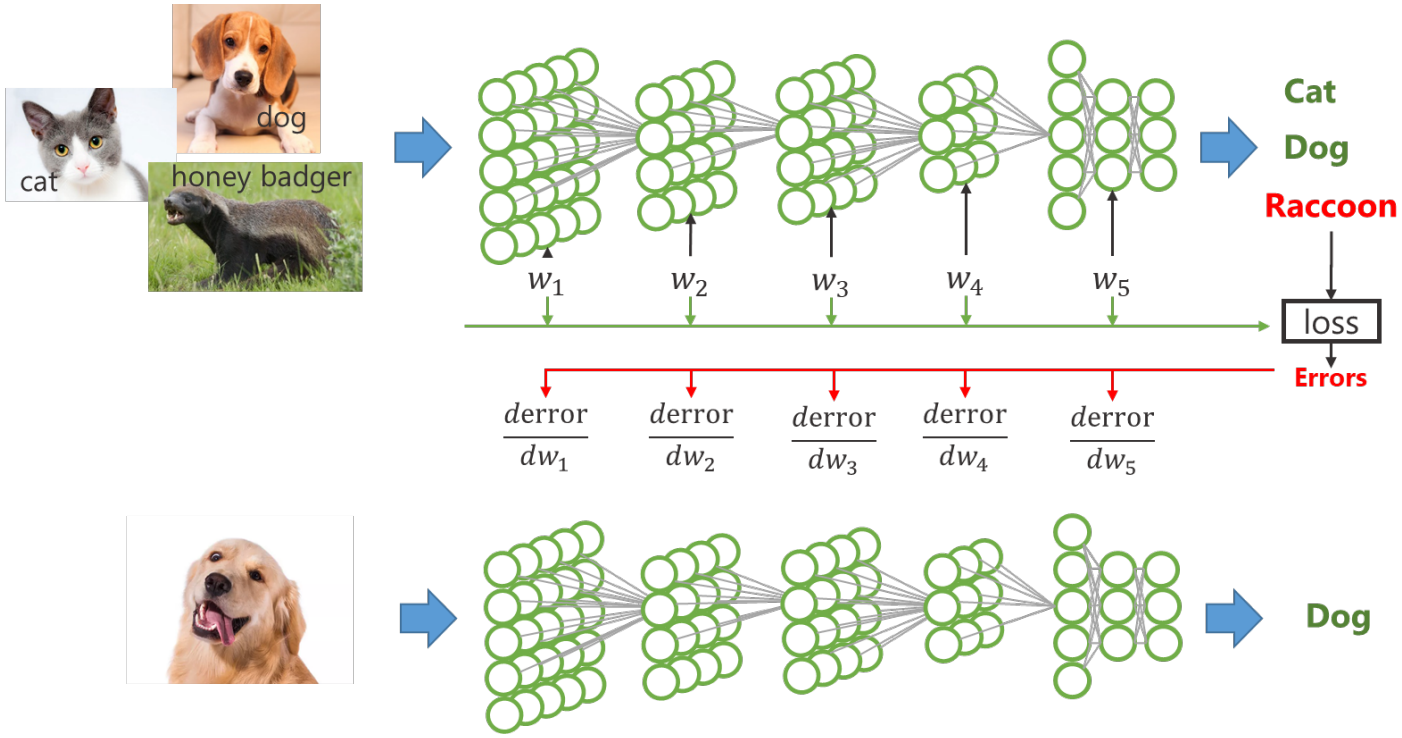
TinyInfer:
Tiny Inference Engine for Neural Network

CS 133 Course Project

上海科技大学
ShanghaiTech University

Team Member:
Jiadi Cui, Jianxiong Cai, Zhiqiang Xie

1 Neural Network Inference



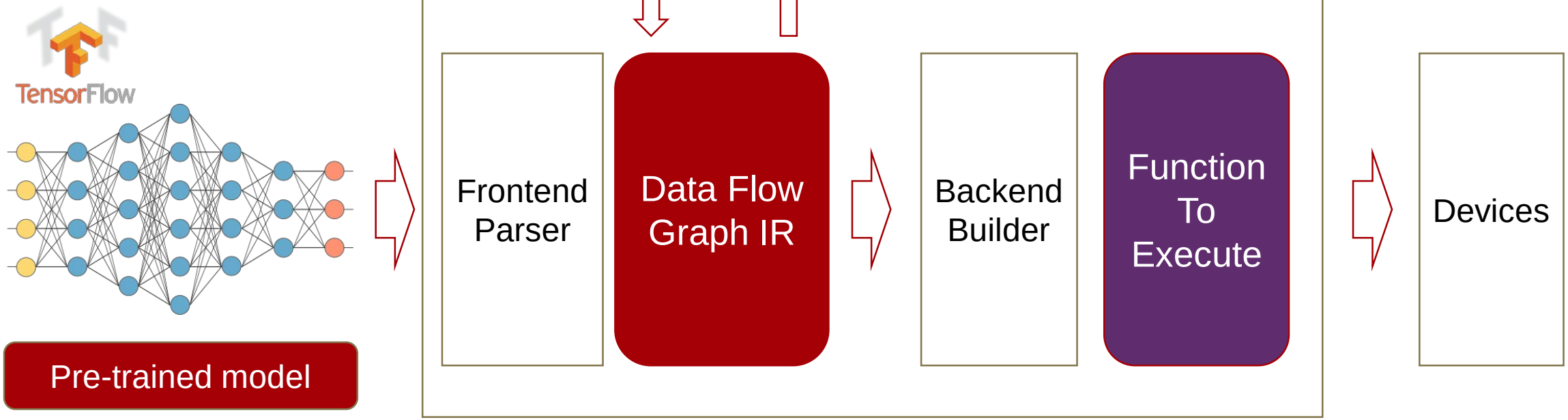
Training

- Build once

Inference

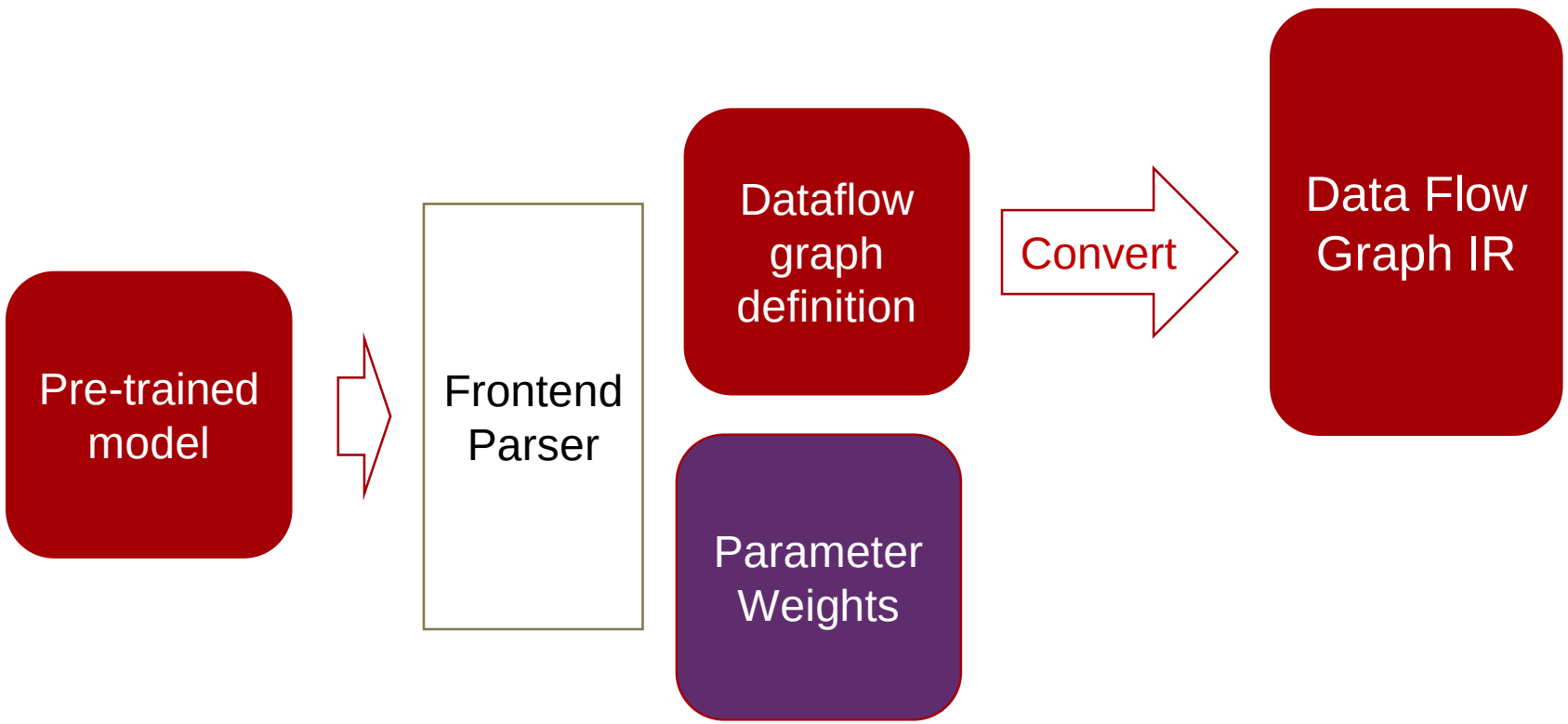
- Run many times

2 Overview


TensorFlow
Pre-trained model
Optimization Pass
Frontend Parser
Data Flow Graph IR
Backend Builder
Function To Execute
Devices

3 Frontend Parser

```
node {
  name: "dense_1/MatMul"
  op: "MatMul"
  input: "flatten_1/Reshape"
  input: "dense_1/kernel/read"
  attr {
    key: "T"
    value {
      type: DT_FLOAT
    }
  }
  attr {
    key: "transpose_a"
    value {
      b: false
    }
  }
  attr {
    key: "transpose_b"
    value {
      b: false
    }
  }
}
```



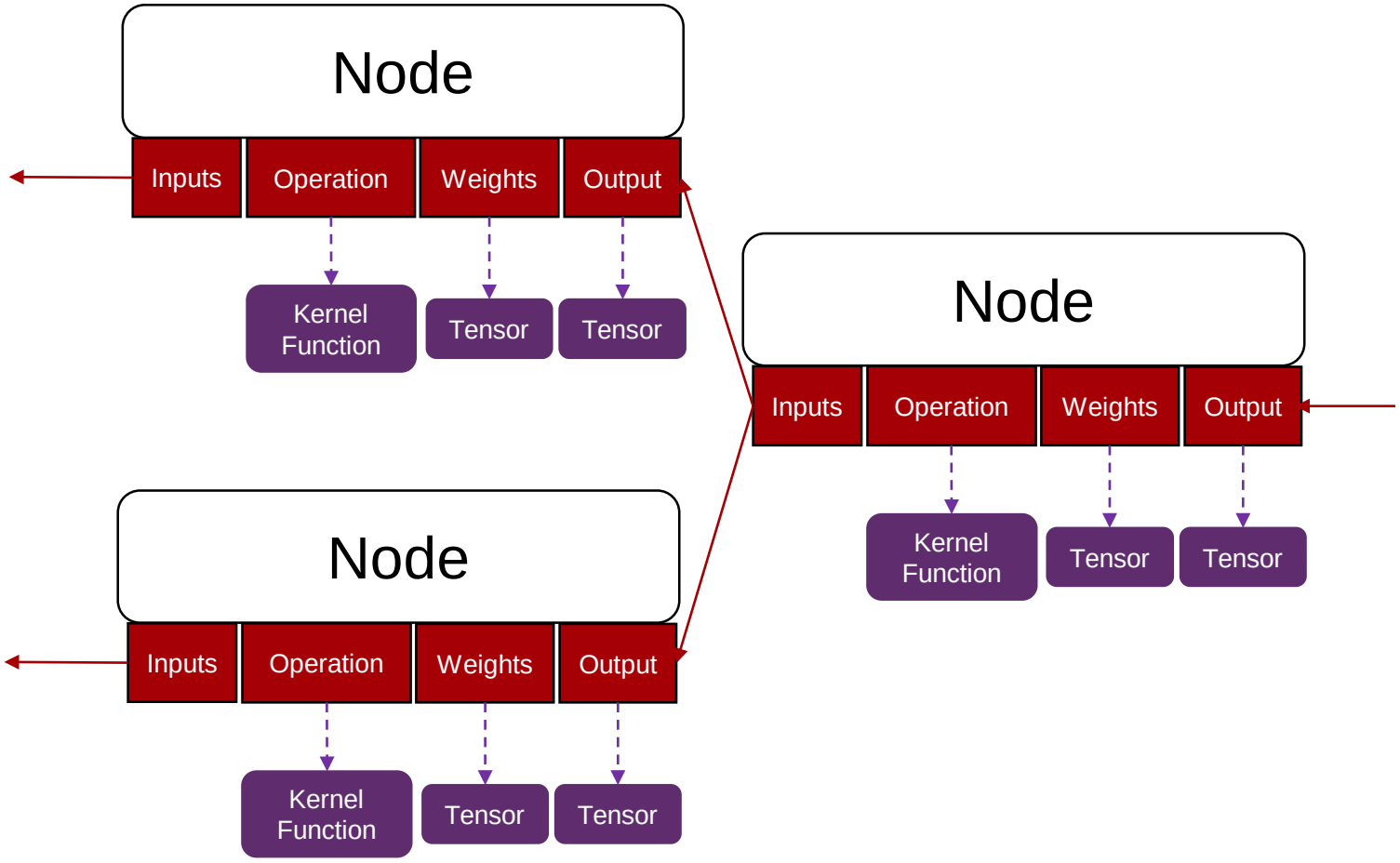
3 IR design

Construct time:

Build and optimize target independent staffs (in red)

Runtime:

Calculation on target specific staffs (in purple)

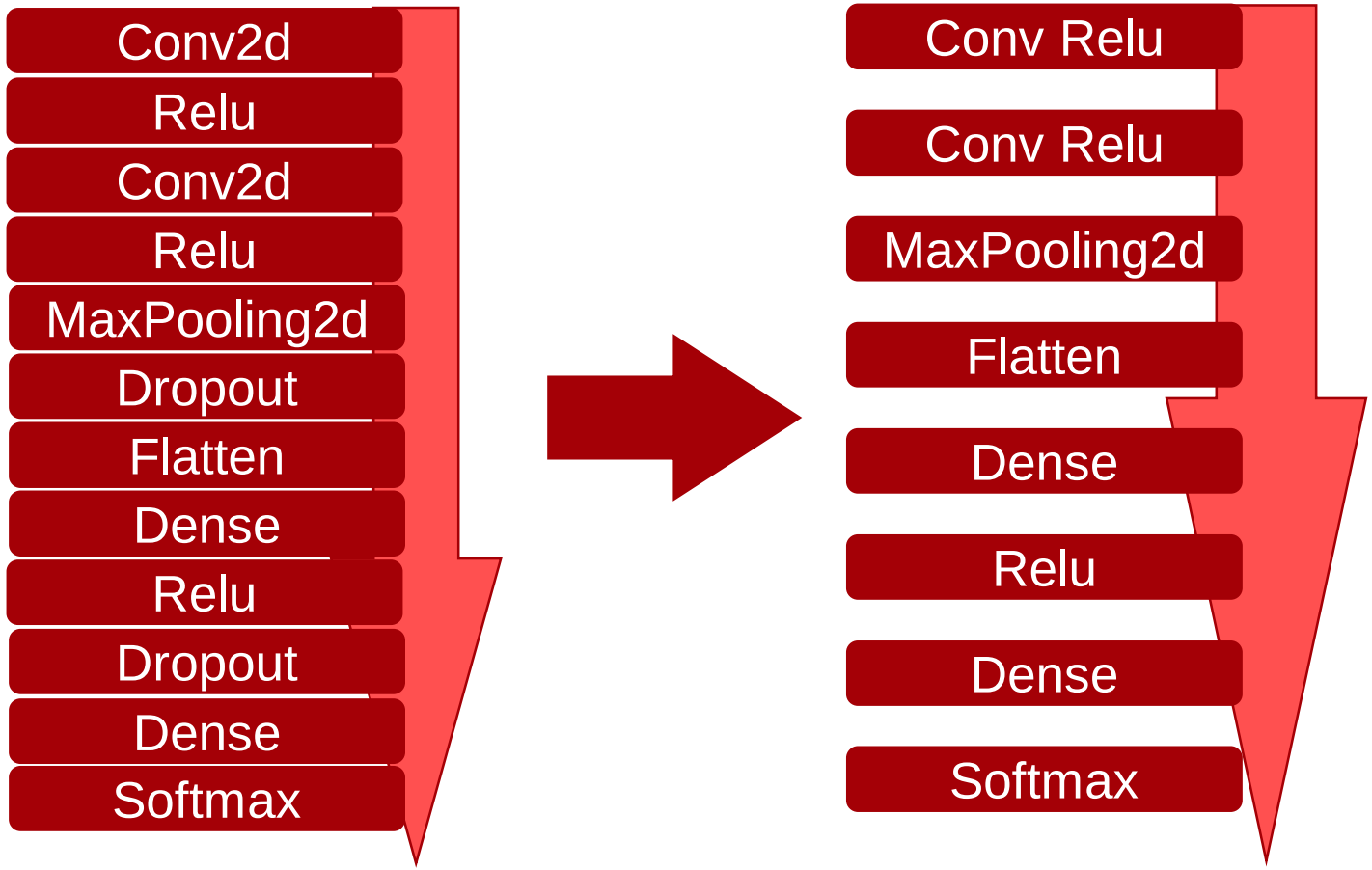


4 Optimization Pass

Currently supported:

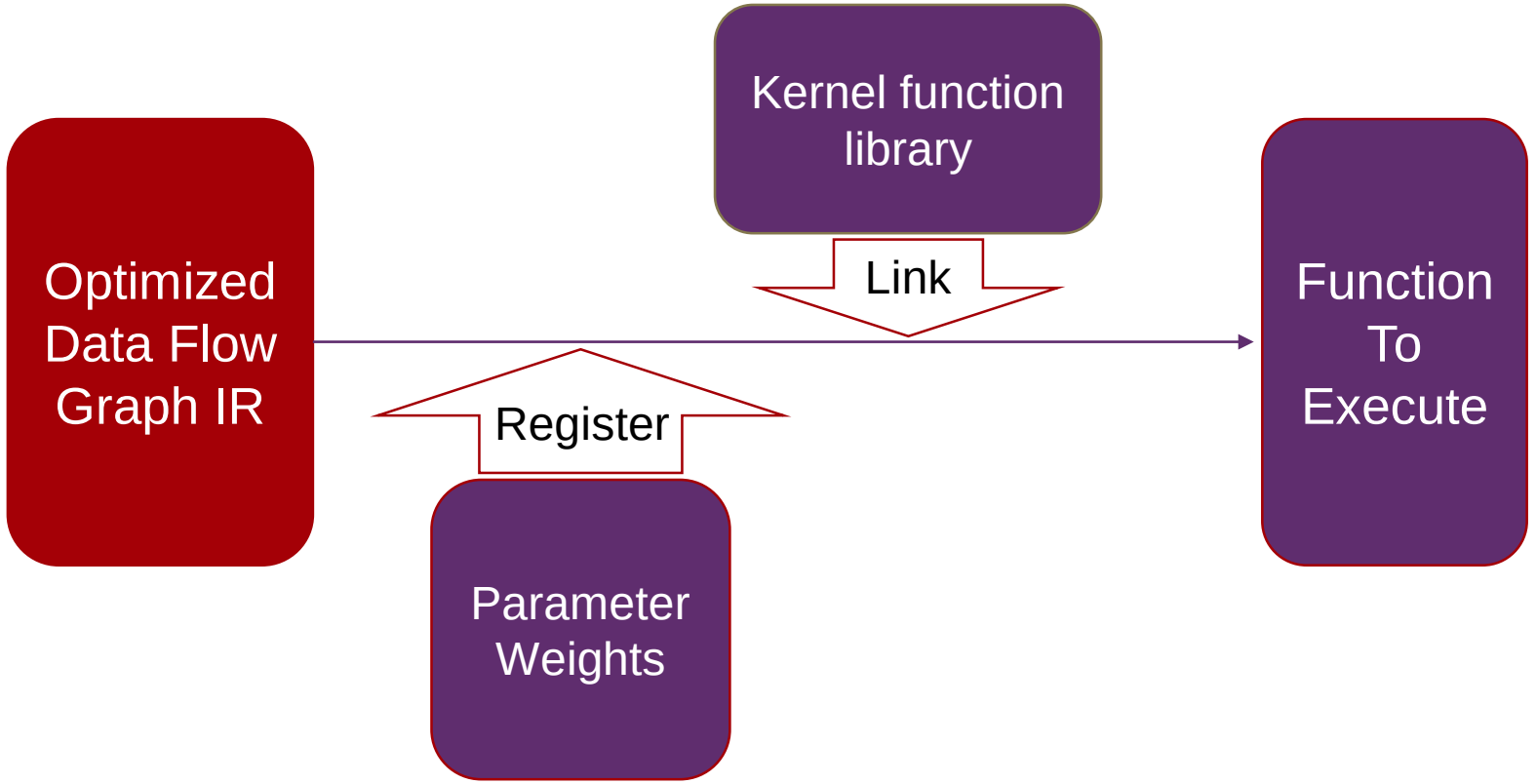
- Futile node elimination
- Convolution, Relu node fusion

… more to be added



5 Backend Builder

Construct the function to be executed on specific device, e.g. CPU, GPU



6 Evaluation

- MNIST dataset
  - 10000 images
  - Accuracy: 98.49% (vs TF keras: 98.49%)

- Demo

7 Conclusion

Tinyinfer

- Modularity
  - Decoupled representation and implementation
  - Target independent IR design
  - Target specific backend builder
- Expendability
  - Easy to introduce new operator
  - Free to replace kernel implementation for different devices
- Scalability

Team Members:
Jiadi Cui
Jianxiong Cai
Zhiqiang Xie

THANKS!

上海科技大学
ShanghaiTech University