Intel® Optimized AI Libraries & Frameworks
2023 Q1

Using optimized AI software can significantly improve AI workload performance, developer productivity, and compute resource usage costs. Intel® oneAPI libraries enable the AI ecosystem with optimized software, libraries, and frameworks. Software optimizations include leveraging accelerators, parallelizing operations, and maximizing core usage.

We encourage you to check out Intel's full suite of AI tools and framework optimizations. For more optimization packages and tuning guides, visit the Intel® Optimization Hub.

TensorFlow* v2.5+

Intel optimizations delivering up to 3x faster deep learning[1] are **upstreamed** into the main branch:

```
pip install tensorflow
```

For versions 2.5-2.8: **enable with the** environment variable:

```
export TF_ENABLE_ONEDNN_OPTS=1
```

For versions 2.9+: **ON** by default.

Cheat Sheet | v2.5-2.8 Tuning Guide | v2.9+ Blog Post

XGBoost* v1.x+

Optimizations for training and prediction on CPU are **upstreamed.**

Download the latest XGBoost – newer versions have the most optimizations.

Optimized **methods** include: split, partitioning, and hist tree method.

```
'tree_method: hist,' #try hist tree
```

Docs | Cheat Sheet | Example

PyTorch* v1.5+

Intel **upstreams** optimizations to PyTorch. These features often debut in Intel® Extension for PyTorch*, which can speed performance up to 2.7x.[2]

Install open source PyTorch (Guide). Then install Intel Extension for PyTorch, choosing from:

```
pip install intel-extension-for-pytorch
conda install -c intel intel-extension-for-pytorch
docker pull intel/intel-optimized-pytorch
```

For previous versions of PyTorch, be sure to install the corresponding version of the extension. Details in the Installation Guide.

PyTorch Version	v1.13.*	v1.12.*	v1.11.*	v1.10.*	v1.9.0	v1.8.0	v1.7.0	v1.5.0-rc3
Extension Version	v1.13.*	v1.12.*	v1.11.*	v1.10.*	v1.9.0	v1.8.0	v1.2.0	v1.1.0

To enable these extensions, add these **two lines** to your Python* code:

```
import intel_extension_for_pytorch as ipex
model = ipex.optimize(your_model)
```

Documentation | Cheat Sheet | Examples | Tuning Guide

Support Forums:
AI Frameworks | SDKs and Libraries

*Names and brands may be claimed as the property of others.

Intel® Optimized AI Libraries & Frameworks
2023 Q1

scikit-learn*

Works with scikit-learn v0.22.x+

Install the extension, choosing from:

```
pip install scikit-learn-intelex
conda install scikit-learn-intelex
docker pull intel/intel-optimized-ml:scikit-learn
```

Activate the patch in your Python code:

```
from sklearnex import patch_sklearn
patch_sklearn()
```

Or run it without changing code:

```
python -m sklearnex my_application.py
```

With Intel® Extension for Scikit-learn*, you can accelerate up to 10-100x,[3] while conforming to scikit-learn APIs.

All it takes is **two lines of code**!

Documentation | Get Started | Cheat Sheet | Examples

Pandas*

Works with Pandas v1.3.4+

Install Modin*, choosing from:

```
pip install modin[ray]
conda install -c conda-forge modin-ray
```

Replace **import** in your Python code:

```
import modin.pandas as pd
```

Scale your Pandas workflows by changing **one line of code**.

Intel® Distribution of Modin* uses all your cores to speed DataFrame processing up to 10-100x[4].

Documentation | Get Started | Cheat Sheet | Examples

PaddlePaddle*

v2.1+

Intel's optimizations are **upstreamed** into the main branch, delivering automatic acceleration on Intel processors.

Blog Post | Documentation

DGL*

v0.8+

Intel's optimizations are **upstreamed** to Deep Graph Library (DGL) and on by default in compile.

Try distributed training on CPU!

Blog Post | Documentation

Intel® Distribution for Python*

v3.7.4+

```
conda install -c intelpython3_full python=3.x
```

Cheat Sheet

SciPy

v1.3.3+

Intel® oneAPI Math Kernel Library (oneMKL) optimizations accelerate scientific compute.

Install (Currently only available via **conda**):

```
conda install scipy
```

Learn More | Cheat Sheet

NumPy

v1.17.5+

Intel's optimizations use oneMKL to accelerate numerical compute.

Install (Currently only available via **conda**):

```
conda install numpy
```

Learn More | Cheat Sheet

Support Forums:
AI Frameworks | SDKs and Libraries

*Names and brands may be claimed as the property of others.

Intel® Optimized AI Libraries & Frameworks
2023 Q1

Apache Spark*

The Optimized Analytics Package (OAP) for Spark* Platform can optimize Spark, with open-source packages for RayDP integration, execution engine, and MLlib.

Install using **conda** for Spark v3.x:

```
conda create -n oapenv -c conda-forge -c intel -y oap=1.5.0.spark32
```

Then add configuration settings listed in the installation guide to this file:

```
$SPARK_HOME/conf/spark-defaults.conf
```

Installation Guide | Tuning Guide | OAP Project GitHub

Apache Spark MLlib v3.x

OAP MLlib accelerates machine learning algorithms in **Spark MLlib** up to 3x.[5]

It's **compatible** with Spark MLlib and **open-source**.

Installation Guide | Docs | GitHub

Apache Spark SQL v3.x

Gazelle Plugin is a native engine for Spark SQL.

It utilizes **Apache Arrow**, **SIMD kernels**, and **LLVM expression** for up to 2.5x faster performance.[6]

Installation Guide | Docs | Github

Apache Kafka* v3.x

Get the most out of your Kafka performance.

Tuning Guide

More optimizations: CatBoost, Apache MXNet*, ONNX RT*, Numba*, LightGBM

Intel® Neural Compressor

An open-source library to **compress and optimize** your AI models, with an average speedup of 2.2x.[7]

Available techniques include auto-quantization, pruning for sparsity, and knowledge distillation.

You can set accuracy loss limits, combine multiple compression techniques, and utilize built-in strategies to achieve objectives with expected accuracy.

Use the web application, or install for command-line use:

```
pip install neural-compressor
```

Publications | Documentation | Cheat Sheet | Examples

For **best practices**, check out open-source AI reference kits, which are end-to-end AI solution examples, optimized for Intel hardware.

Learn more about Intel's full suite of AI development tools and resources.

For **performance analysis** and profiling, see Intel® VTune™ Profiler.

Support Forums:
AI Frameworks | SDKs and Libraries

*Names and brands may be claimed as the property of others.

Intel® Optimized AI Libraries & Frameworks
2023 Q1

Notices & Disclaimers

Performance varies by use, configuration, and other factors. Learn more at www.Intel.com/PerformanceIndex. Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure. Your costs and results may vary. Intel technologies may require enabled hardware, software, or service activation. © Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.

Performance Claims

[1]https://medium.com/intel-analytics-software/leverage-intel-deep-learning-optimizations-in-tensorflow-129faa80ee07

[2]https://intel.github.io/intel-extension-for-pytorch/cpu/latest/tutorials/performance.html

[3]https://www.intel.com/content/www/us/en/developer/articles/technical/benchmarking-intel-extension-for-scikit-learn.html#gs.jnlmgq

[4]https://modin.readthedocs.io/en/latest/getting_started/why_modin/modin_vs_dask_vs_koalas.html#performance-comparison

[5] https://oap-project.github.io/gazelle_plugin/latest/User-Guide/#performance-data

[6]https://oap-project.github.io/gazelle_plugin/latest/User-Guide/#performance-data

[7]https://www.intel.com/content/www/us/en/developer/articles/technical/pytorch-inference-with-intel-neural-compressor.html#gs.k9o31y

Support Forums:
AI Frameworks | SDKs and Libraries
*Names and brands may be claimed as the property of others.