LUDWIG 0.8

HANDS-ON WEBINAR

Efficiently Build Custom LLMs on Your Data

Welcome

Webinar Logistics

- All lines are muted
- Today’s session is recorded and will be made available
- Please submit questions in the panel for the live Q&A
- Visit https://pbase.ai/GetStarted to get access
- Join the open-source community at Ludwig.ai

Today’s speakers

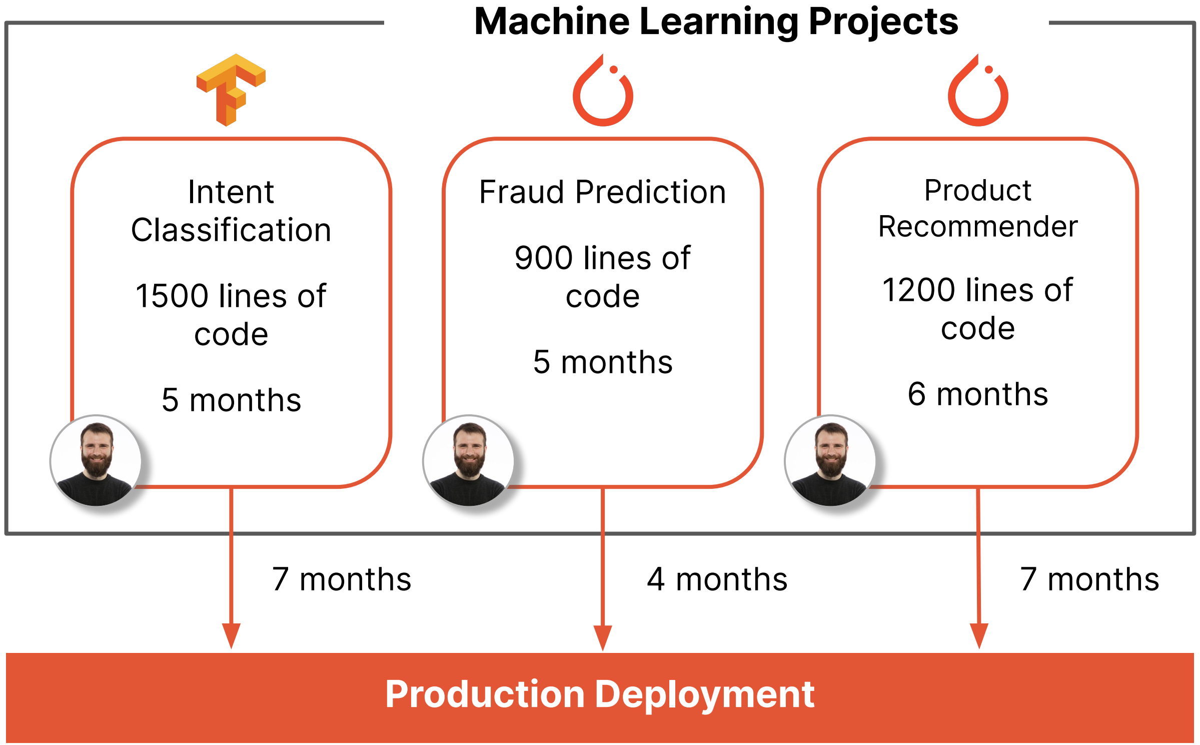
Piero Molino

Creator of Ludwig //
CEO and Cofounder,
Predibase

Arnav Garg

Ludwig Maintainer //
ML Engineer,
Predibase

My experience building ML apps at Uber

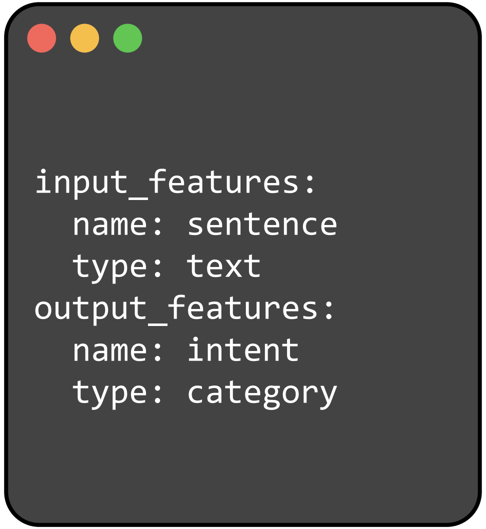


There must be a better way

Unblocking engineers with LUDWIG

An open-source declarative ML framework started at Uber

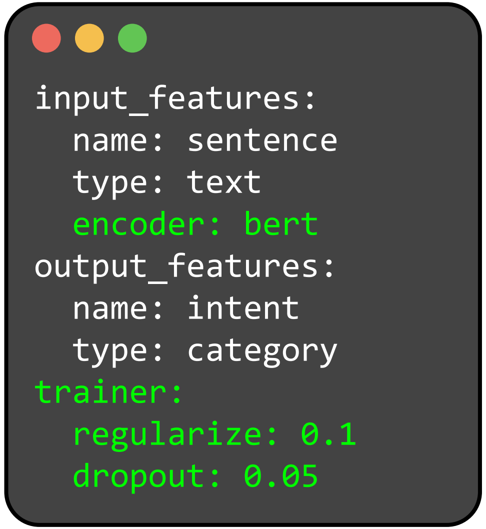
Easy to start



```
input_features:
  name: sentence
  type: text
output_features:
  name: intent
  type: category
```

From months to days
No ML code required
Readable & Reproducible

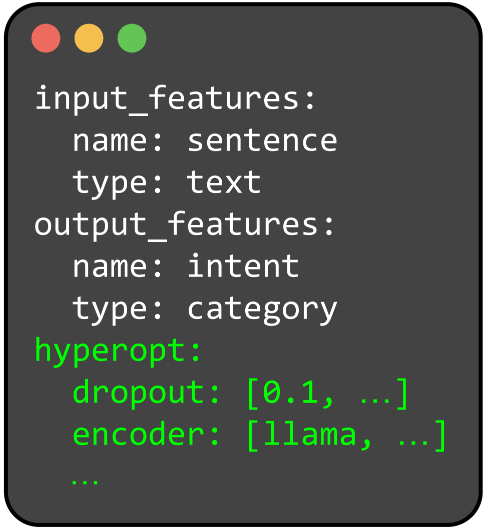
Expert level control



```
input_features:
  name: sentence
  type: text
  encoder: bert
output_features:
  name: intent
  type: category
trainer:
  regularize: 0.1
  dropout: 0.05
```

Easy to Iterate
Extensible

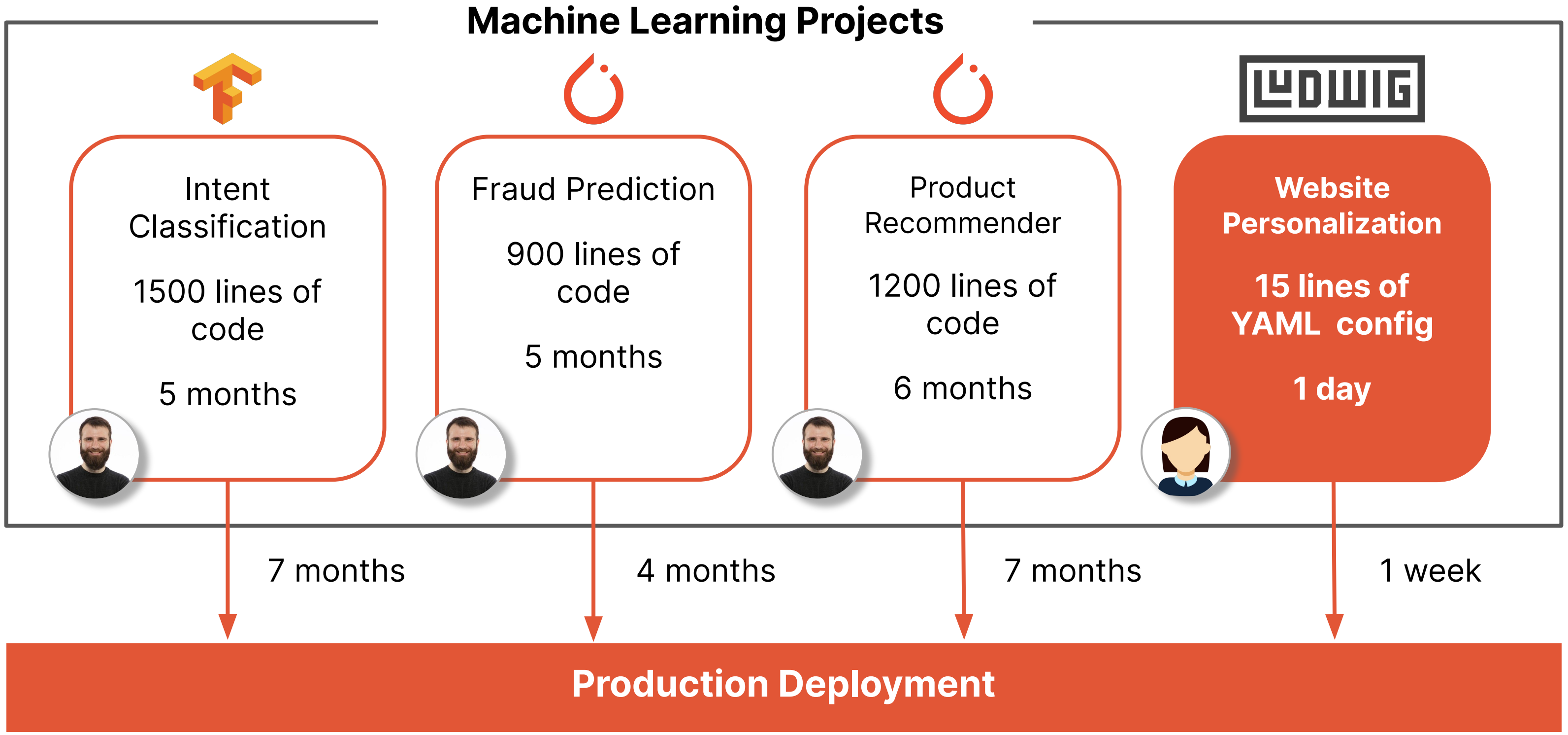
Advanced functionalities



```
input_features:
  name: sentence
  type: text
output_features:
  name: intent
  type: category
hyperopt:
  dropout: [0.1, …]
  encoder: [llama, …]
  …
```

Hyperparameter search
State-of-the-art models
Distributed training

Making engineers the new ML team



Machine Learning Projects

Intent Classification

1500 lines of code

5 months

7 months

Fraud Prediction

900 lines of code

5 months

4 months

Product Recommender

1200 lines of code

6 months

7 months

Website Personalization

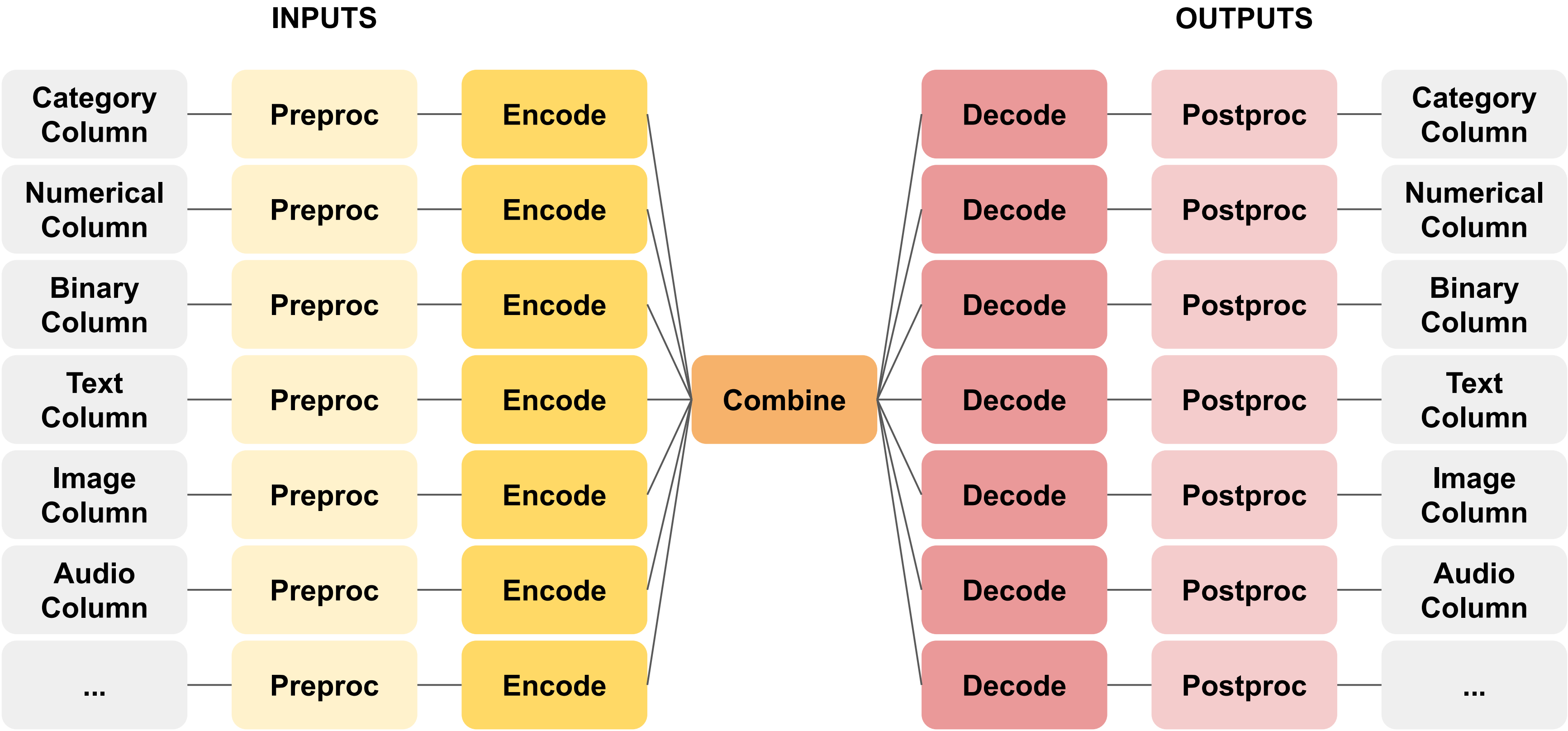
15 lines of YAML config

1 day

1 week

Production Deployment

Ludwig Architecture



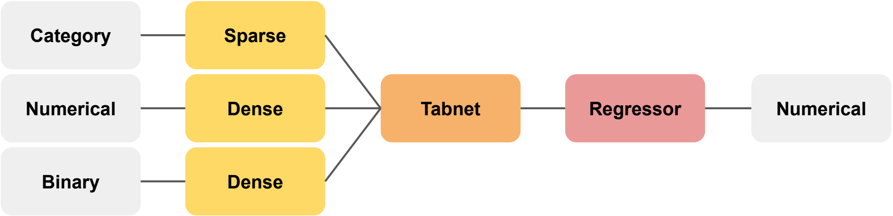
INPUTS

OUTPUTS

Category Column	Preproc	Encode		Decode	Postproc	Category Column
Numerical Column	Preproc	Encode		Decode	Postproc	Numerical Column
Binary Column	Preproc	Encode		Decode	Postproc	Binary Column
Text Column	Preproc	Encode	Combine	Decode	Postproc	Text Column
Image Column	Preproc	Encode		Decode	Postproc	Image Column
Audio Column	Preproc	Encode		Decode	Postproc	Audio Column
...	Preproc	Encode		Decode	Postproc	...

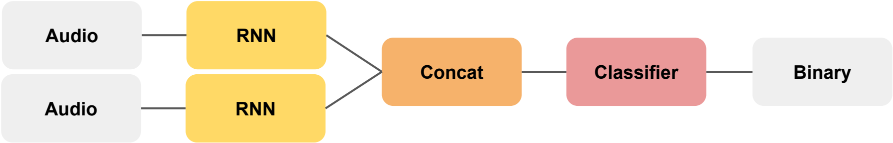
Ludwig Task Flexibility

Regression



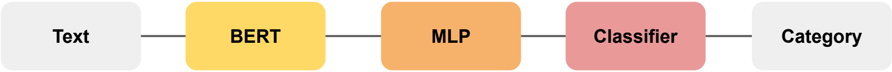
Category — Sparse
Numerical — Dense
Binary — Dense
Tabnet — Regressor — Numerical

Speech Verification



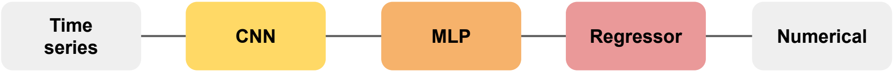
Audio — RNN
Audio — RNN
Concat — Classifier — Binary

Text Classification



Text — BERT — MLP — Classifier — Category

Forecasting



Time series — CNN — MLP — Regressor — Numerical

Image Captioning

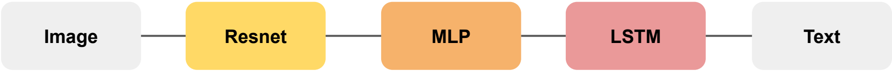
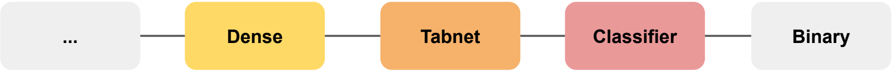


Image — Resnet — MLP — LSTM — Text

Binary Classification



... — Dense — Tabnet — Classifier — Binary

Ludwig v0.8 new features

- Prompt Templating
- Zero-Shot and Few-Shot In-Context Learning
- Declaratively Fine-Tune Large Language Models
- Large Model Training with Deepspeed
- Parameter efficient fine-tuning (PEFT)

Prompt Templating

Prompt Template Definition

```
model_type: llm
base_model: Llama-2-7b-hf
prompt:
 task: "Rate this book review with from 1 to 5"
 template: |
   Task: {task}.
   Review: “{title} {review}”.
   What score would you assign?
```

Data

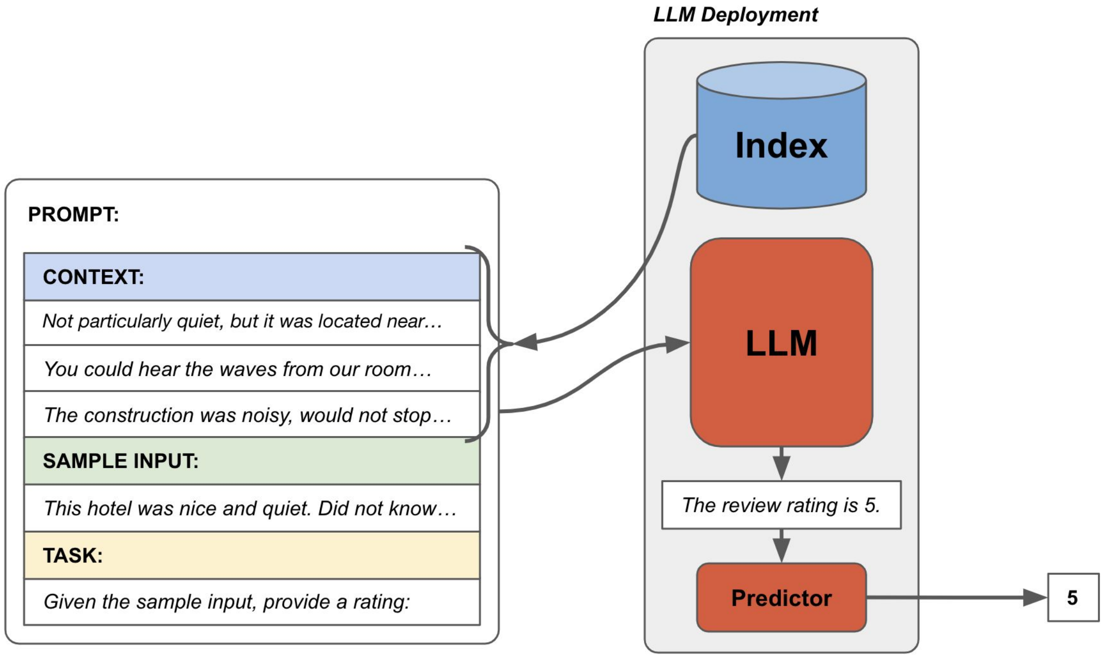
title	review	score
Amazing story!	This book made me dream of …	4

Input to LLM

```
Task: Classify this book review with a score from 1 to 5.
Review: “Amazing story! This book made me dream of …”.
What score would you assign?
```

```
llm = LudwigModel(config)
llm.create_model()
results = llm.predict(df)
```

Zero-Shot and Few-Shot In-Context Learning



PROMPT:

CONTEXT:

Not particularly quiet, but it was located near…

You could hear the waves from our room…

The construction was noisy, would not stop…

SAMPLE INPUT:

This hotel was nice and quiet. Did not know…

TASK:

Given the sample input, provide a rating:

The review rating is 5.

Zero-Shot and Few-Shot In-Context Learning

Prompt Template Definition

```
model_type: llm
base_model: Llama-2-7b-hf
prompt:
 task: "Rate this book review with from 1 to 5"
 template: |
   Task: {task}. Examples: {__context__}.
   Review: “{title} {review}”.
   What score would you assign?
retrieval:
   type: semantic
   k: 2
   model_name: paraphrase-MiniLM-L3-v2
```

Retrieved Data

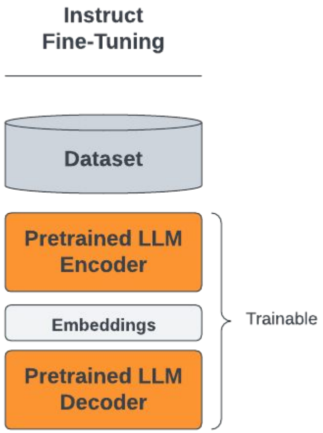
title	review	score
Great Sci-Fi	Asimov always delivers …	5
Boring	Not the best Asimov book …	2

Input to LLM

```
Task: Classify this book review with a score from 1 to 5.
Examples: [{title: “Great Sci-Fi”, review: “Asimov always delivers …”,
score “5”}, {title: “Boring”, review: “Not the best Asimov book …”,
score “2”}].
Review: “Sci-fi masterpiece. Second Foundation series book…”.
What score would you assign?
```

```
llm = LudwigModel(config)
llm.create_model()
results = llm.predict(df)
```

Declaratively Fine-Tune LLMs

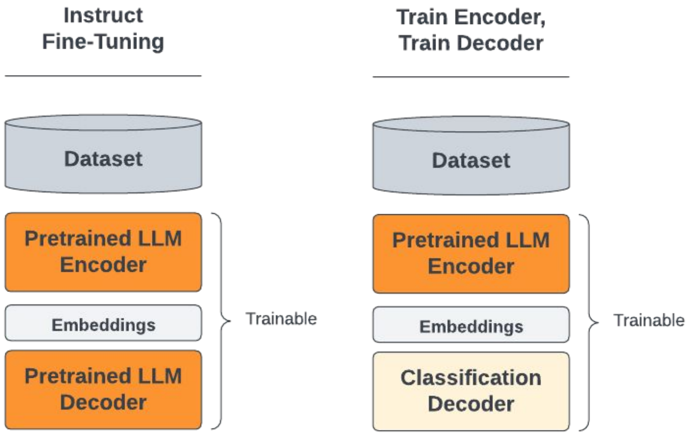


Instruct Fine-Tuning

```yaml
model_type: llm
base_model: Llama-2-7b-hf

input_features:
  - name: input
    type: text

output_features:
  - name: output
    type: text

trainer:
  type: finetune
  learning_rate: 0.0003
  batch_size: 1
  gradient_accumulation_steps: 8
  epochs: 3
```

```python
llm = LudwigModel(config)
results = llm.train(df)
```

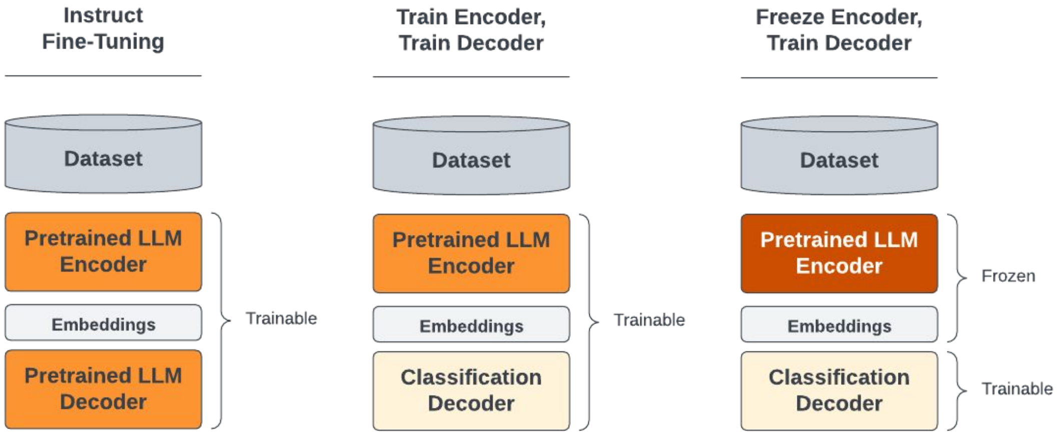
Declaratively Fine-Tune LLMs



```yaml
input_features:
  - name: review
    type: text
    encoder:
      type: auto_transformer
      pretrained_model_name_or_path: Llama-2-7b-hf
      trainable: true

output_features:
  - name: sentiment
    type: category
```

```python
llm = LudwigModel(config)
results = llm.train(df)
```

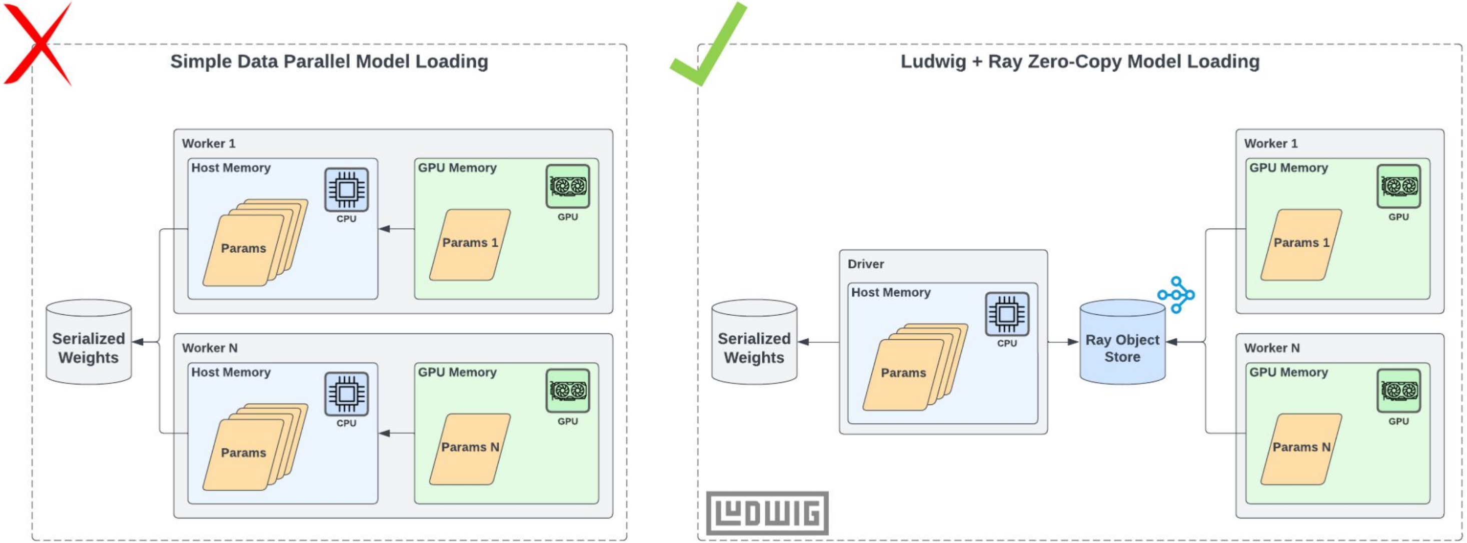
Declaratively Fine-Tune LLMs



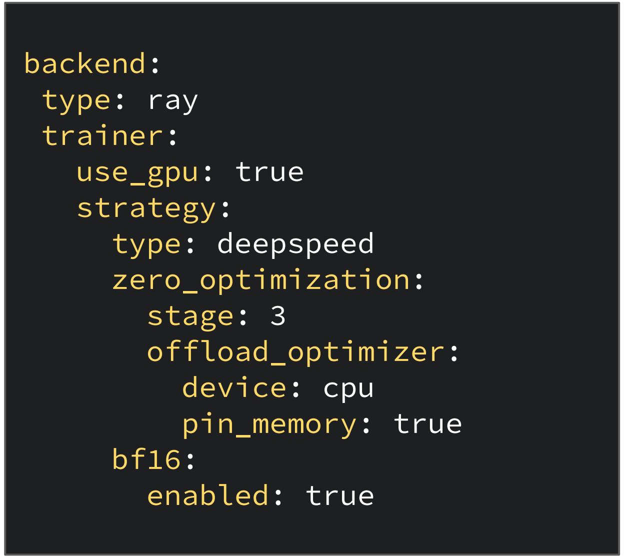
```
input_features:
  - name: review
    type: text
    encoder:
      type: auto_transformer
      pretrained_model_name_or_path:
Llama-2-7b-hf
      trainable: false
      preprocessing:
        cache_encoder_embeddings: true


output_features:
  - name: sentiment
    type: category
```

```
llm = LudwigModel(config)
results = llm.train(df)
```

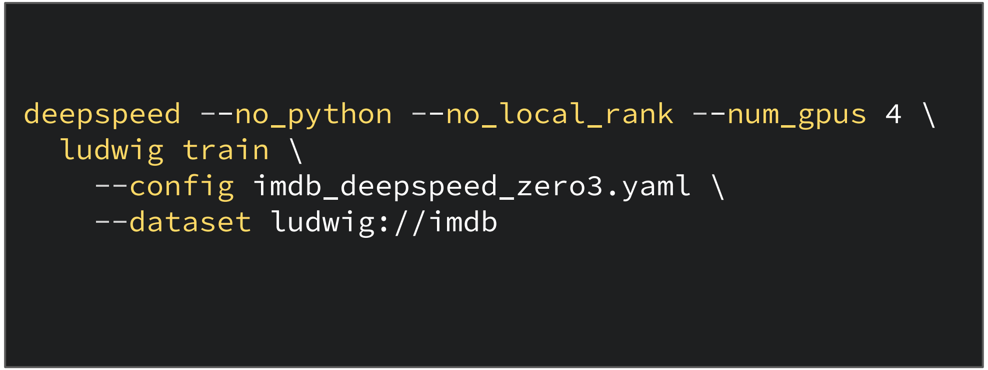
Large Model Training with Deepspeed



Large Model Training with Deepspeed



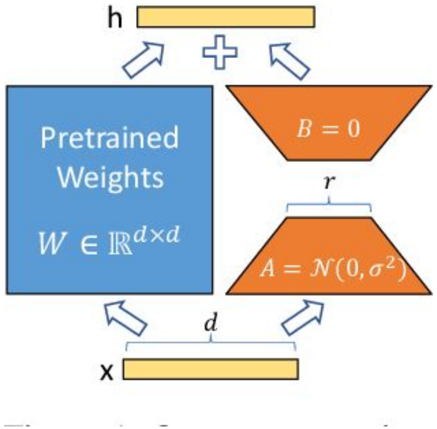
```yaml
backend:
 type: ray
 trainer:
   use_gpu: true
   strategy:
     type: deepspeed
     zero_optimization:
       stage: 3
       offload_optimizer:
         device: cpu
         pin_memory: true
     bf16:
       enabled: true
```

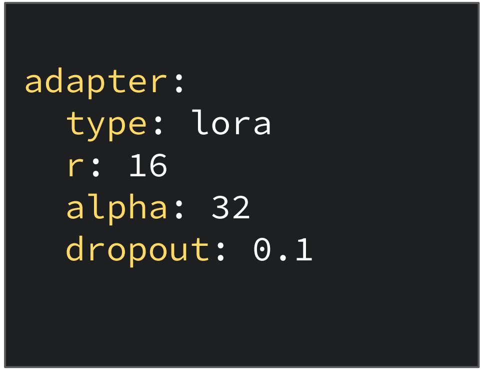


```
deepspeed --no_python --no_local_rank --num_gpus 4 \
  ludwig train \
    --config imdb_deepspeed_zero3.yaml \
    --dataset ludwig://imdb
```

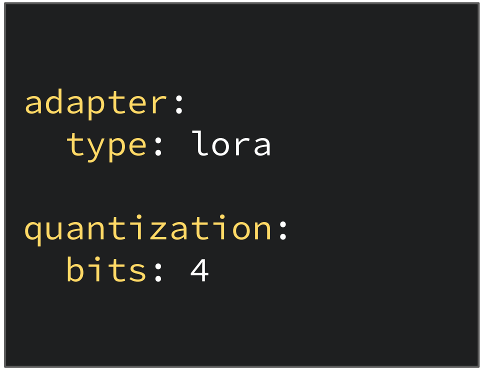
Parameter efficient fine-tuning

- LoRA
- AdaLoRA
- Adaptation Prompt
  (aka, LLaMA Adapter)
- QLoRA


h
Pretrained Weights
$W \in \mathbb{R}^{d\times d}$
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
d
x

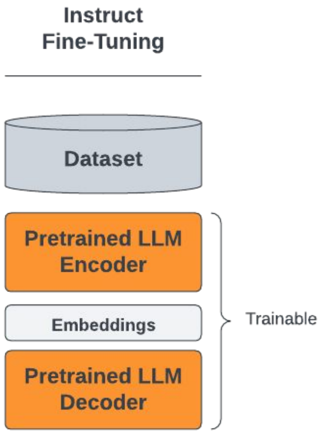


```
adapter:
  type: lora
  r: 16
  alpha: 32
  dropout: 0.1
```



```
adapter:
  type: lora

quantization:
  bits: 4
```

Putting it all together



```yaml
model_type: llm
base_model: Llama-2-7b-hf

adapter:
  type: lora
quantization:
  bits: 4

prompt:
  template: |
    ### Instruction:
    {instruction}

    ### Input:
    {input}

    ### Response:
```

```yaml
input_features:
  - name: input
    type: text

output_features:
  - name: output
    type: text

trainer:
  type: finetune
  learning_rate: 0.0003
  batch_size: 1
  gradient_accumulation_steps: 8
  epochs: 3
```

```python
llm = LudwigModel(config)
results = llm.train(df)
```

Hands-on Tutorial

Notebooks available at: https://pbase.ai/3YDMrcz

Predibase

LUDWIG

9,100+
★ on GitHub

3000+
downloads/month

130+
contributors

~80
commits/month

Learn more: www.ludwig.ai

Predibase

The Low-code Declarative ML Platform

Build customized, privately hosted LLMs in just a few lines of code

Request a demo or free trial:
https://pbase.ai/GetStarted