# TVM BENCHMARK

2025-10-21 김규진

# TVM BenchMark

```python
class PyTorchModel(torch_nn.Module):
    def __init__(self):
        super(PyTorchModel, self).__init__()
        self.fc1 = torch_nn.Linear(784, 256)
        self.relu1 = torch_nn.ReLU()
        self.fc2 = torch_nn.Linear(256, 10, bias=False)

    def forward(self, x):
        x = self.fc1(x)
        x = self.relu1(x)
        x = self.fc2(x)
        return x
```

## Tensor

```
================================================================
BENCHMARKING PYTORCH MODEL
================================================================
✓ Warming up PyTorch inference...
✓ Benchmarking PyTorch inference...

PyTorch Results (over 100 runs):
  Average: 0.02 ms
  Std Dev: 0.01 ms
  Min:     0.01 ms
  Max:     0.10 ms

================================================================
BENCHMARKING TVM MODEL
================================================================
✓ Warming up TVM inference...
✓ Benchmarking TVM inference...

TVM Results (over 100 runs):
  Average: 0.28 ms
  Std Dev: 0.05 ms
  Min:     0.25 ms
  Max:     0.63 ms
```

## Resnet 18 - TVM

```
================================================================
TVM CPU-OPTIMIZED INFERENCE TIME (over 100 runs)
================================================================
  Average: 1959.64 ms
  Std Dev: 22.03 ms
  Min:     1914.91 ms
  Max:     2058.70 ms
================================================================

✓ Inference completed - Output shape: (1, 1000)

================================================================
TOP-5 PREDICTIONS FOR DOG IMAGE
================================================================
1. Pomeranian                  39.73%
2. Samoyed                     22.64%
3. keeshond                     6.05%
4. Japanese spaniel             5.42%
5. Arctic fox                   4.58%
```

## Resnet 18 - Torch

```
================================================================
PYTORCH INFERENCE TIME (over 100 runs)
================================================================
  Average: 13.89 ms
  Std Dev: 1.69 ms
  Min:     11.18 ms
  Max:     19.58 ms
================================================================

✓ Inference completed - Output shape: (1, 1000)

================================================================
TOP-5 PREDICTIONS FOR DOG IMAGE
================================================================
1. Pomeranian                  39.73%
2. Samoyed                     22.64%
3. keeshond                     6.05%
4. Japanese spaniel             5.42%
5. Arctic fox                   4.58%
```

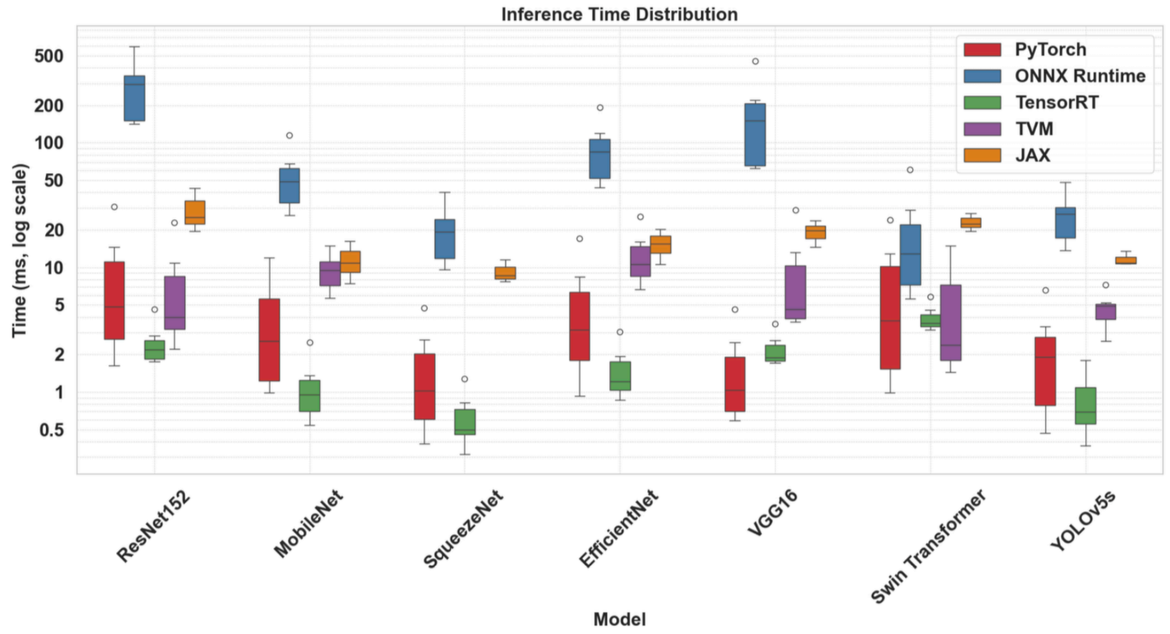
# 진행사항

- PyTorch: Input → C++ Kernel → Output
- TVM: Input → VM Interpreter → Compiled Function → Output
- Function OverHead? ->  딜레이에 미치는 요소가 큰 비율은 아님
- MetaSchedule (AutoTuning)? -> 이게 핵심인데 tvm relax 의 스케쥴링이 모델 로그 스케일이 커짐에 따라 런타임 딜레이가 줄어들도록 설계하고있음  but 아직 0.22dev 버전이기때문에 지원하지 않는 op가 많아  큰 모델을 테스트해볼 수가 없음
- Tensor IR 을 그대로 사용했을때  llvm core 스케쥴링이 들어가도 torch 가 더 빠른게 의문

# 참고

- https://www.mdpi.com/2079-9292/14/15/2977?utm_source=chatgpt.com

**Figure 1.** Inference Time Distribution (y axis in log scale).



# 다음 단계

- Tensor RT Pipeline
- TVM Virtual Machine Scheduling