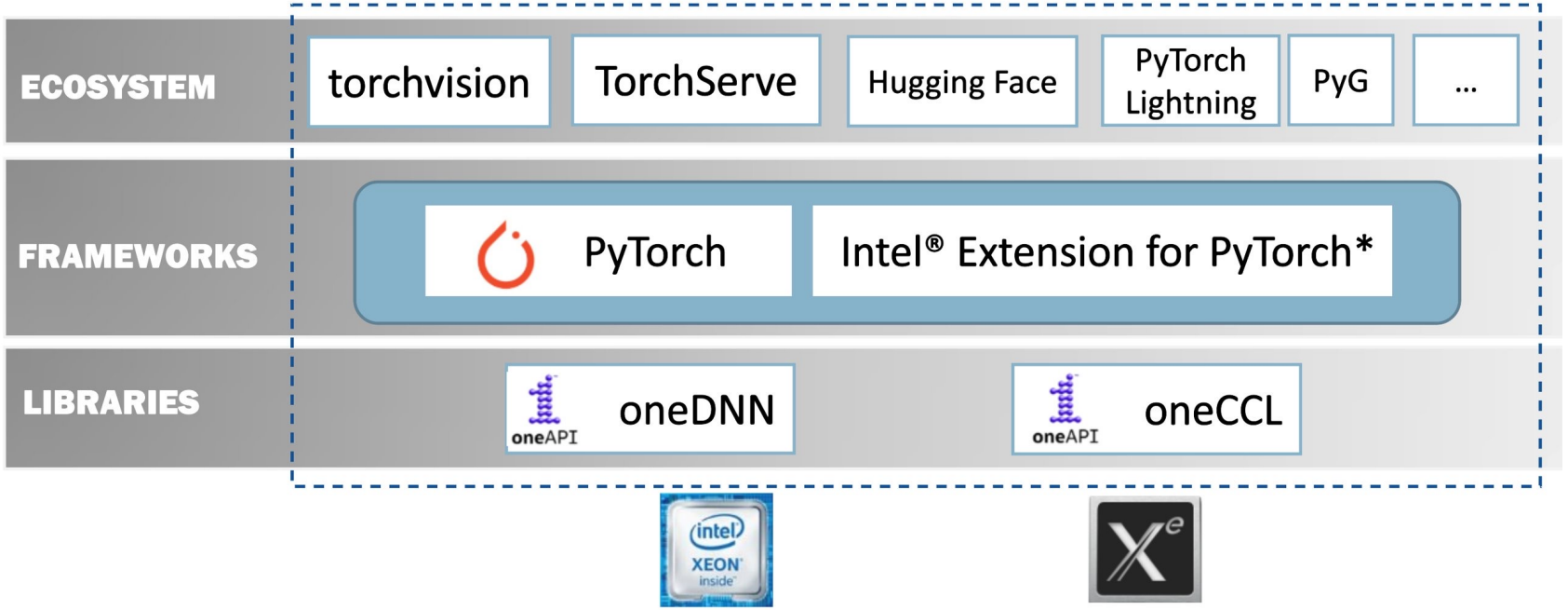
Accelerating PyTorch Deep Learning Models on Intel XPU

Agenda

- Overview
- Intel Optimizations for PyTorch
- Intel Extension for PyTorch
- Performance showcase & ecosystem

- Hands-on lab

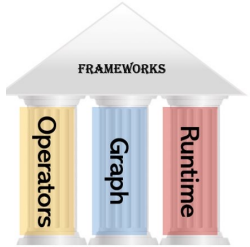
Intel Optimizations for PyTorch - Overview



Other names and brands may be claimed as the property of others

Major Optimization Methodologies

3-Pillar Framework Optimization Techniques



Op

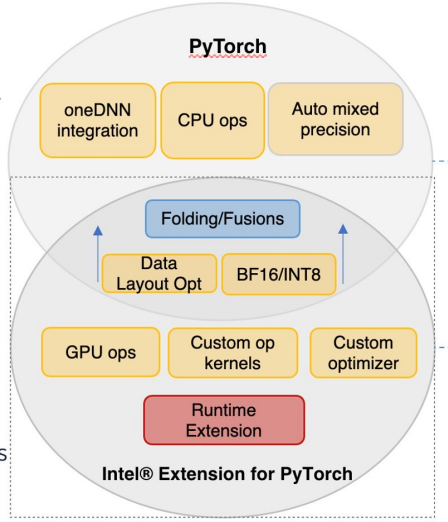
- Vectorization and Multi-threading
- Low-precision BF16/INT8 compute
- Ease-of-use BF16 compute with Auto-Mixed-Precision (AMP)
- Data layout optimization for better cache locality

Graph

- Constant folding to reduce compute
- Op fusion for better cache locality

Runtime

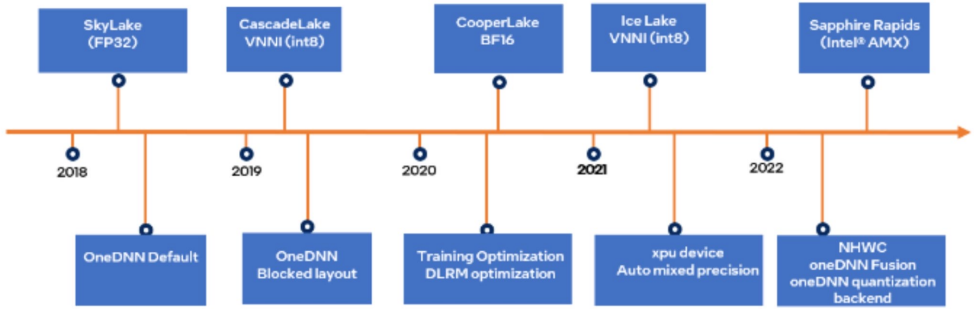
- Thread affinitization and multi-streams
- Memory buffer pooling
- GPU runtime
- Launcher



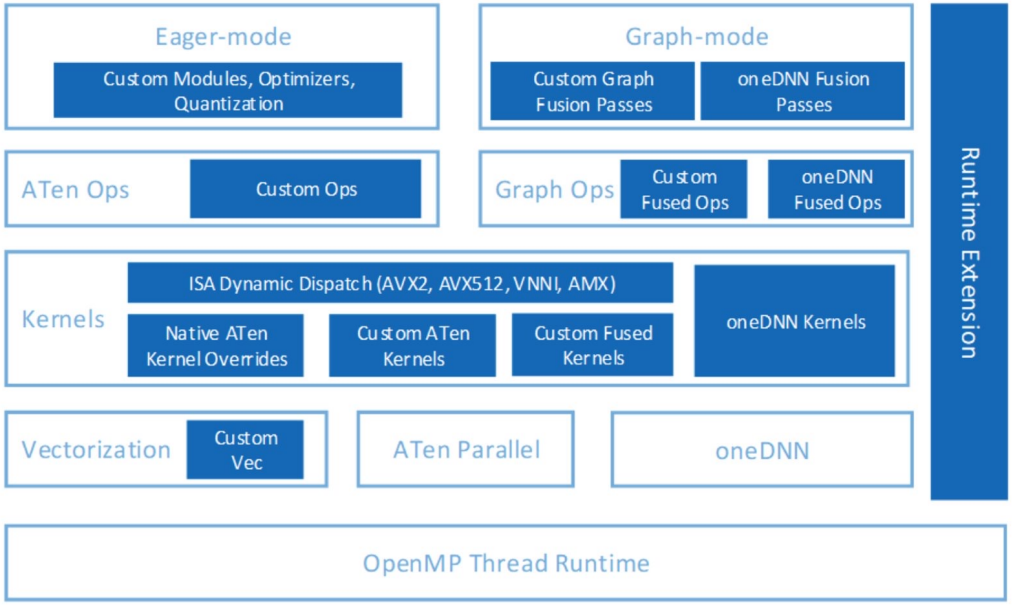
PyTorch Upstream
General Intel new feature enabling and performance optimizations

Intel Extension for PyTorch
Early access/adoption of aggressive optimizations & GPU support

PyTorch Upstream Optimizations



Intel* Extension for PyTorch* - Overview



Intel* Extension for PyTorch* usage

- FP32

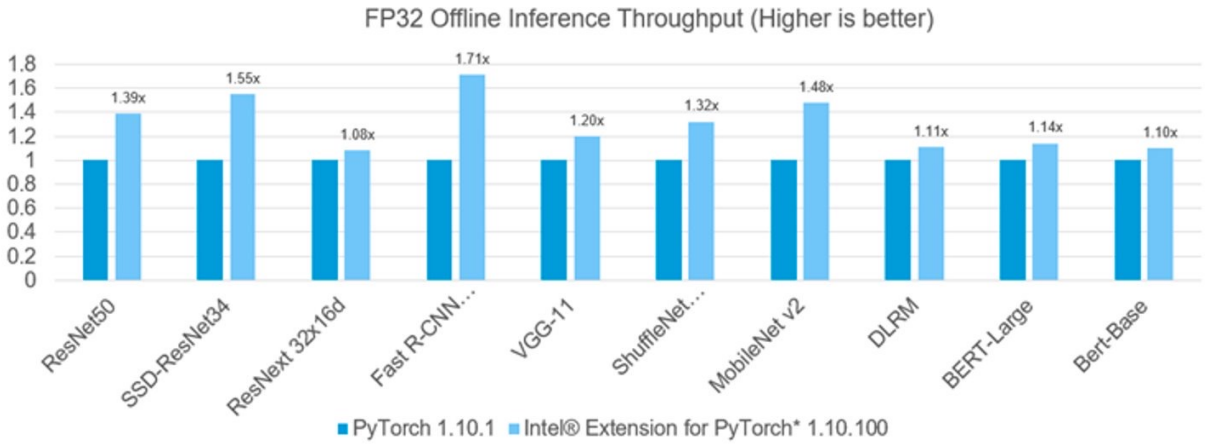
```
import torch
import torchvision.models as models

model = models.resnet50(pretrained=True)
model.eval()
data = torch.rand(1, 3, 224, 224)

import intel_extension_for_pytorch as ipex
model = model.to(memory_format=torch.channels_last)
model = ipex.optimize(model)
data = data.to(memory_format=torch.channels_last)

with torch.no_grad():
  model(data)
```

- BFloat16

```
import torch
from transformers import BertModel

model = BertModel.from_pretrained(args.model_name)
model.eval()

vocab_size = model.config.vocab_size
batch_size = 1
seq_length = 512
data = torch.randint(vocab_size, size=[batch_size, seq_length])

import intel_extension_for_pytorch as ipex
model = ipex.optimize(model, dtype=torch.bfloat16)

with torch.no_grad():
  with torch.cpu.amp.autocast():
    model(data)
```

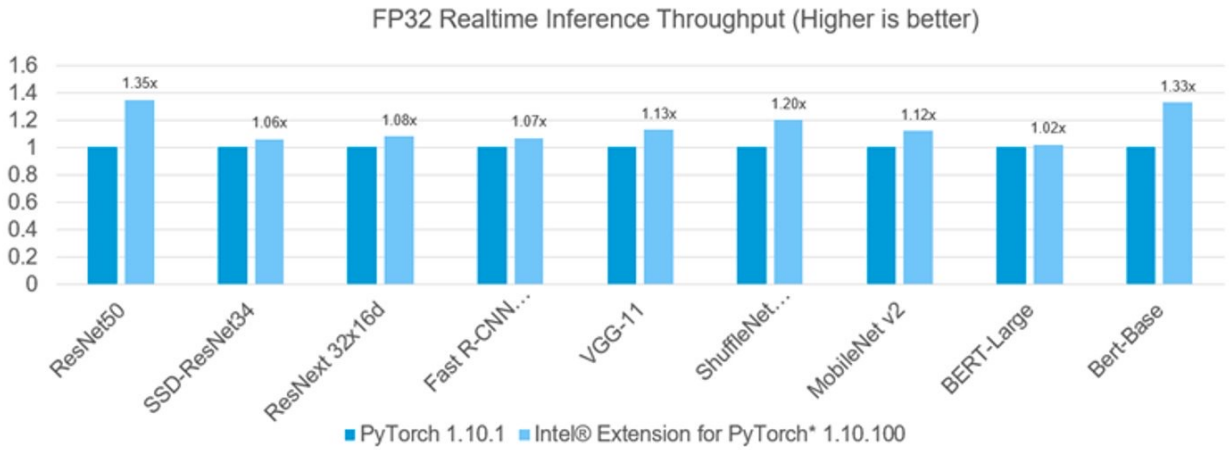
Intel* Extension for PyTorch*

Performance Snapshot on Intel(R) Xeon(R) Platinum 8380 CPU @ 2.3 GHz

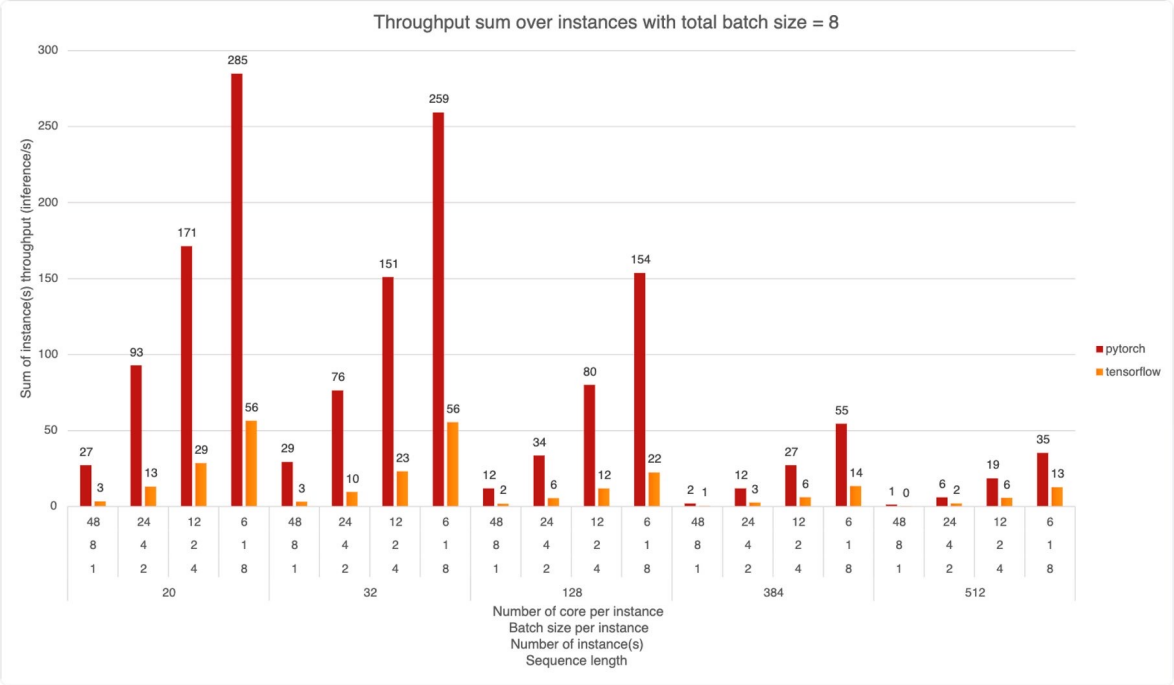


Intel* Extension for PyTorch*

Performance Snapshot on Intel(R) Xeon(R) Platinum 8380 CPU @ 2.3 GHz



Scaling up Hugging Face BERT-like model Inference



Source - https://huggingface.co/blog/bert-cpu-scaling-part-1

Hands-on Lab

PyTorch and Intel Extension for PyTorch *

Call to action

- Recommended to use latest PyTorch release
- https://github.com/pytorch/pytorch
- Intel® Extension for PyTorch*
- https://www.intel.com/content/www/us/en/developer/tools/oneapi/extension-for-pytorch.html
- Model Zoo
- https://github.com/IntelAI/models/tree/pytorch-r1.12-models


intel