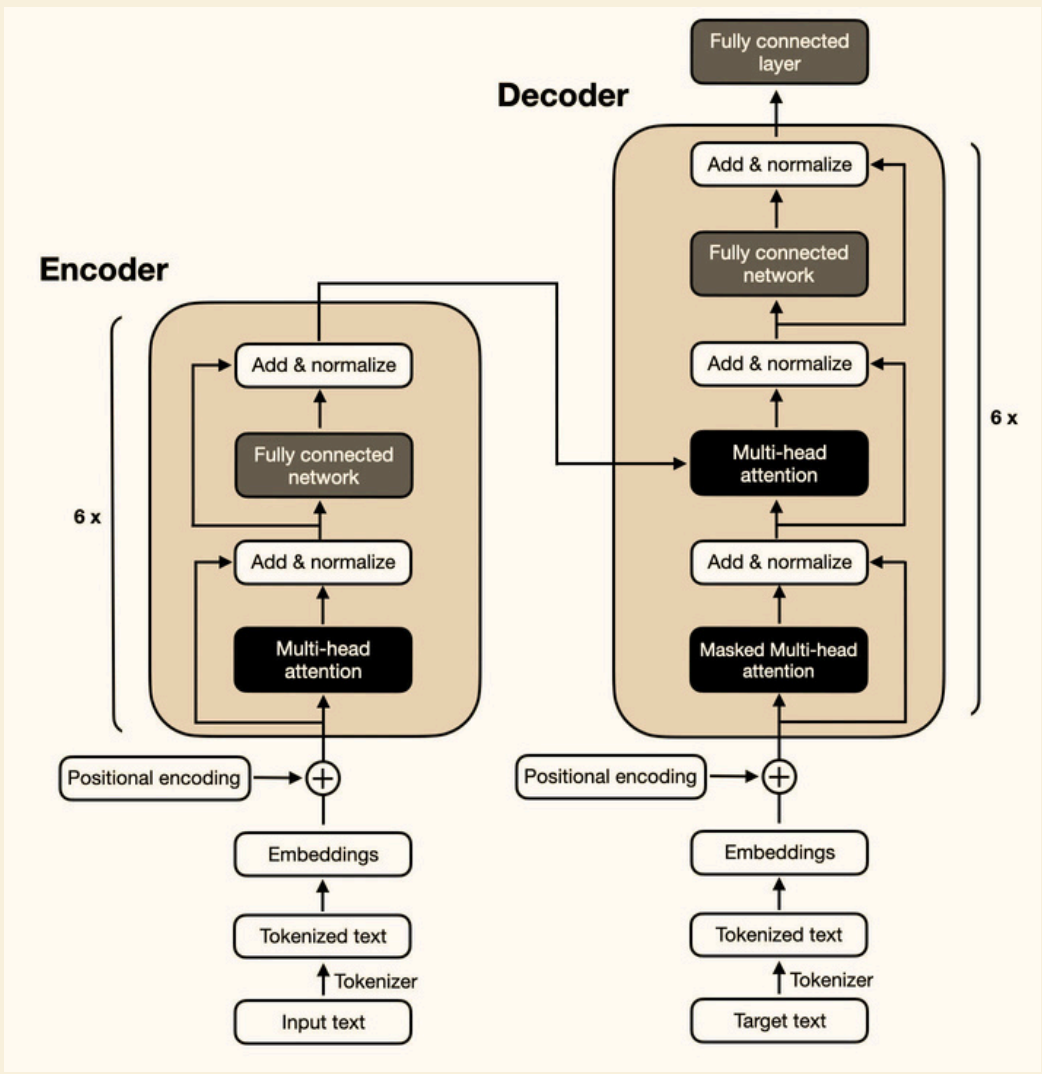


Next, I'd evaluate the model's architecture, hyperparameters, and fine-tuning to identify any structural issues. Solutions could include adversarial training, debiasing, data augmentation, or retraining with a more balanced dataset.

LLM

Q5. How is the encoder different from the decoder?

Ans -



In the transformer architecture used in large language models, the encoder and decoder serve different purposes. The encoder processes the input data and transforms it into a set of abstract representations. The decoder then takes these representations and generates the output, using both the information from the encoder and the previously generated elements in the sequence. Essentially, the encoder is responsible for understanding the input, while the decoder focuses on producing the final output.

Q6. What are the main differences between LLMs and traditional statistical language models?

Ans -

- **Architecture:** LLMs are based on transformers with self-attention, which captures long-range dependencies, unlike traditional models like N-grams or HMMs that struggle with this.

- **Scale:** LLMs have billions of parameters and train on massive datasets, enabling better generalization. Traditional models are smaller and task-specific.

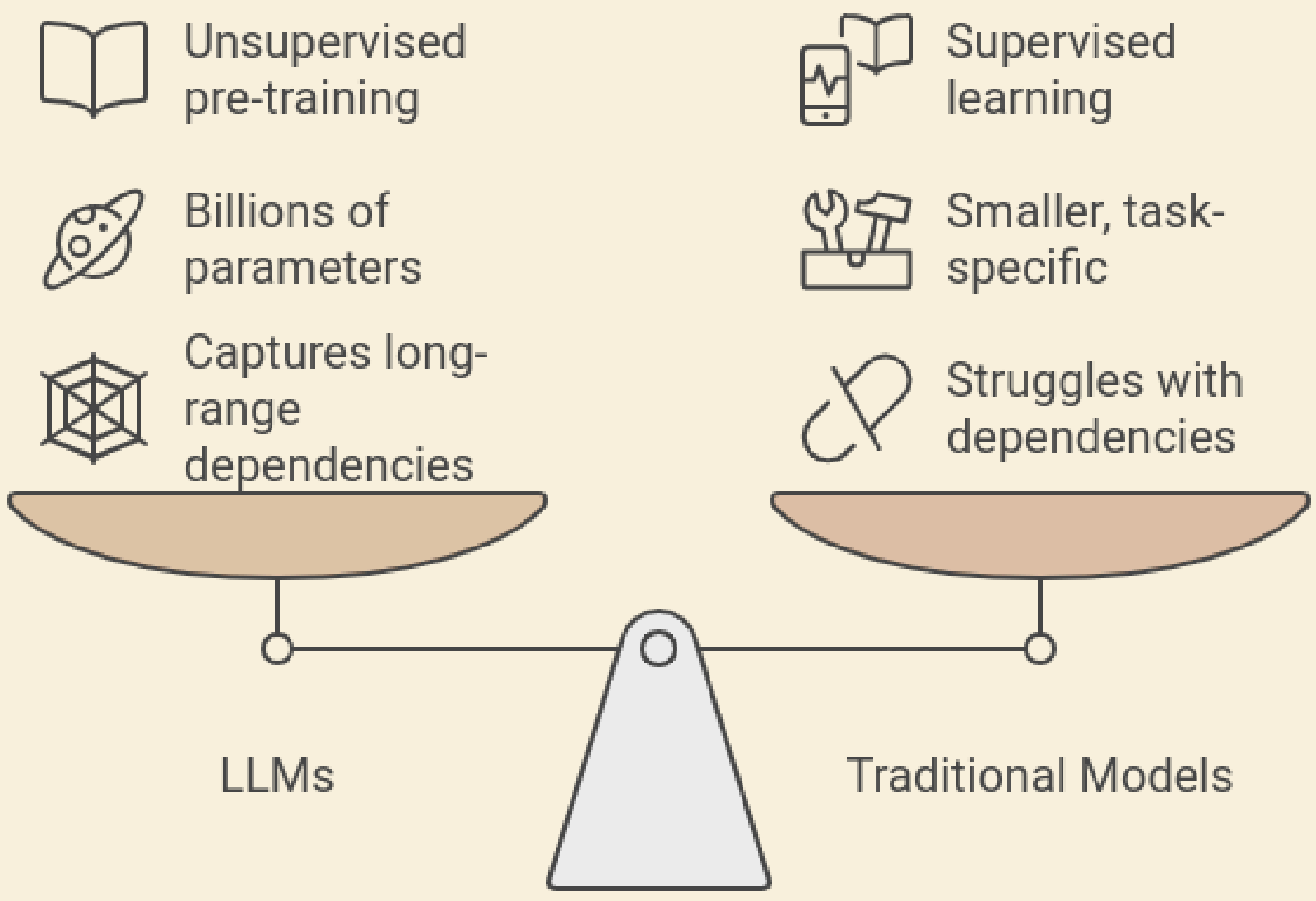
- **Training:** LLMs undergo unsupervised pre-training and fine-tuning. Traditional models rely on supervised learning with labeled data for each task.

- **Input:** LLMs handle variable-length inputs using advanced tokenization like BPE, whereas traditional models often use fixed-length inputs and simpler tokenization.

- **Context:** LLMs generate contextual embeddings, adapting to word meaning based on context. Traditional

models use static embeddings.

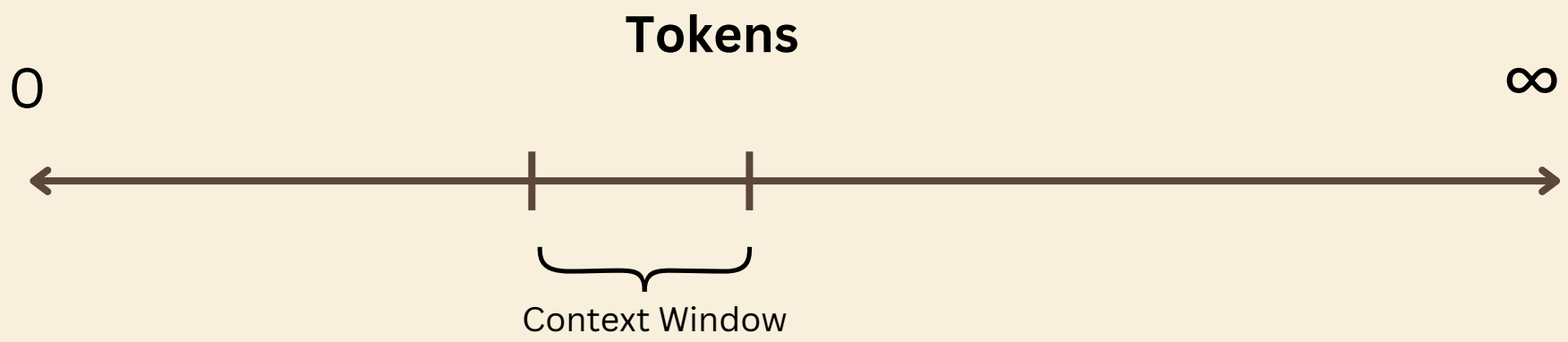
- **Flexibility:** LLMs can tackle multiple NLP tasks with little fine-tuning, while traditional models are designed for specific tasks.
- **Resources:** LLMs demand high computational power, requiring GPUs or TPUs, whereas traditional models are more lightweight.



LLM

Q7. What is a “context window”?

Ans - The "context window" in large language models (LLMs) is the span of text—measured in tokens or words—that the model can process at any given moment when generating or interpreting language. The importance of the context window lies in its influence on the model's ability to produce coherent and contextually relevant responses.

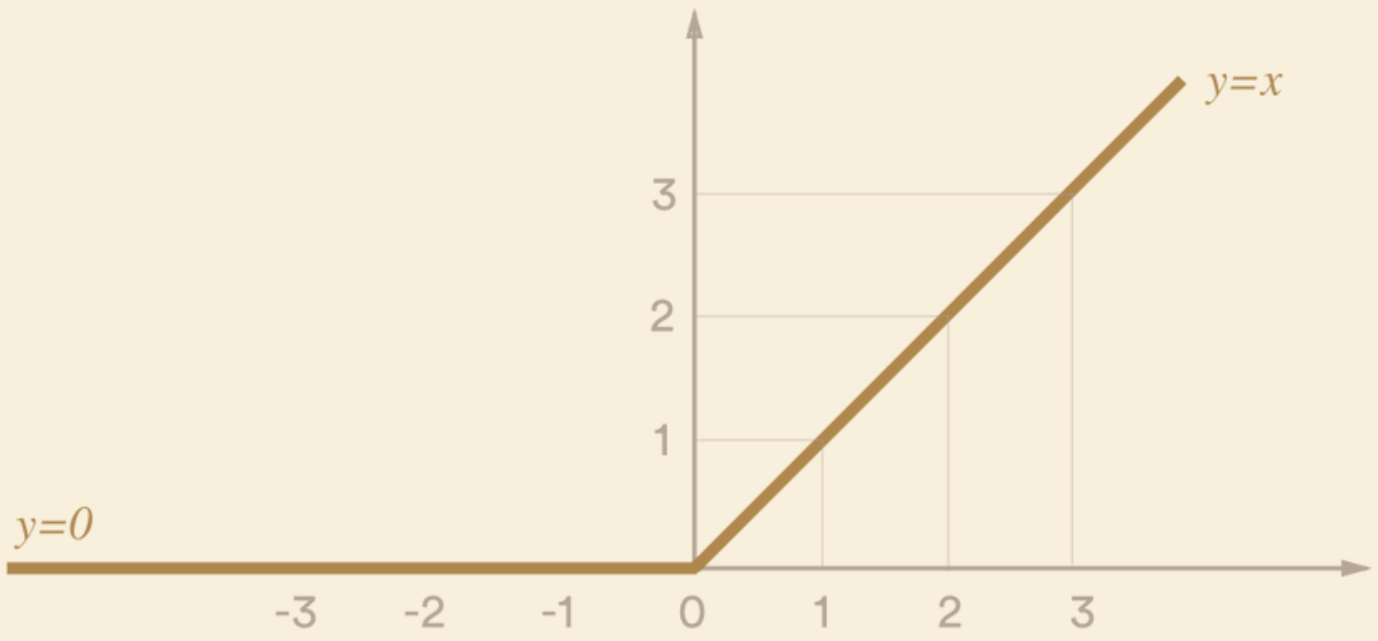


A larger context window means the model can incorporate more surrounding information, which enhances its understanding and ability to generate text, particularly in more complex or extended interactions. However, increasing the context window also raises computational demands, so there’s a trade-off between improved performance and resource efficiency.

LLM

Q8. What is a hyperparameter?

Ans - A hyperparameter is a parameter that is set before the training process begins and influences how the model is trained. It controls aspects of the training process and is chosen by the developer or researcher based on prior knowledge or experimentation. Common examples of hyperparameters include the model's architecture, batch size, regularization strength, and learning rate.

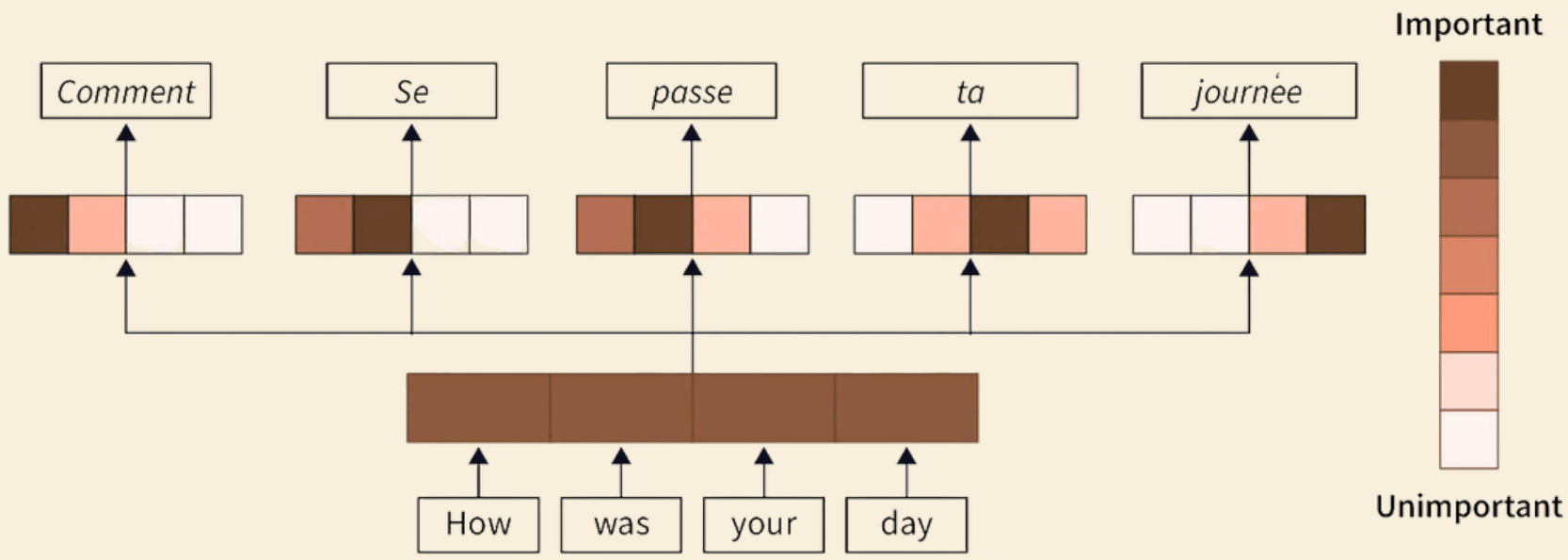


Some common examples

- Train-test split ratio
- Learning rate in optimization algorithms (e.g. gradient descent)
- Choice of optimization algorithm (e.g., gradient descent, stochastic gradient descent, or Adam optimizer)
- Choice of activation function in a neural network (nn) layer (e.g. Sigmoid, ReLU, Tanh)

Q9. Can you explain the concept of attention mechanisms in transformer models?

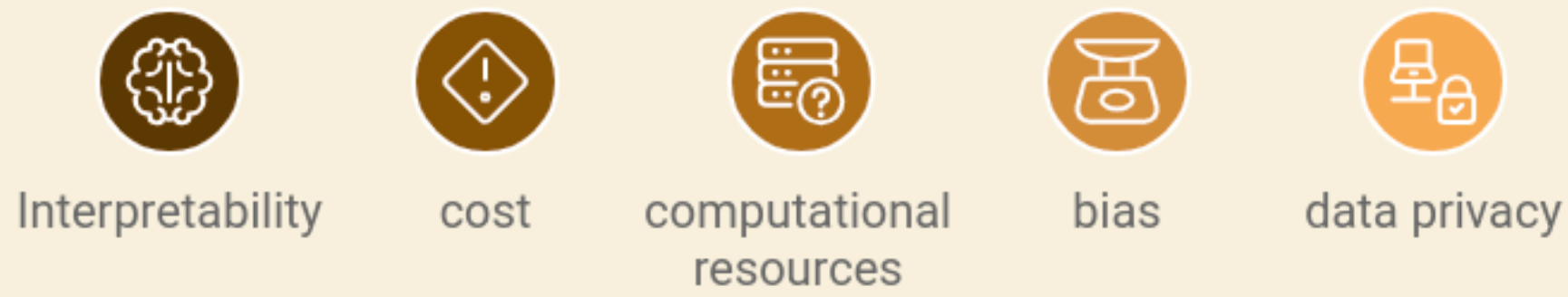
Ans - At a high level, attention allows the model to focus on different parts of the input sequence when making predictions. Instead of treating every word or token equally, the model learns to "attend" to relevant words that contribute most to the current prediction, regardless of their position in the sequence.



For example, in a sentence like "The dog chased the ball because it was fast," the word "it" could refer to either the dog or the ball. The attention mechanism helps the model figure out that "it" is likely referring to "the ball" based on the context.

Q10. What are some common challenges associated with using LLMs?

Ans - Using LLMs presents several common challenges:



- **Computational Resources:** They require substantial computing power and memory, making both training and deployment demanding.

- **Bias and Fairness:** LLMs might learn and reproduce biases from their training data, potentially leading to biased or unfair outputs.

- **Interpretability:** It can be challenging to understand and explain how LLMs make their decisions due to their complex and often opaque nature.

- **Data Privacy:** Training on large datasets can raise concerns about data privacy and security.

- **Cost:** Developing, training, and deploying LLMs can be costly, which may limit their use for smaller organizations.

Follow for more AI/ML posts

SAVE

LIKE

SHARE

Bhavishya Pandit