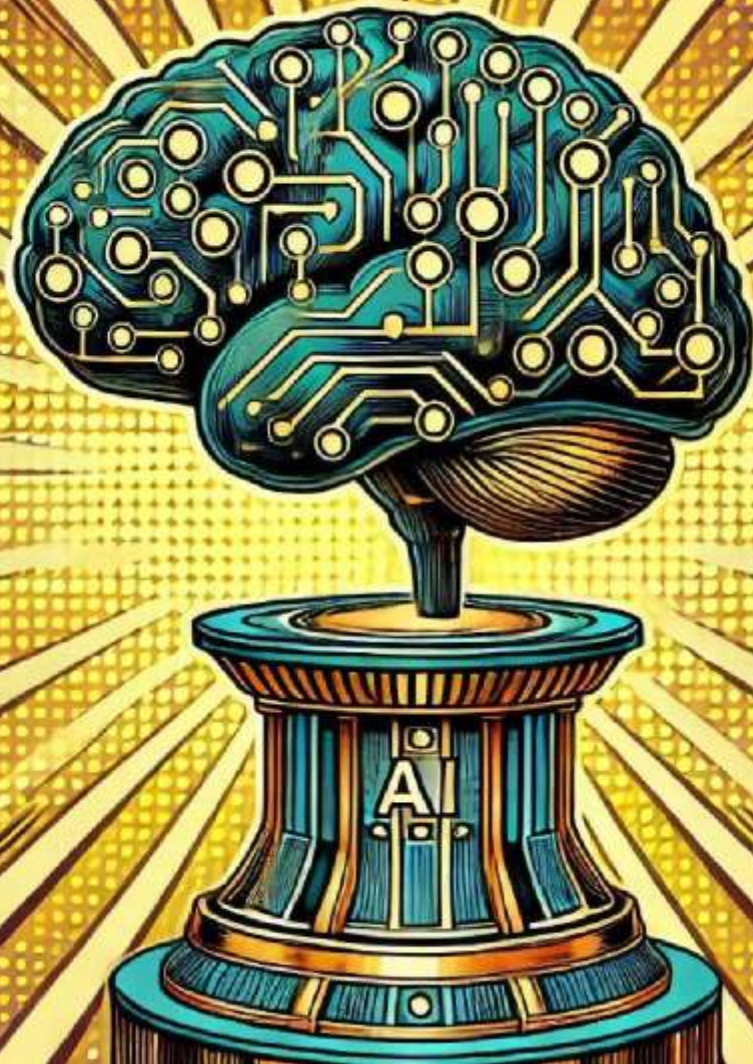


Natural Language Processing

LARGE LANGUAGE MODELS





NLP: 12/06/2025

Báo cáo môn học: Xử lý ngôn ngữ tự nhiên

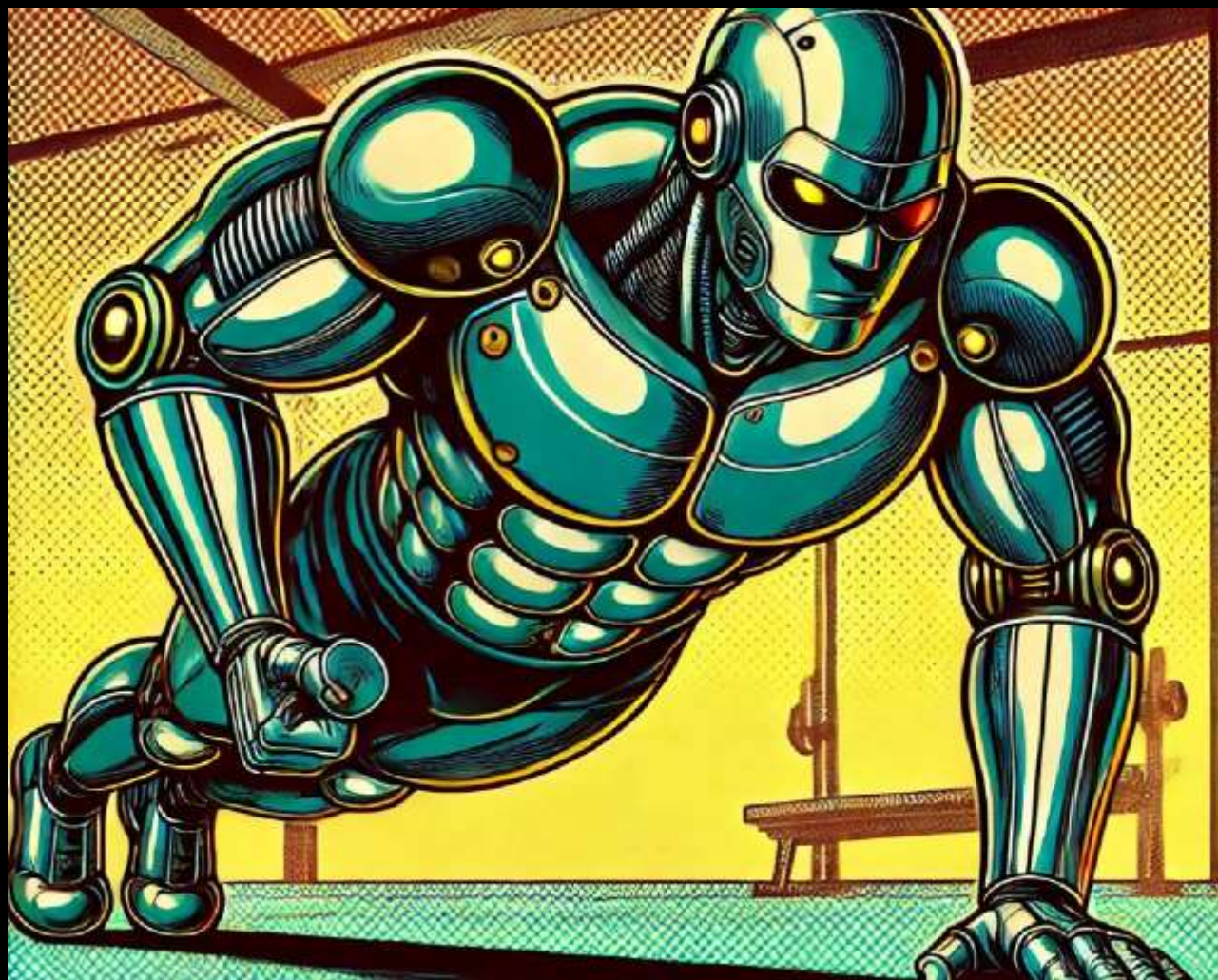
Tên các thành viên nhóm

- Cao Xuân Nhật Huy – 22DH111204
- Nguyễn Thế Kiên – 22DH111782
- Trần Tấn Kiệt – 22DH111836
- Trần Hà Minh Thư – 22DH113653

Giảng viên hướng dẫn:

- TS. Trần Khải Thiện

TRAINING LLMs



- Huấn luyện một mô hình có hàng tỷ tham số từ đầu sẽ tốn hàng chục cho đến hàng trăm triệu đô \$
- Thay vào đó, tận dụng **Transfer Learning**
- Lấy một mô hình đã được huấn luyện trước làm nền tảng và sử dụng dữ liệu huấn luyện bổ sung để tinh chỉnh nó cho một nhiệm vụ cụ thể

Giới thiệu dự án

Một bài toán kinh doanh

Cho một mô tả sản phẩm và dự đoán giá của nó

- Đối với một sàn thương mại điện tử cần ước tính giá hàng hóa
- Thông thường thường dùng mô hình Hồi quy để dự đoán giá, nhưng có lý do chính

Có thể huấn luyện LLM và đánh giá một cách minh bạch. Điều này nghĩa là chúng ta có thể đối đầu với GPT-4o



Tìm kiếm tập dữ liệu



Data riêng biệt của bạn



Kaggle



HuggingFace datasets




Data tổng hợp



Các công ty chuyên biệt như Scale.com

HuggingFace là một kho tàng dữ liệu

 **Datasets:**  McAuley-Lab / **Amazon-Reviews-2023**   like 58

Languages:  English Size: 10B<n<100B Tags: recommendation reviews

 **Dataset card**  Files  Community **7**

Dataset Viewer

 View in Dataset Viewer

The viewer is disabled because this dataset repo requires arbitrary Python code execution. Please consider removing the loading_script and relying on automated data support (you can use convert to parquet from the datasets library). If this is not possible, please open a discussion for direct help.

Amazon Reviews 2023


Please also visit amazon-reviews-2023.github.io/ for more details, loading scripts, and preprocessed benchmark files.

Downloads last month **24,354**

 Edit dataset card

⋮

 Models trained or fine-tuned on McAuley...

 hyp1231/blair-roberta-base

 Feature Extraction • Upd... •  3.57k •  1

Đào sâu vào dữ liệu



Investigate



Parse



Visualize



Assess Data Quality



Curate



Save

Đánh giá hiệu suất

Từ bảng so sánh Giá dự báo và Giá thực tế



Chỉ số theo hướng mô hình hoặc Chỉ số kỹ thuật

Training loss Validation

loss

Root Mean Squared Log Error (RMSLE)



Chỉ số theo hướng kinh doanh hoặc Chỉ số hiệu quả

Average price difference

% price difference

% estimates that are "good"

Chiến lược gồm 5 bước

Để việc lựa chọn, huấn luyện và áp dụng một LLM vào bài toán thương mại



Understand



Prepare



Select



Customize

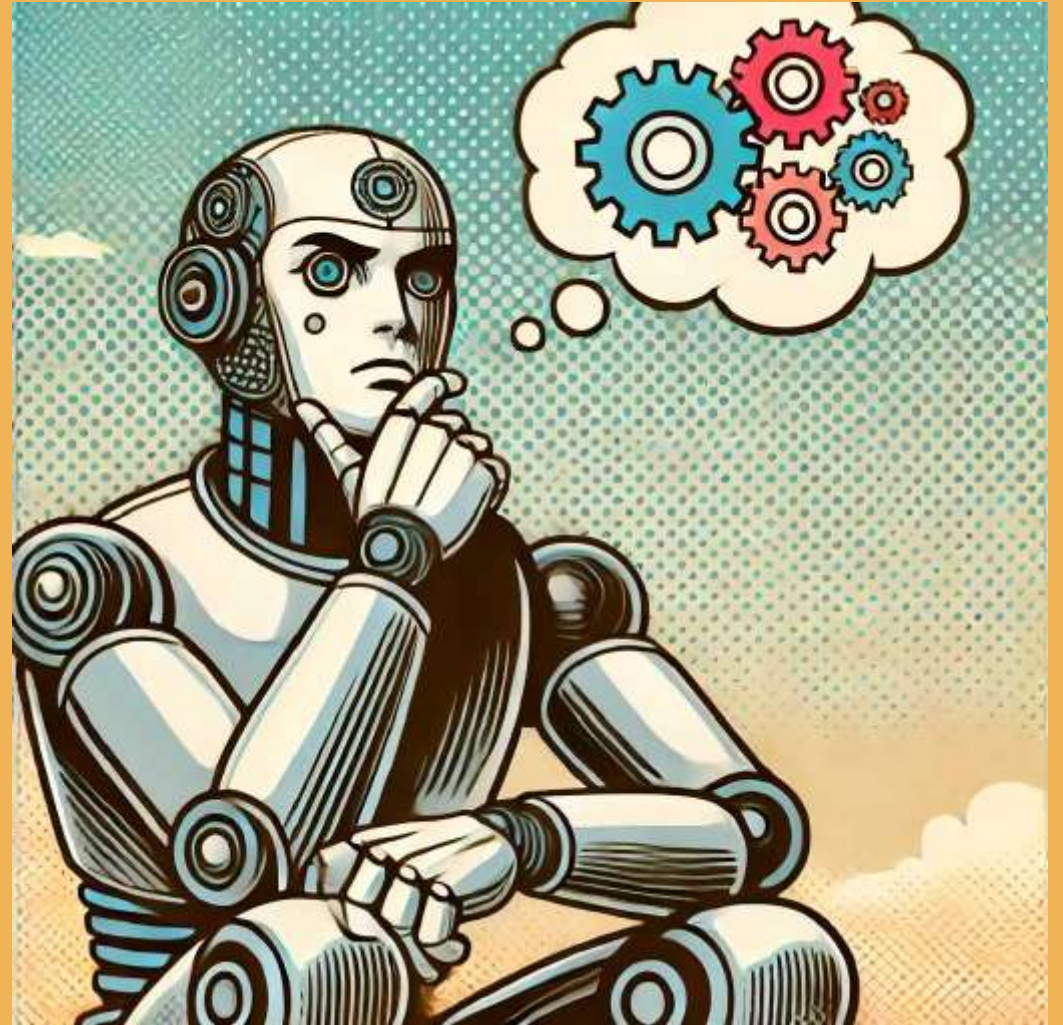


Productionize

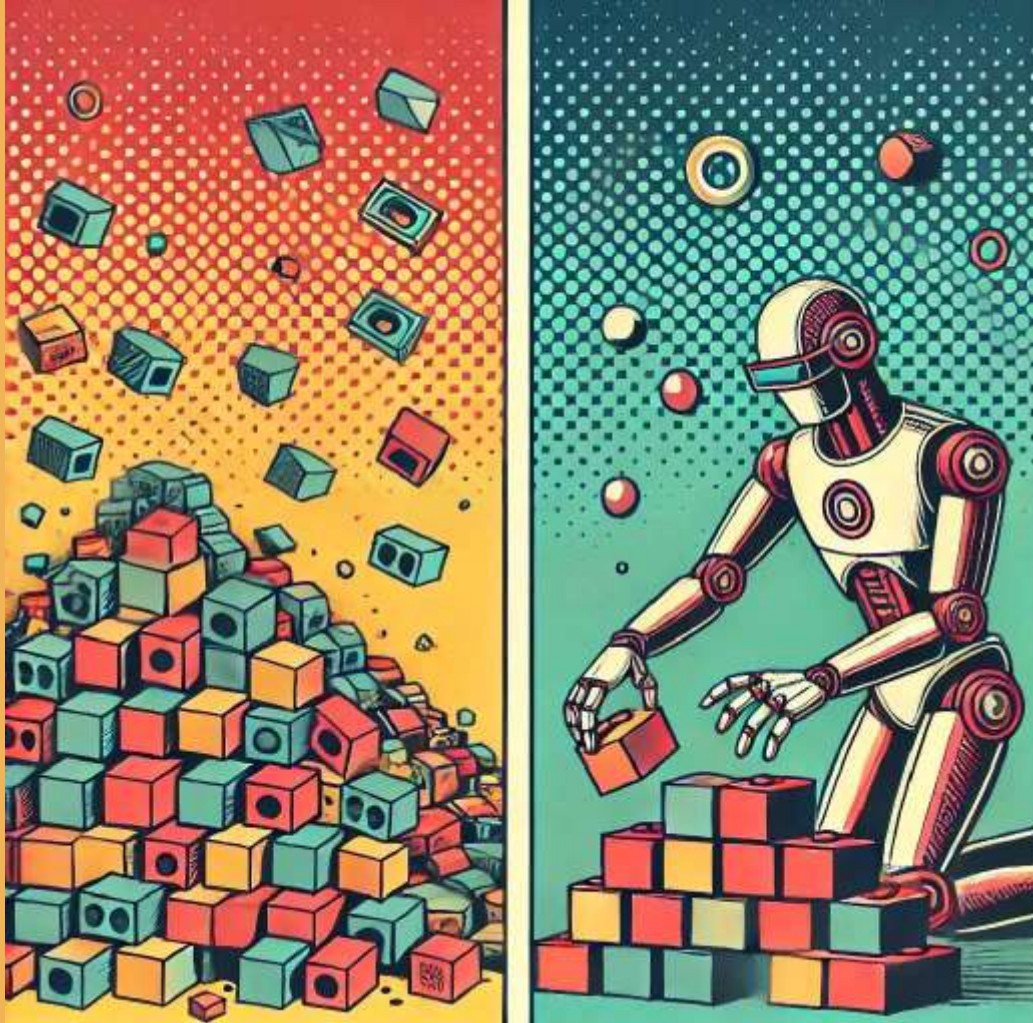
1. Understand

Các hoạt động:

- Thu thập yêu cầu kinh doanh cho
- Xác định tiêu chí hiệu suất
Đặc biệt là các chỉ số tập trung vào kinh doanh
- Tìm hiểu dữ liệu: số lượng, chất lượng, định dạng
- Xác định các yêu cầu phi chức năng
Ràng buộc về chi phí, khả năng mở rộng, độ trễ
Ngân sách R&D/phát triển và lộ trình triển khai



2. Prepare



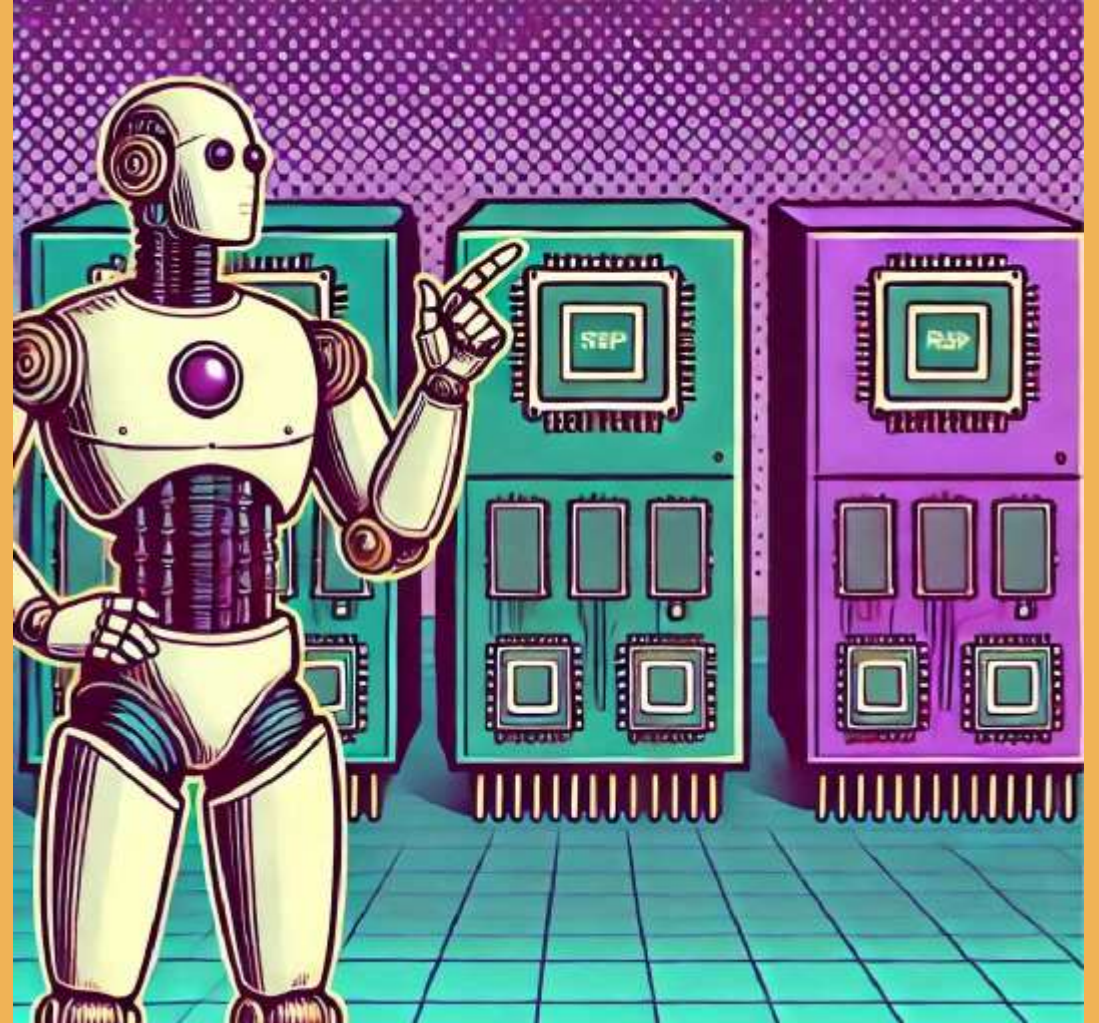
Các hoạt động:

- Nghiên cứu giải pháp hiện có / không sử dụng LLM
- So sánh các LLM liên quan
 - Những yếu tố cơ bản, bao gồm độ dài ngữ cảnh, giá cả và giấy phép
 - Điểm chuẩn, Bảng xếp hạng và Arena
 - Điểm chuyên môn cho nhiệm vụ cụ thể
- Chọn dữ liệu: làm sạch, tiền xử lý và phân chia

3. Select

Các hoạt động:

- Lựa chọn LLM(s)
- Thực nghiệm
- Huấn luyện và thực nghiệm với dữ liệu đã chọn



4. Customize

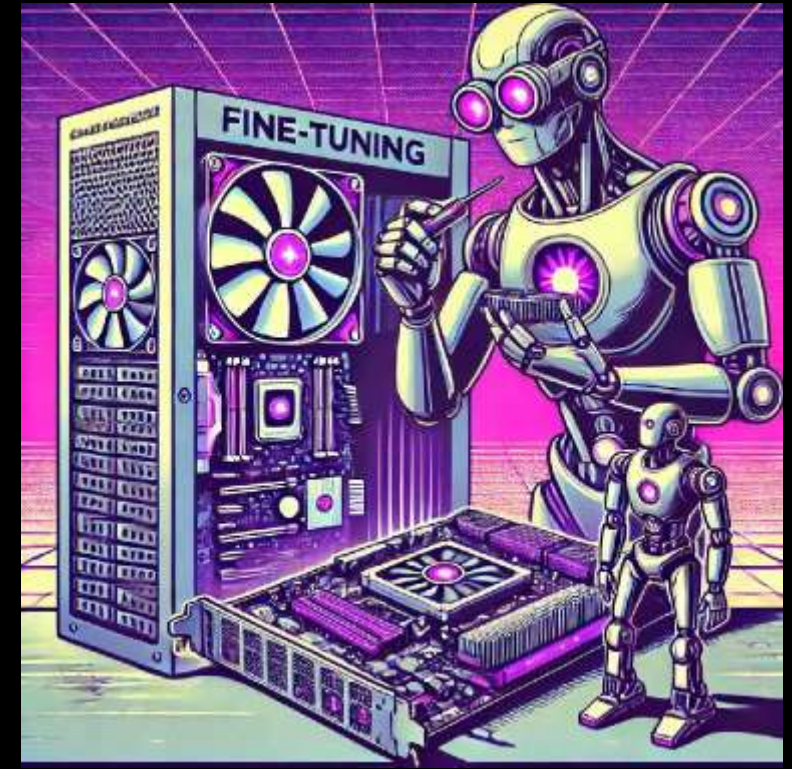
Three techniques to optimize the performance of the model



Prompting
multi-shot, chaining and
tools



RAG

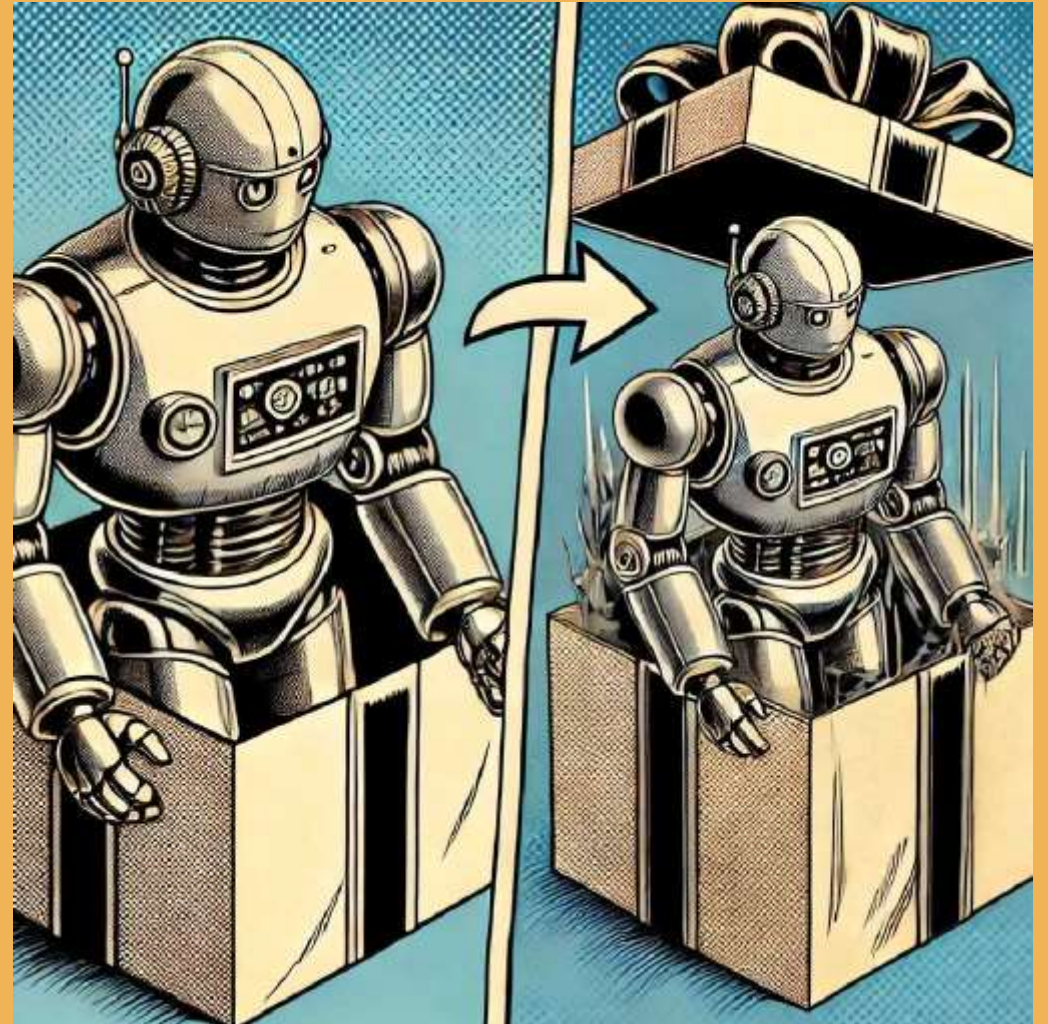


Fine-tuning

5. Productionize

Hoạt động:

- Xác định API giữa mô hình và nền tảng
- Xác định kiến trúc lưu trữ và triển khai mô hình
- Giải quyết các vấn đề về khả năng mở rộng, giám sát, bảo mật và tuân thủ
- Đo lường các chỉ số tập trung vào doanh nghiệp đã xác định ở bước 1
- Liên tục huấn luyện lại và đo lường hiệu suất



Tầm quan trọng của một mô hình cơ sở (Baseline)

- Bắt đầu với giải pháp rẻ và đơn giản
 - Tạo mốc so sánh để cải tiến
 - Một LLM có thể không phải là giải pháp phù hợp
-



Traditional ML models



Feature engineering & Linear Regression



Bag of Words & Linear Regression



word2vec & Linear Regression



word2vec & Random Forest



word2vec & SVR

Sai số tuyệt đối trung bình từ các mô hình





To the
Frontier!

Sai số tuyệt đối trung bình từ các mô hình



Ba giai đoạn

Fine-Tuning OpenAI



Tạo tập huấn luyện ở định dạng jsonl và tải lên OpenAI



Bắt đầu training - training loss và validation loss nên giảm dần



Đánh giá kết quả, điều chỉnh và lặp lại quy trình

Chuẩn bị dữ liệu

OpenAI yêu cầu dữ liệu định dạng JSONL

Các dòng JSON chứa tin nhắn theo định dạng prompt thông thường

Sai số tuyệt đối trung bình từ các mô hình



Tại sao mô hình lại tệ đi

Mục tiêu chính của Fine-Tuning dành cho mô hình Frontier

- 1 | Thiết lập phong cách hoặc giọng điệu theo cách mà việc dùng prompt thông thường không thể đạt
- 2 | Cải thiện độ tin cậy khi tạo ra một loại đầu ra nhất định
- 3 | Sửa lỗi khi mô hình không làm theo các prompt phức tạp
- 4 | Xử lý các trường hợp đặc biệt
- 5 | Thực hiện một kỹ năng hoặc nhiệm vụ mới mà khó có thể diễn đạt đầy đủ chỉ bằng prompt

Tại sao trường hợp này không hưởng lợi từ việc Fine Tuning

- Trường hợp này và kiểu đầu ra có thể xác định rõ ràng thông qua prompt
- Mô hình có thể tận dụng kiến thức rộng lớn từ giai đoạn pre-train, việc cung cấp vài trăm dòng dữ liệu không mang lại nhiều giá trị

Giải thích tổng quát về LoRa

Sử dụng Llama 3.1 với 8B tham số - quá lớn để huấn luyện trên GPU



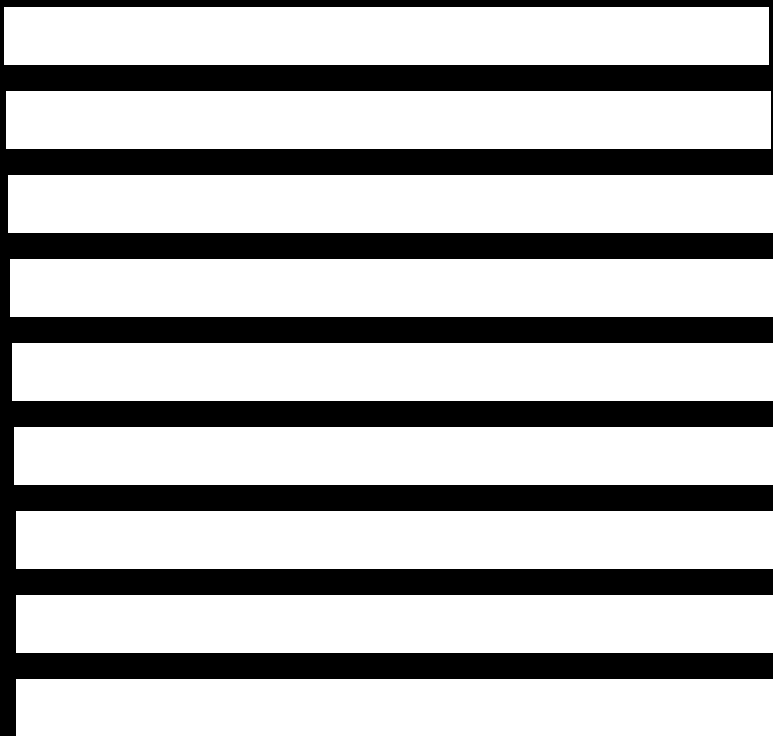
- Kiến trúc Llama 3.1 8B bao gồm 32 nhóm module xếp chồng lên nhau, gọi là 'Llama Decoder Layers'

Mỗi layer có các lớp self-

- attention, các lớp multi-layer perceptron, hàm kích hoạt SiLU và layer norm
- Các tham số này chiếm 32GB bộ nhớ

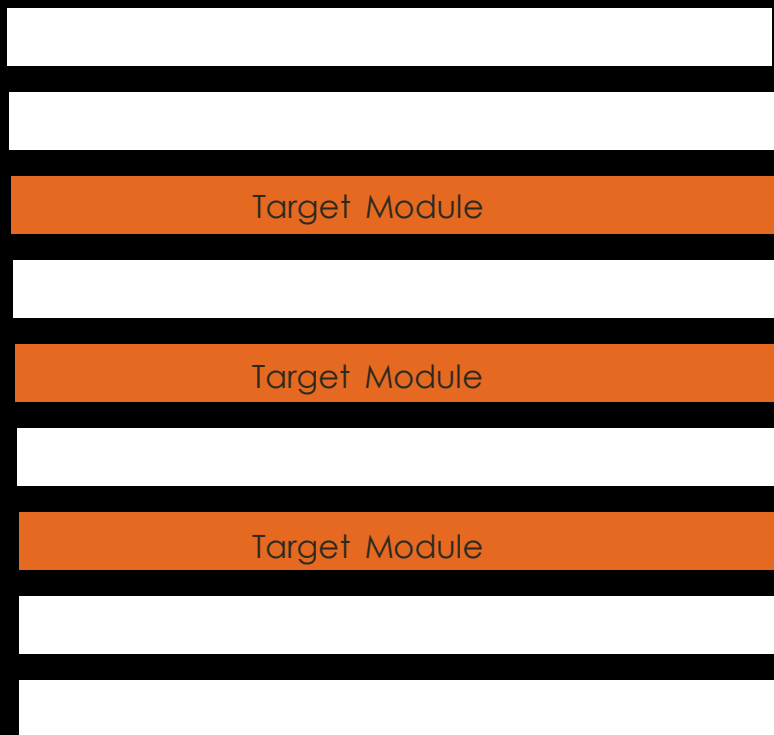
Giải thích tổng quát về LoRa

Bước 1: Đóng băng các trọng số - sẽ không tối ưu hóa chúng



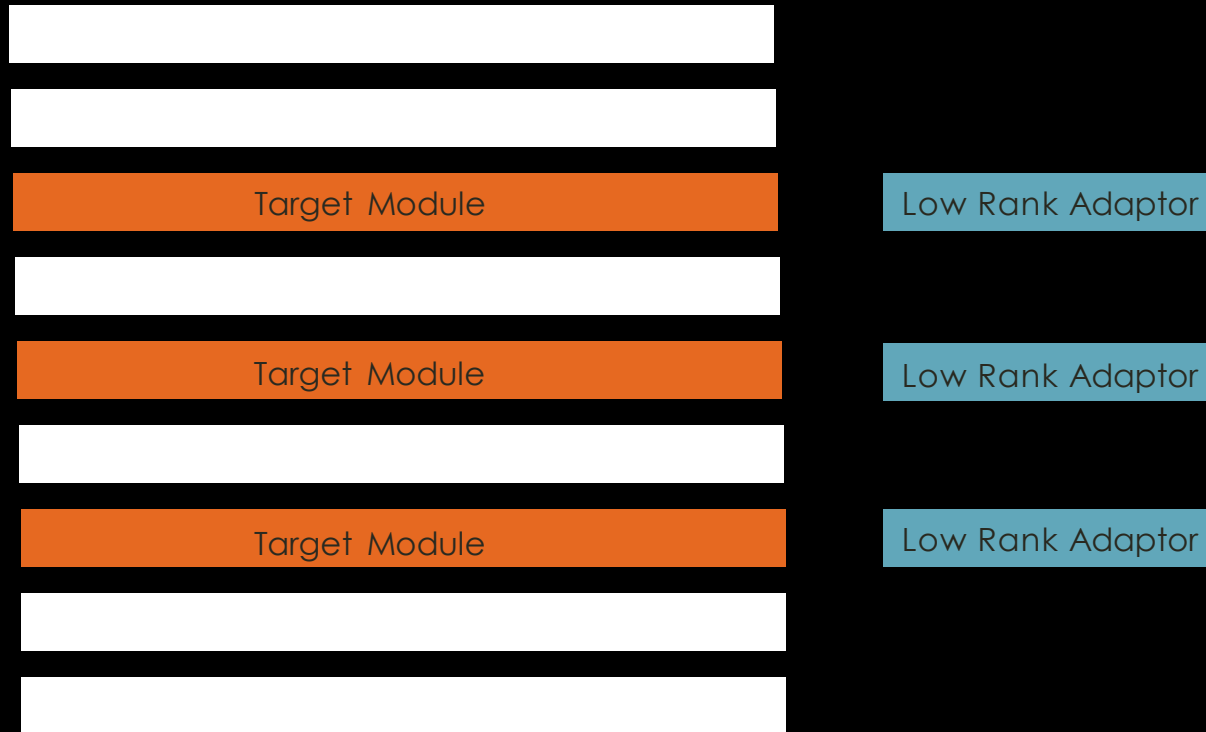
Giải thích tổng quát về LoRa

Bước 2: Chọn một số layers để nhắm mục tiêu, gọi là "Target Modules"



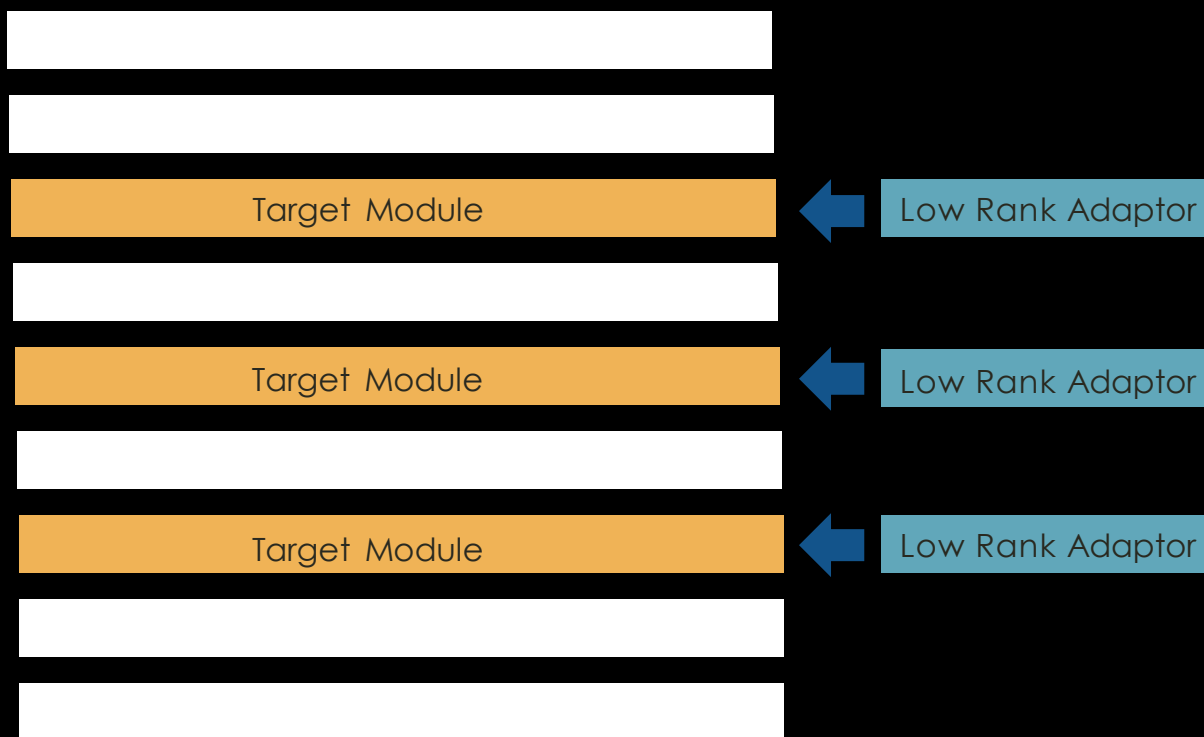
Giải thích tổng quát về LoRa

Bước 3: Tạo các ma trận "adaptor" mới với chiều thấp hơn, ít tham số hơn



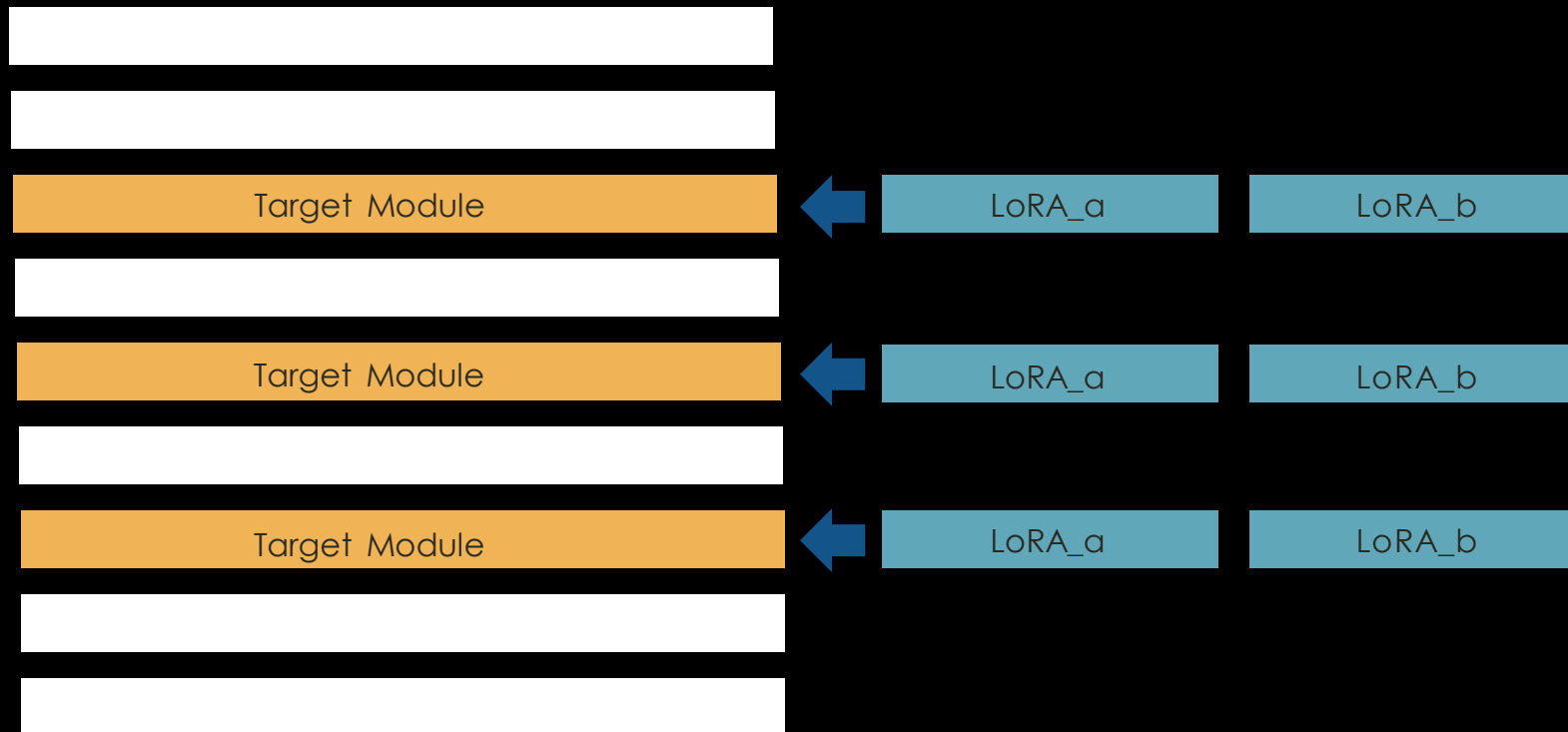
Giải thích tổng quát về LoRa

Bước 4: Áp dụng những adaptors vào Target Modules để điều chỉnh chúng – và những lớp này sẽ được huấn luyện



Giải thích tổng quát về LoRa

Để chính xác hơn: có 2 ma trận LoRa được áp dụng



Công thức ma trận Adaptor

Cho ma trận gốc $W \in R^{d \times k}$

LoRa phân rã sự thay đổi Ma trận ΔW thành tích của hai ma trận nhỏ, $\Delta W = A \cdot B$ với:

$A \in R^{d \times r}$ với giá trị ngẫu nhiên (phân phối Gaussian)

$B \in R^{r \times k}$ khởi tạo bằng 0

Trong đó:

d : Chiều đầu vào (vd: d_{model} trong Transformer).

k : Chiều đầu ra (vd: d_{head} với Self-Attention).

r : Rank của phép phân rã ($r \ll \min(d, k)$, thường chọn 8, 16).

Ma trận hiệu dụng sau khi áp dụng LoRA:

$$W_{\text{eff}} = W + \frac{\alpha}{r} \cdot \Delta W = W + \frac{\alpha}{r} \cdot AB$$

α : Hệ số tỉ lệ (thường đặt $\alpha = r$)

Quá trình training

Forward Pass

Đầu ra y được tính bằng:

$$y = x \cdot W_{\text{eff}} = x \cdot \left(W + \frac{a}{r} \cdot AB \right)$$

x : Input vector (batch size \times sequence length \times d).

Backward Pass (Cập nhật A , B)

Gradient của loss τ được truyền ngược như sau:

$$\nabla_A \tau = (\nabla W_{\text{eff}} \tau) \cdot B^T \cdot \frac{a}{r}$$

$$\nabla_B \tau = A^T \cdot (\nabla W_{\text{eff}} \tau) \cdot \frac{a}{r}$$

Chỉ **A** và **B** được cập nhật, W giữ nguyên

Công thức số tham số

Số lượng tham số cần cập nhật trong LoRA:

$$\text{Params}_{\text{LoRA}} = (d \times r) + (r \times k) = r(d+k)$$

So với full fine-tuning ($d \times k d \times k$), LoRA giảm:

$$\text{Tỉ lệ tiết kiệm} = \frac{r(d+k)}{d \times k} \approx \frac{r}{\min(d, k)} \text{ vì } r \ll d, k$$

Ví dụ:

Với $W \in \mathbb{R}^{4096 \times 4096}$, $r=8$:

$$\text{Params}_{\text{LoRA}} = 8 \times (4096 + 4096) = 65,536 \text{ (chỉ 0.4\% so với 16.8M của } W \text{)}.$$

Ba siêu tham số cần thiết

For LoRA Fine-Tuning



r

The rank, or how many dimensions in the low-rank matrices

RULE OF THUMB:

Start with 8, then double to 16, then 32, until diminishing returns



Alpha

A scaling factor that multiplies the lower rank matrices

RULE OF THUMB:

Twice the value of r



Target Modules

Which layers of the neural network are adapted

RULE OF THUMB:

Target the attention head layers

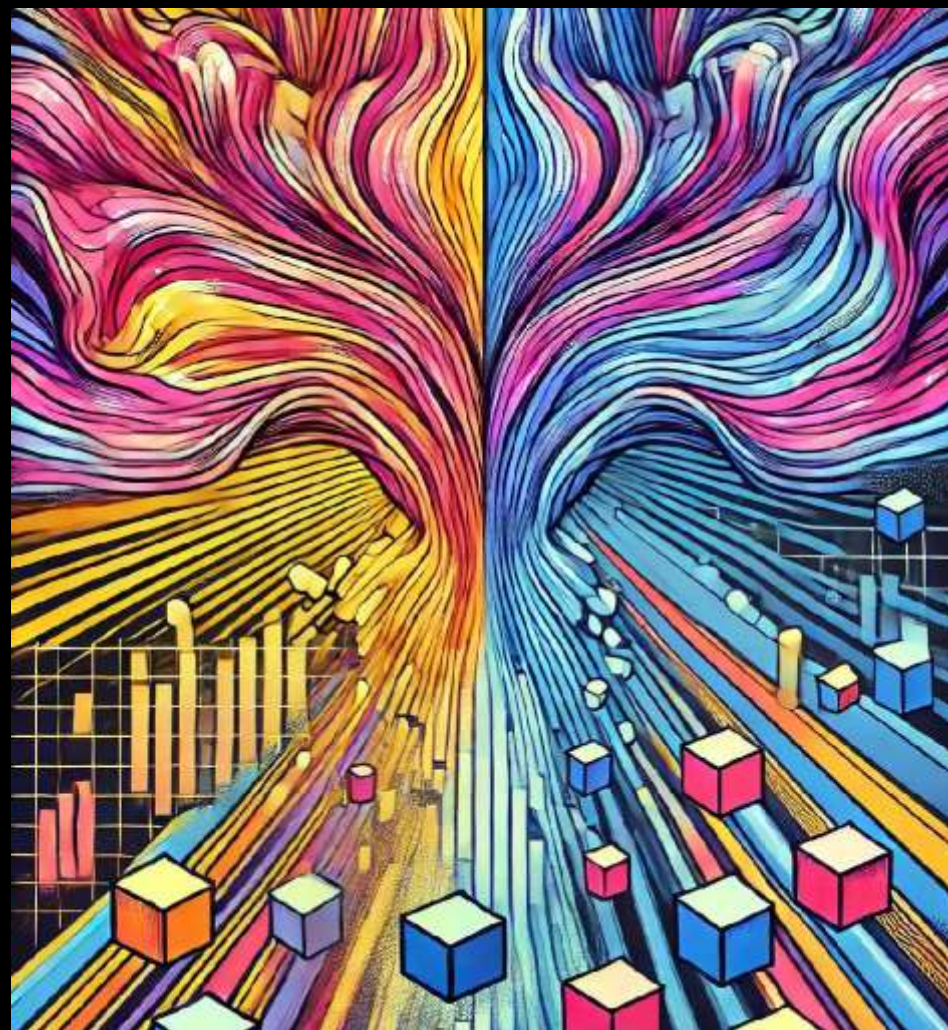
Quantization - the Q in QLoRA

Ngay cả biến thể 8B cũng rất lớn

- $8 \text{ tỷ} * 32 \text{ bits} = 32\text{GB}$
- Ý tưởng: giữ nguyên số lượng trọng số nhưng giảm độ chính xác của chúng
- Hiệu suất mô hình kém hơn, nhưng tác động một cách đáng ngạc nhiên
- Giảm xuống 8 bits, hoặc xuống 4 bits

Note 1: 4 bits được hiểu float, không phải int

Note 2: ma trận adaptor vẫn là 32 bit



Công thức trong QLoRA (Quantized LoRA)

QLoRA thêm lượng tử hóa 4-bit cho W :

Lượng tử hóa:

$$W_{4\text{bit}} = \text{quantize}(W)$$

Giải lượng tử khi tính toán:

$$W_{\text{dequant}} = \text{dequantize}(W_{4\text{bit}})$$

Forward Pass:

$$W_{\text{eff}} = W_{\text{dequant}} + \frac{a}{r} AB$$

Sau training, có thể gộp AB vào W để tăng tốc inference:

$$W_{\text{merged}} = W + \frac{a}{r} AB$$

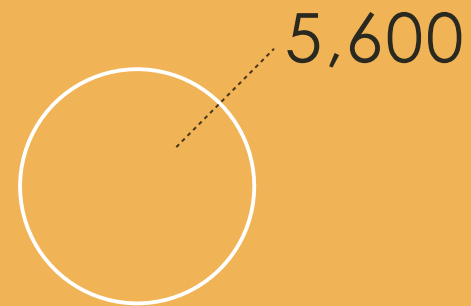
Size of Weights in MB



Llama 3.1 8B



Quantized to 8 bit



Quantized to 4 bit



QLoRA with r=32

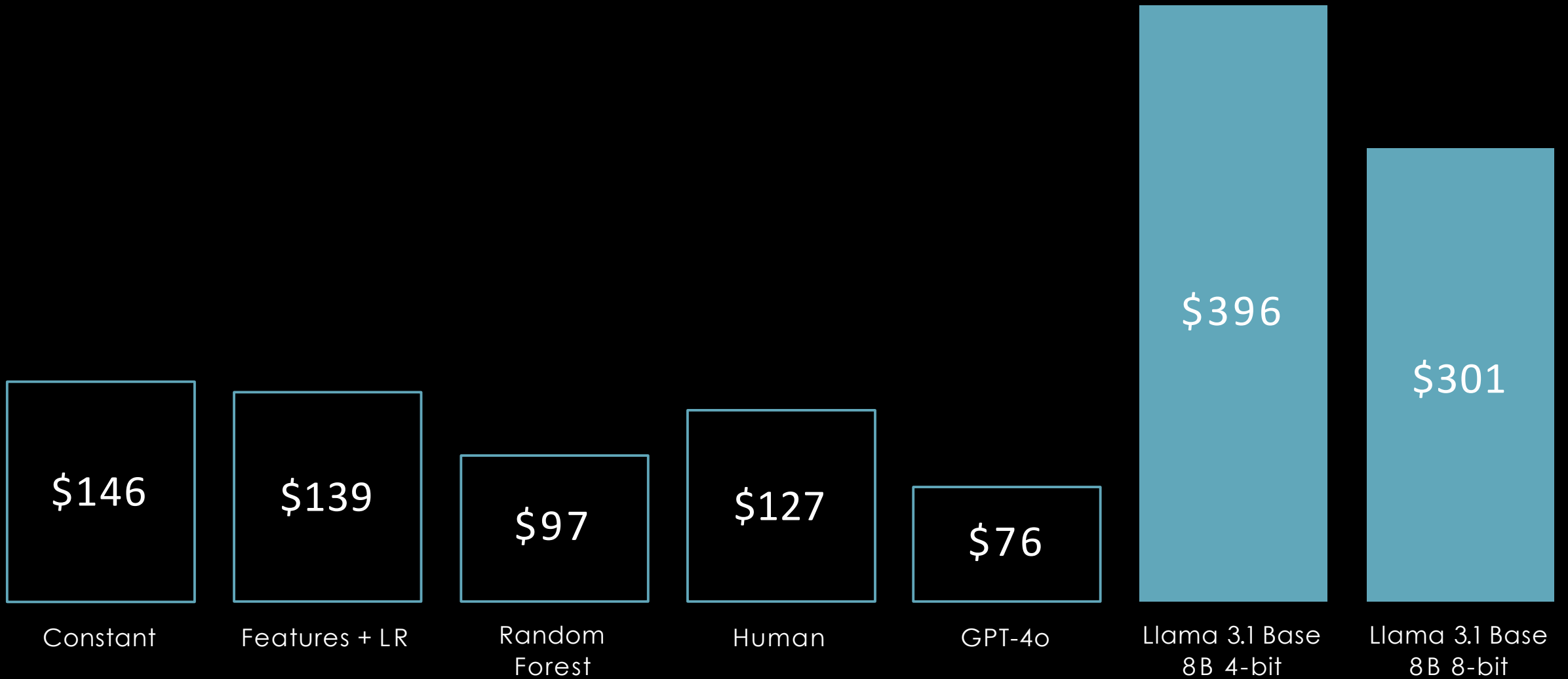
Chọn mô hình nào?

Chọn mô hình dựa trên

- Số lượng parameters
- Llama vs Qwen vs Phi vs Gemma
- Base hoặc Instruct variants



Sai số tuyệt đối trung bình từ các mô hình



5 tham số Hyper-parameters cho QLoRA...



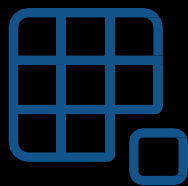
Target Modules



r



Alpha



Quantization



Dropout

... và 5 Hyper-parameters cho quá trình Training



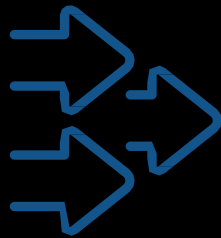
Epochs



Batch Size



Learning Rate



Gradient Accumulation



Optimizer

4 bước quá trình Training



Forward pass

Predict the next token in training data



Loss calculation

How different was it to the true next token



Backward pass

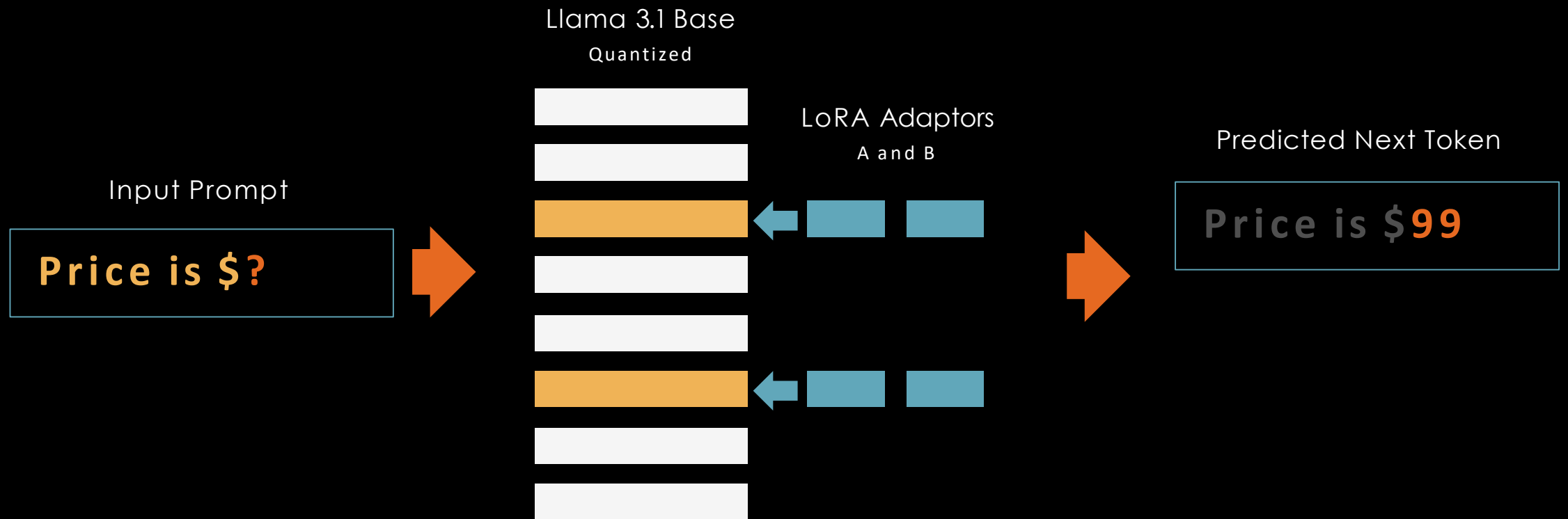
How much should we tweak parameters to do better next time (the "gradients")



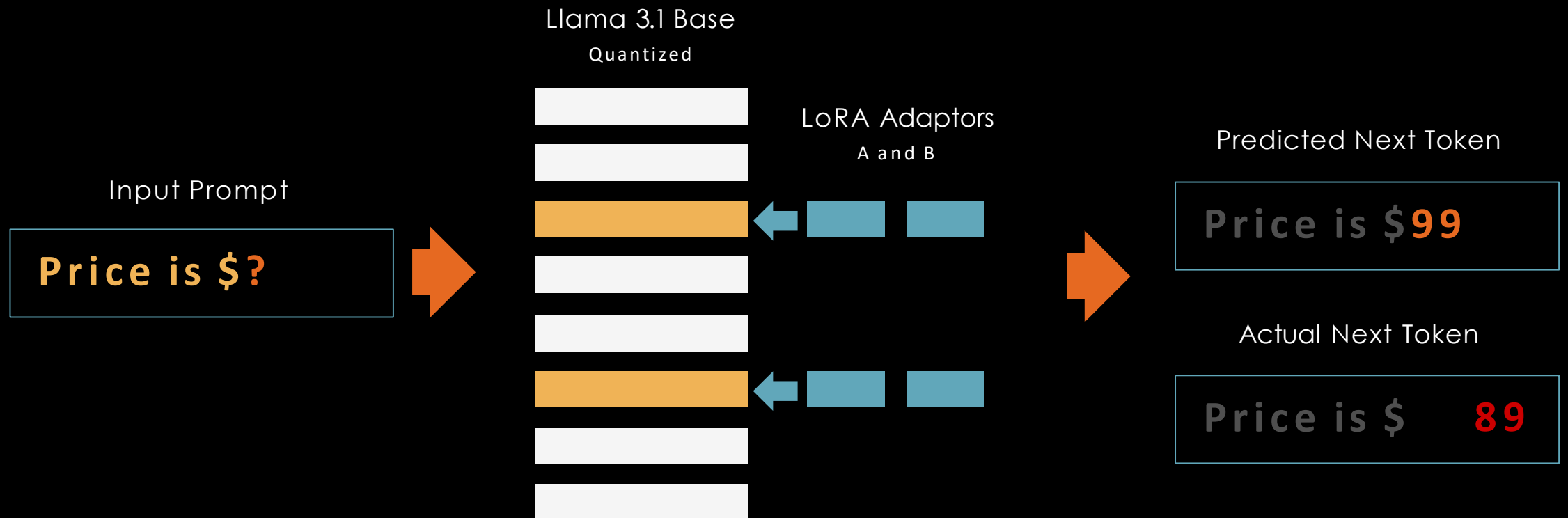
Optimization

Update parameters a tiny step to do better next time

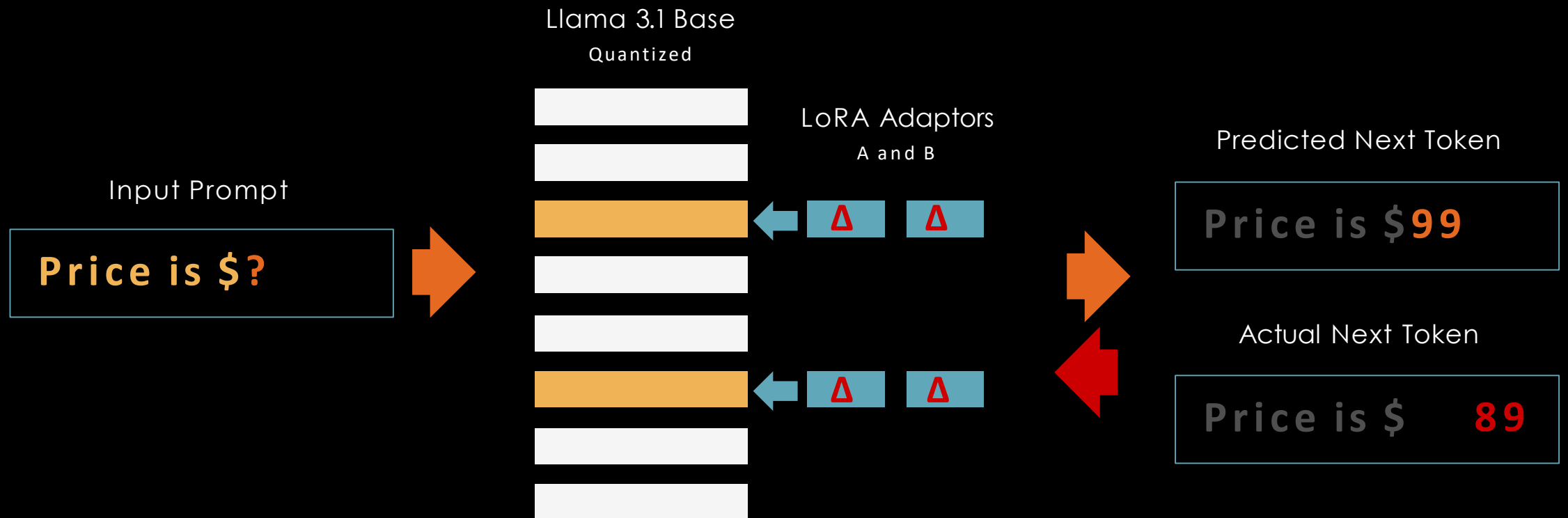
1. The Forward Pass



2. The Loss Calculation

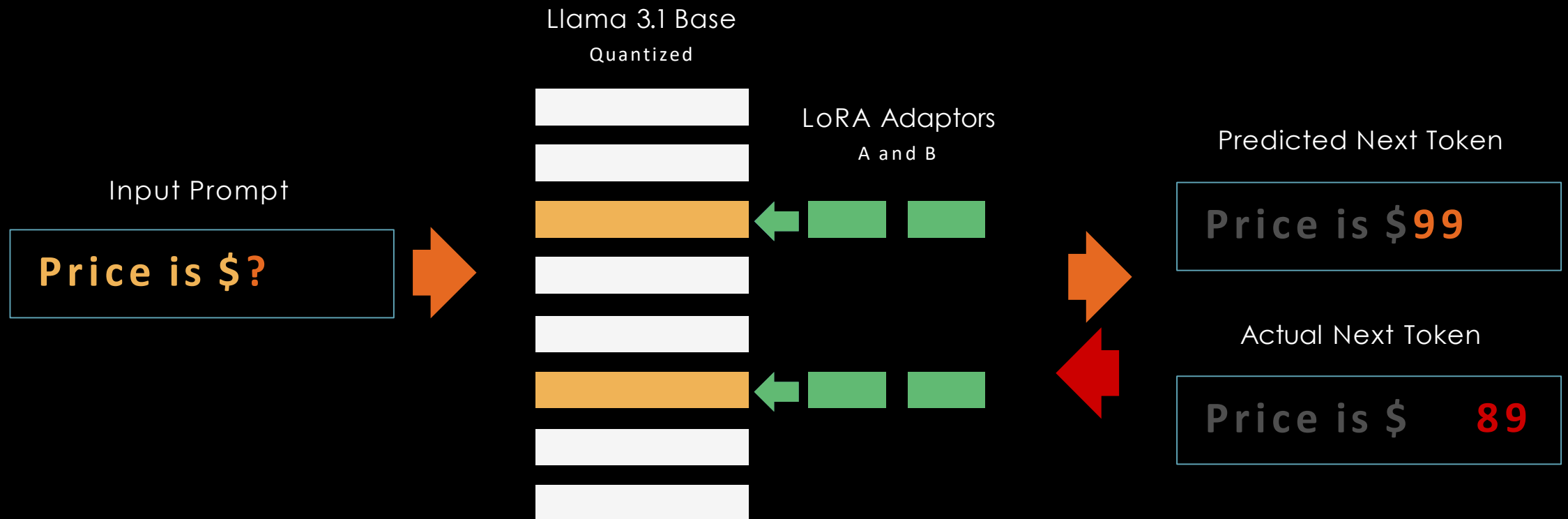


3. The Backward Pass ("Backprop")



4. Optimization

Shift weights a tiny amount (the Learning Rate) for a slightly higher chance of predicting the right token next time



Kết quả!

