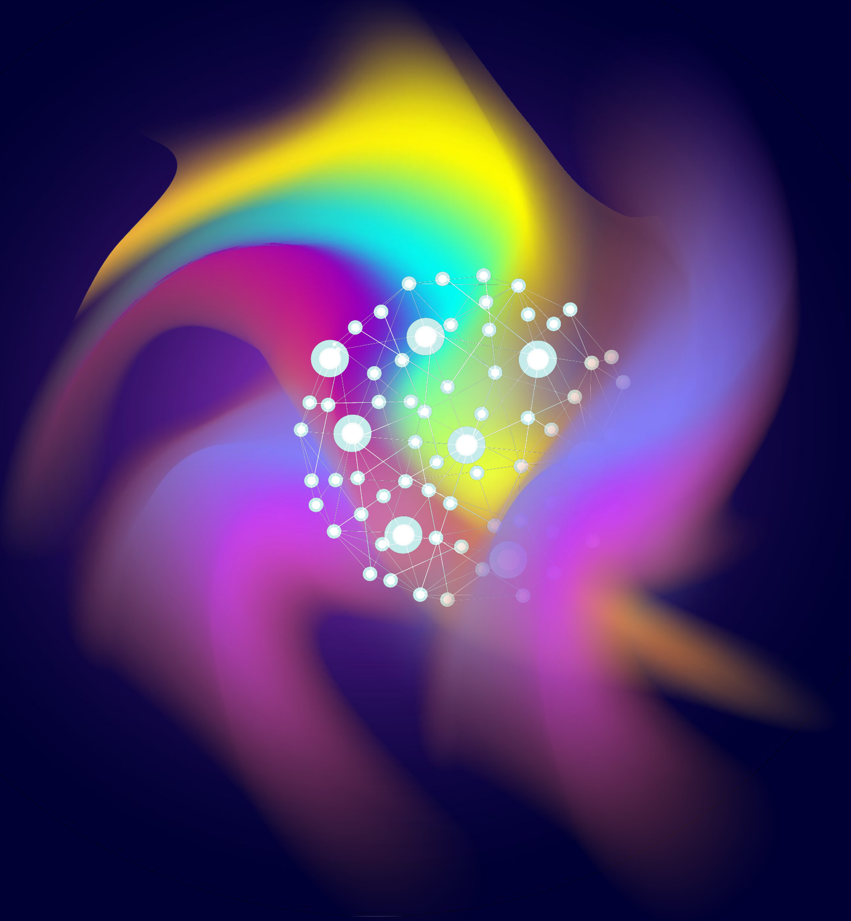
Lag-Llama

The Future of Forecasting!



Presented by: Besma Guesmi

- Software Engineer Specialising in Computer Vision
- Machine Learning Trainer

Outlines

01

LLM

02

Fine Tuning LLM

03

LLAMA

04

Time series Analysis

05

Lag-Llama

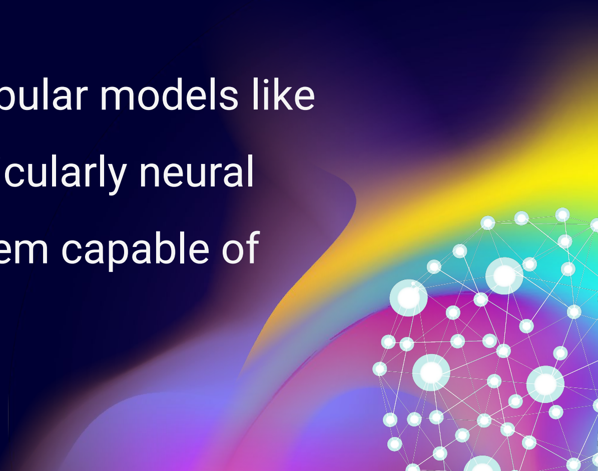
06

Let’s Practice

LLM

Large Language Model

- An LLM, is an advanced artificial intelligence algorithm designed to understand, generate, and interact with human language.
- These models are trained on huge amounts of text data, enabling them to perform a wide range of natural language processing (NLP) tasks such as text generation, translation, summarisation, and question-answering.
- LLMs, like Generative Pre-trained Transformer (GPT) – with popular models like OpenAI's Chat GPT-3.5 or 4, use deep learning techniques, particularly neural networks, to analyse and predict language patterns, making them capable of producing remarkably coherent and contextually relevant text.

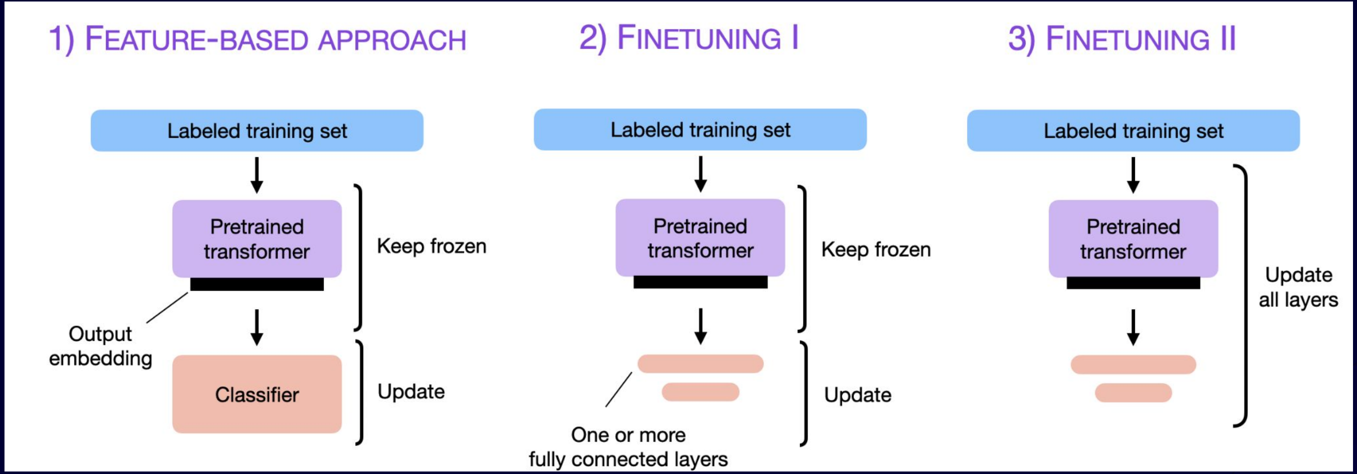


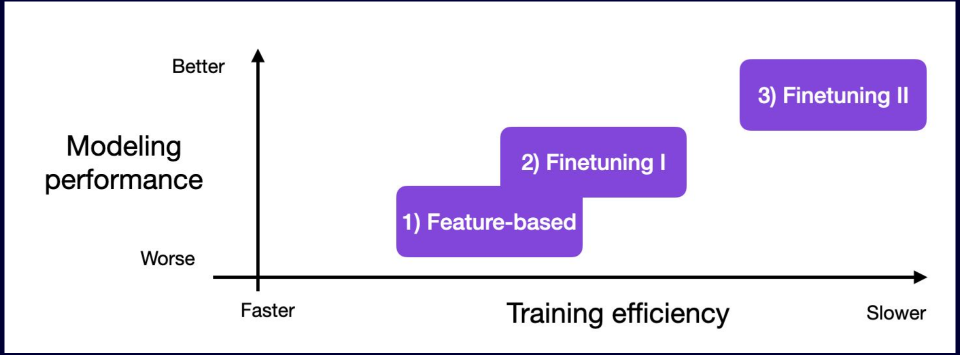
Fine Tuning LLM

- Models like GPT-2 and GPT-3, once pretrained on diverse text, demonstrate the ability for in-context learning, enabling them to handle new tasks without additional training or fine-tuning.
- In-context learning is a valuable and user-friendly method for situations where direct access to the large language model (LLM) is limited, such as when interacting with the LLM through an API or user interface.
- However, if we have access to the LLM, adapting and fine tuning it on a target task using data from a target domain usually leads to superior results.

So, how can we adapt a model to a target task?

Fine Tuning LLM





BERT params approx 110 Billion

LLaMA

Large Language Model Meta AI

➢ LLaMA is not a single model, but rather a suite of LLMs with sizes ranging from 7 billion to 65 billion parameters.

➢ **Natural language understanding**: LLaMA can understand the meaning of text, including the nuances of human language.

➢ **Natural language generation:** LLaMA can generate text that is grammatically correct and makes sense.

➢ **Translation**: LLaMA can translate text from one language to another.

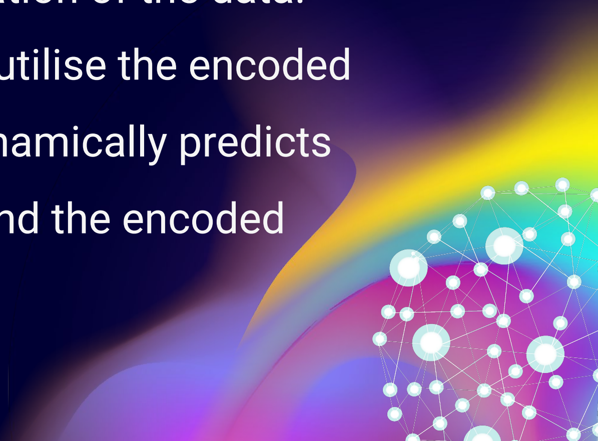
➢ **Code generation**: LLaMA can generate code in a variety of programming languages.

➢ **Question answering**: LLaMA can answer your questions in an informative way.

LLaMA

Architectural Foundation

- The foundation of Llama LLM lies in the Transformer-based encoder-decoder architecture.
    - **Encoder:** The encoder utilises multiple self-attention layers to analyze the relationships between words within an input sequence. This process captures contextual information and builds a rich internal representation of the data.
    - **Decoder:** The decoder leverages attention mechanisms to utilise the encoded representation while generating the output sequence. It dynamically predicts the next word based on the previously generated content and the encoded context.



LLaMA

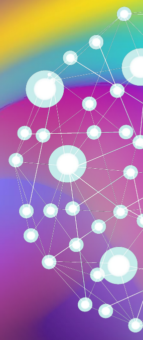
Architectural Foundation

Several architectural enhancements further amplify Llama LLM's performance and efficiency.

➢ **Massively Parallel Processing:** This technique employs multiple Graphic Processing Units) GPUs or Tensile Processing Units (TPUs) to operate on the model simultaneously, enabling faster computation and handling of large datasets.

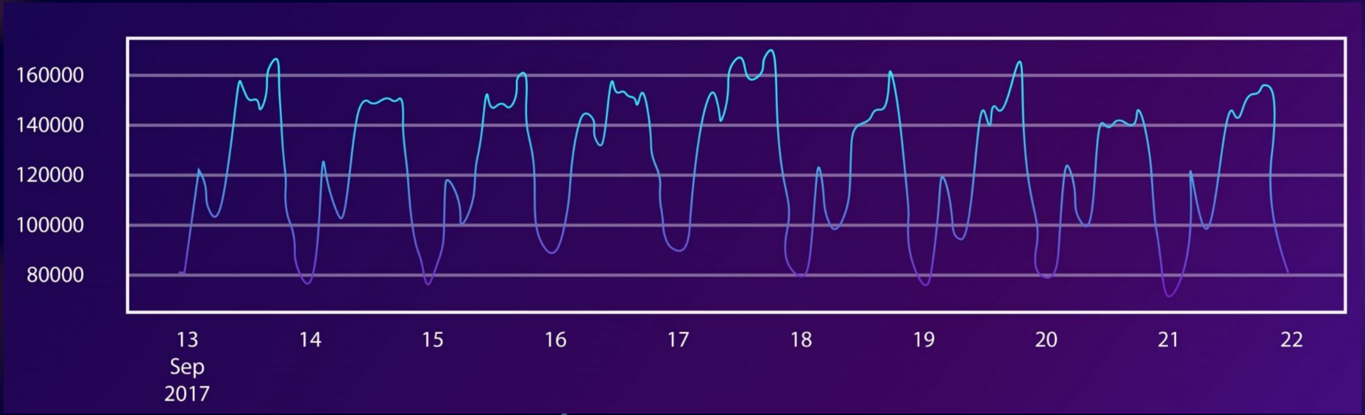
➢ **Mixed Precision Training:** This optimisation involves a combination of data types (e.g., 32-bit and 16-bit floating-point) during training, improving computational efficiency without sacrificing accuracy.

➢ **Checkpoint Ensembling:** By periodically saving checkpoints and averaging their predictions/weights, the model achieves more robust and accurate outputs.



Time Series Analysis

- ➢ A time series analysis is a way of studying a group of data points accumulated over time.
- ➢ Time series analysis is used to forecast future data based on prior data
- ➢ Statistical Models: "ARIMA, Exponential Smoothing."
- ➢ Machine Learning Models: "LSTM, RNNs, Transformer-based models (e.g., TFT, DeepAR)



Time series Forecasting

Time series forecasting models in natural language processing (NLP) often involve leveraging techniques from both time series analysis and NLP to handle sequential and temporal aspects of text data.

- ➢ Recurrent Neural Networks (RNNs)
- ➢ Long Short-Term Memory (LSTM) Networks
- ➢ Gated Recurrent Units (GRUs)
- ➢ Transformer Models
- ➢ BERT for Time Series (BERTTS)

Lag-Llama

- Lag-Llama is built for univariate (predict a single variable) probabilistic forecasting.
- Lag-Llama uses a general method for tokenizing time series data that does not rely on frequency. That way, the model can generalize well to unseen frequencies.

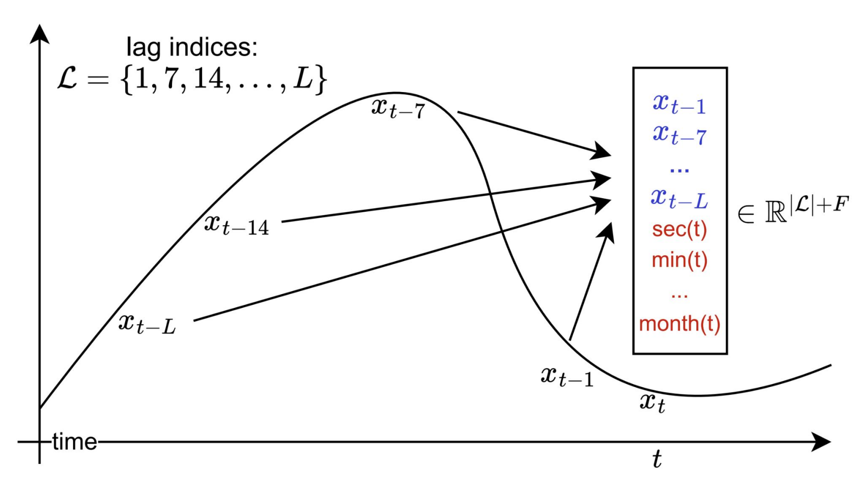
Tokenization with lag features

❖ **Tokenization Strategy**
- Lag-Llama employs a tokenization strategy for time series data.

❖ **Lagged Features**
- The strategy involves constructing lagged features using historical values.

❖ **Specified Set of Lags**
- Lag-Llama uses a predefined set of time lags (= fixed time window).

❖ **List of Appropriate Frequencies**
- The model considers a list of frequencies, including quarterly, monthly, weekly, daily, hourly, and every second.

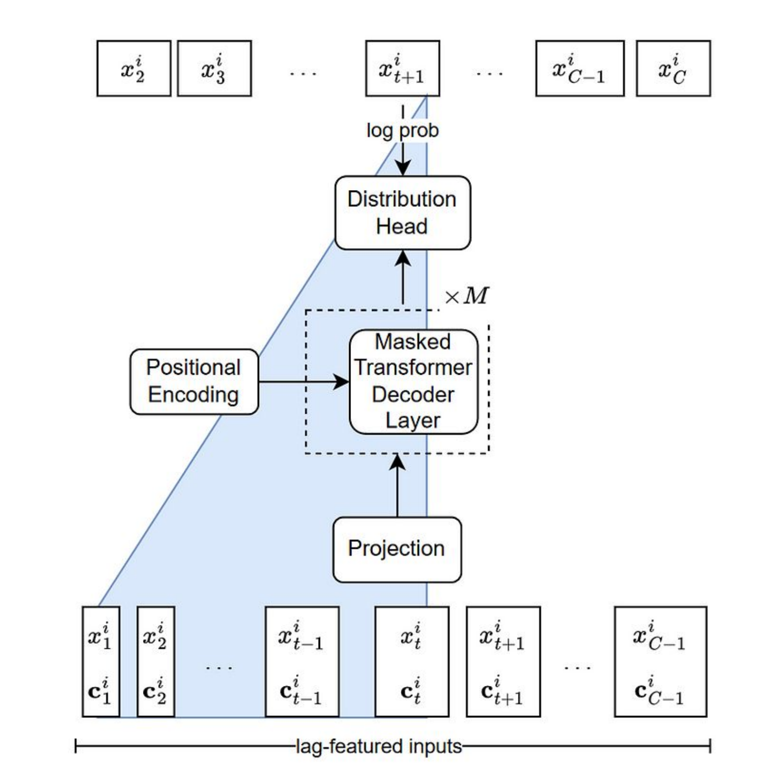


Tokenization strategy of Lag-Llama. Image from Lag-Llama: Towards Foundation Models for Probabilistic Time Series Forecasting by K. Rasul, A. Ashok, A. Williams, H. Ghonia, R. Bhagwatkar, A. Khorasani, M. Bayazi, G. Adamopoulos, R. Riachi, N. Hassen, M. Bilos, S. Garg, A. Schneider, N. Chapados, A. Drouin, V. Zantedeschi, Y. Nevmyvaka, I. Rish

This means that if we feed a dataset with a daily frequency, Lag-Llama will attempt to build features using a daily lag (t-1), a weekly lag (t-7), a monthly lag (t-30), and so on.

Lag-Llama Architecture

- Lag-Llama is a **decoder-only Transformer-based** model, and takes inspiration from the architecture of the large language model LLaMA.
- Input token: Lagged time steps + static covariante (const)
- Linear Projection layer is used to map the features to the hidden dimension within the attention mechanism in the decoder
- The attention mechanism process the input sequence and send it to the distribution head.
- The distribution head will generate the probability distribution



Architecture of Lag-Llama. Image from Lag-Llama: Towards Foundation Models for Probabilistic Time Series Forecasting by K. Rasul, A. Ashok, A. Williams, H. Ghonia, R. Bhagwatkar, A. Khorasani, M. Bayazi, G. Adamopoulos, R. Riachi, N. Hassen, M. Bilos, S. Garg, A. Schneider, N. Chapados, A. Drouin, V. Zantedeschi, Y. Nevmyvaka, I. Rish

Let’s Practice!

Thank you!

Any Questions?
bessmagsm@gmail.com
LinkedIn: Besma Guesmi