

HW3: Tensor Library

Weijie Zhao

11/01/2022

HW 3: Tensor Library

- Write a tensor library that is callable from python
- No 3rd party code is allowed. **Numpy is not allowed.**
- 10 test cases. Each case weights 1 pt.
- The compilation is considered failed if it does not finish in **5 minute**.
- A test case is considered **incorrect** if it does not finish in **2 minutes**.
- The numeric error of each printed value must be within $1e-3$ to the correct result.
- **Correct GPU solutions will get 5 pts bonus.**
- **Correct computation graph building solution with lazy evaluations will get 10 pts bonus**
- The **summation** of the execution time across 10 cases will be used to rank **correct** solutions.

- Due: 11/11/2022 11:59 pm EST

Testing Environment

- `ssh yourusername@granger.cs.rit.edu`
- Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz
- 48 threads in total (2 sockets, 12 cores per socket, 2 threads per core)
- 251 GB memory
- GPU: Tesla P4
- **pybind11 2.10.0 installed (`pip3 install pybind11`)**
- Testing limit:
 - **8 threads** `taskset -c`
 - **1 GPU**

pybind11

```
#include <pybind11/pybind11.h>
namespace py = pybind11;
int add(int i, int j) {
    return i + j;
}
#include <pybind11/pybind11.h>
```

```
int add(int i, int j) {
    return i + j;
}
```

```
PYBIND11_MODULE(example, m) {
    m.doc() = "pybind11 example plugin"; // optional module docstring

    m.def("add", &add, "A function that adds two numbers");
}
```

```
$ python
>>> import example
>>> example.add(1, 2)
3
>>>
```

```
m.def("add", &add, "A function which adds two numbers", py::arg("i"), py::arg("j"));
```

```
int add(int i = 1, int j = 2) {
    return i + j;
}
```

```
m.def("add", &add, "A function which adds two numbers",
      py::arg("i") = 1, py::arg("j") = 2);
```