

Tips for Serving Language Model Adapters at Scale



Christian Johnson

MLOps Engineer



Agenda

1. Personal Introduction
2. Presentation
 - a. Overview of Language Model Adapters
 - b. Summary of MLOps Challenges
 - c. Tips for Serving
 - d. Tips for Model Management
 - e. Additional Resources
3. Q&A



Personal Introduction

- Background in Computational Linguistics
- Worked as Data Analyst / Cloud Solutions Architect
- Recent experience working heavily with Language Models for various use-cases, necessitating attention to scalability and automation



Christian Johnson

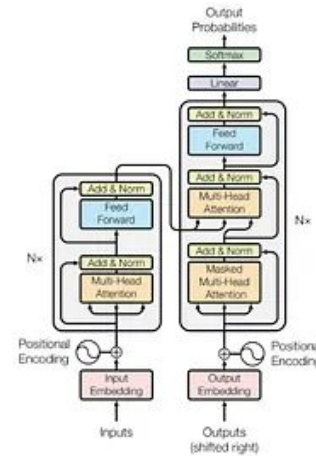
MLOps Engineer



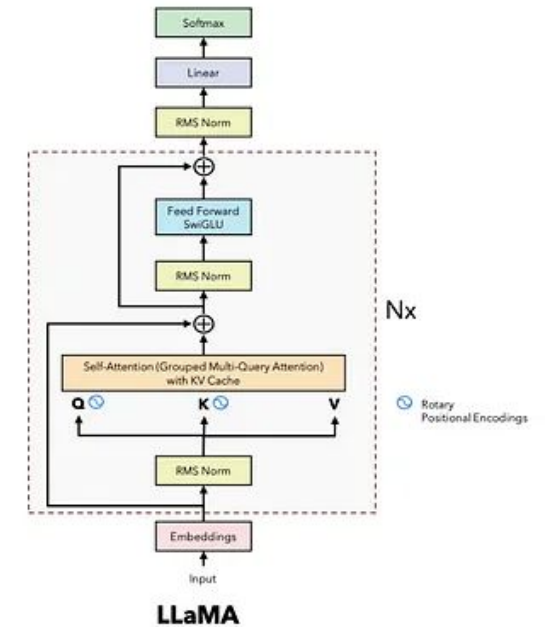
Overview of Language Model Adapters

- What is a Language Model?
- What is a Language Model Adapter?
- For what use-cases are Language Model Adapters relevant?

Transformer vs LLaMA

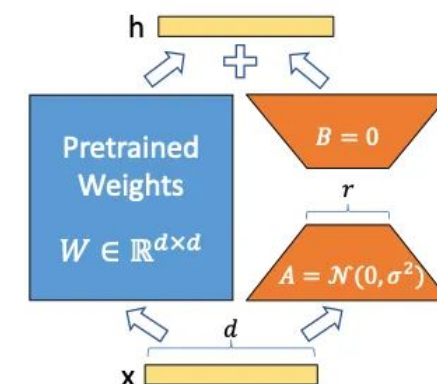
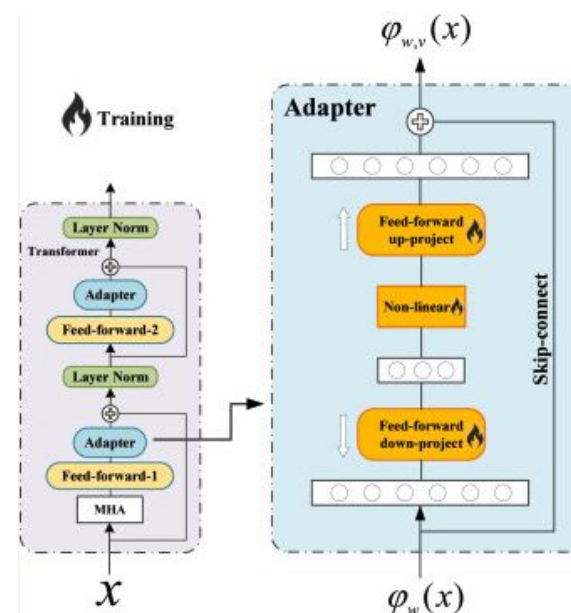


Transformer
("Attention is all you need")



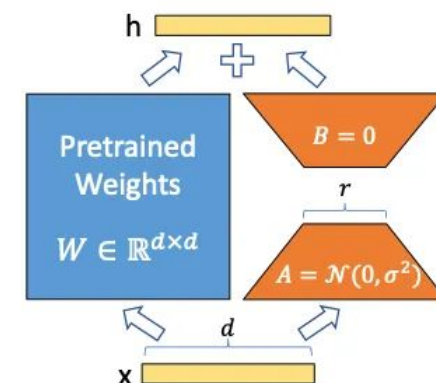
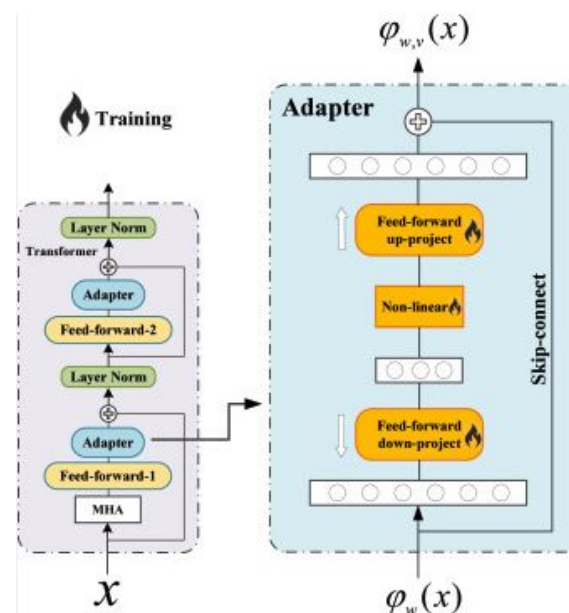
Overview of Language Model Adapters

- What is a Language Model?
 - Probabilistic model of word sequences; often used for text generation.
 - Often refers to Transformer architectures which leverage self-attention.
- What is a Language Model Adapter?
 - Additional trainable parameters added to linear layers.

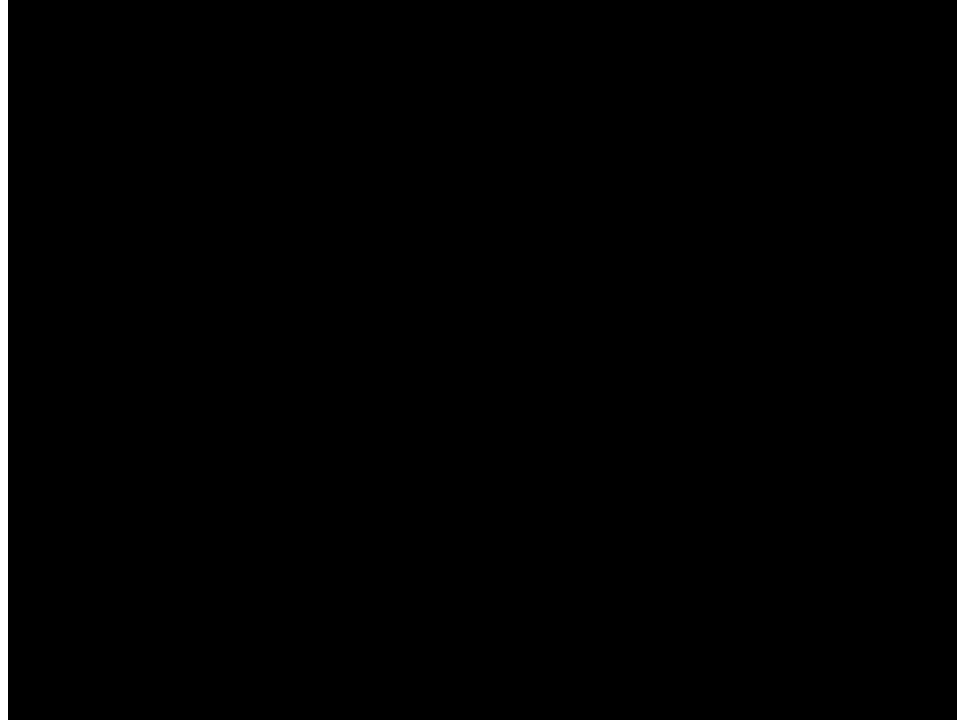


Overview of Language Model Adapters

- What is a Language Model?
 - Probabilistic model of word sequences; often used for text generation.
 - Often refers to Transformer architectures which leverage self-attention.
- What is a Language Model Adapter?
 - Additional trainable parameters added to linear layers.
- For what use-cases are Language Model Adapters relevant?
 - Efficient fine-tuning.
 - **Multi- domain/language/task adaptation.**



Overview of Language Model Adapters



Summary of MLOps Challenges

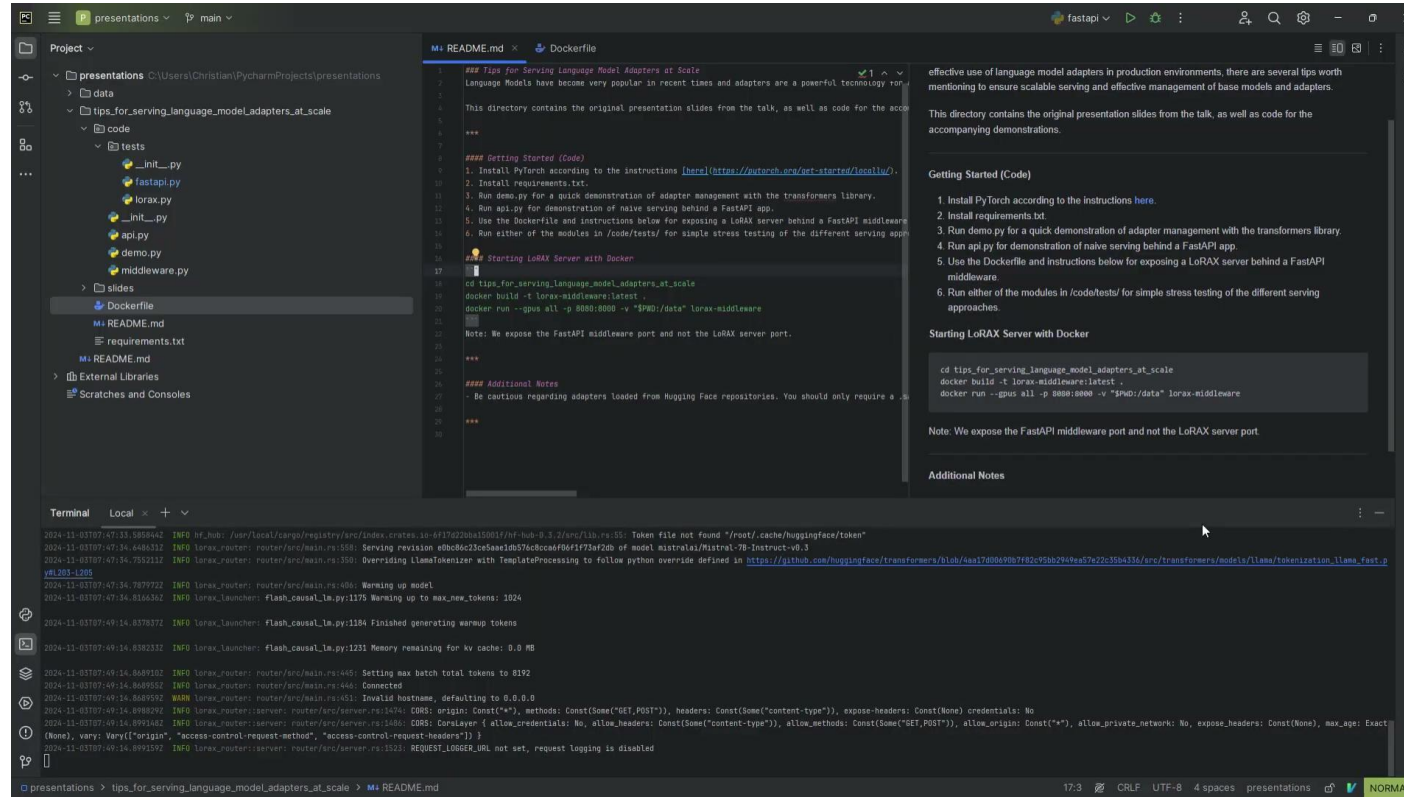
- Efficient Serving
 - Depending on the size of the base model, hosting multiple model instances for different adapters is not resource efficient.
 - Requires a serving solution which can manage the large parameters of the base model and orchestrate adapters dynamically.
 - Batching and adapter caching.
- Model / Adapter Versioning and Management
 - Similar to other multi-model scenarios where versioning and model registration tools can provide a framework for model management.
 - Unique in that adapters may need independent versioning which must be coordinated with base model. Requires unified versioning approach with base model awareness.

Tips for Serving

- Open Source LLM Servers with adapter support
 - vLLM
 - Text Generation Inference
 - LoRAX



Tips for Serving



```
## Tips for Serving Language Model Adapters at Scale
Language Models have become very popular in recent times and adapters are a powerful technology for
This directory contains the original presentation slides from the talk, as well as code for the accompanying demonstrations.

### Getting Started (Code)
1. Install PyTorch according to the instructions [here](https://pytorch.org/get-started/locally/).
2. Install requirements.txt.
3. Run demo.py for a quick demonstration of adapter management with the transformers library.
4. Run api.py for demonstration of naive serving behind a FastAPI app.
5. Use the Dockerfile and instructions below for exposing a LoRA server behind a FastAPI middleware.
6. Run either of the modules in /code/tests/ for simple stress testing of the different serving approaches.

### Starting LoRA Server with Docker
cd tips_for_serving_language_model_adapters_at_scale
docker build -t lorax-middleware:latest
docker run --gpus all -p 8080:8080 -v "$PWD:/data" lorax-middleware

Note: We expose the FastAPI middleware port and not the LoRA server port.

### Additional Notes
- Be cautious regarding adapters loaded from Hugging Face repositories. You should only require a subset of the adapters.
```

```
2024-11-03T07:47:33.585942 INFO hf_hub: /usr/local/cargo/registry/src/index.crates.io-6f1762220a150017/hf-hub-0.3.2/src/lib.rs:55: Token file not found "/root/.cache/huggingface/token"
2024-11-03T07:47:34.648332 INFO lorax_router: router/src/main.rs:586: Serving revision 68b086c23c6a61d0576c8cc6f06f173e22b of model mistralai/mistral-7B-Instruct-v0.3
2024-11-03T07:47:34.753232 INFO lorax_router: router/src/main.rs:350: Overriding LlamaTokenizer with TemplateProcessing to follow python override defined in https://github.com/huggingface/transformers/blob/4aa1740b40b7482c49bb2948a7e72c35b433e/src/transformers/models/llama/tokenization_llama_fast.py#L203-L205
2024-11-03T07:47:34.787722 INFO lorax_router: router/src/main.rs:406: Warming up model
2024-11-03T07:47:34.816362 INFO lorax_launcher: flash_causal_lm.py:1175: Warming up to max_new_tokens: 1024
2024-11-03T07:49:14.837072 INFO lorax_launcher: flash_causal_lm.py:1231: Finished generating warmup tokens
2024-11-03T07:49:14.838232 INFO lorax_launcher: flash_causal_lm.py:1231: Memory remaining for kv cache: 0.0 MB
2024-11-03T07:49:14.868182 INFO lorax_router: router/src/main.rs:546: Setting max batch total tokens to 8192
2024-11-03T07:49:14.868952 INFO lorax_router: router/src/main.rs:446: Connected
2024-11-03T07:49:14.868952 WARN lorax_router: router/src/main.rs:451: Invalid hostname, defaulting to 0.0.0.0
2024-11-03T07:49:14.898292 INFO lorax_router: server: router/src/server.rs:1474: CORS: origin: Const(*), methods: Const(Some("GET,POST")), headers: Const(Some("content-type")), expose_headers: Const(None), credentials: No
2024-11-03T07:49:14.899342 INFO lorax_router: server: router/src/server.rs:1480: CORS: CorsLayer { allow_credentials: No, allow_headers: Const(Some("content-type")), allow_methods: Const(Some("GET,POST")), allow_origin: Const(*), allow_private_network: No, expose_headers: Const(None), max_age: Exact(
(None), vary: Vary(("origin", "access-control-request-method", "access-control-request-headers")) }
2024-11-03T07:49:14.899392 INFO lorax_router: server: router/src/server.rs:1523: REQUEST_LOGGING_URL not set, request logging is disabled
```

1:12, 3:41

<https://loraexchange.ai/>

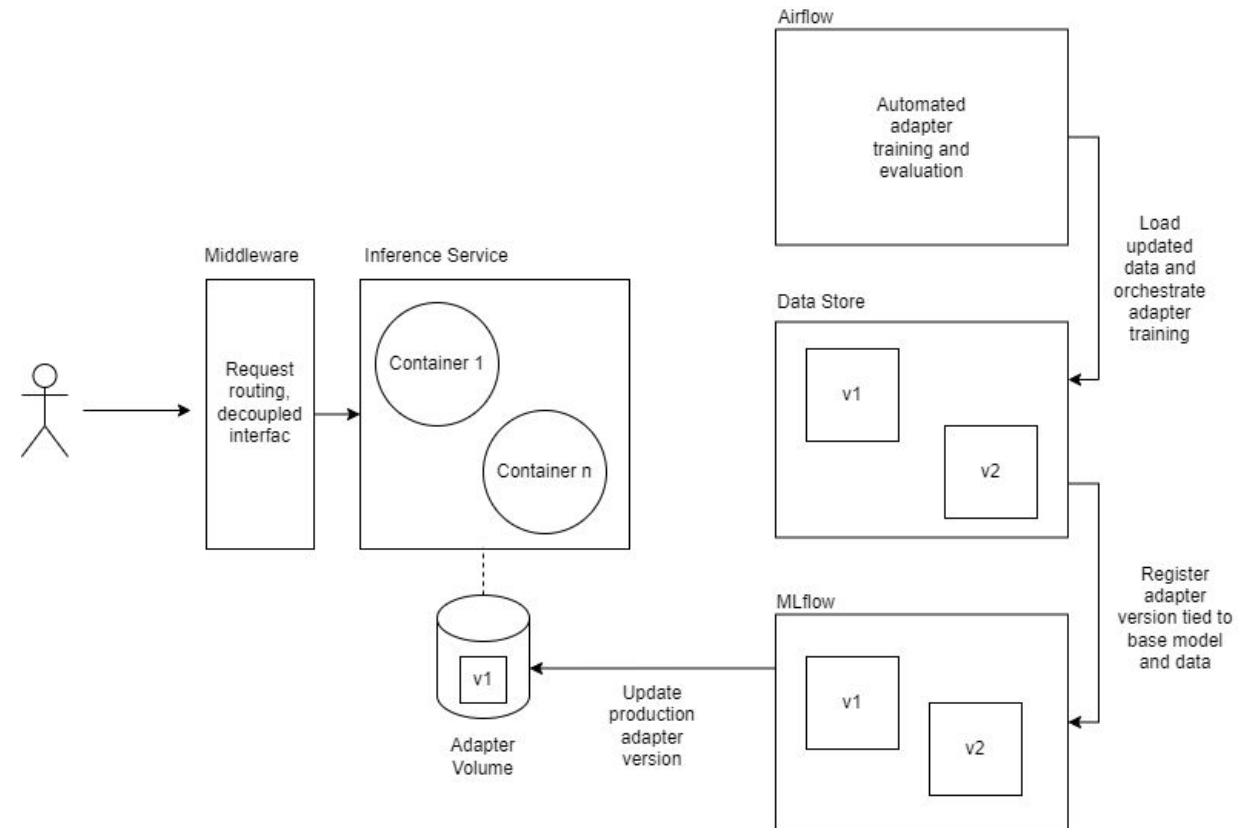
Tips for Serving

- Open Source LLM Servers with adapter support
 - vLLM
 - Text Generation Inference
 - LoRAX
- Important to keep in mind
 - Load testing
 - Adapter cache parameters
 - Adapter loading / attachment
 - Middleware
 - Managed solutions



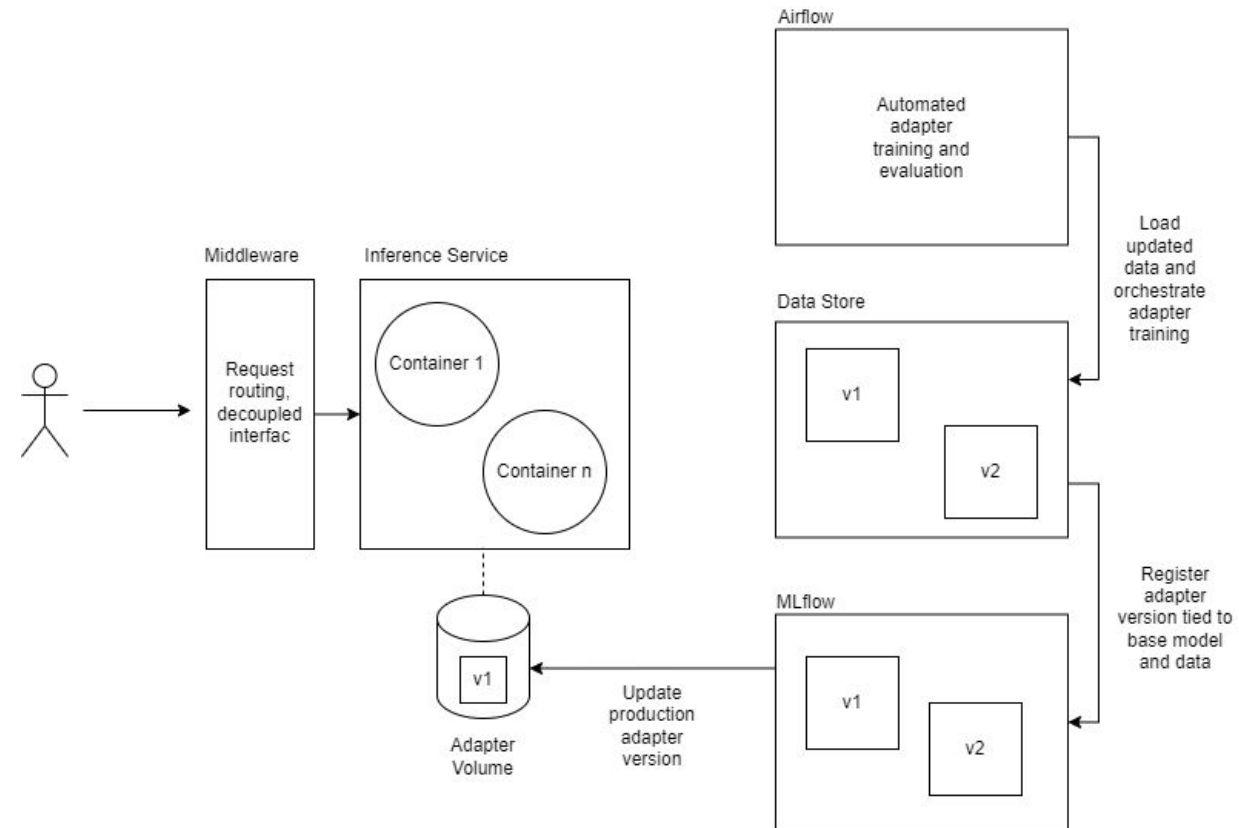
Tips for Model Management

- General Tips
 - Version base model with data
 - Version adapter with data and base model
 - Register adapters as independent models according to the use-case and ensure traceability with data and base model
- Important to keep in mind
 - Emphasis on automation



Summary

- Language Model Adapters
 - Allows adaptation of large models with efficient number of parameters.
 - Potentiates dynamic adaptation of base model behaviour through inclusion of small matrices.
- Serving
 - Ensure proper orchestration of adapters and batching.
 - Wide support in open source servers.
- Model Management
 - Attend to proper versioning of adapters with respect to base models.
 - Emphasis on automation, especially for higher numbers of adapters.



Additional Resources

- Informational
 - [Overview of the LoRA Family](#)
 - [Apple Intelligence](#)
- Tutorials
 - [Hugging Face](#)
 - [TensorRT](#)
 - [Slides and Code](#)
- Serving Solutions
 - [LoRAX](#)
 - [Text Generation Inference](#)
 - [vLLM](#)
- Managed Serving Solutions
 - [Predibase](#)
 - [Empower](#)



Q&A

