## Design Principles

- Be Pythonic
- Put Researchers First
- Pragmatic Performance
- 'Worse is Better'

#### Features

- Everything is a program
   Models, optimizers, data loader, training loops, etc... are python code
- Interoperability
   Efficient data exchange with other tools
- Autodiff
   Native support for reverse-mode automatic

## Example

differentiation

```
class LinearLayer(Module):
  def __init__(self, in_sz, out_sz):
     super().__init__()
     t1 = torch.randn(in_sz, out_sz)
     self.w = nn.Parameter(t1)
     t2 = torch.randn(out_sz)
     self.b = nn.Parameter(t2)
  def forward(self, activations):
     t = torch.mm(activations, self.w)
     return t + self.b
class FullBasicModel(nn.Module):
  def __init__(self):
      super().__init__()
      self.conv = nn.Conv2d(1, 128, 3)
      self.fc = LinearLayer(128, 10)
  def forward(self, x):
      t1 = self.conv(x)
      t2 = nn.functional.relu(t1)
      t3 = self.fc(t1)
      return nn.functional.softmax(t3)
```

facebook Artificial Intelligence

# PyTorch

An Imperative Style, High-Performance Deep Learning Library

Designed to be highly usable

Implemented carefully to be fast

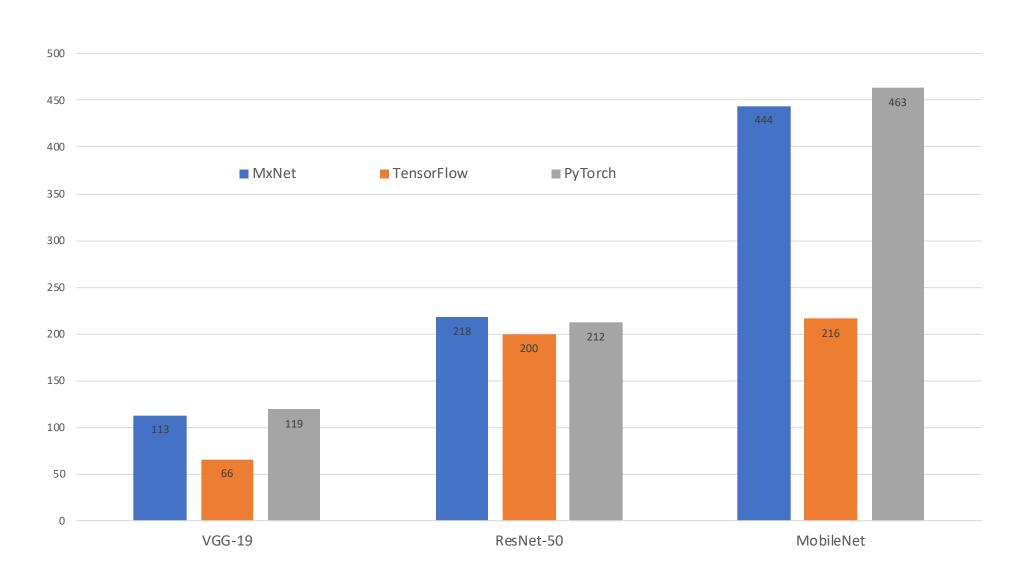


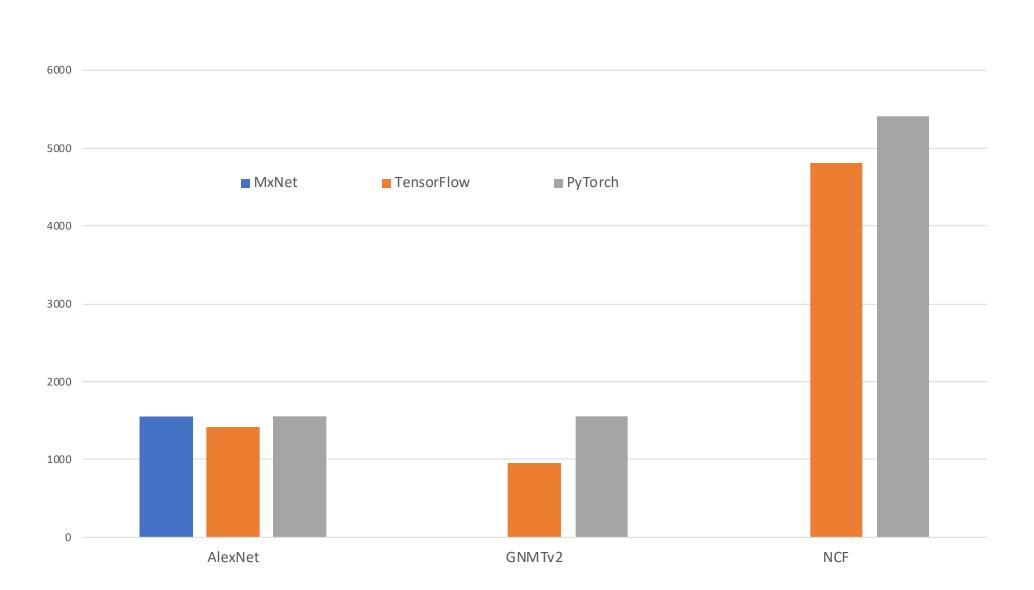
Full paper at http://github.com/benoitsteiner/misc/blob/master/pytorch\_paper.pdf

## Implementation

- Efficient C++ core
  - Avoids most of the Python overhead, especially the GIL
  - Can be used independently of Python when Python isn't practical (e.g mobile devices)
- Separate control and data flow
  - Execute operators asynchronously
- Overlaps interpretation of the program with computation of the op kernels
- Custom tensor allocator
  - Tuned for typical ML workloads
  - Limits memory fragmentation
- Multiprocessing
- Leverages shared memory to communicate between processes
- Reference Counting
- Frees user from managing memory
- Unlike GC, releases memory as soon as possible thus decreasing peak usage

### Benchmarks





Training speed for 6 models using 32bit floats. Throughput is measured in images per second for the VGG-19, ResNet-50, MobileNet, and AlexNet models, in tens of tokens per second for the GNMTv2 model, and in kilo-samples per second for the NCF model.