

# ECE408 / CS483 / CSE 408 Final Project Kickoff

Carl Pearson

pearson@illinois.edu

# Outline

Fast convolution layer forward pass in MxNet

# Timeline

Tue 11/07/2017: Project Released

Fri 11/10/2017: Milestone 1 "Due"

Fri 11/17/2017: Milestone 2 Due with draft report

Fri 12/1/2017: Milestone 3 Due with updated report

12/15/2017: Final Submission with final report

# Project Release

## Project Landing Page

[https://github.com/webgpu/2017fa\\_ece408](https://github.com/webgpu/2017fa_ece408)

## Includes:

Instructions

RAI Client

Final Project Rubric

Other Guidelines

Skeleton Code

# rai Client

The client may be downloaded from here: <https://github.com/rai-project/rai>

Download from the links in the README, not releases.

You will need your `.rai_profile` file in your home directory.

- `~/.rai_profile` on Linux/macOS
- `%HOME%/.rai_profile` on Windows

This should have been emailed to you.

(demo)

# Teams

## Team of 3 People

Modify `.rai_profile` with your team name under `team.name`:

```
profile:
  firstname: Carl
  lastname: Pearson
  username: carlwpearson
  email: carlwpearson@gmail.com
  access_key: auth0|5a0130d32327ea70420b71ef
  secret_key: <snip>
  affiliation: uiuc
  team:
    name: staff
```

# Basic `rai`

`rai -p <project folder>` uploads your folder to AWS.

Your code in `rai_build.yml` is executed on AWS in a specific docker container.

The results are streamed back to you in *real time*.

# Milestone 1 (Friday 11/10/2017)

Nothing to turn in until Milestone 2

- Not much work on your part!
- Just making sure rai is working for you.

## Run MxNet baseline CPU code

- Make sure MxNet is working.
- Report execution time.

## Run MxNet baseline GPU code

- Make sure MxNet GPU is working
- Report execution time.
- Use `nvprof` to make a profile

(demo)



# Submitting

```
rai -p <project folder> --submit
```

- \* Enforces a particular `rai_build.yml`
- \* Records timing information
- \* You will need a ``report.pdf``.

This was designed for a final submission, so the Milestone submissions may fail to execute. That's okay, we just want to collect the report.

# Milestone 2 (Friday 11/17/2017)

Turn in a draft of `report.pdf`

## Implement CPU forward pass

- Make sure you can compile/run MxNet CPU code.
- Execution time

Should be pretty straightforward copy from slides / chapter 16

# Milestone 3 (Friday 12/1/2017)

Turn in an updated draft of `report.pdf`.

## Implement GPU forward pass

- Make sure you can compile/run MxNet GPU code.
- Execution time, profile

Doesn't have to be fast, but it should work. Small changes to milestone 2.

# Final Submission (Friday 12/15/2017)

- The **real deal**.

Optimize that GPU convolution.

Turn in a final report.

# Final report

See project page for up-to-date rubric.

- Baseline Results
  - M1.1: mxnet CPU layer performance results (time)
  - M1.2: mxnet GPU layer performance results (time, nvprof profile)
- M2.1: your baseline cpu implementation performance results (time)
- M3.1: your baseline gpu implementation performance results (time, nvprof profile)
- Optimization Approach and Results
  - how you identified the optimization opportunity
  - why you thought the approach would be fruitful
  - the effect of the optimization. was it fruitful, and why or why not. Use nvprof as needed to justify your explanation.
  - Any external references used during identification or development of the optimization
- References (as needed)

# Comparing against your peers

rai rankings

Shows *anonymized* performance results for you and other teams.

# rai tips and tricks

## ``rai -p version`

Prints the date rai was built

Check piazza, if your rai is old, download a newer one

## ``rai -p -d -v`

Debug / verbose mode. Prints a bit of additional info while running

- the queue submitted to (should be `rai_amd64`)
- what it thinks your username and team name is

# Notes on batch\_size

Don't modify the `batch_size` in the python script.

You can loop over `B` in the C/CUDA code and split it up there however you want.