



ORGANISATION →  **IVY**
ACCELERATE YOUR AI

PROJECT TITLE : TensorLy - Multi-backend Tensor API Libraries

Name- [Amitesh Vatsa](#)

Github- [vtsamit](#)

Mentors- [Haris Mahmood](#), [Deeptendu Santra](#), [James Keane](#)

Merged Pull Requests:

- <https://github.com/unifyai/tensorly/pull/1>
- <https://github.com/unifyai/tensorly/pull/2>

Abstract:

The aim of this project is to use IVY as a framework and build that into TensorLy project so that TensorLy would be usable with any of the frameworks supported by IVY, instead of relying on TensorLy to extend the available backend themselves.

Consequently, this would allow TensorLy users to switch backends via IVY's backend handler rather than their own native backend handler and new frameworks which were currently not present in TensorLy but are present in Ivy (Example Paddle Framework) will get incorporated into TensorLy.

TensorLy is simply a wrapping framework which supports only a subset of dtypes across all backends e.g. it breaks when trying to switch between backends and testing with unsupported dtypes of that backend. It is brittle in its design such that it does not perform rigorous casting rules to make sure that switching between multiple backends provides them support of not only a subset of dtypes but all the dtypes that the user would expect to use TensorLy with. Ivy ensures this since it unifies all of the frameworks in its backends and implements superset behaviors s.t. a behavior existing in one backend is replicated in all the others as well for providing a consistent and unified interface to the user. This would be a huge utility to TensorLy after this integration.

Progress of the Project:

Implemented the functions from Tensorly's API in `tensorly/backend/ivy_backend.py` which will enable users to now use IVY as a framework in TensorLy.

- **Context of a tensor**
- **Array creation**
- **Array manipulation**
- **Algebraic operations**

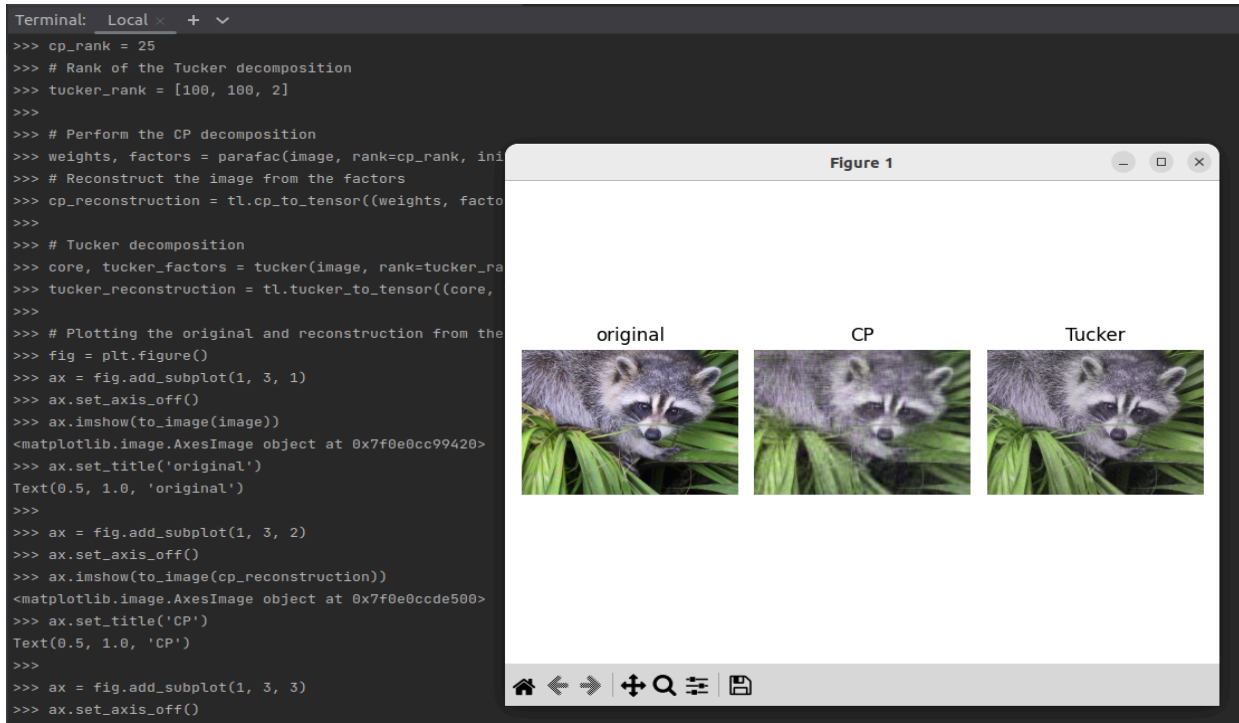
Doing so enables the user to now set ivy as the backend handler for TensorLy projects using **`tensorly.set_backend('ivy')` function.**

Demonstration:

Google Colab Link-

<https://colab.research.google.com/drive/1rYf1eYY8LBwfWZpj0sGRSerfe92mgcxh?usp=sharing>

Image compression via tensor decomposition by setting backend as 'ivy' in TensorLy



Testing:

By continuous debugging and making required changes, final test results:

359 passed and 1 failed

```
===== short test summary info =====
FAILED tensorly/tests/test_backend.py::test_lstsq - AssertionError:
FAILED tensorly/decomposition/tests/test_tucker.py::test_masked_tucker - AssertionError: norm 2 of reconstruction higher than 0.001
===== 2 failed, 358 passed, 2 skipped, 10 warnings in 2344.37s (0:39:04) =====
```

- Ivy doesn't skip any of the tests from TensorLy's test suite whereas testing with TensorFlow backend skips tests wherever we try to index into a TensorLy Tensor since Tensorflow doesn't support it natively but Ivy handles this internally as well.

Demo Notebook for Tensor Regression Networks with TensorLy and Paddle as a backend:

This demo helps a user in understanding that when we added Ivy as the backend in TensorLy, all the frameworks supported by Ivy got incorporated into TensorLy. In this example, we used the Paddle framework along with TensorLy for implementing and training the model.

Google Colab Link-

<https://colab.research.google.com/drive/1nOsbVWBpkAhUvNw4z-8KuCsLviuXrwA-?usp=sharing>

Tasks Remaining:

- 1 out of 360 Tests (test_masked_tucker) is failing. So, the fixing part is remaining.
- Few fixes are remaining in the model part of the demo.