Open-source LLMs

Quantitative Comparison, Finetuning, and Evaluation

Xyrel De Mesa
AI Republic

Large Languade Models

The idea of "large" in LLMs is a moving target, dependent on the current state of technology and available computational resources.

Criteria	Description
Parameters	LLMs have billions of parameters for nuanced language understanding.
Training Data	Trained on vast, diverse text data (gigabytes to terabytes).
Architecture	Uses deep neural network architectures, often Transformer-based, to capture complex language patterns.
Computation	Requires significant computational power (GPUs/TPUs) for training.
Generalization	Can perform various language tasks without specific tuning due to large-scale training.

How do we quantitatively compare different language models?

Metrics in Comparing Models

Quality Index

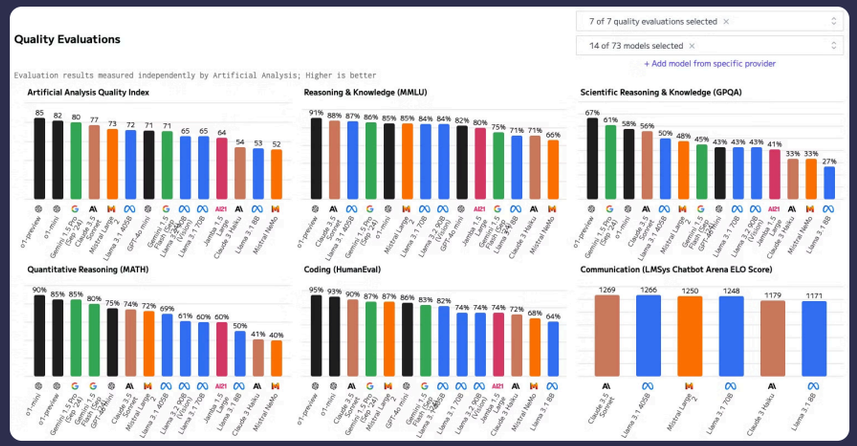
- A combined score that evaluates the overall quality of a model based on the more granular benchmarks.

Reasoning & Knowledge (MMLU)

- MMLU (Massive Multitask Language Understanding) evaluates a model's ability to perform across a diverse set of tasks in 57 subjects.
- It reflects the model's ability to **understand and reason across different contexts**, similar to human knowledge application.

Scientific Reasoning & Knowledge (GPQA)

- GPQA (Grade School Problem Solving with Multiple Choice Questions) assesses a model's ability to perform **scientific reasoning and problem-solving at a grade-school level**.
- It focuses on understanding and applying basic scientific concepts to answer questions correctly.



artificialanalysis.ai

Metrics in Comparing Models

Quantitative Reasoning (MATH)

- MATH evaluates quantitative reasoning and **mathematical problem-solving abilities**.
- It includes a range of problems from basic arithmetic to advanced mathematics.

Coding (HumanEval)

- HumanEval evaluates a model's ability to generate **correct and functional code based on natural language prompts**.
- *https://github.com/openai/human-eval*

Communication (LMSys Chatbot Arena ELO Score)

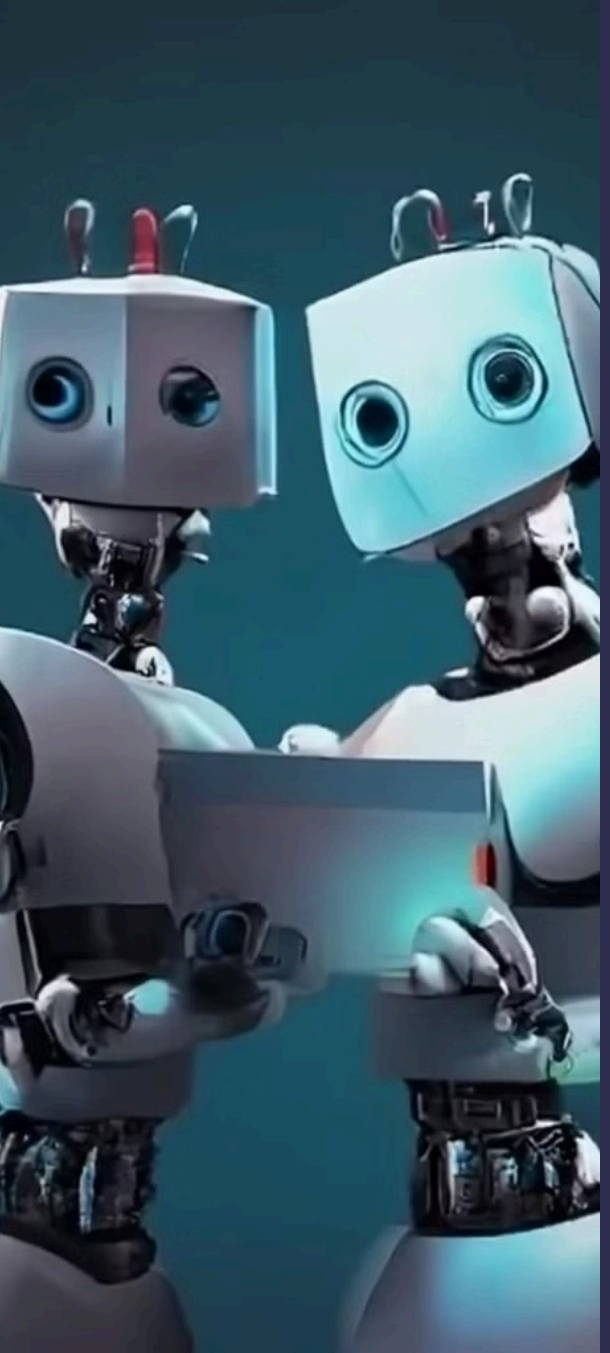
- The LMSys Chatbot Arena ELO Score is a metric derived from the ELO rating system, traditionally used in competitive games, adapted to evaluate chatbot performance.
- It measures the model's ability to engage in **coherent, contextually appropriate, and user-satisfying** conversations.

Maths (MGSM)

- MGSM (Mathematical General Syllabus Modelling) is focused on a more structured, curriculum-based approach to math, evaluating how well a model can solve problems aligned with educational syllabi.
- It evaluates the model’s capability to handle math problems typically taught in schools, ensuring alignment with general educational standards. It is a more **structured and pedagogical** evaluation, meant to see how well the model can assist in academic settings.

Task-Specific Prompt Examples

Metric	Example Prompt
Reasoning & Knowledge (MMLU)	“Explain the differences between classical and operant conditioning in psychology.”
Scientific Reasoning & Knowledge (GPQA)	“Describe the process of photosynthesis, and explain the role of chlorophyll.”
Quantitative Reasoning (MATH)	“Solve the equation 3 ▯ 2 − 12 ▯ + 9 = 0 3x 2 −12x+9=0, and explain each step in your solution.”
Coding (HumanEval)	“Generate a Python function that takes a list of integers and returns the sum of all even numbers.”
Communication (LMSys Chatbot Arena ELO Score)	“Can you help me troubleshoot my internet connection? I keep losing signal.”
Maths (MGSM)	“Find the area of a triangle with a base of 8 cm and a height of 5 cm.”



What's the difference between base and instruct models?

Base vs. Instruct Models

huggingface.co/google

huggingface.co/meta-llama

Base Models: Offer versatility and general-purpose capabilities suitable for a wide range of applications.

Instruct Models: Provide enhanced performance for tasks requiring precise and directive responses.

Base vs. Instruct Models

	Base Models	Instruct Models
Training Data	Large, diverse, unlabeled datasets.	Base model training plus supervised instruction-response pairs.
Primary Characteristics	Versatile and general-purpose.	Enhanced ability to understand and execute user instructions.
Performance	Generates coherent and fluent text across various topics.	Produces more accurate and aligned responses based on prompts.
Use Cases	Creative writing, brainstorming, language translation, general conversation.	Customer support, task automation, educational tools, precise information retrieval.
Flexibility	High flexibility for a wide range of applications.	Optimized for tasks requiring clear and directive outputs.

Sample Training Data

Base Model Training Data

Text from a Novel:
"Once upon a time, in a land far away, there lived a young prince who dreamed of adventure."

News Article Excerpt:
"The stock market saw a significant increase today, with major indices closing higher."

Wikipedia Entry:
"The Great Barrier Reef is the world's largest coral reef system, located in Australia."

- **Predictive Generation:** The model generates the next word or sequence of words that are statistically likely to follow the input prompt.
- **Coherent and Fluent Text:** Outputs are coherent and contextually relevant but may lack specificity or adherence to particular instructions.
- **Open-Ended:** The generation is open-ended and can vary widely depending on the input prompt.

Instruct Model Training Data

Instruction:
"Translate the following sentence to French: 'Hello, how are you?'"
Response:
"Bonjour, comment ça va?"

Instruction:
"Summarize the main points of the article about climate change."
Response:
"The article discusses the impacts of climate change, including rising temperatures and increased frequency of extreme weather events."

Instruction:
"Generate a friendly email response to a customer complaint about a delayed shipment."
Response:
"Dear [Customer], I apologize for the delay in your shipment. We are working to ensure your order arrives soon."

- **Task-Specific:** Outputs are tailored to perform specific tasks such as answering questions, summarizing text, translating languages, etc.
- **Consistent and Reliable:** Responses are more consistent and reliable in following the provided instructions compared to base models.

Finetuning Language Models

"I want to use GPT for a specific use case."

Finetuning LLM

- Retrain an LLM with a lot of private or proprietary data to tailor its responses for specific needs.
- Static knowledge base; requires retraining to update information.

vs.

RAG (Knowledge Base)

- Creating a vector database stores data as embeddings, of which relevant information which are retrieved to enhance the LLM's response with contextual knowledge.
- Dynamic knowledge base; easily updated by modifying external data sources.

Finetuning LLM

- Ideal for applications requiring single, deep expertise (e.g., medical diagnosis, legal analysis).
- Ensures knowledge is embedded within the model for consistency and reliability.
- Suitable when integrating and maintaining an external retrieval system is impractical.
- Critical for real-time applications needing fast, low-latency responses (e.g., interactive chatbots).

vs.

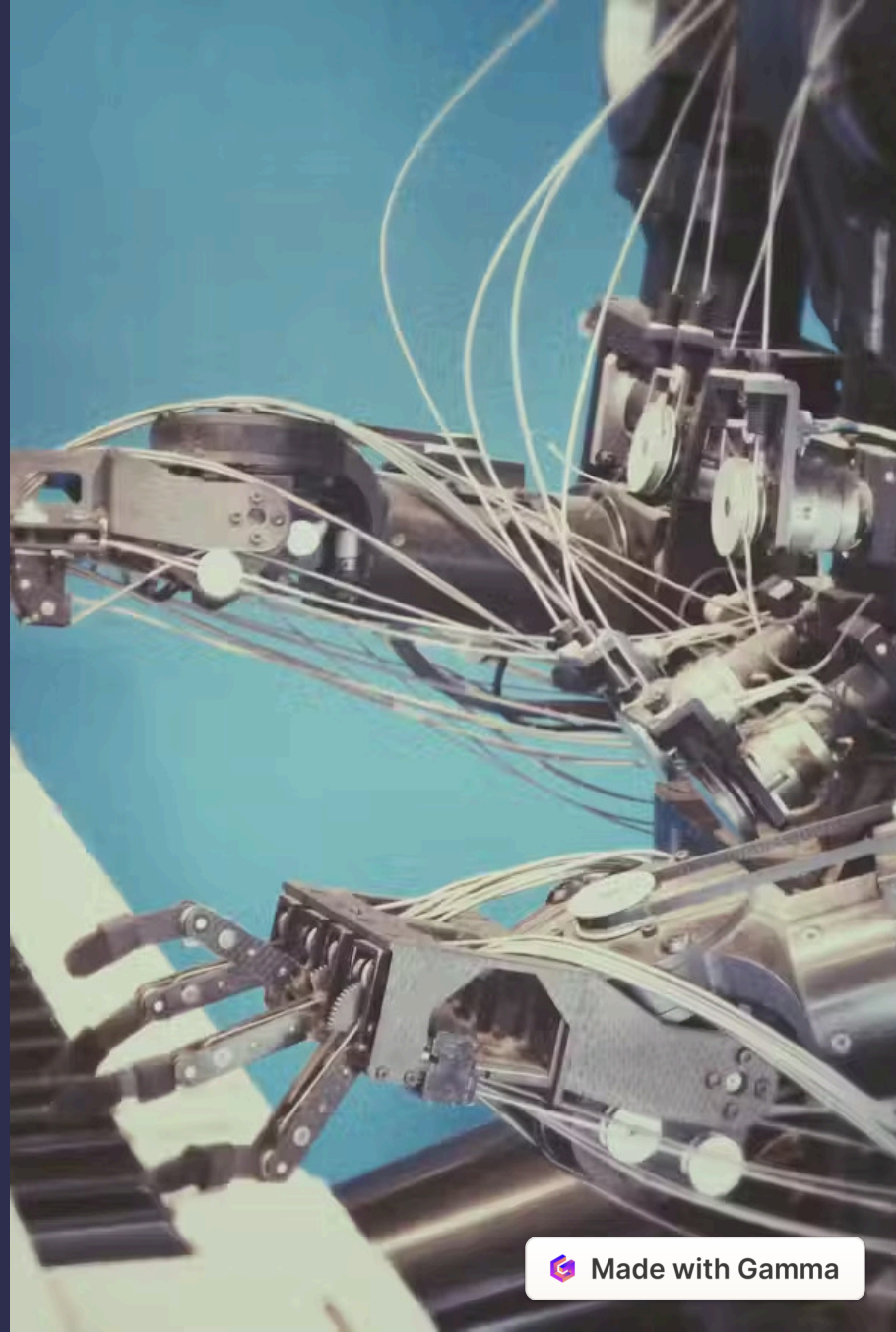
RAG (Knowledge Base)

- Ideal for applications requiring the latest information (e.g., news aggregation, live updates).
- Update external data sources without modifying the underlying model.
- Provides more accurate answers by sourcing relevant information during generation.

SuperAnnotate

What do we need to know before finetuning a language model?

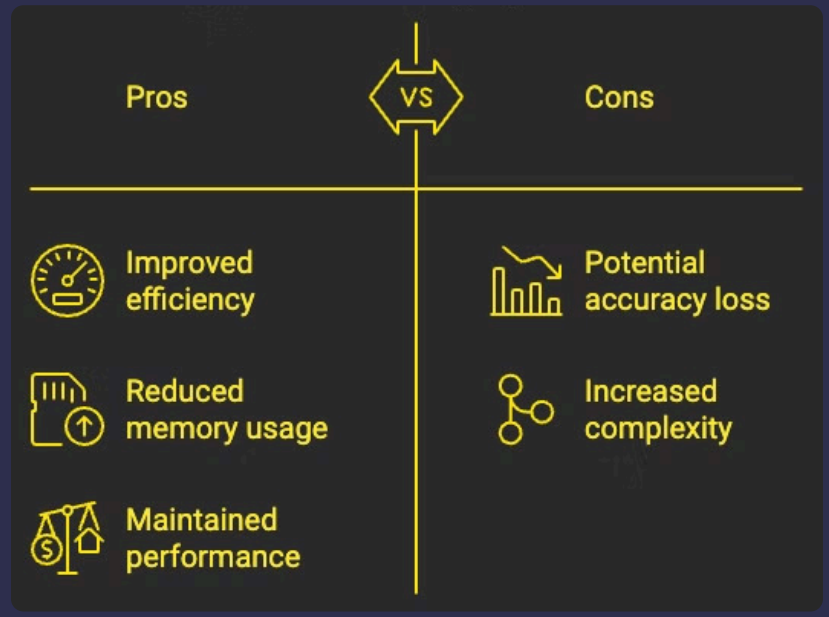
Quantization, LoRA, PEFT, Hyperparameters, and Quantitative Evaluation



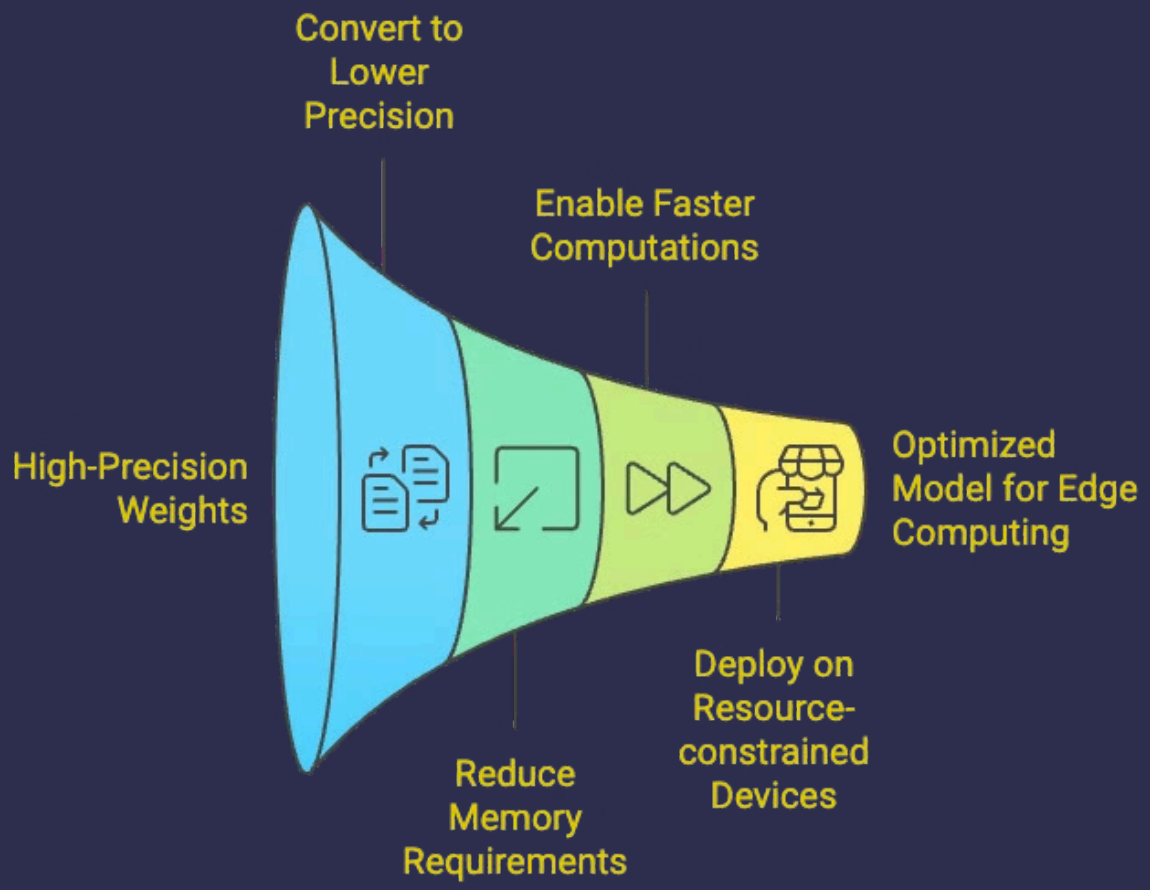
Quantization in Language Models

Quantization is a technique in the optimization of language models, allowing them to run efficiently on resource-constrained devices while maintaining performance.

- **Relevance:** Essential for deploying language models models on resource-constrained devices such as smartphones, edge devices, and embedded systems.



Quantization



Parameter-Efficient Fine-Tuning (PEFT)

Adopt PEFT

More efficient and scalable fine-tuning with reduced resources.

Continue with traditional fine-tuning

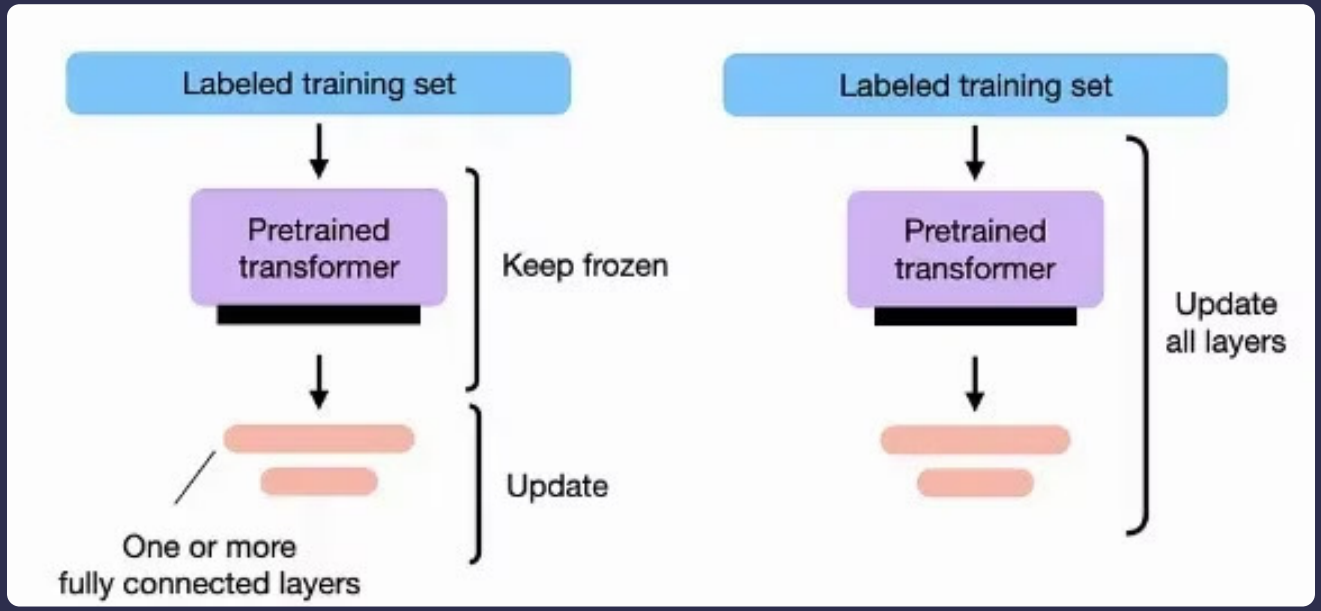
Higher computational costs and potentially lower efficiency.

PEFT is a strategy that focuses on optimizing only a small subset of parameters in a pre-trained language model, rather than adjusting all parameters during the fine-tuning process.

Key Advantages of PEFT:

- **Reduced Computational Cost**
- **Faster Training Times**
- **Improved Generalization**

PEFT vs. Traditional Finetuning



source: https://www.analyticsvidhya.com/blog/2023/08/fine-tuning-large-language-models/

Low-Rank Adaptation (LoRA)

LoRA is a technique designed to efficiently fine-tune large language models by introducing low-rank matrices into the model's architecture.

Instead of updating all the parameters of the model during fine-tuning, LoRA allows for the addition of a small number of trainable parameters that capture the essential adaptations needed for a specific task.

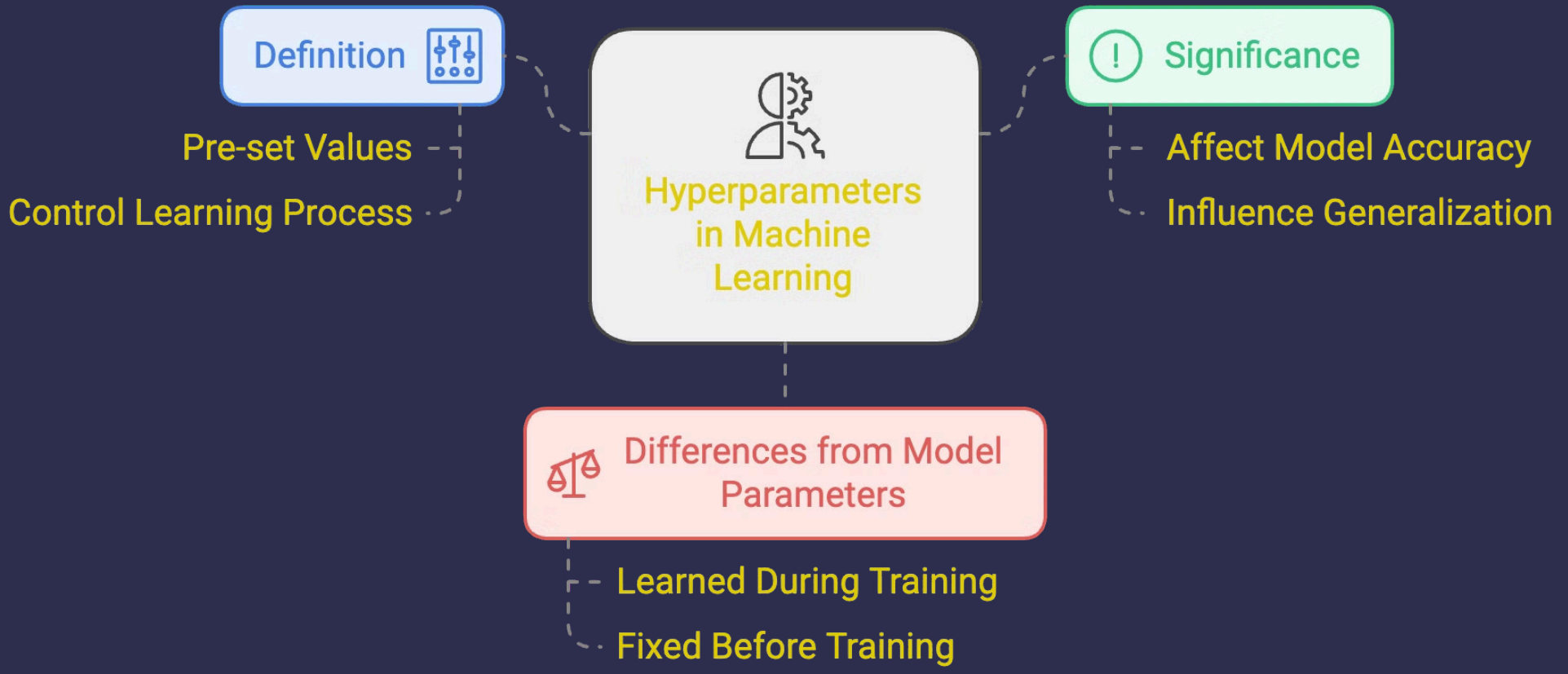
How LoRA works

LoRA modifies the weight matrices of the model by decomposing them into two low-rank matrices. This decomposition allows the model to learn task-specific information without the need to adjust the entire set of parameters. The main steps involved in implementing LoRA include:

1. **Decomposition**: The original weight matrix (W) is approximated as (W + \Delta W), where (\Delta W) is the low-rank adaptation.
2. **Training**: During training, only the low-rank matrices are updated, while the original weights remain frozen. This results in a smaller memory footprint and faster training times.
3. **Inference**: At inference time, the adapted weights are combined with the original weights to produce the final output.

Hyperparameters in ML

Hyperparameters are the configuration settings used to control the learning process of a machine learning algorithm.



Key Hyperparameters

Hyperparameter	Definition	Higher Value	Lower Value
Epochs	The number of complete passes through the entire training dataset.	Allows the model to learn more from the data, but too many epochs can lead to overfitting.	Model may not learn enough from the data (underfitting), leading to poor performance.
Learning Rate	Determines the size of the steps the optimizer takes while updating the model's weights.	Can speed up training but may overshoot optimal solutions	Ensures more precise convergence but can slow down the training process.
Batch Size	The number of training samples processed before the model's internal parameters are updated.	Can lead to more stable gradient estimates and faster computation through parallelism, but they require more memory.	Offers more frequent updates and can help escape local minima but may result in noisier gradient estimates.
Warmup Steps	The number of initial training steps during which the learning rate gradually increases from a lower value to the set learning rate.	More stable training by preventing large weight updates early in training.	Higher risk of unstable training dynamics; reduces the warmup period, speeding up the early stages of training.
Weight Decay	A regularization technique that adds a penalty to the loss function based on the magnitude of the model's weights to prevent overfitting.	Improved generalization (helps the model perform better on unseen data) but excessive weight decay can hinder the model's ability to learn complex patterns.	Allows the model to capture more complex patterns in the data, but may perform poorly on unseen data due to overfitting.

Evaluation Metrics

Metric	Definition	Use Case
Perplexity	A measure of how well a probability model predicts a sample. It is the exponential of the average negative log-likelihood of a sequence. Lower perplexity indicates better performance.	**Language Modeling**: Evaluates the model's ability to predict the next word in a sequence. Commonly used to assess the overall quality of language models.
Semantic Similarity	It evaluates how close the model's generated responses are to the expected answers in terms of meaning.	**Content Relevance:** Measures how semantically close the model's output is to the expected answer, ensuring that the generated text conveys the intended meaning.
BLEU	Measures the precision of n-grams in the generated text against reference texts. It calculates the proportion of overlapping n-grams between the candidate and reference sentences.	**Machine Translation & Text Generation:** Evaluates the fluency and accuracy of generated text by comparing it to one or more reference translations or outputs.
ROUGE	Focuses on the recall of n-grams in the generated text compared to reference texts. It measures the overlap of n-grams, longest common subsequences, and skip-bigrams.	**Summarization & Text Generation:** Assesses the coverage and completeness of the generated summaries or responses by comparing them to reference summaries.
METEOR	Considers exact word matches, stemmed matches, synonym matches, and paraphrases between the generated and reference texts. It aligns the two texts to compute precision and recall.	**Machine Translation & Paraphrasing:** Provides a more flexible and semantically aware evaluation compared to BLEU by accounting for synonyms and different phrasings.
Exact Match	Checks whether the generated text exactly matches the expected answer. It is a binary metric where 1 indicates a perfect match and 0 indicates otherwise.	**Question Answering & Retrieval Tasks:** Useful in scenarios where precise, verbatim answers are required, ensuring that the model's output matches the reference exactly.