Train deep learning models faster with Amazon SageMaker

Robert Van Dusen

Senior Product Manager - AWS AI

Agenda

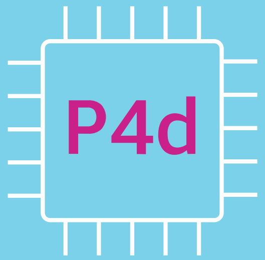
- Challenges in training deep learning models
- Introducing Amazon SageMaker Training Compiler
- Demo
- Wrap-up and Q&A

Deep learning models have a size problem

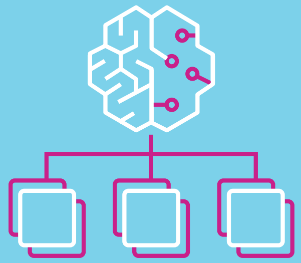
Large datasets take a **long time to train**, creating a bottleneck

Training costs are an obstacle to experimentation and innovation

Optimizing for cost and speed is challenging



Infrastructure



Distributed training

Compilers offer efficiency gains

Convert high-level representation to hardware-optimized instructions

Enable more efficient use of hardware without scaling out or up

Dedicated compilers for inference offer performance gains of 25x

Compilers offer efficiency gains

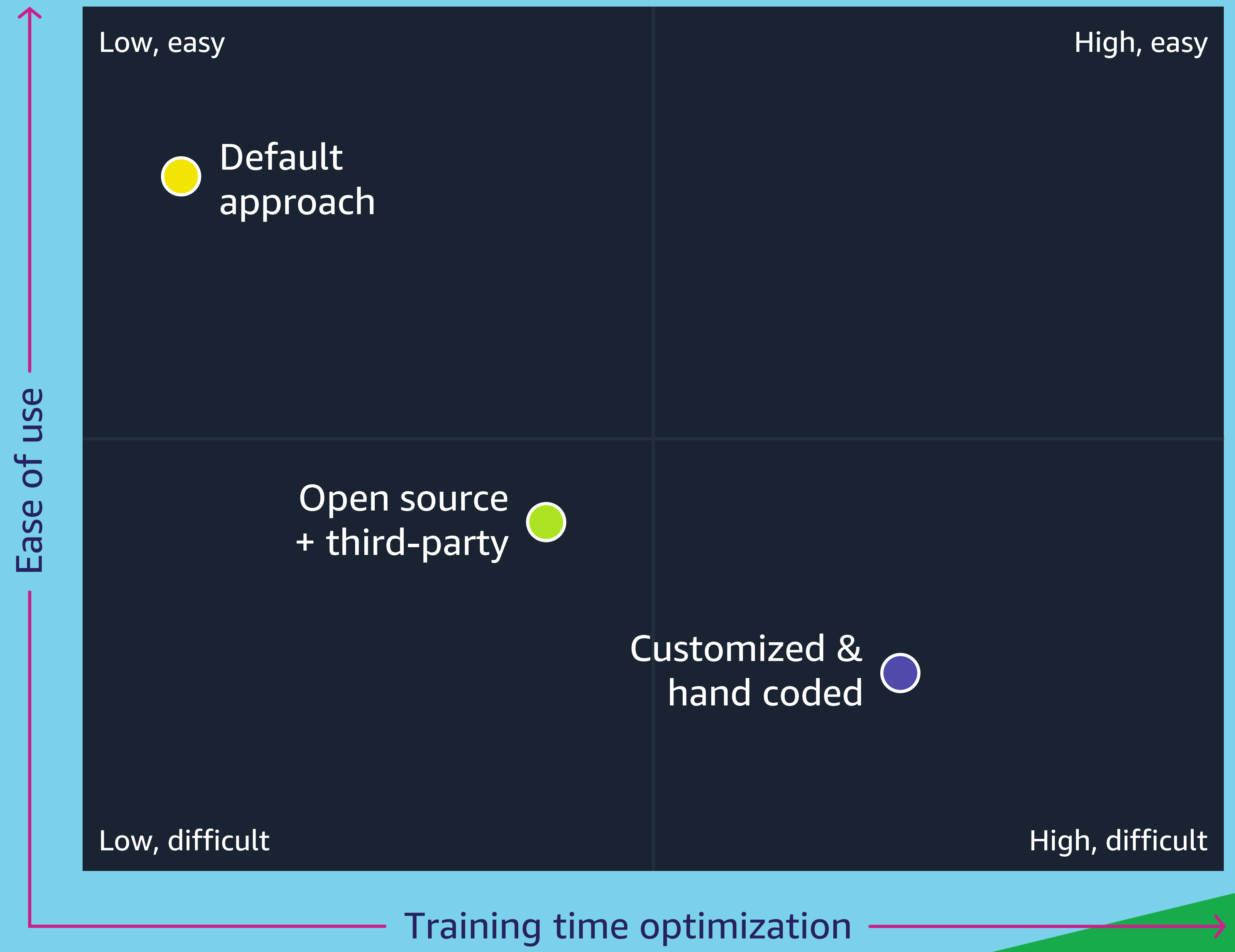
Convert high-level representation to hardware-optimized instructions

Enable more efficient use of hardware without scaling out or up

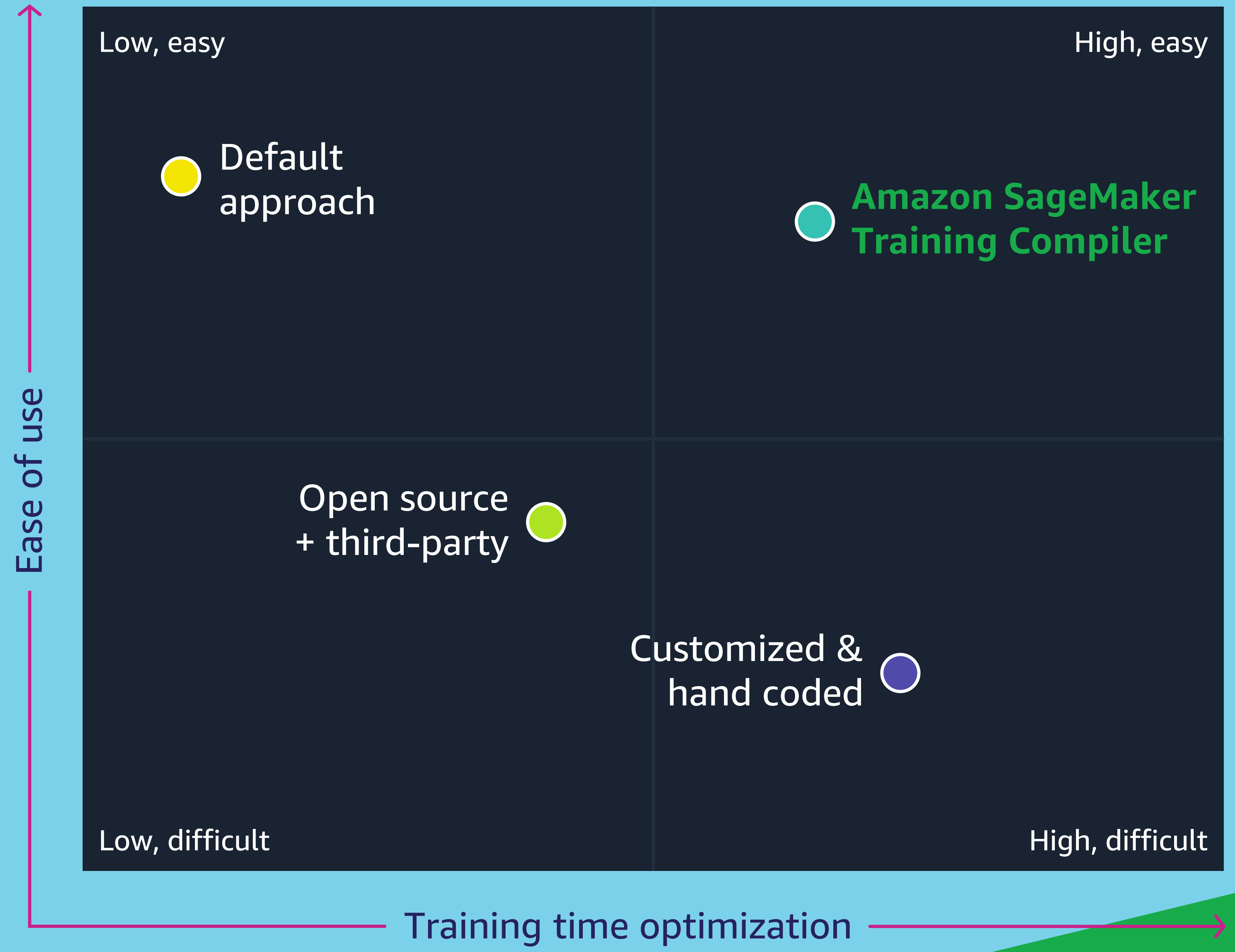
Dedicated compilers for inference offer performance gains of 25x

No good built-in solution for training available

Compilation for deep learning training is a challenge



Compilation for deep learning training is a challenge



NEW

Amazon SageMaker Training Compiler

The fast and easy way to train large deep learning models on GPUs

GENERALLY AVAILABLE

Accelerate deep learning model training

Speed up training by as much as 50%

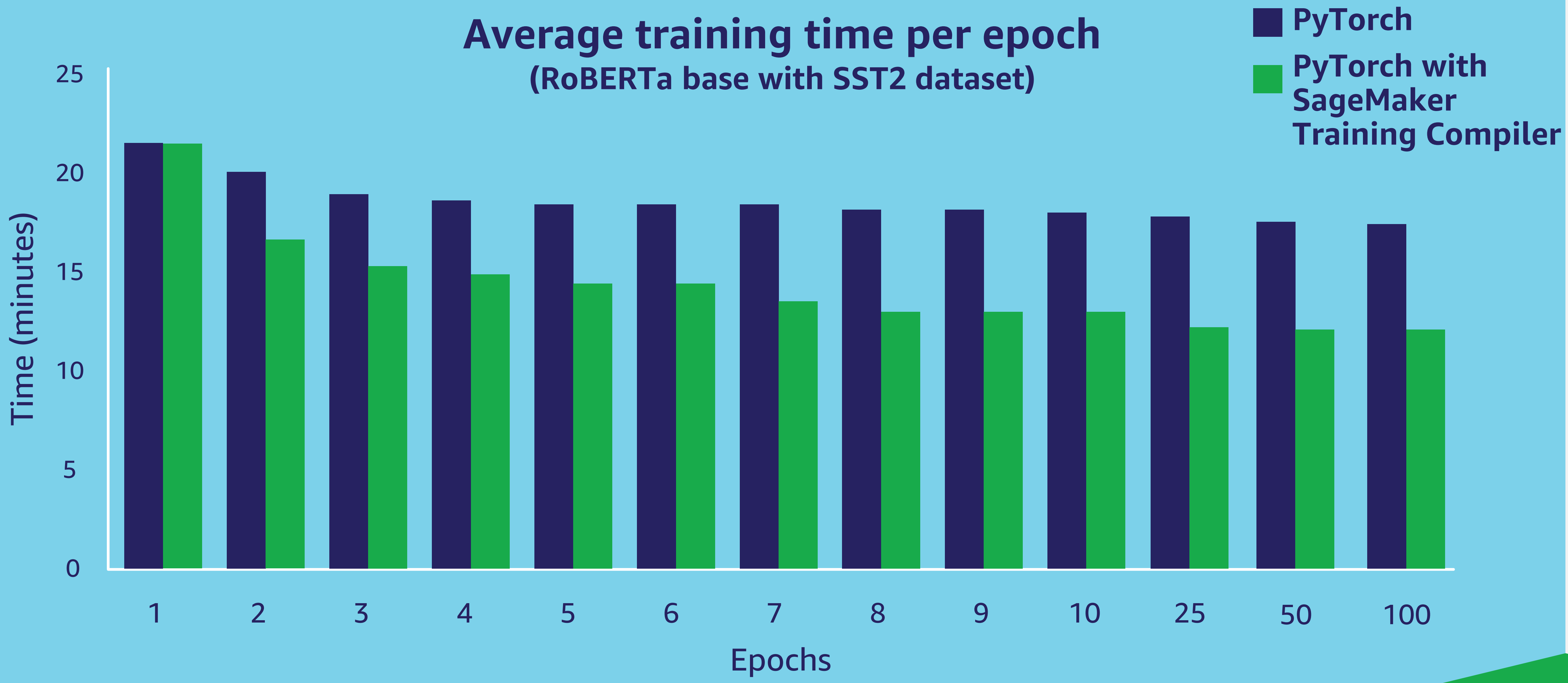
Minimal code changes required

Enable in minutes without any changes to workflow

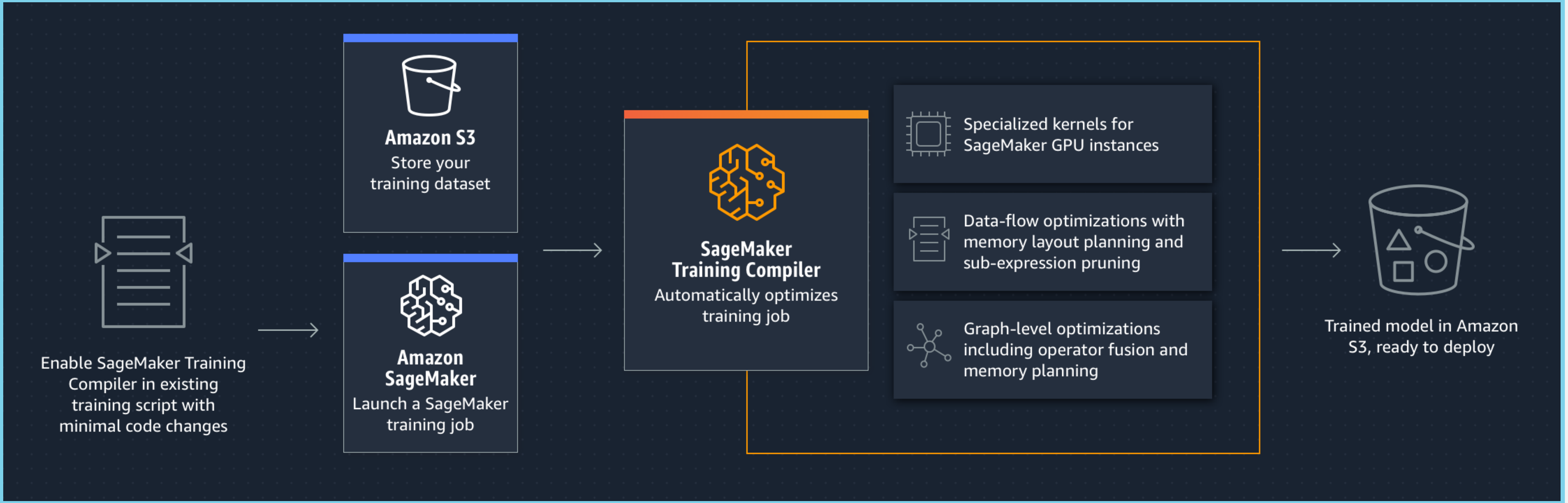
Lower training costs

Free to use on SageMaker and additional savings from shortened training jobs

Specialized compilation accelerates training



Acceleration without workflow disruption



Enable SageMaker Training Compiler in existing training script with minimal code changes

Amazon S3
Store your training dataset

Amazon SageMaker
Launch a SageMaker training job

SageMaker Training Compiler
Automatically optimizes training job

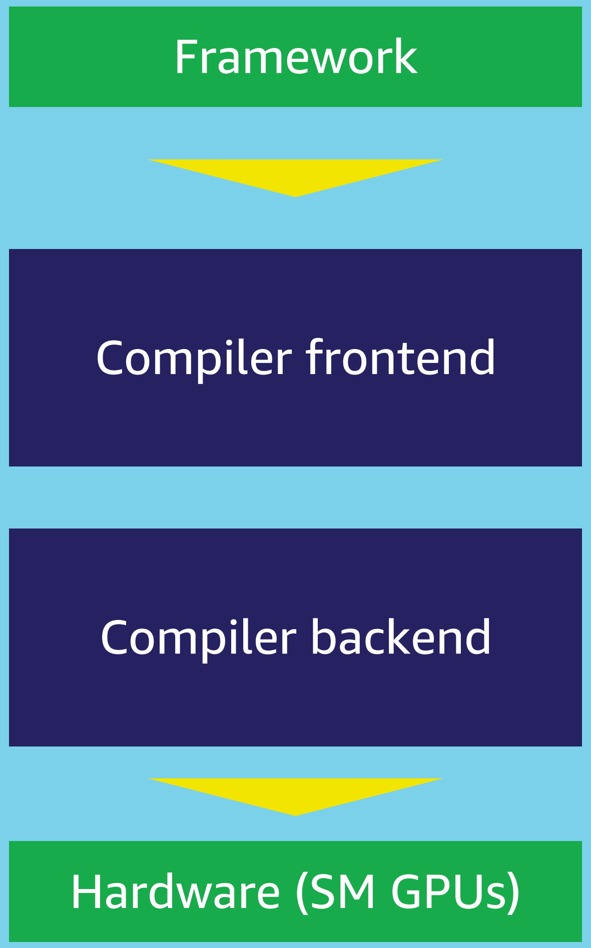
Specialized kernels for SageMaker GPU instances

Data-flow optimizations with memory layout planning and sub-expression pruning

Graph-level optimizations including operator fusion and memory planning

Trained model in Amazon S3, ready to deploy

Deep learning compiler architecture



Framework

Compiler frontend

Compiler backend

Hardware (SM GPUs)

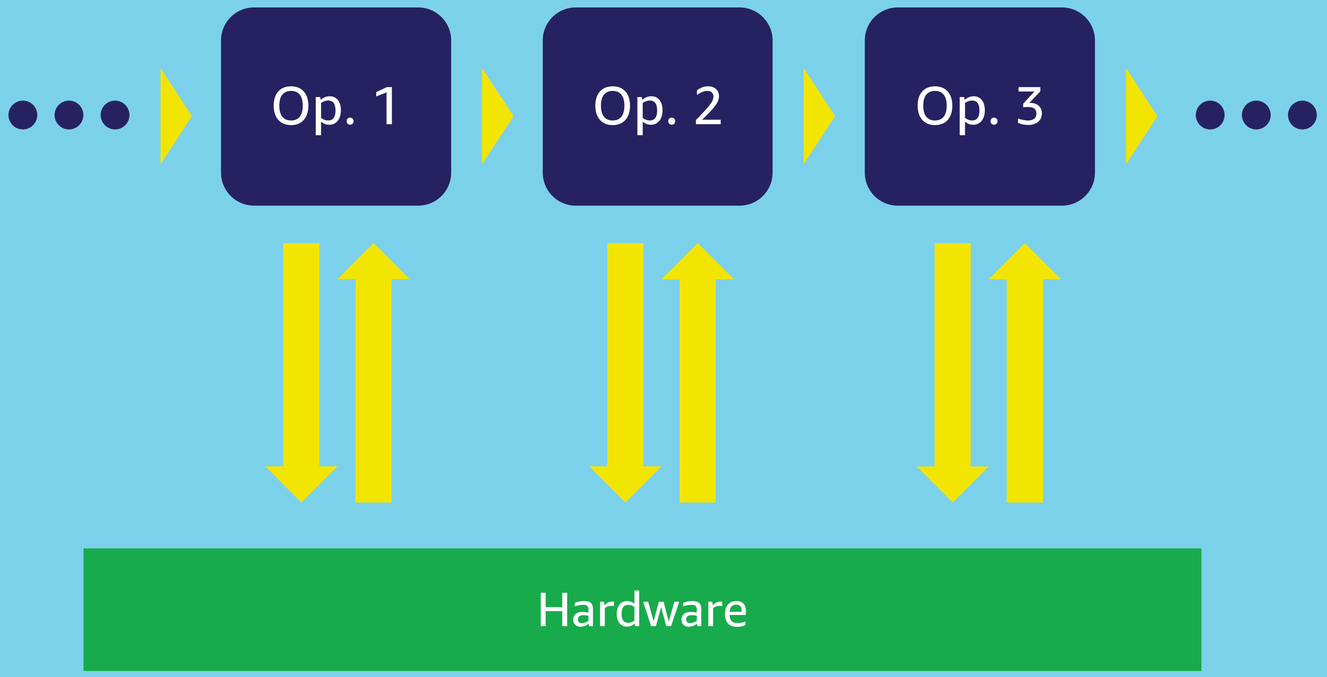
PyTorch, TensorFlow

- High-level IR (Graph IR)
  - Hardware independent optimizations
  - Optimizations on computation and control flow
  - E.g., static memory planning, operator fusion

- Low-level IR
  - Hardware specific optimizations
  - E.g., memory latency hiding
  - Hardware code generation

ml.g4dn, ml.p3, and ml.p4d instances on SageMaker

Key DL compiler optimization: operator fusion

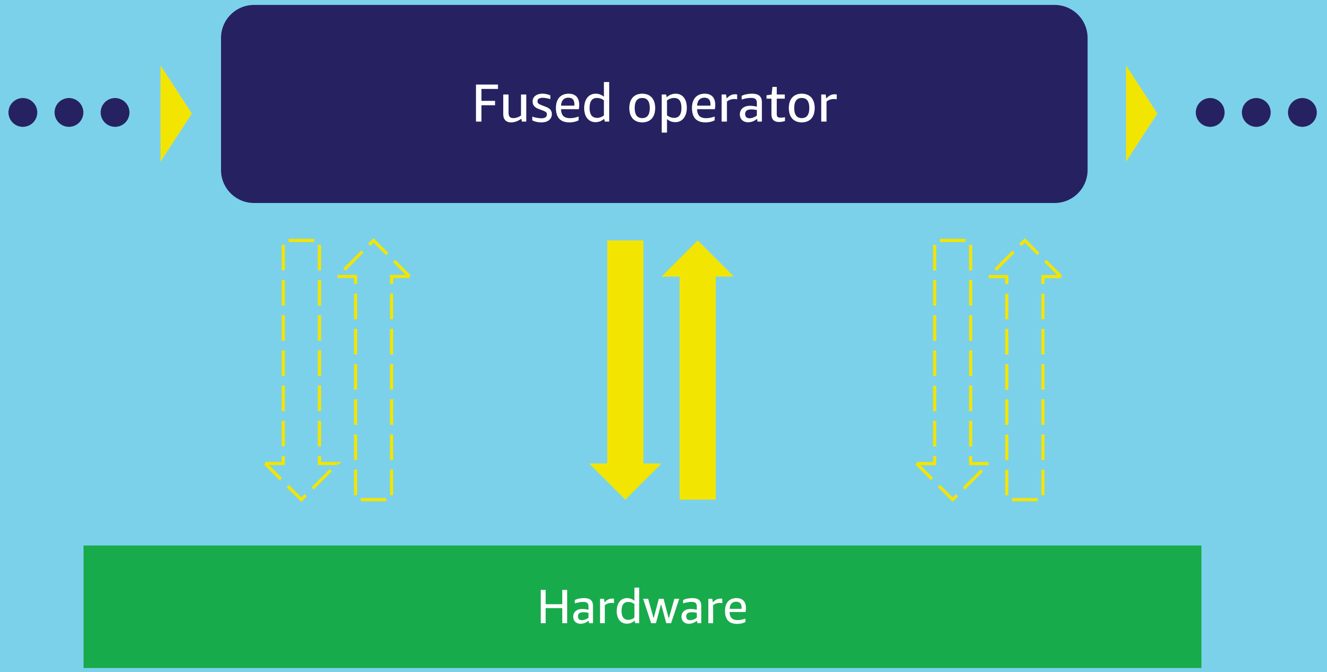


Computation graph

Kernel launches

Hardware

Key DL compiler optimization: operator fusion

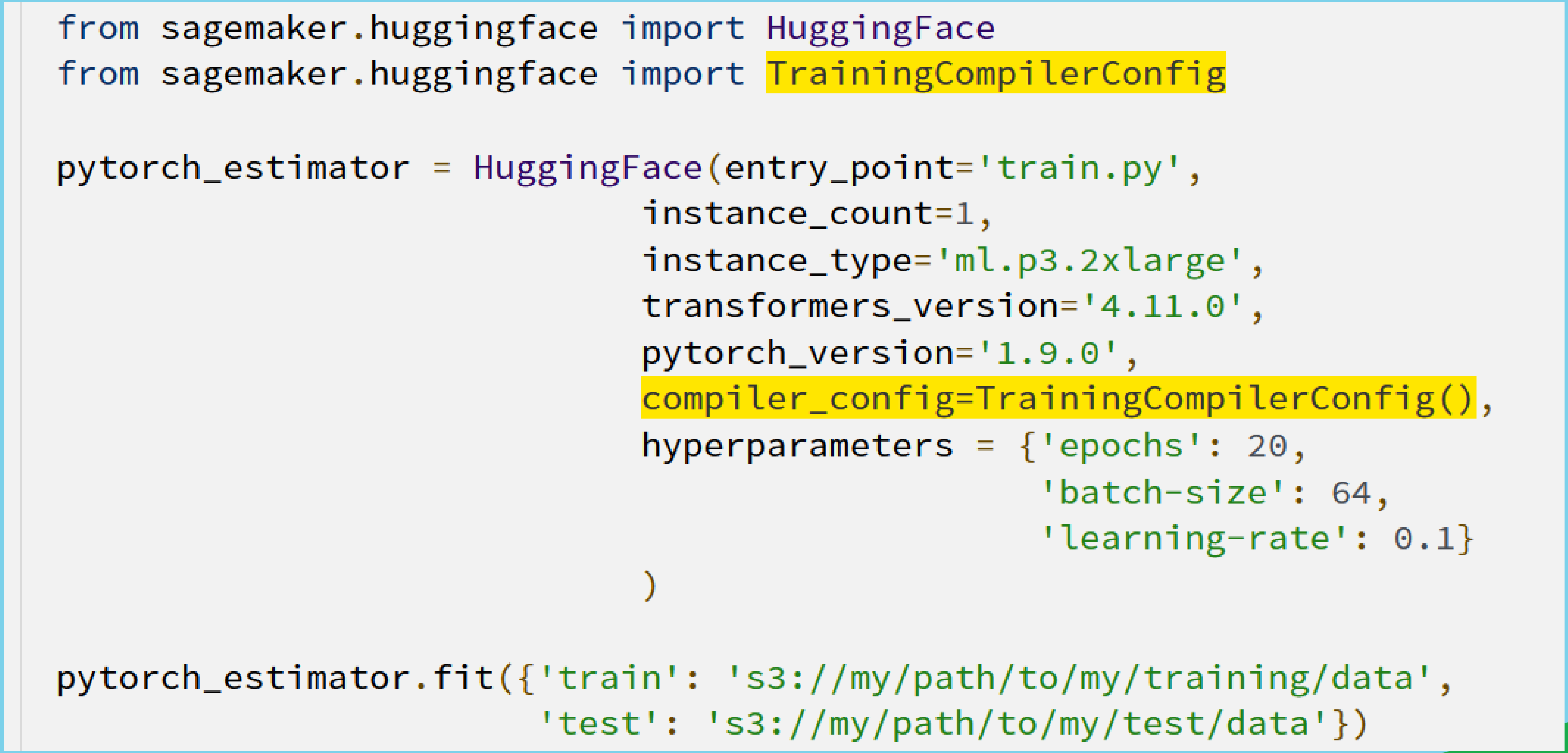


Computation graph

Kernel launches

Hardware

SageMaker Training Compiler can be enabled in minutes

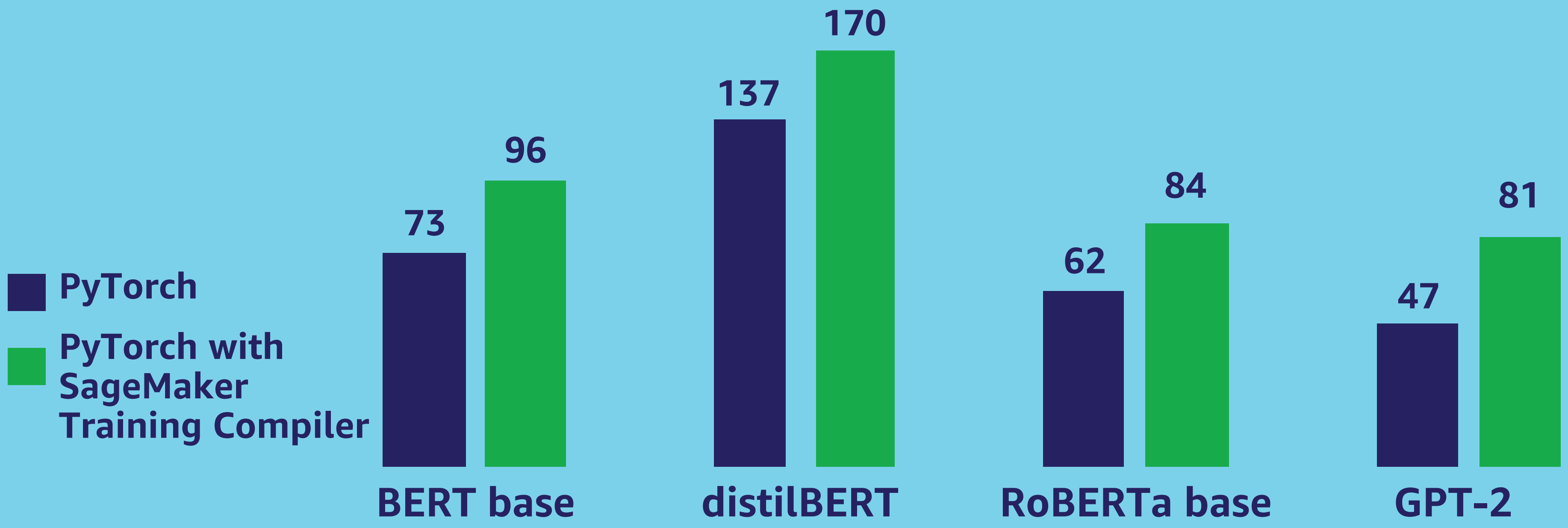


```
from sagemaker.huggingface import HuggingFace
from sagemaker.huggingface import TrainingCompilerConfig

pytorch_estimator = HuggingFace(entry_point='train.py',
                            instance_count=1,
                            instance_type='ml.p3.2xlarge',
                            transformers_version='4.11.0',
                            pytorch_version='1.9.0',
                            compiler_config=TrainingCompilerConfig(),
                            hyperparameters = {'epochs': 20,
                                               'batch-size': 64,
                                               'learning-rate': 0.1}
                            )

pytorch_estimator.fit({'train': 's3://my/path/to/my/training/data',
                       'test': 's3://my/path/to/my/test/data'})
```

Up to 50% speedup with Training Compiler

Training sample throughput[1]

(samples/second)



	BERT base	distilBERT	RoBERTa base	GPT-2
Reduction in billable time	38%	26%	27%	54%

[1] Test parameters: ml.p3.2xlarge, PyTorch with Hugging Face Trainer API, 25 epochs, sequence length of 512
Baseline used the Hugging Face AWS Deep Learning Container from Amazon ECR

SageMaker Training Compiler accelerates the most popular NLP models

- bert-base-uncased
- bert-large-uncased
- roberta-base
- gpt2
- bert-base-cased
- xlm-roberta-base
- bert-base-chinese
- roberta-large
- distilbert-base-uncased
- distilbert-base-uncased-finetuned-sst-2-English
- cl-tohoku/bert-base-japanese-whole-word-masking
- bert-base-multilingual-cased
- distilgpt2
- albert-base-v2
- gpt2-large

GUIDEWIRE

Guidewire is the platform P&C insurers trust to engage, innovate, and grow efficiently; the company combines digital, core, analytics, and AI to deliver its platform as a cloud service, and it enables its customers to do advanced analytics and machine learning for their industry-specific workloads; more than 450 insurers, from new ventures to the largest and most complex in the world, run on Guidewire

> “
>
> *One of Guidewire’s services is to help customers develop cutting-edge NLP models for applications like risk assessment and claims operations. Amazon SageMaker Training Compiler is compelling because it offers time and cost savings to our customers while developing these NLP models. We expect it to help us reduce training time by more than 20% through more efficient use of GPU resources. We are excited to implement SageMaker Training Compiler in our NLP workloads, helping us to accelerate the transformation of data to insight for our customers.”*
>
> **Matt Pearson, Principal Product Manager, Analytics and Data Services, Guidewire Software**

musixmatch

Leading music data company providing data, tools, and services that enrich the way we experience music; the largest in the world with over 80 million users and over 8 million distinct lyrics

> “*We are always looking for ways to accelerate training time while also lowering training costs, which is why we are excited about Amazon SageMaker Training Compiler. SageMaker Training Compiler provides more efficient ways to use GPUs during the training process and, with the seamless integration between SageMaker Training Compiler, PyTorch, and high-level libraries like Hugging Face, we have seen a significant improvement in training time of our transformer-based models – going from weeks to days – as well as lower training costs.”*
>
> **Loreto Parisi, AI Engineering Director, Musixmatch**

Quantum Health is on a mission to make healthcare navigation smarter, simpler, and more cost-effective for everyone; they use Amazon SageMaker for use cases like text classification, text summarization, predictive models, classification problems, and Q&A to help the Quantum team and the members they serve

> *Iterating with NLP models can be a challenge because of their size; long training times bog down workflows, and high costs can discourage our team from trying larger models that might offer better performance. Amazon SageMaker Training Compiler is exciting because it has the potential to alleviate these frictions. Achieving a speedup with SageMaker Training Compiler is a real win for our team that will make us more agile and innovative moving forward.”*

Jorge Lopez Grisman, Senior Data Scientist, Quantum Health

Demo

SageMaker Training Compiler in summary

Accelerate DL model training by up to 50%

Free to use on Amazon SageMaker

Generally available
us-east-1, us-east-2
us-west-2, eu-west-1

Roadmap

Launch

- Tested with HF encoder models
- Hugging Face DLCs on SM
- Native distributed data parallelism

2022

- More testing with more models
- PT, TF DLCs on SM
- SM distributed training library support

Resources

- Example notebooks
  https://github.com/aws/amazon-sagemaker-examples/tree/master/sagemaker-training-compiler/

- Technical documentation
  https://docs.aws.amazon.com/sagemaker/latest/dg/training-compiler.html

- Amazon SageMaker Model Training web page
  https://aws.amazon.com/sagemaker/train

Thank you!

Robert Van Dusen
robdusen@amazon.com