TensorFlow and PyTorch on Arm Servers

Linaro Virtual Connect 2020

Ashok Bhat, Sr Product Manager
Sep 2020
Agenda

TensorFlow and PyTorch on Arm Servers for on-CPU inference

• What is the end goal?
• What is available today for end users?
• What is being worked on to improve usability and performance?
• How to get involved?

Not being covered

• Training use-case
• Machine learning using Arm + GPU
• Benchmarks and performance comparison
On-CPU Machine Learning (Inference)

Goal: Easy to use, best-in-class performance, ML inference solution on Arm servers using ML specific CPU features

<table>
<thead>
<tr>
<th>Easy to use</th>
<th>Wide variety of inference workloads</th>
<th>Using Arm architecture features</th>
<th>On latest Arm based hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>Container images and Python Packages</td>
<td>Popular ML frameworks support Arm as a first-class citizen</td>
<td>Large core count, INT8, Bfloat16, FP32, Matrix Multiplier Extension, SVE/2</td>
<td>Arm Neoverse N1, V1 (Zeus), N2 (Perseus)</td>
</tr>
<tr>
<td>Image classification</td>
<td>Object detection</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Wide variety of inference workloads:
- Image classification
- Object detection
- ...
AArch64 packages and images

For machine learning users on Arm servers
Ready-to-use Python Packages

Goal: Readily available TensorFlow and PyTorch packages from standard repositories

Current status

• TensorFlow 1.15 and 2.3 package snapshots available
  • https://snapshots.linaro.org/hpc/python/tensorflow/latest/

• PyTorch nightly package snapshots available
  • https://snapshots.linaro.org/hpc/python/pytorch/latest

Next steps

• Snapshots – Add support for PyTorch v1.6
• Linaro hosted packages – Host ready-to-use packages for TensorFlow and PyTorch
• Work with upstream to provide AArch64 ready packages
Docker recipes

Goal: Recipe to build your own Docker images

Current status

- **TensorFlow**
  - [https://github.com/ARM-software/Tool-Solutions/tree/master/docker/tensorflow-aarch64](https://github.com/ARM-software/Tool-Solutions/tree/master/docker/tensorflow-aarch64)
  - Versions – 1.15 and 2.3
  - Configurations – Eigen backend, oneDNN(ArmPL), oneDNN(OpenBLAS)
- **PyTorch**
  - [https://github.com/ARM-software/Tool-Solutions/tree/master/docker/pytorch-aarch64](https://github.com/ARM-software/Tool-Solutions/tree/master/docker/pytorch-aarch64)
  - Versions – 1.6
  - Configurations – OpenBLAS backend

Next steps

- Upgrade recipes to newer releases
- Add oneDNN(ACL) based configuration
Docker images

Goal: Readily available docker images on par with other architectures

Current status

- Images for Arm Neoverse N1 is available in a staging area
  - TensorFlow 2.3 with Eigen, oneDNN (ArmPL)
  - PyTorch 1.6 with OpenBLAS

Next steps

- Upgrade images to newer releases
- Add oneDNN(ACL) based configuration
- Add images tuned for Fujitsu A64FX
- Provide images on Linaro Docker Hub repo
- Work with upstream to provide images in standard repositories
Best-in-class performance using ML-specific Arm features
Key open source projects

- TensorFlow
- PyTorch
- OpenBLAS
- Eigen
- oneDNN
- Arm Compute Library
Key open source projects for ML on Servers

Frameworks

• ML Framework - TensorFlow
  • Popular open source ML framework
  • Has multiple backends on x86 – Eigen GEBP, oneDNN (with BLAS, direct kernels, JIT)

• ML Framework - PyTorch
  • Popular ML framework
  • Has multiple backends on x86 – NNPACK, OpenBLAS, oneDNN (with BLAS, direct kernels, JIT)
Key open source projects for ML on Servers

Libraries

• Library - Eigen
  • Eigen is a C++ template library for linear algebra: vectors, matrices, and related algorithms
  • TensorFlow heavily uses Eigen to represent internal data structures and their operations.
  • Eigen’s GEBP kernel is used as a default CPU backend for FP32 contraction kernel

• Library - oneDNN
  • Intel’s ML acceleration open-source library – Integrated with all major frameworks
  • Experimental support for AArch64

• Library - Arm Compute Library
  • Open source ML acceleration library for Arm used in edge/mobile use-cases
  • Contains high level operators which can be used in oneDNN

• Library - OpenBLAS
  • Most common open source BLAS backend
Arm Compute Library

A software library for computer vision and machine learning

- Collection of low-level functions
  - Optimized for Arm CPU and GPU architectures
  - Targeted at image processing, computer vision, and machine learning.
- Available free of charge under a permissive MIT open source license.
- Used to accelerate ArmNN (Arm’s inference engine for CPUs, GPUs and NPUs)
oneDNN primitives – Implementation options

<table>
<thead>
<tr>
<th>Primitive implementation</th>
<th>Notes</th>
<th>x86_64 libraries</th>
<th>Arm libraries</th>
</tr>
</thead>
<tbody>
<tr>
<td>C++ reference code</td>
<td>Slowest, provided for correctness</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>GEMM based</td>
<td>Use BLAS library via CBLAS interface</td>
<td>Intel MKLML (a subset of Intel MKL)</td>
<td>Arm Performance Libraries (ArmPL)</td>
</tr>
<tr>
<td>Optimized primitives</td>
<td>Typically uses hand-coded intrinsic or assembly</td>
<td>xybak (JIT assembler)</td>
<td>Arm Compute Library</td>
</tr>
</tbody>
</table>
TensorFlow software stack on Arm – Status and Plan

**Timeline** | **Library options**
---|---
2.2 (May 20) | Eigen (FP32)
2.3 (Jul 20) | Eigen (FP32)
2.4* (Q4, 20) | Eigen (FP32)
oneDNN (ArmPL via CBLAS)
2.5* (Q1, 21) | Eigen (+SVE)
oneDNN (ArmPL via CBLAS)
oneDNN (ACL) (FP32, INT8)

* Future release information (version and date) is Arm’s estimate based on previous releases.

Work in progress | Available
---|---
PyTorch software stack on Arm – Status and Plan

ML inference workloads

PyTorch

OpenBLAS

oneDNN

Arm backend

Arm Compute Library (ACL) operators

Hardware

Timeline | Library options
--- | ---
1.6 (Jul 20) | OpenBLAS (FP32)
1.7* (Oct 20) | OpenBLAS (FP32)
1.8* (Dec 20) | OpenBLAS (FP32), oneDNN (ACL) (FP32, INT8)

* Future release information (version and date) is Arm’s estimate based on previous releases.

Work in progress | Available
## Data type support

### Status and plan

<table>
<thead>
<tr>
<th>Data type</th>
<th>TF Eigen</th>
<th>TF ACL (via oneDNN)</th>
<th>PyTorch OpenBLAS</th>
<th>PyTorch ACL (via oneDNN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP32 type</td>
<td>Yes</td>
<td>Q4 CY20</td>
<td>Yes</td>
<td>Q4 CY20</td>
</tr>
<tr>
<td>INT8 type</td>
<td>Not planned</td>
<td>Q4 CY20</td>
<td>Not planned</td>
<td>Q4 CY20</td>
</tr>
<tr>
<td>BF16 type</td>
<td>Q4 CY20</td>
<td>Future</td>
<td>Not planned</td>
<td>Future</td>
</tr>
</tbody>
</table>

[Not planned]
Wrap Up
Get involved in Machine Learning on Arm

<table>
<thead>
<tr>
<th>Try</th>
<th>Try the Docker recipes/images to run TensorFlow and PyTorch on AArch64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learn</td>
<td>Learn about the Arm Compute Library</td>
</tr>
<tr>
<td>Provide Feedback</td>
<td>Provide feedback on performance for your applications on Arm machines</td>
</tr>
<tr>
<td>Get involved</td>
<td>Get involved in the open source development of ML inference on Arm</td>
</tr>
</tbody>
</table>
Thank You
Danke
Merci
Merci
Merci
Merci
Merci
감사합니다
धन्यवाद
شكرًا
ধন্যবাদ
תודה