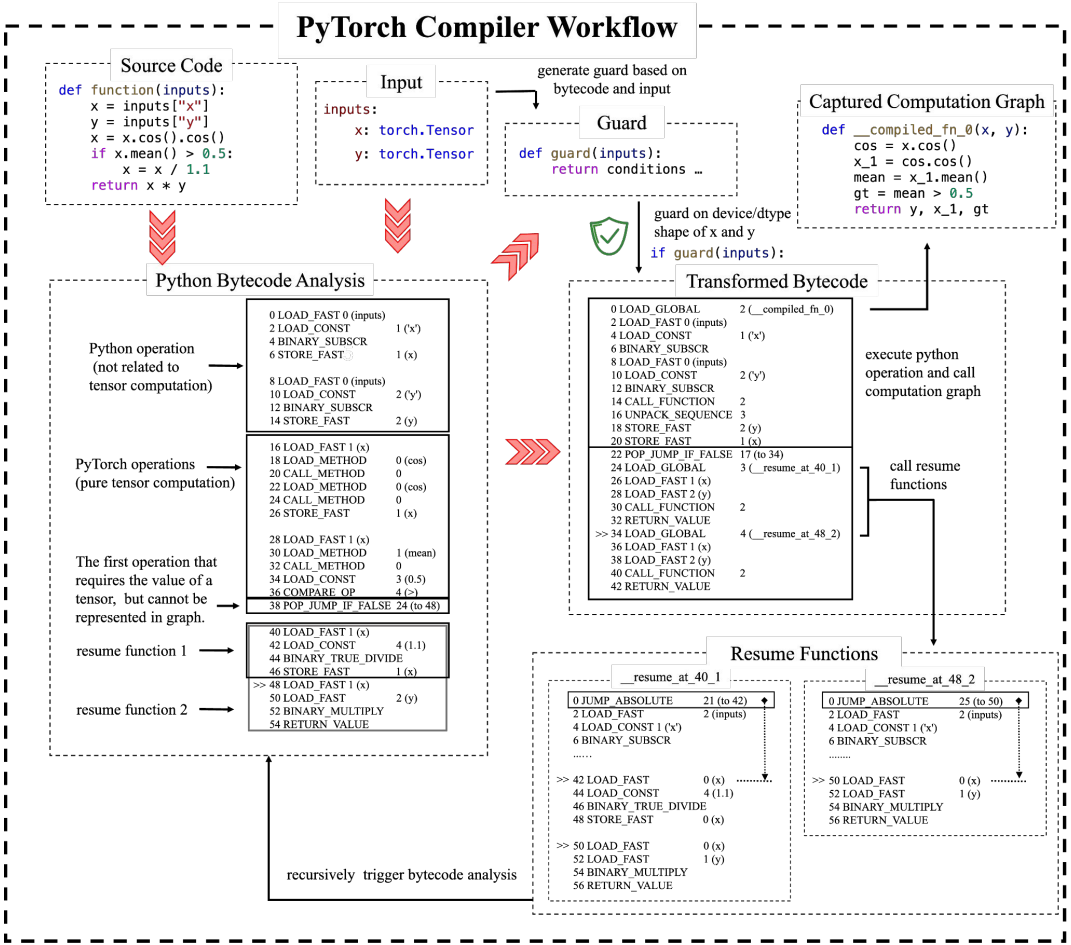
Exploring JIT compilation in TorchDynamo

Andrzej Szablewski

How to make my PyTorch models faster?

torch.compile

- JIT
- Operates on the Python Bytecode level
- Supports various target architectures through different backends

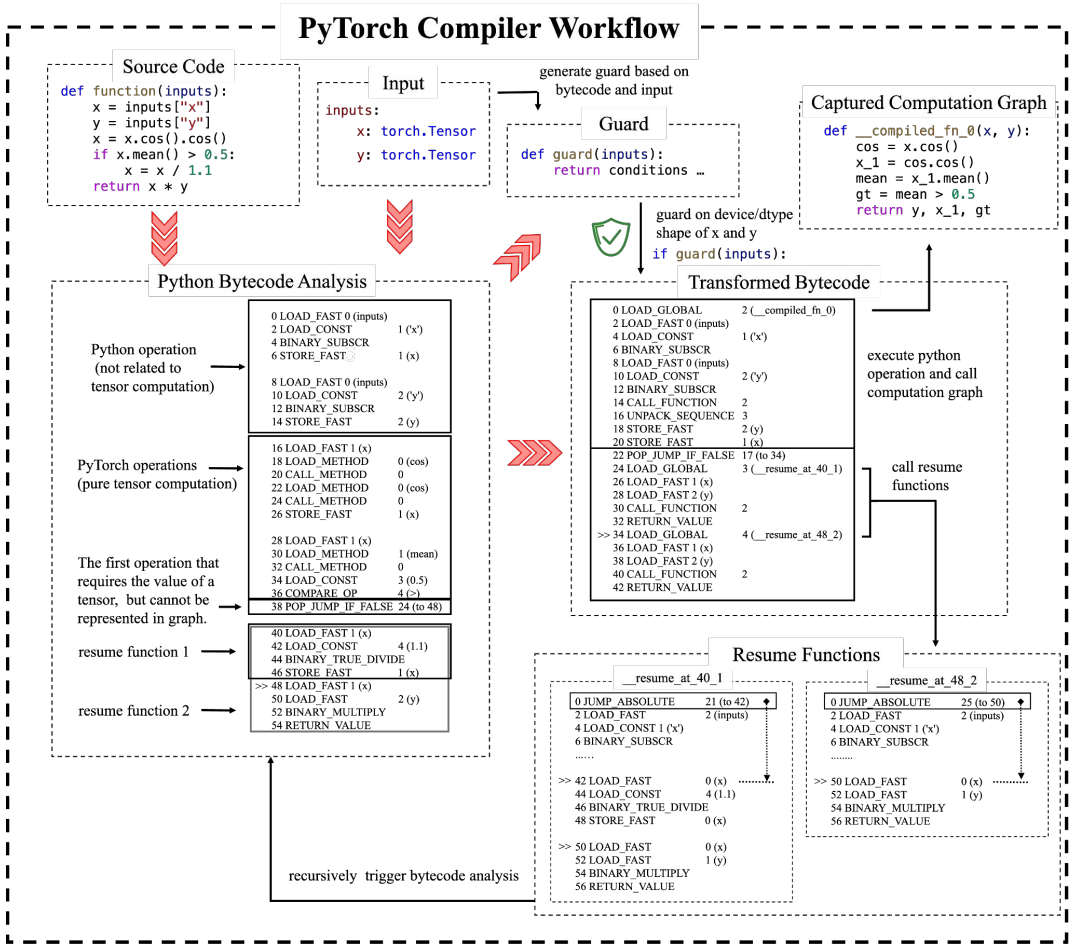


https://depyf.readthedocs.io/en/latest/walk_through.html

torch.compile

3 elements:

- TorchDynamo - tracer
- AoT Autograd - another tracer
- Inductor - compiler backend



https://depyf.readthedocs.io/en/latest/walk_through.html

torch.compile

```python
# ex.py
import torch

@torch.compile
def lerp(x, y, w):
    return w * x + (1 - w) * y

x = torch.randn(200)
y = torch.randn(200)
w = torch.rand(200)
result = lerp(x, y, w)
```

```cpp
// Simplified output of running
// TORCH_LOGS=output_code python ex.py
void lerp(const float* w_ptr,
          const float* x_ptr,
          const float* y_ptr,
          float* out_ptr0) {
    for(long x0=0; x0<200; x0+=1L) {
        auto tmp0 = w_ptr[x0];
        auto tmp1 = x_ptr[x0];
        auto tmp5 = y_ptr[x0];
        auto tmp2 = tmp0 * tmp1;
        auto tmp3 = c10::convert<double>(1.0);
        auto tmp4 = tmp3 - tmp0;
        auto tmp6 = tmp4 * tmp5;
        auto tmp7 = tmp2 + tmp6;
        out_ptr0[x0] = tmp7
    }
}
```

https://www.youtube.com/watch?v=Z3dL2D2BEHE

Modes of Execution in Pytorch

Modes of Execution in Pytorch

Eager mode

- Operations are executed immediately when they are encountered
- Intuitive and flexible

Modes of Execution in Pytorch

Eager mode

- Operations are executed immediately when they are encountered
- Intuitive and flexible

Graph mode

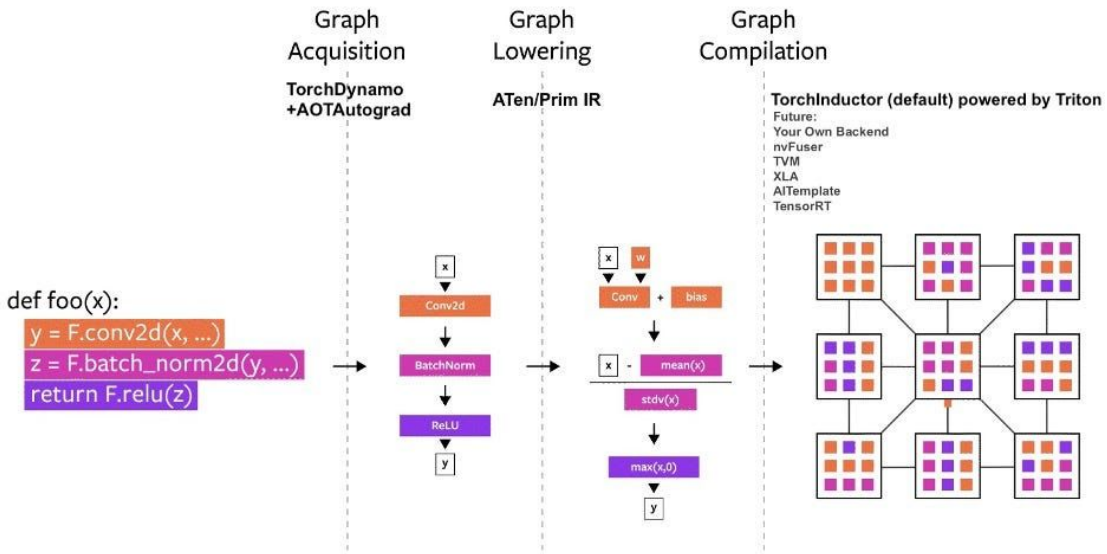
- Delays execution after the forward pass is performed
- Creates a computational graph from operations, allowing for optimisations

Modes of Execution in Pytorch

Eager mode

```
def f1(x, y):
    if x.sum() < 0:
        return -y
    return y
```

Graph mode



https://pytorch.org/get-started/pytorch-2.0/

How to Capture a Computation Graph in PyTorch?

Eager mode

- Graph created on the fly!
- Super helpful for debugging

A graph is created on the fly


```
W_h = torch.randn(20, 20, requires_grad=True)
W_x = torch.randn(20, 10, requires_grad=True)
x = torch.randn(1, 10)
prev_h = torch.randn(1, 20)
```

the nuances of pytorch graph capture

https://pytorch.org/tutorials/beginner/Intro_to_TorchScript_tutorial.htm

How to Capture a Computation Graph in PyTorch?

Graph mode

```
def f1(x, y):
    x = torch.sin(x)
    x = torch.matmul(x, y)
    if x.sum() < 0:
        return -y
    return y
```

How to Capture a Computation Graph in PyTorch?

Graph mode

- ???

```
def f1(x, y):
    x = torch.sin(x)
    x = torch.matmul(x, y)
    if x.sum() < 0:
        return -y
    return y
```

How to Capture a Computation Graph in PyTorch?

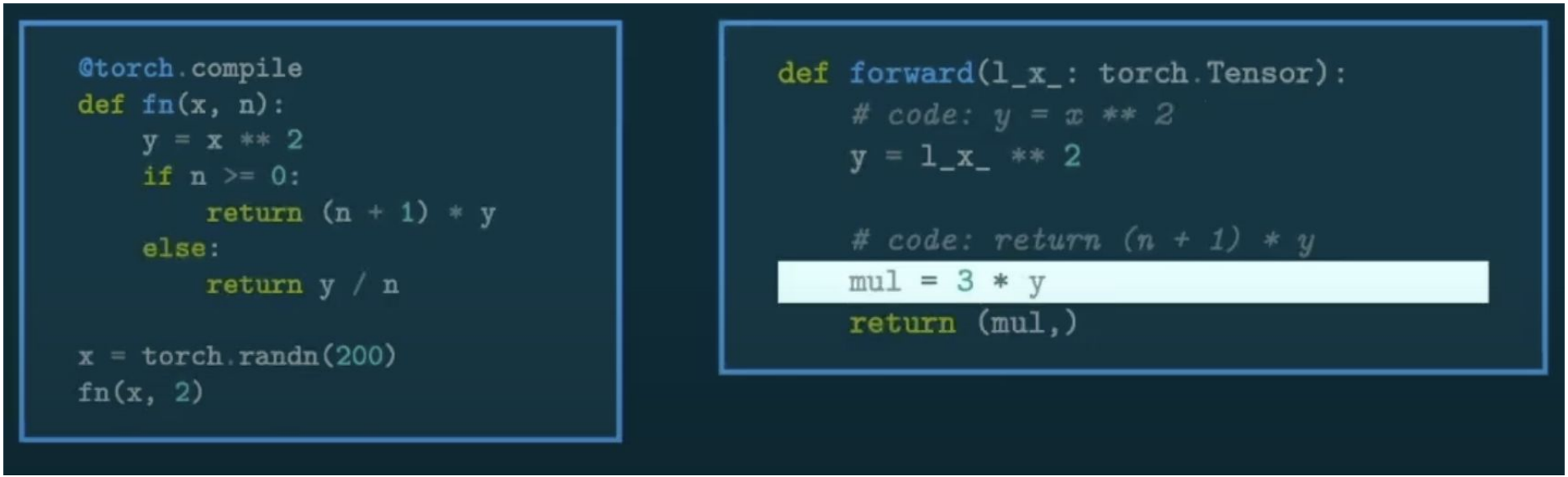
```python
@torch.compile
def mse(x, y):
    z = (x - y) ** 2
    return z.sum()

x = torch.randn(200)
y = torch.randn(200)
mse(x, y)
```

```python
def forward(l_x_: torch.Tensor,
            l_y_: torch.Tensor):
    # File: ex.py:30, code: z = (x - y) ** 2
    sub = l_x_ - l_y_
    z = sub ** 2
    # File: ex.py:31, code: return z.sum()
    sum_1 = z.sum()
    return (sum_1,)
```

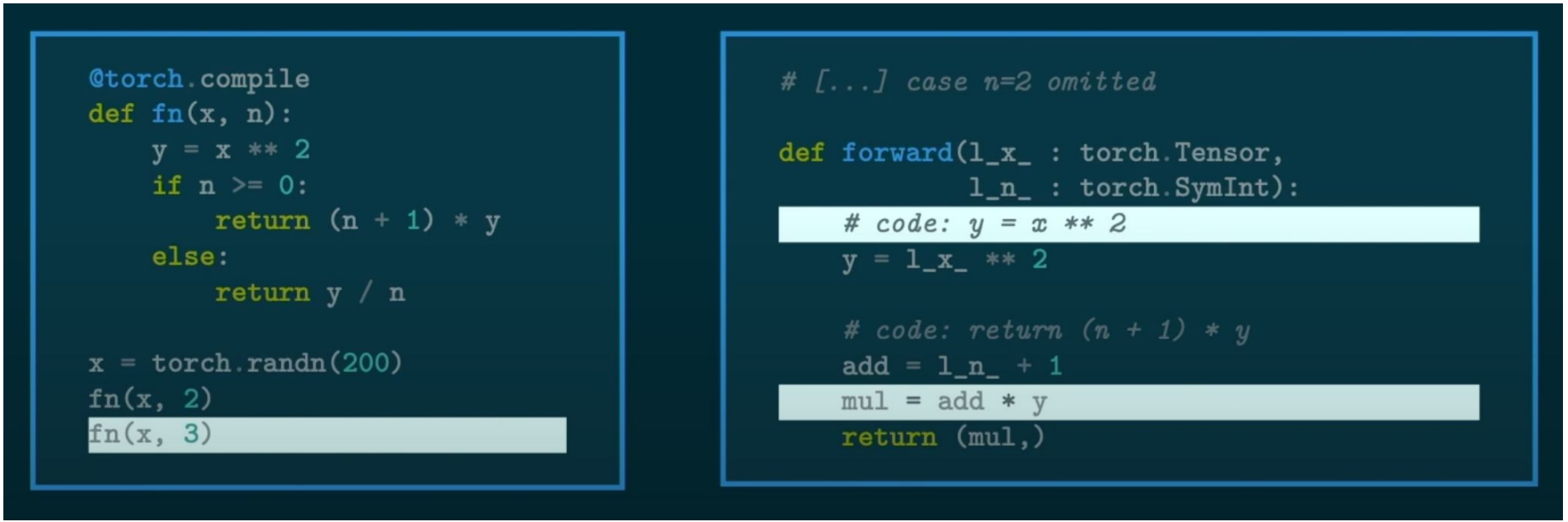
https://www.youtube.com/watch?v=Z3dL2D2BEHE

How to Capture a Computation Graph in PyTorch?



https://www.youtube.com/watch?v=Z3dL2D2BEHE

How to Capture a Computation Graph in PyTorch?



https://www.youtube.com/watch?v=Z3dL2D2BEHE

How to Capture a Computation Graph in PyTorch?

```python
@torch.compile
def fn(x, n):
    y = x ** 2
    if n >= 0:
        return (n + 1) * y
    else:
        return y / n

x = torch.randn(200)
fn(x, 2)
fn(x, 3)
fn(x, 6)   # can use n >= 0
fn(x, -3) # retrace!
```

```python
# [...] case n==2 omitted
# [...] case n>=0 omitted

def forward(l_x_ : torch.Tensor,
            l_n_ : torch.SymInt):
    # code: y = x ** 2
    y = l_x_ ** 2

    # code: return y / n
    truediv = y / l_n_
    return (truediv,)
```

https://www.youtube.com/watch?v=Z3dL2D2BEHE

How to Capture a Computation Graph in PyTorch?

Graph mode (static graphs)

- `torch.jit.script`
- `torch.jit.trace`

- Require changes to the code

```
def f1(x, y):
    x = torch.sin(x)
    x = torch.matmul(x, y)
    if x.sum() < 0:
        return -y
    return y
```

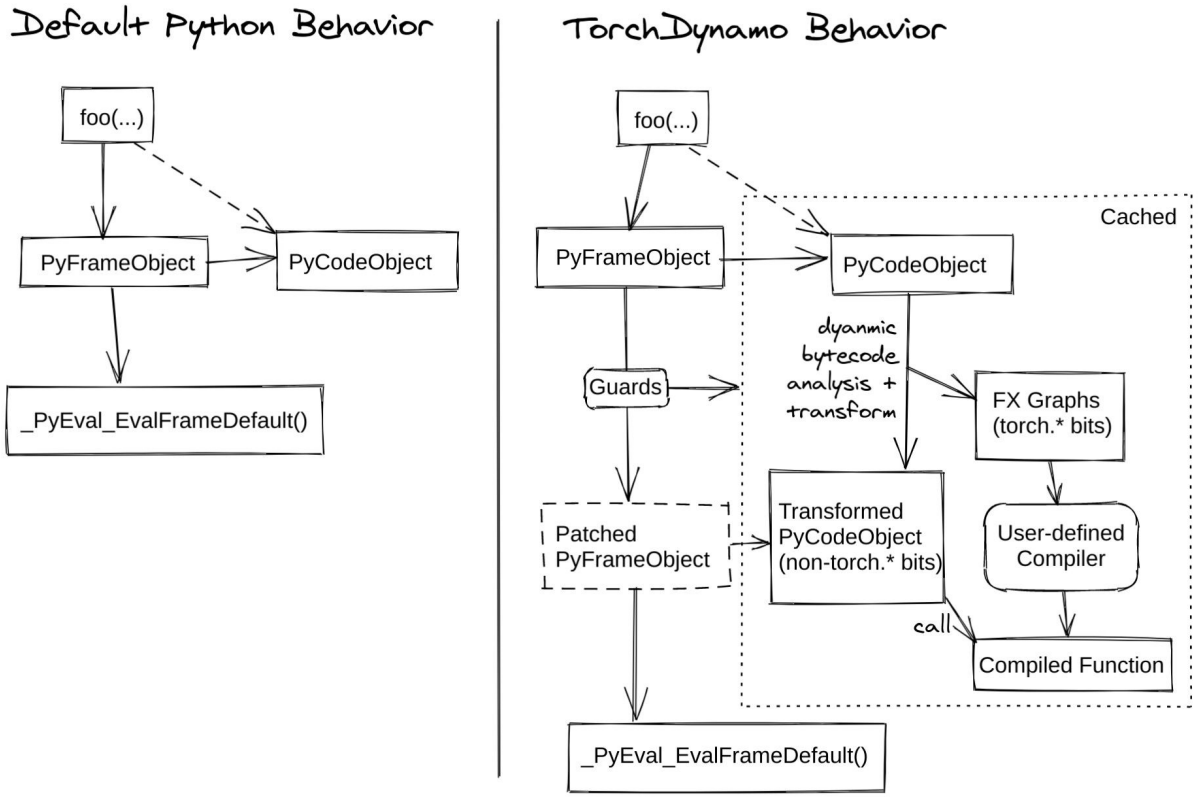
How to Capture a Computation Graph in PyTorch?

Graph mode (TorchDynamo)

- Uses CPython frame evaluation API
- Splits code into graph and non-graph parts
- Optimises the graph

```
def f1(x, y):
    x = torch.sin(x)
    x = torch.matmul(x, y)
    if x.sum() < 0:
        return -y
    return y
```

TorchDynamo



https://pytorch.org/docs/stable/torch.compiler_dynamo_overview.html

Question

How different neural architectures are represented in FX graph and what optimisations they allow for?

Question

How different neural architectures are represented in FX graph and what optimisations they allow for?

- Explore the graph capture and optimisation of NN components (e.g. FFNN, attention), including dynamic model shapes with TorchDynamo and TorchInductor
- Consider custom hardware-specific optimisations (e.g. operation fusion, memory layout)
- Evaluate various configurations of the `torch.compile()` framework on real-world use-cases

Thank you!