Hewlett Packard
Enterprise

Machine Learning in The Chapel Programming Language

Iain Moncrief

August 6, 2023

Background

Lightweight tensor and linear algebra library

Extends NumPy to support ML needs
(backpropogation, CUDA implementation)

Fully featured ML tools (uses TensorFlow)

Chapel

- High performance computing
- Sophisticated array programming
- Automatically utilize maximal resources
- Easily program parallel computations

My Project

- Implement ML programs in Chapel
- See what aspects of Chapel make it easier/harder to implement ML programs
- Performance comparison

My Chapel ML Implementation

- Tensor Library
  - Attempts to replicate much of NumPy's functionality
  - Arithmetic and linear algebra operations
  - Supporting helper functions
- Machine Learning Library
  - Implements various layer types: Dense, Conv2d, MaxPool, SoftMax, ReLU, TanH, Sigmoid, Flatten, …
  - Offers similar user interface as PyTorch or Keras

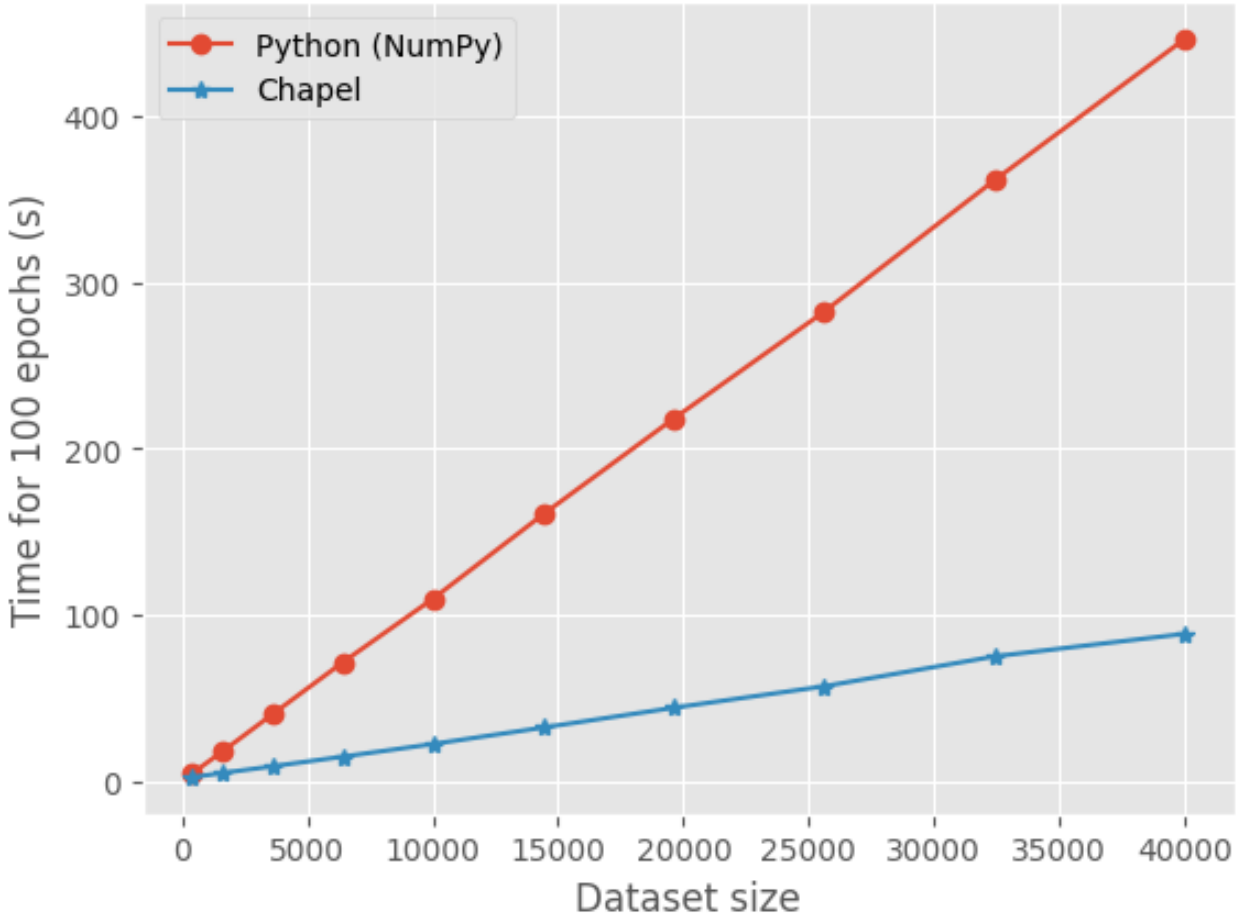
```
model = Sequential(
    Dense(4),
    Sigmoid(),
    Dense(3),
    Sigmoid(),
)
```

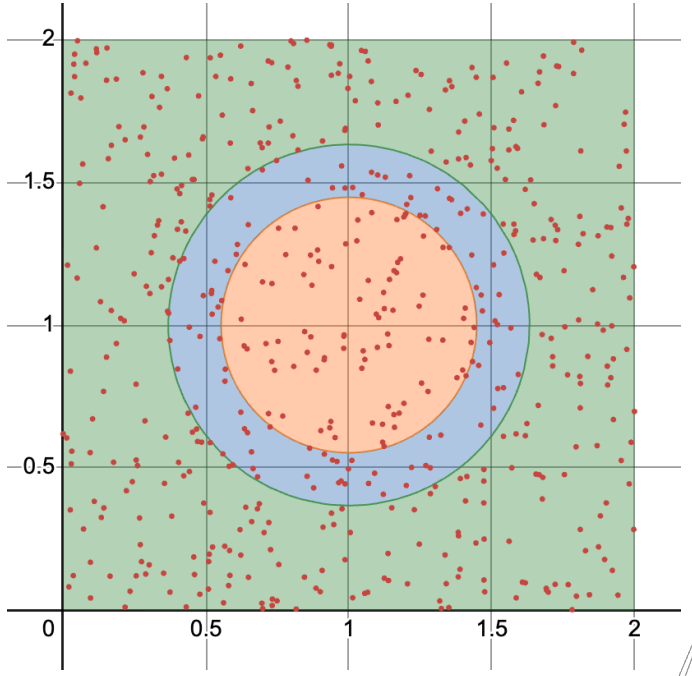
PyTorch

```
var model = new Sequential(
    new Dense(4),
    new Sigmoid(),
    new Dense(3),
    new Sigmoid()
);
```

CHAPEL

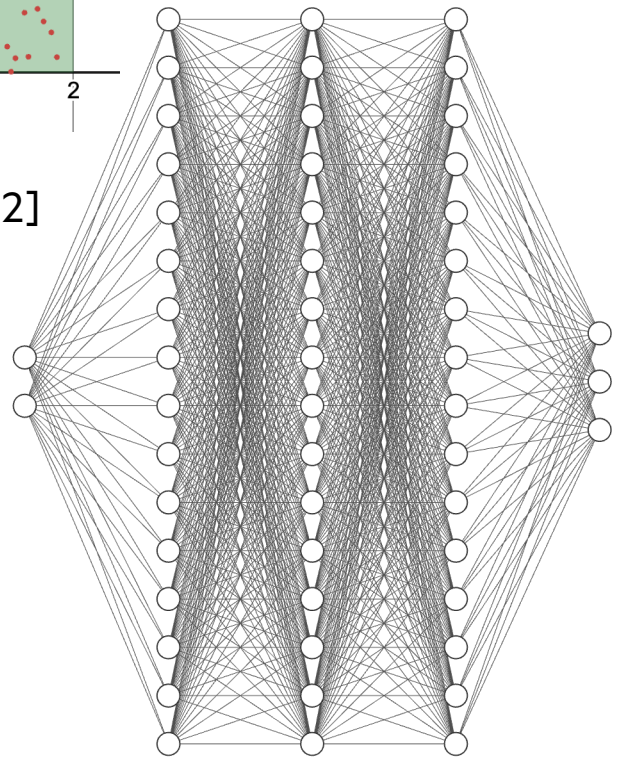
Speed Comparison (Simple Classification)



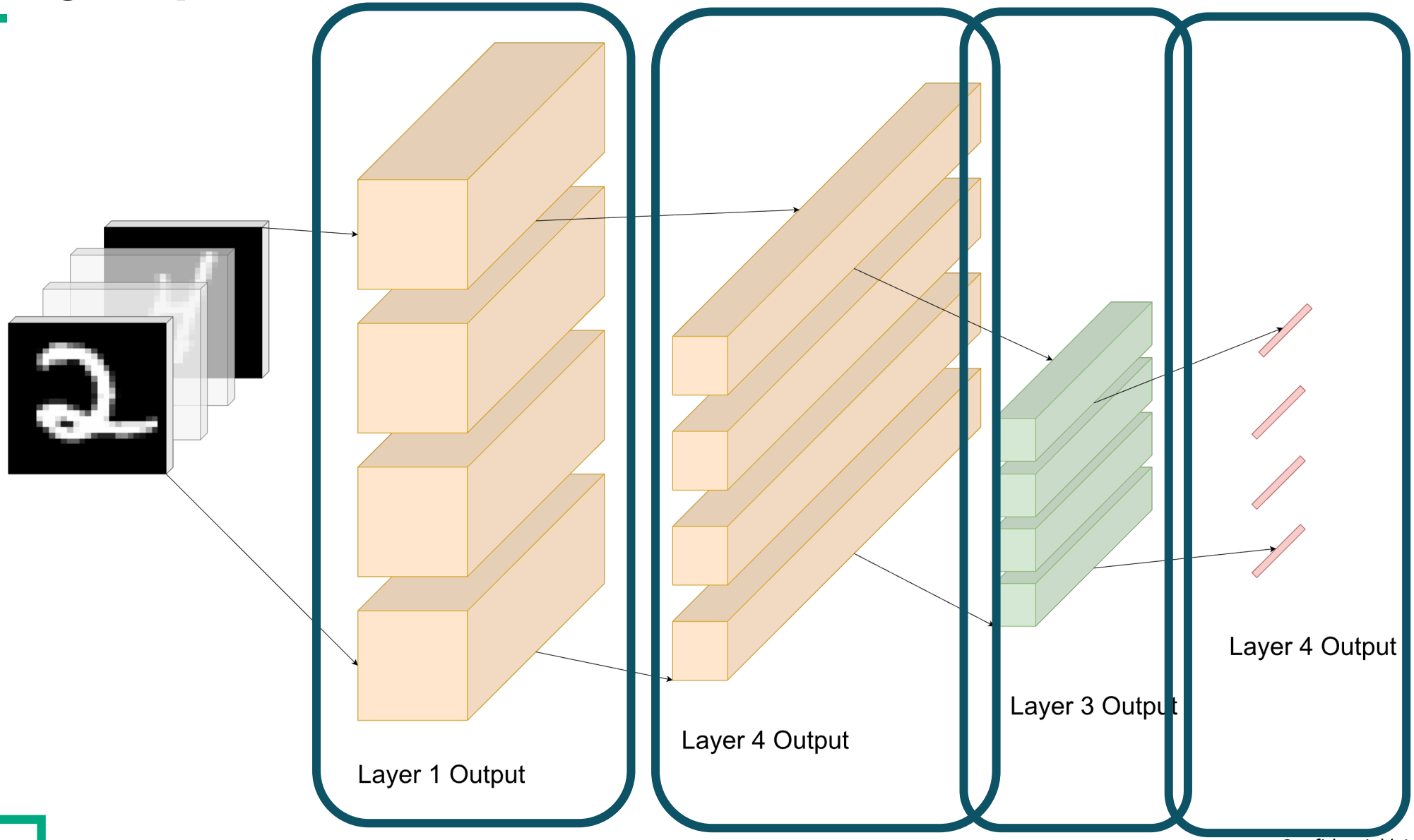


Input: N samples from [0,2] x [0,2]

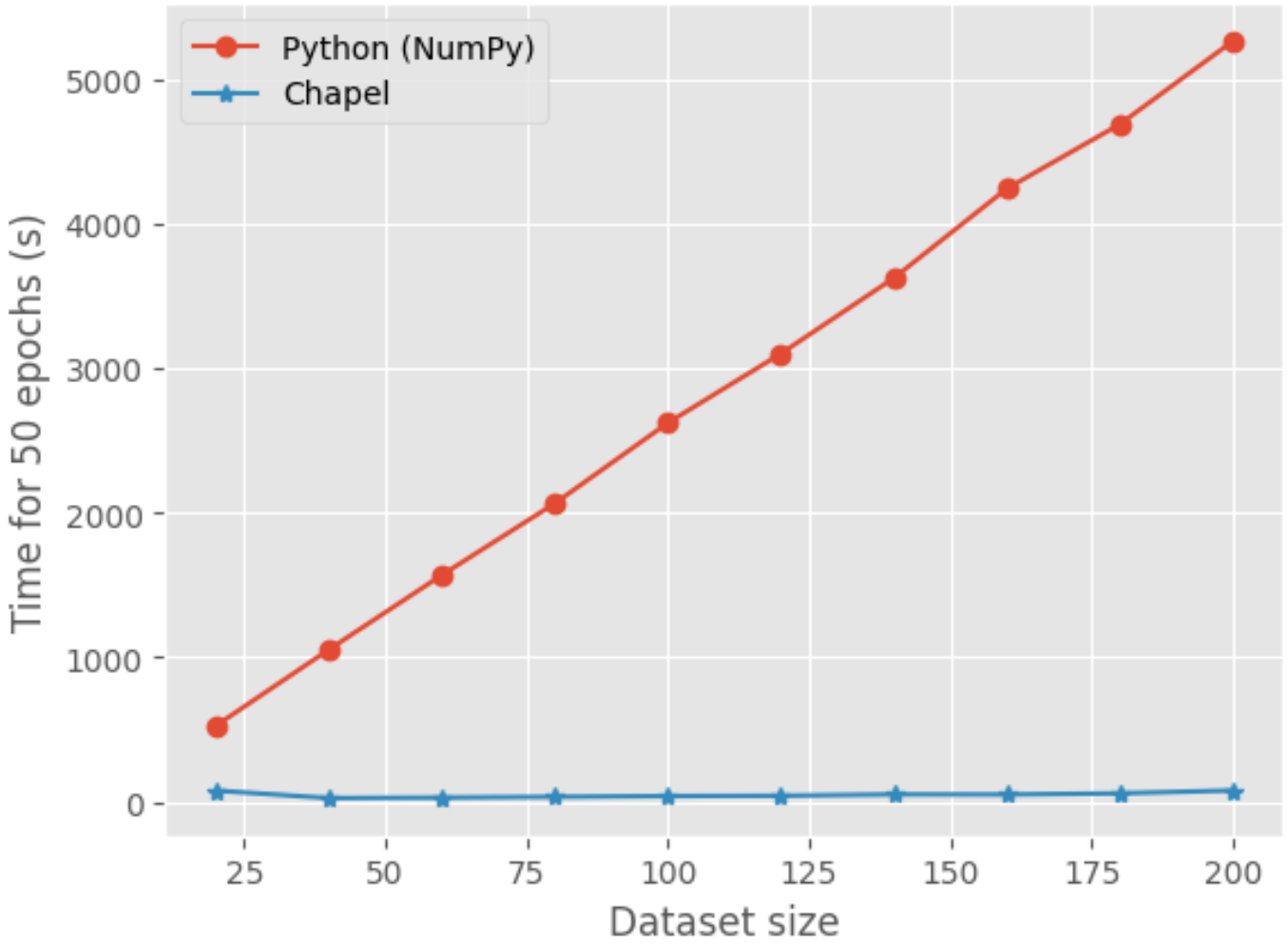
Output: 3 different categories



Utilizing Chapel's Parallelism Constructs

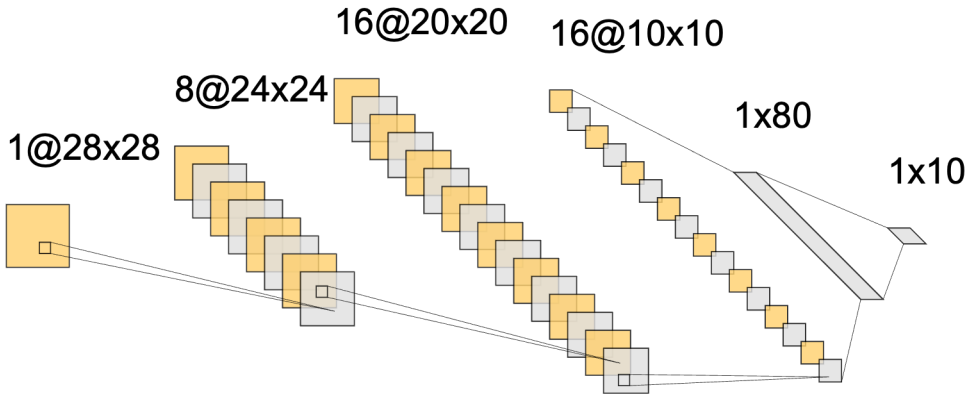


Speed Comparison (MNIST Classification)

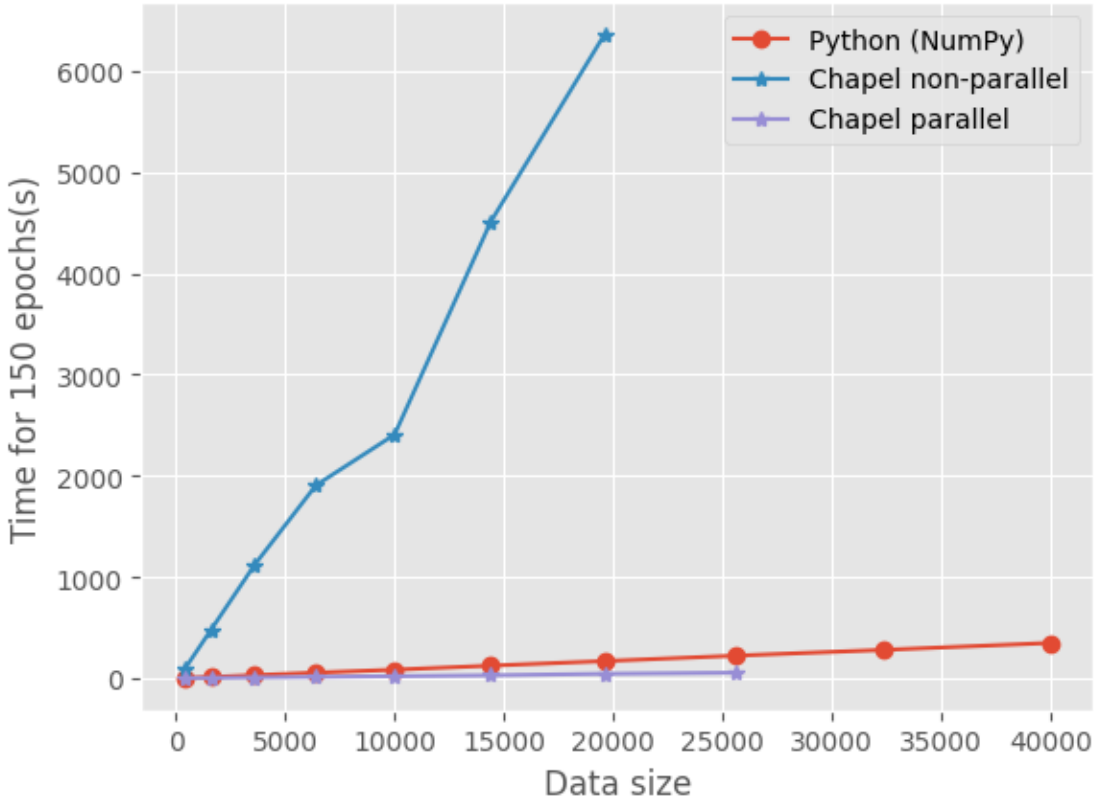


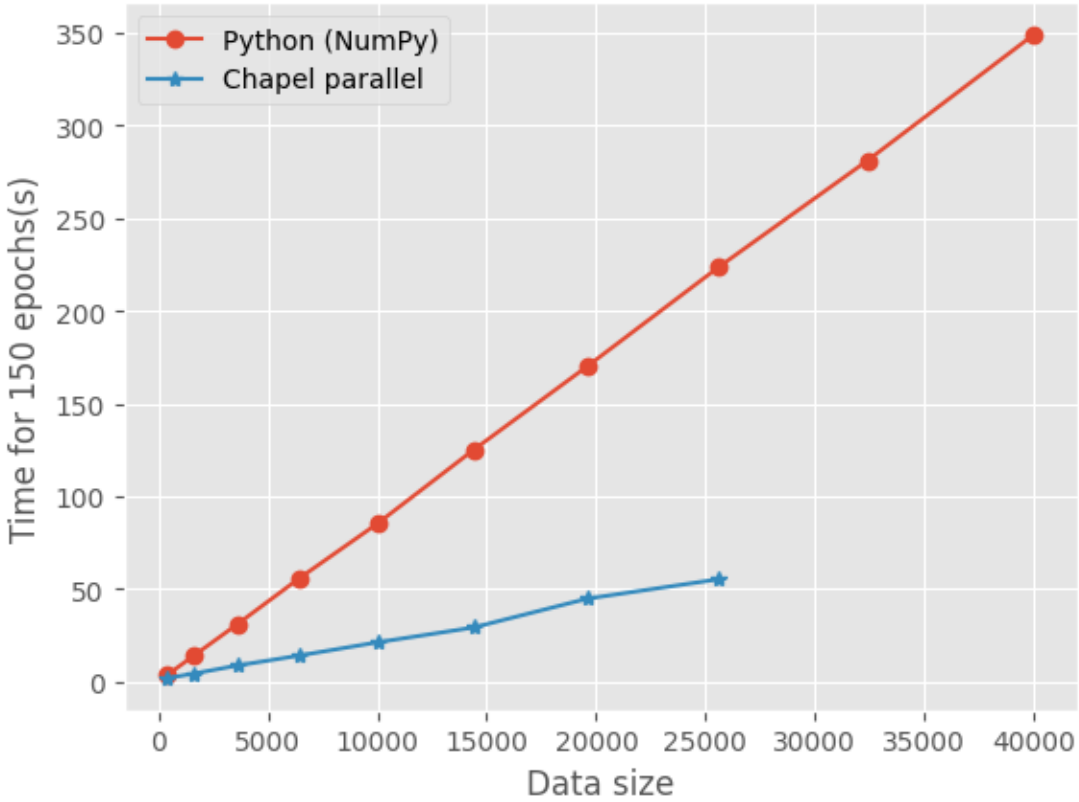
Input:

Output: 0,1,2,3,...,9



Thank you


Python (NumPy)
Chapel non-parallel
Chapel parallel
Time for 150 epochs(s)
Data size
0
1000
2000
3000
4000
5000
6000
0
5000
10000
15000
20000
25000
30000
35000
40000


Python (NumPy)
Chapel parallel
Time for 150 epochs(s)
Data size
0
50
100
150
200
250
300
350
0
5000
10000
15000
20000
25000
30000
35000
40000