PARALLEL AND DISTRIBUTED DEEP LEARNING

PADDLEPADDLE

OUTLINE

- Introduction
- PaddlePaddle Fluid
- Elastic Distributed Training
- Multi-GPU
- Sequence Model and LoDTensor
- Official Resources

INTRODUCTION

PaddlePaddle

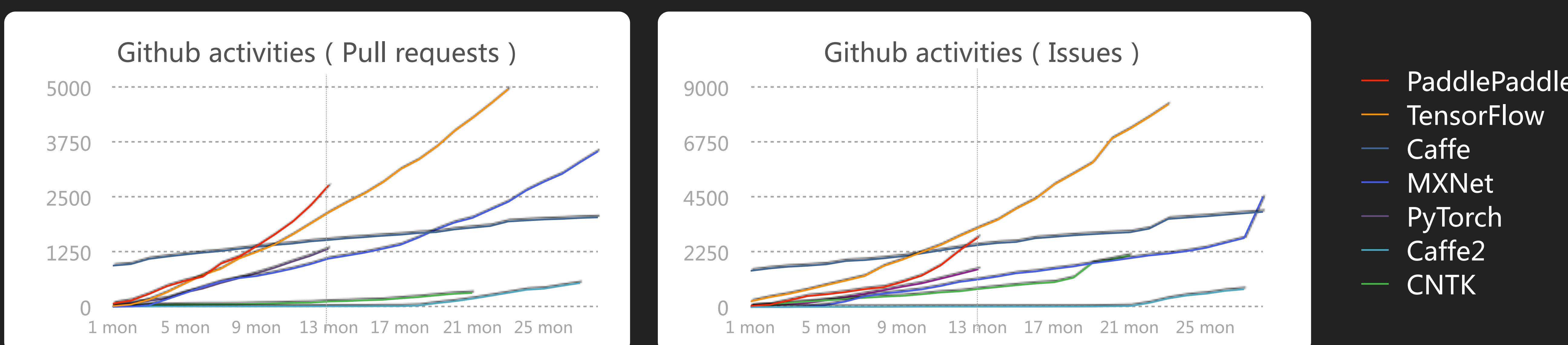
build passing docs latest 中文文档 最新

- Official Website: http://paddlepaddle.org/
- Github: https://github.com/PaddlePaddle/Paddle

- Deep Learning Platform for both Enterprise & Research
- Easy-to-use, Flexibility, Efficiency and Scalability
- Official Release: pypi and Docker image

A BIT HISTORY

- 2013 — Start PaddlePaddle Project
- 2014~2016 — Wins Baidu Highest Award
- 2016~2017 — Open Source and rapid growth



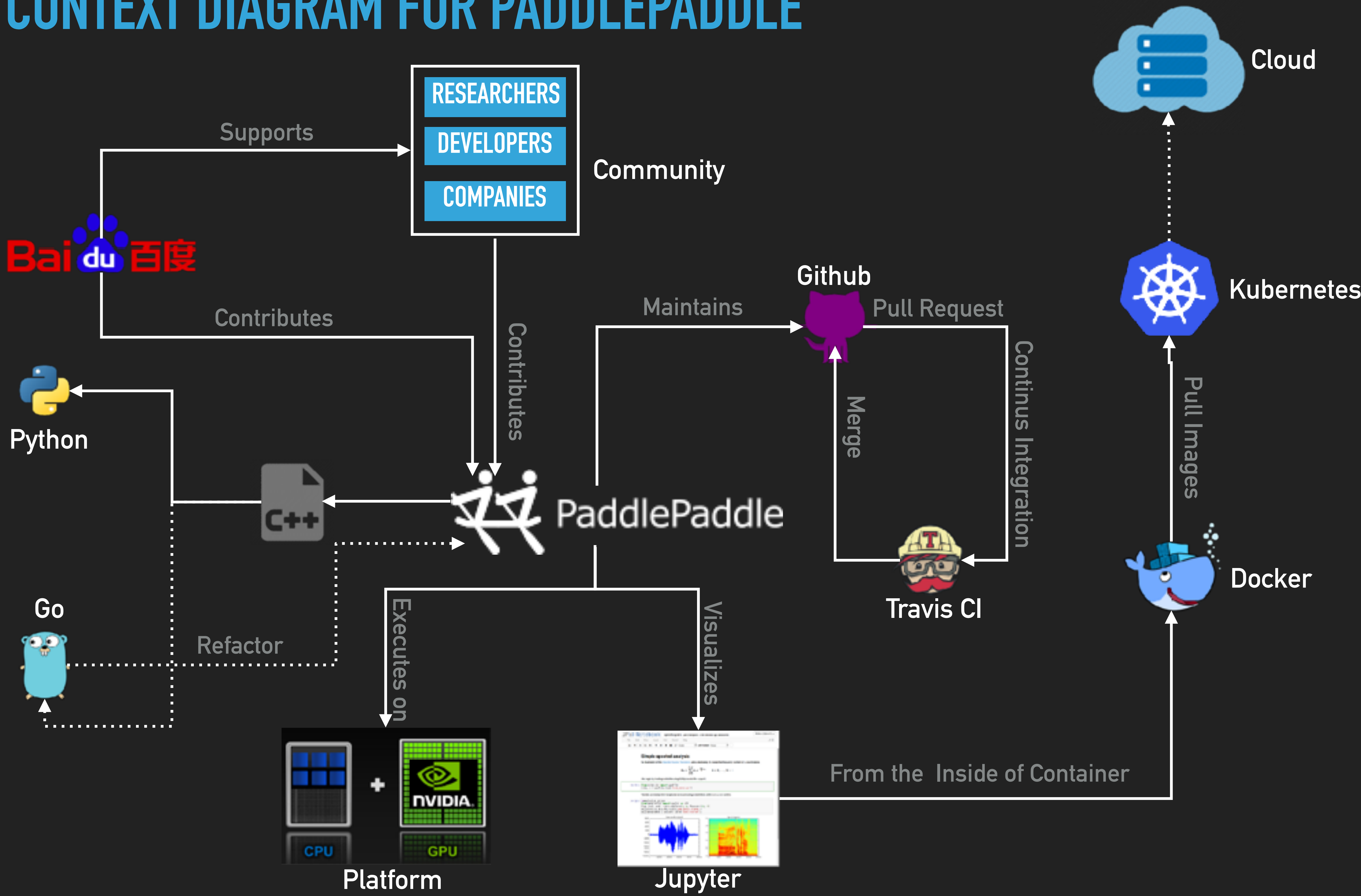
WHY WE START THE PADDLEPADDLE PROJECT ?

- Make use of billion data
- People do not have such open source platforms in 2013
- Toolkits on deep learning were about image recognition in 2013
- Requirement: Products around Our Search Engine
- Text based Problem (Natural Language Understanding)

PADDLEPADDLE IN BAIDU

- supports many online products

CONTEXT DIAGRAM FOR PADDLEPADDLE



OUTLINE

- Introduction
- PaddlePaddle Fluid
- Elastic Distributed Training
- Multi-GPU
- Sequence Model and LoDTensor
- Official Resources

PADDLEPADDLE FLUID

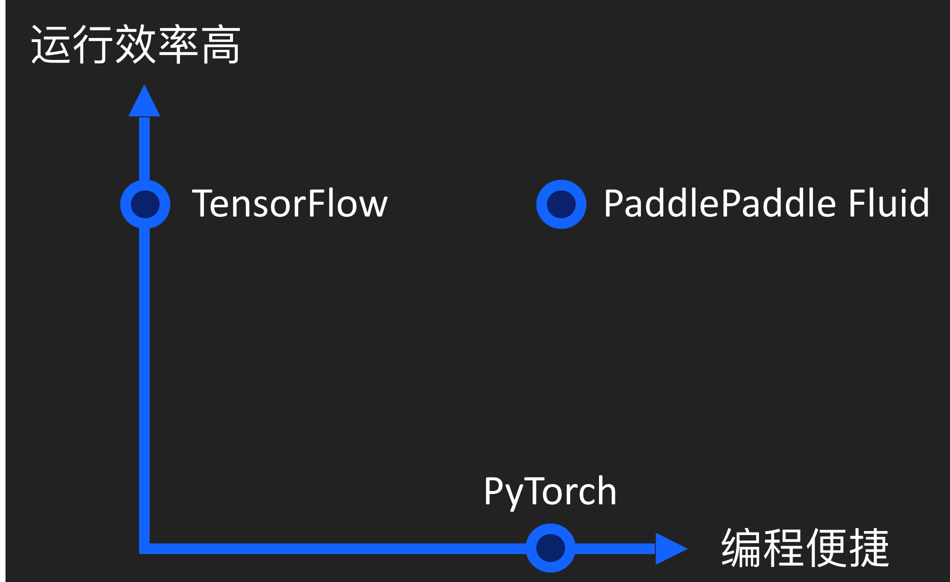
```
                          # block 0
x = sequence([10, 20, 30])
m = var(0)
W = var(0.314, param=true)
U = var(0.375, param=true)



rnn = pd.rnn()
with rnn.step():          # block 1
  x_ = rnn.step_input(x)
  h = rnn.memory(init = m)
  hh = rnn.previous_memory(h)
  a = layer.fc(W, x_)
  b = layer.fc(U, hh)
  s = pd.add(a, b)
  act = pd.sigmoid(s)
  rnn.update_memory(h, act)
  rnn.output(a, b)
o1, o2 = rnn()
```

```
                                       // block 0
int* x = {10, 20, 30};
int* m = {0};
int* W = {0.314};
int* U = {0.375};

int len = sizeof(x) / sizeof(x[0])
int mem[len + 1];
int o1[len + 1];
int o2[len + 1];
for (int i = 1; i <= len; ++i) { // block 1
  int x = x[i-1];
  if (i == 1) mem[0] = m;
  int* hh = &(mem[i-1]);
  int a = W * x;
  int b = Y * *hh;
  int s = fc_out + hidden_out;
  int act = sigmoid(sum);
  mem[i] = act;
  o1[i] = act; o2[i] = hidden_out;
}
```



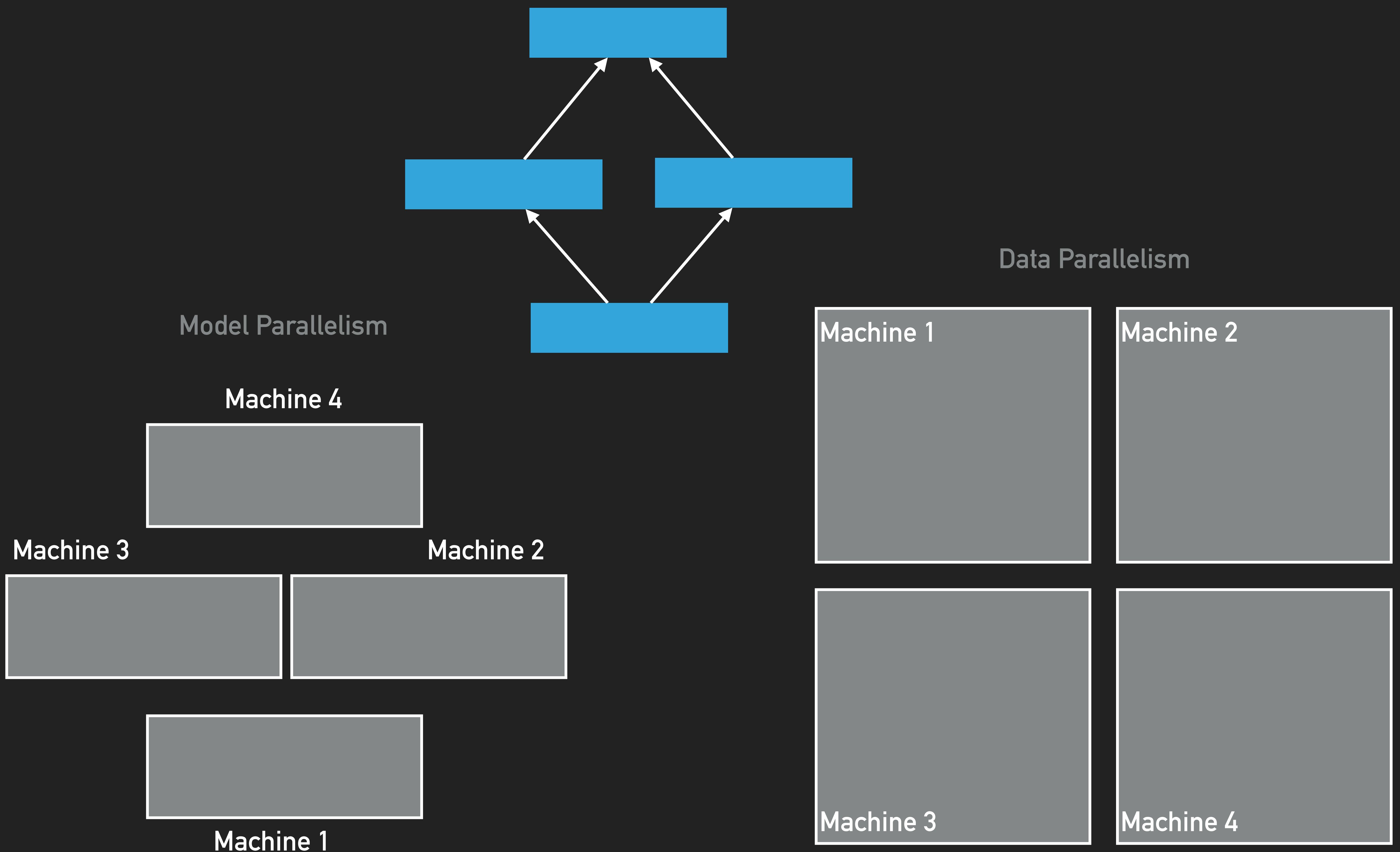
PADDLEPADDLE FLUID

- High level programing API
  - Write like a High-level Programing Language
  - Compatible with legacy API
  - Implemented using operators
- Current Status:
  - Development Complete (single machine)
  - IfElse/While Operators
  - Better support for GAN/RL

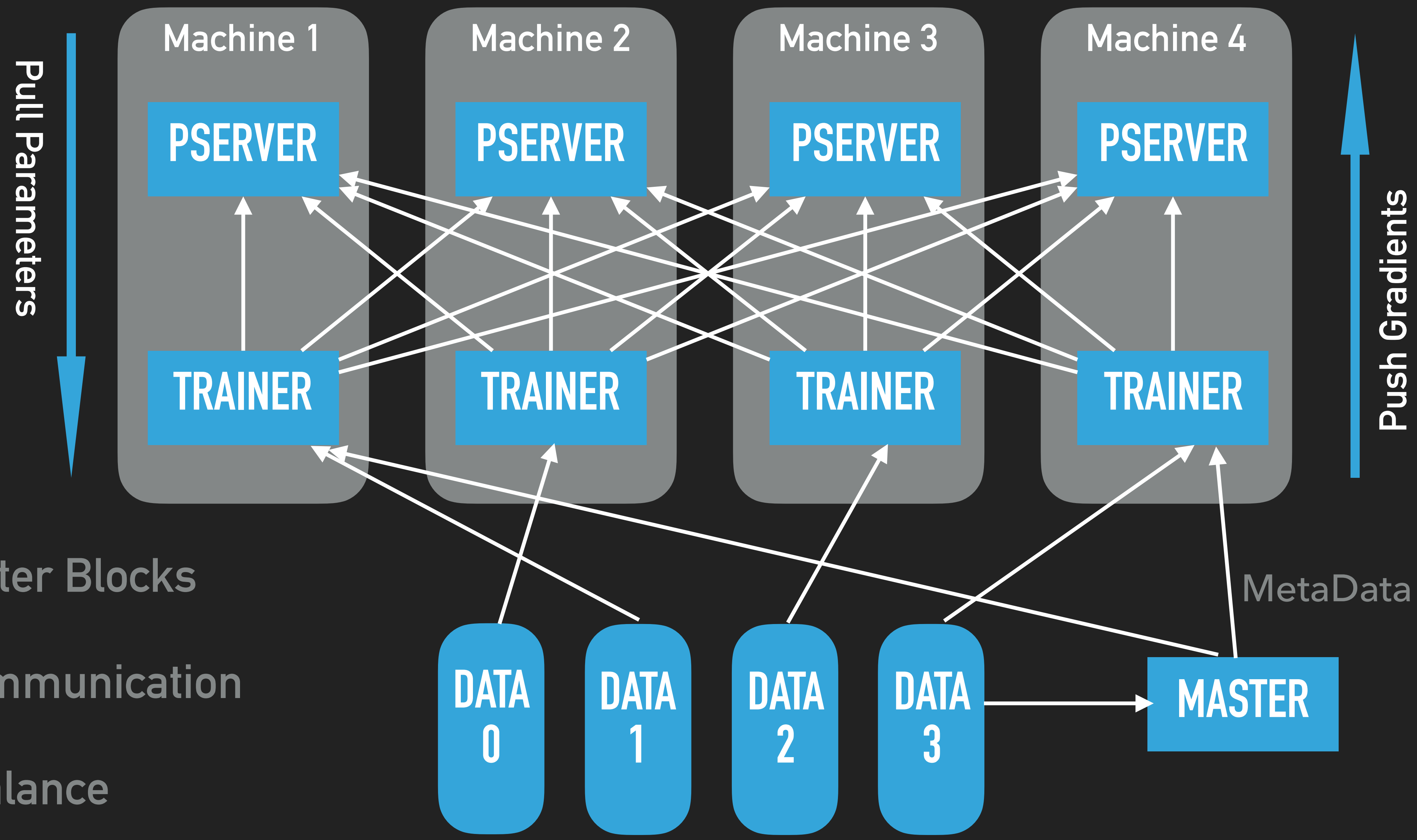
OUTLINE

- Introduction
- PaddlePaddle Fluid
- Elastic Distributed Training
- Multi-GPU
- Sequence Model and LoDTensor
- Official Resources

DISTRIBUTED TRAINING



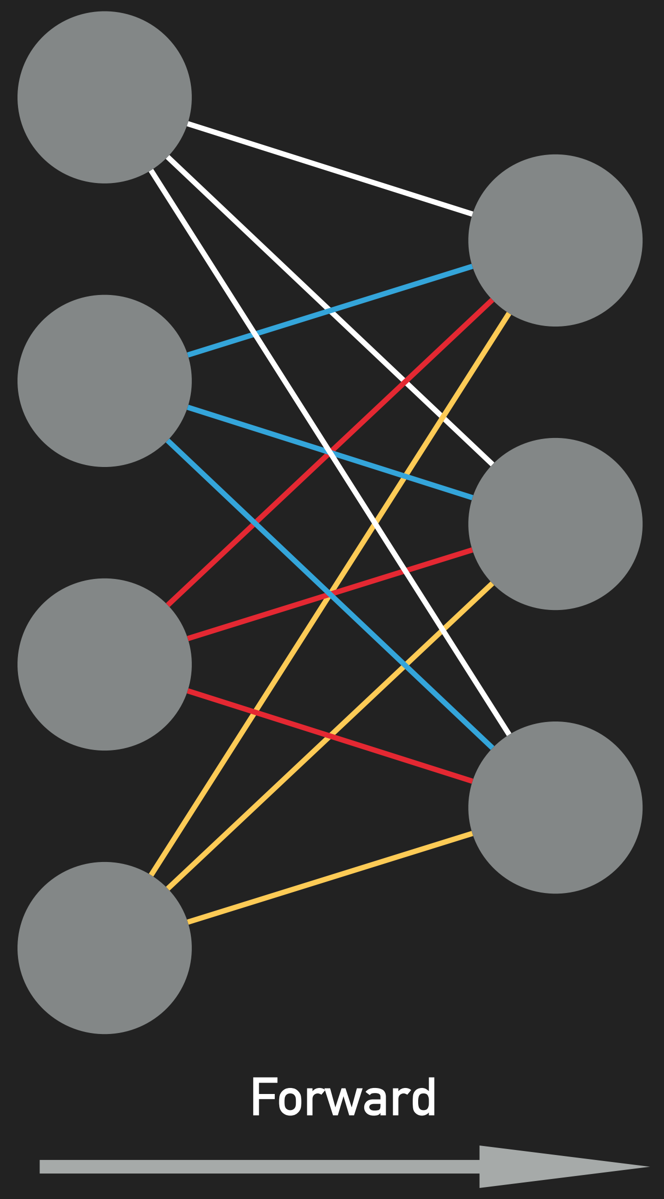
FAULT TOLERANT DISTRIBUTED TRAINING



- Parameter Blocks
- P2P Communication
- Load Balance
- Fault tolerant Async SGD
- Checkpointing

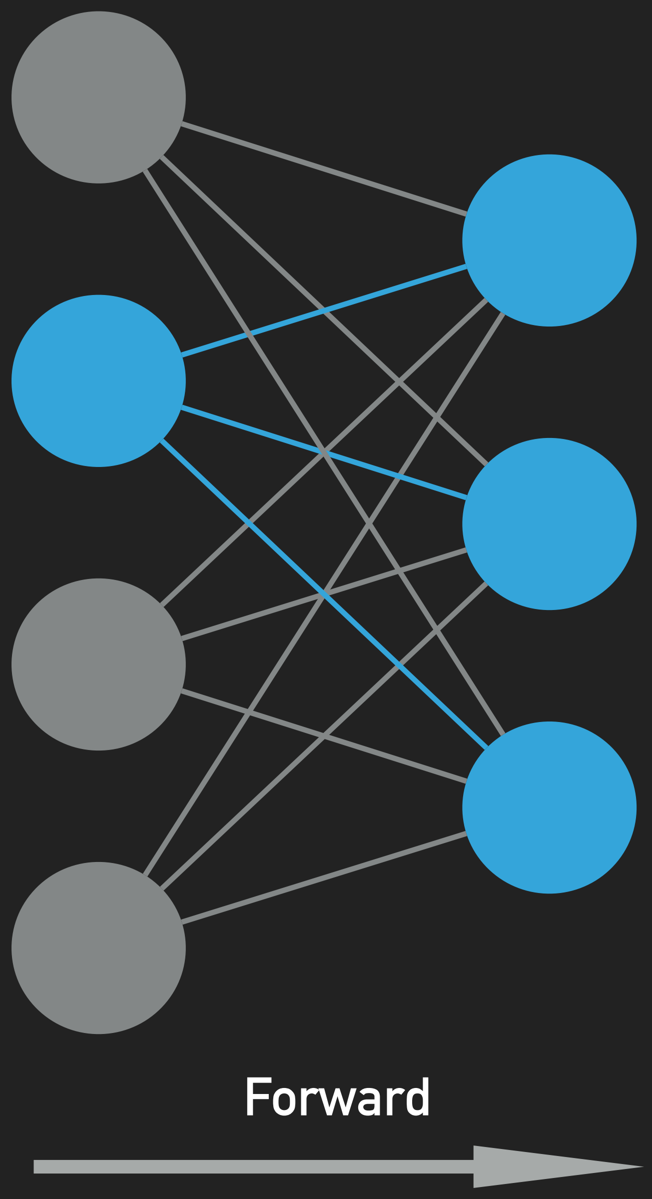
SPLIT PARAMETER BLOCKS

- Parameters: links between two layers
- Parameters will be divided equally
- Division Operation:
  - 4 Parameter Servers:
    {White, Blue, Red, Yellow}
  - 3 Parameter Servers:
    {White, Blue, (Red, Yellow)}
- Sparse Training
- Customizable block hashing method (On Roadmap)



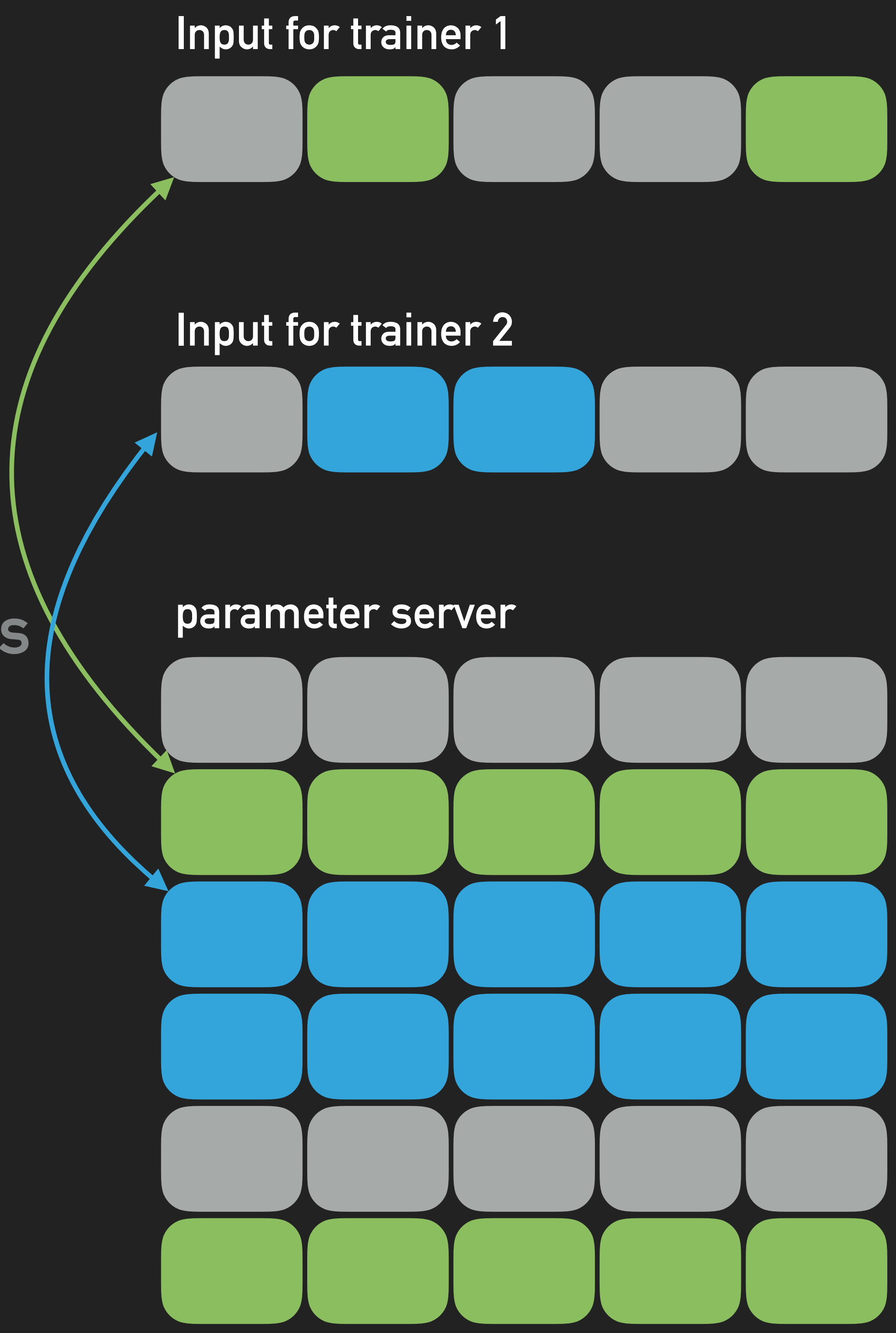
SPARSE TRAINING

- Sparse Training: Input is sparse
- Gray neuron and links: have no effect
  - output is 0, gradient is 0
  - no update (simple SGD)
- Blue neuron and links really matters
  - pull new parameter from PServer
  - calculate gradients, push to PServer



DISTRIBUTED SPARSE TRAINING

- Prefetch Operations
  - First, always beforehand scan training data
  - Label the corresponding parameters
  - Pull the latest parameter from PServer
- Forward/Backward
  - calculate gradients, push them to PServer

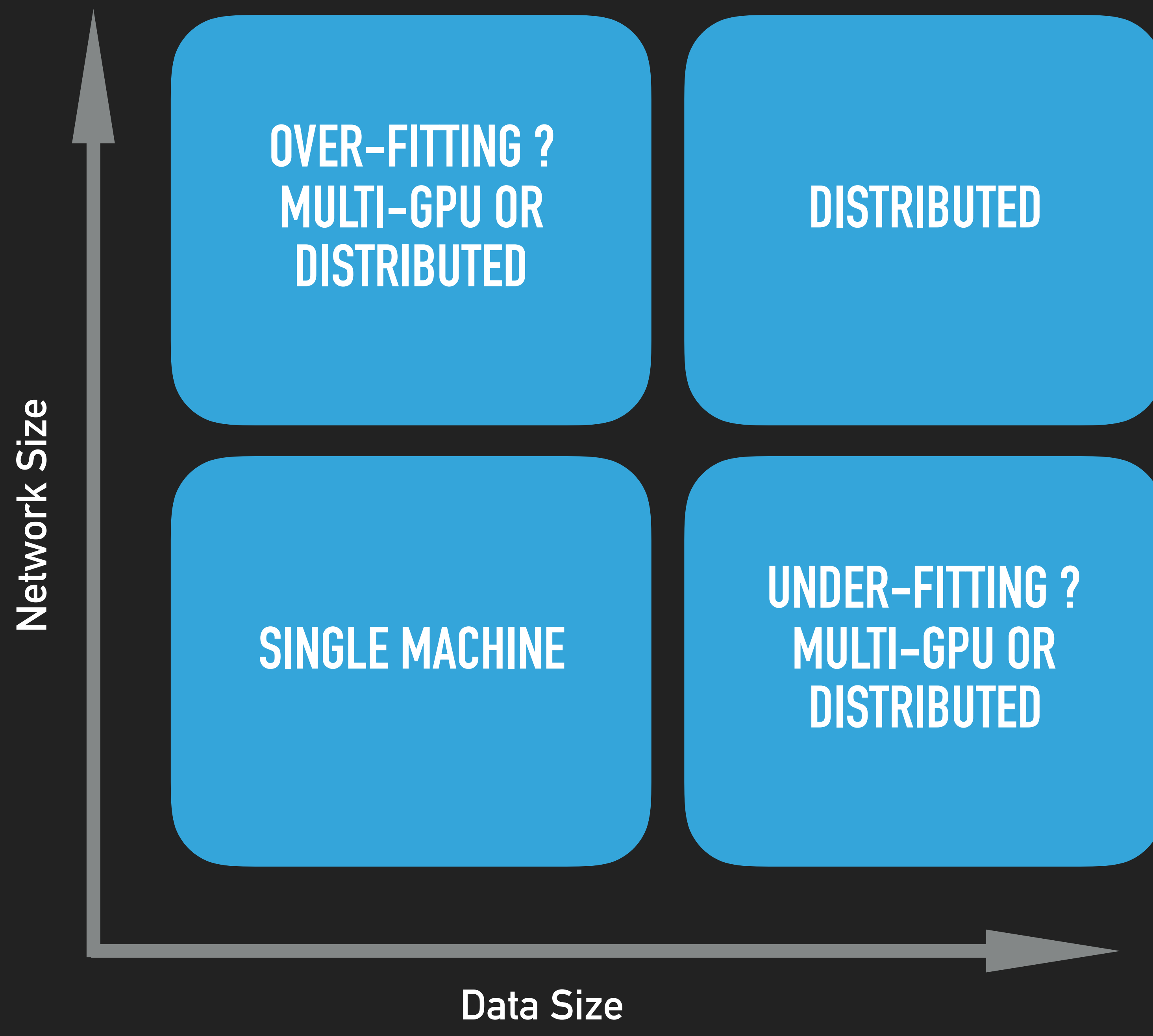


OUTLINE

- Introduction
- PaddlePaddle Fluid
- Elastic Distributed Training
- Multi-GPU
- Sequence Model and LoDTensor
- Official Resources

DISTRIBUTED VS SINGLE MACHINE

- Distributed training isn't free
- Overhead
  - Synchronization
  - Network transfer data
  - Setup time (preparing and loading training data)
  - hyperparameter tuning
- Training on single machine until time becomes prohibitive

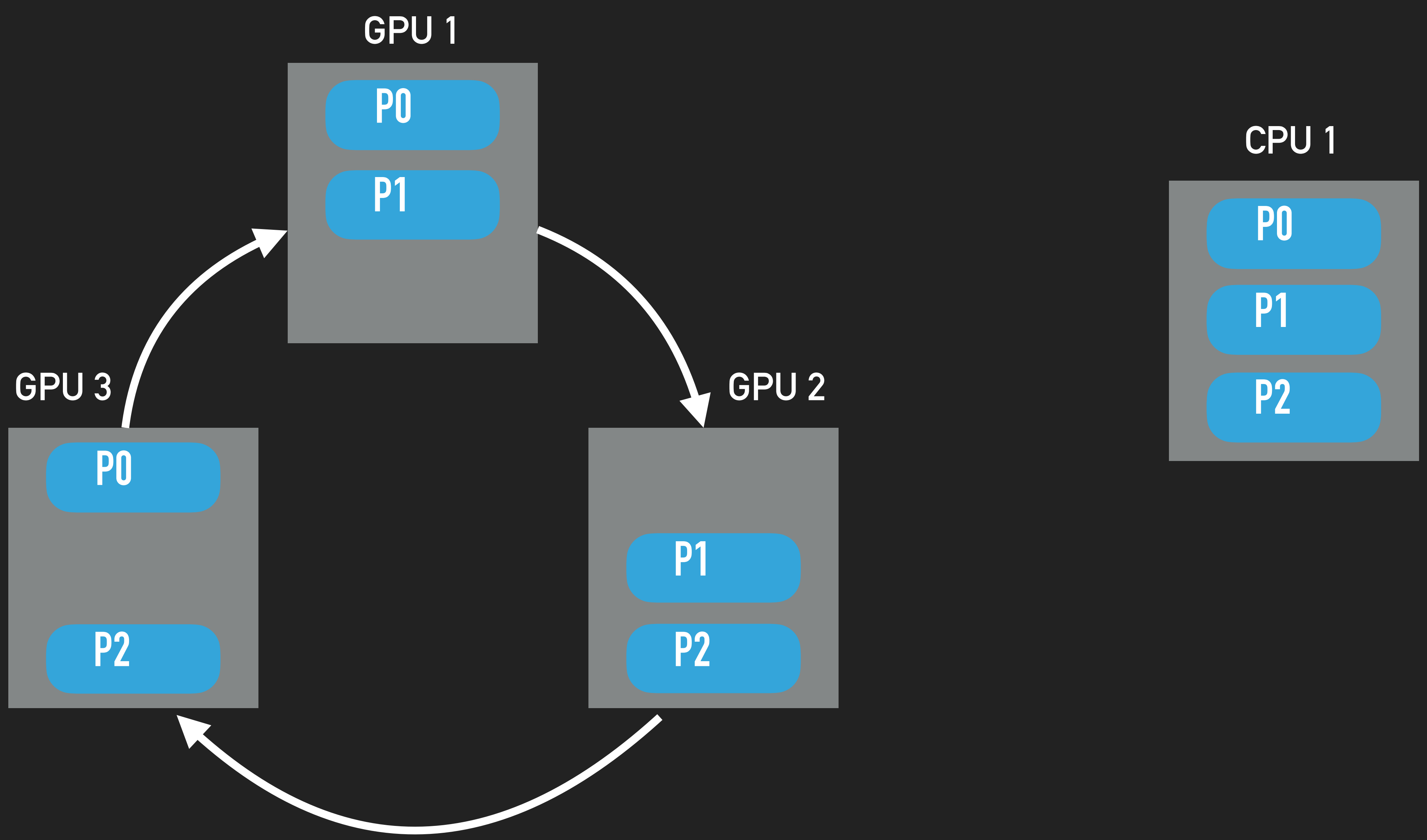


DISTRIBUTED VS SINGLE MACHINE

- Another Perspective:
  - the ratio of network transfers to computation
- Distributed Training is more efficient when the ratio is low
  - small and shallow networks are not good candidates
  - increase batch size and learning rate
- Thus, in some cases, multi-GPU system is considered before

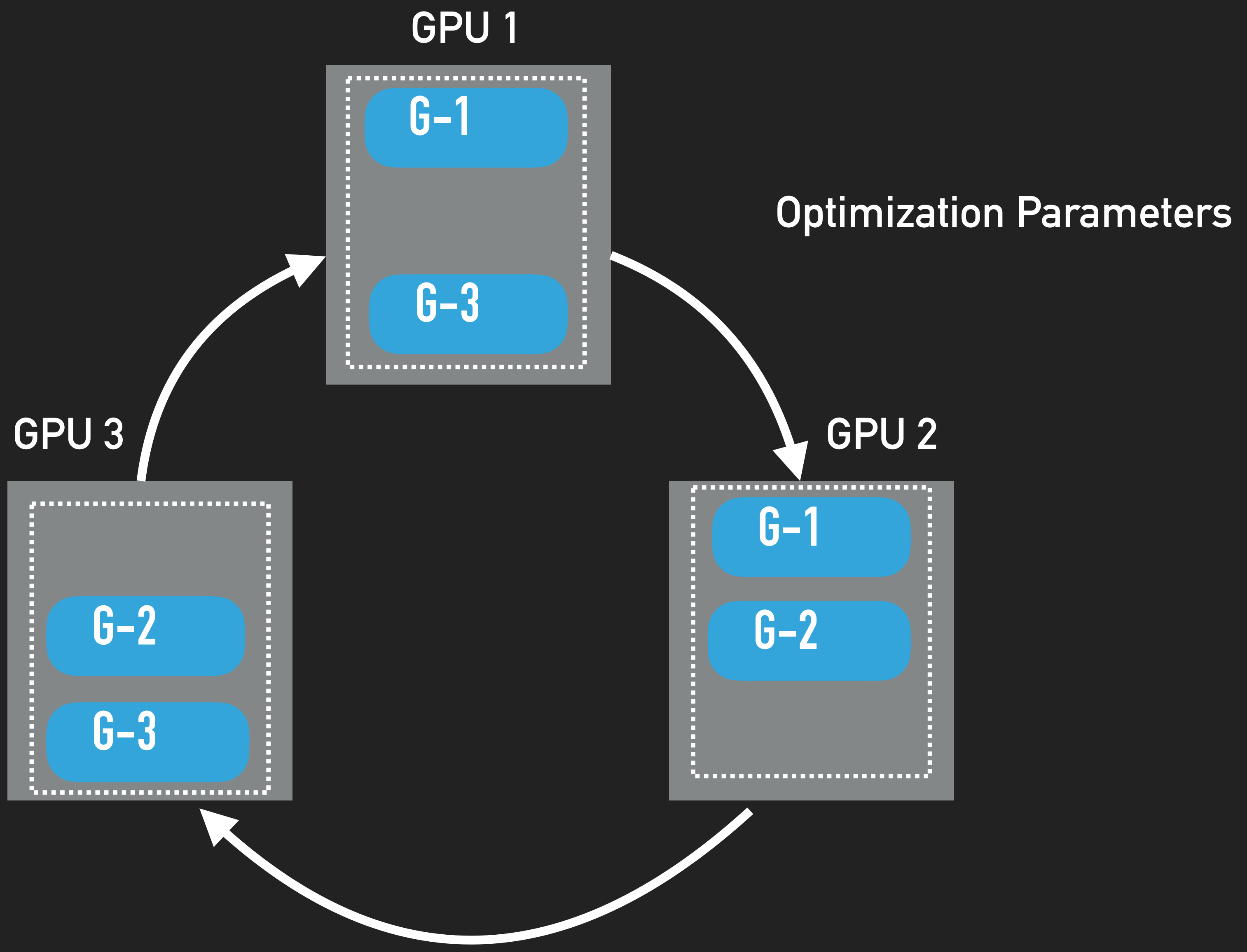
MULTI-GPU COMMUNICATION

- Ring-based network communication
- Hand out Parameters: each parameter has a master card



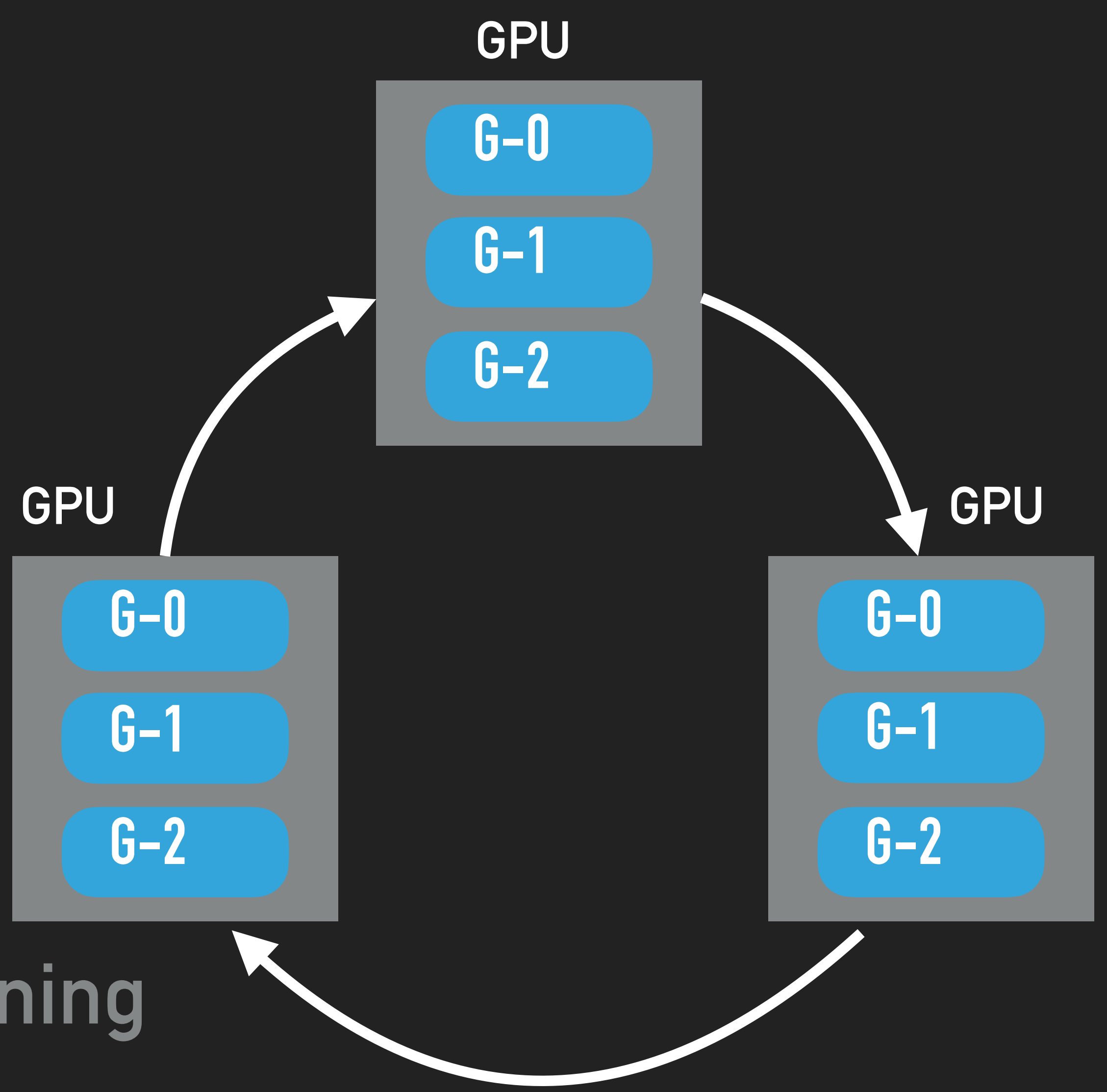
MULTI-GPU COMMUNICATION IN PADDLEPADDLE

- Ring-based network communication



MULTI-GPU COMMUNICATION IN PADDLEPADDLE

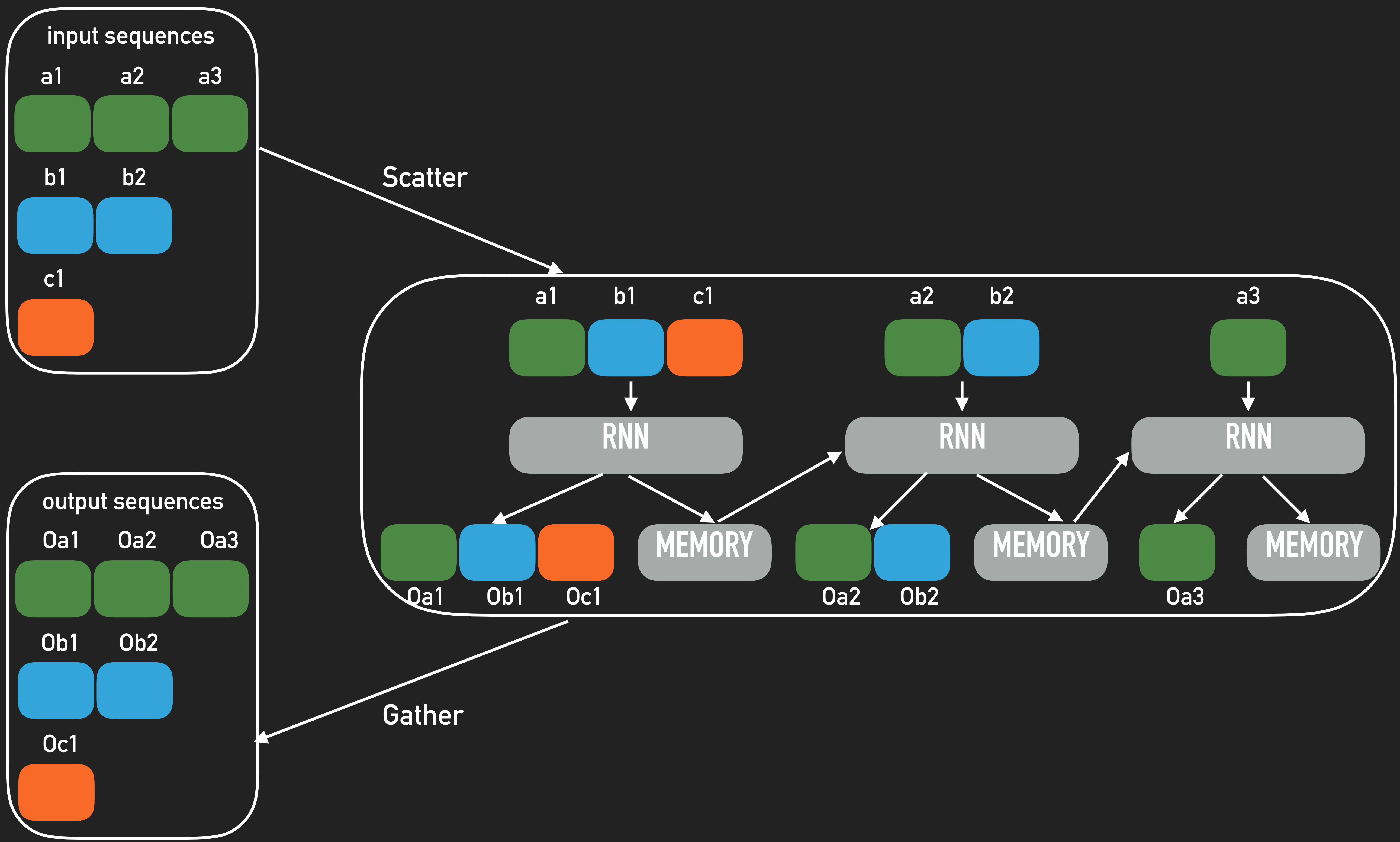
- Ring-based network communication
- Advantages:
  - SGD method is simple
  - network communication utilization
  - coarse-grained synchronization
- Disadvantages:
  - no Async SGD
  - network overhead when sparse training



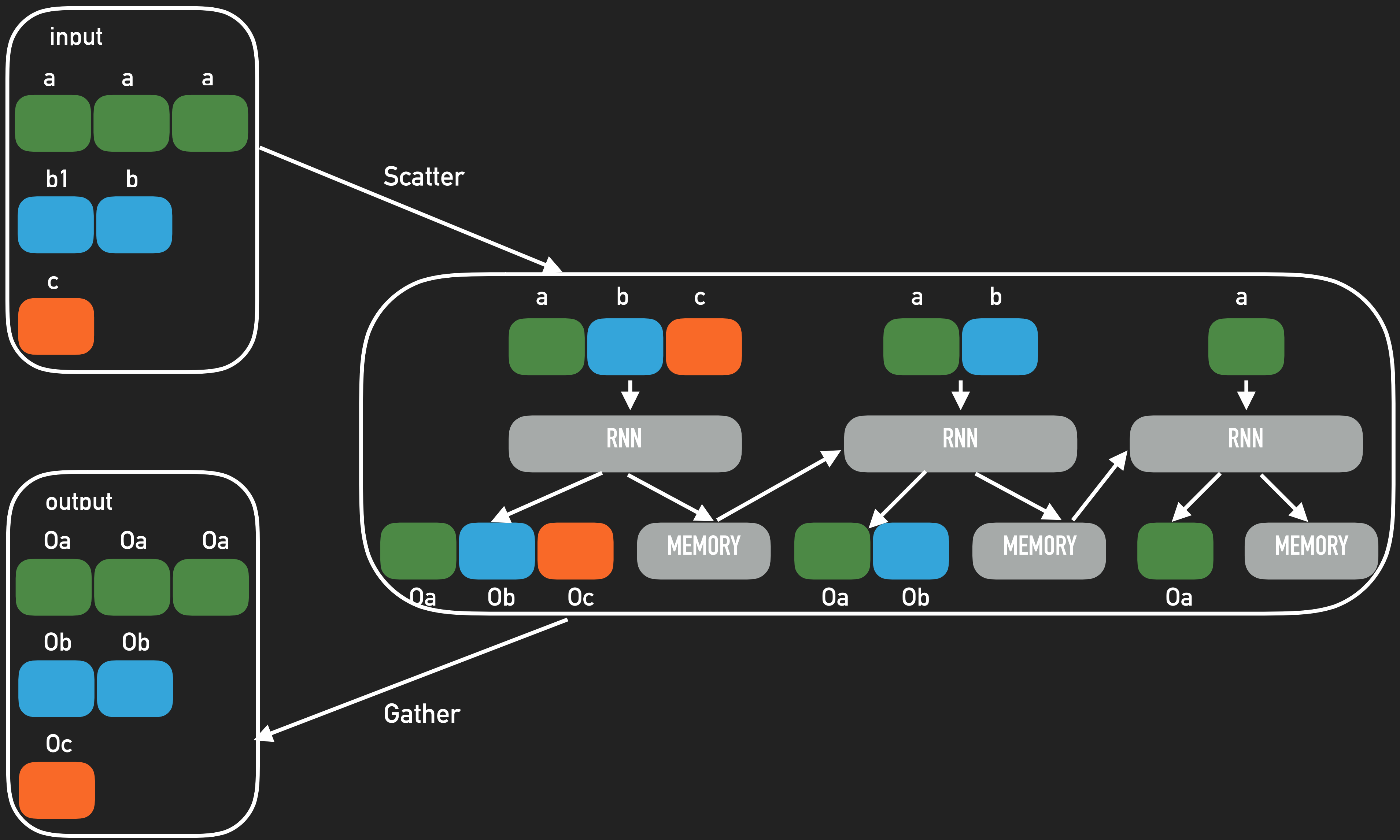
OUTLINE

- Introduction
- PaddlePaddle Fluid
- Elastic Distributed Training
- Multi-GPU
- Sequence Model and LoDTensor
- Official Resources

SEQUENCE MODEL IN PADDLE PADDLE



SEQUENCE MODEL IN PADDLEPADDLE



- Support arbitrary complicated RNN
- No Padding
- Efficiently batch processing

OUTLINE

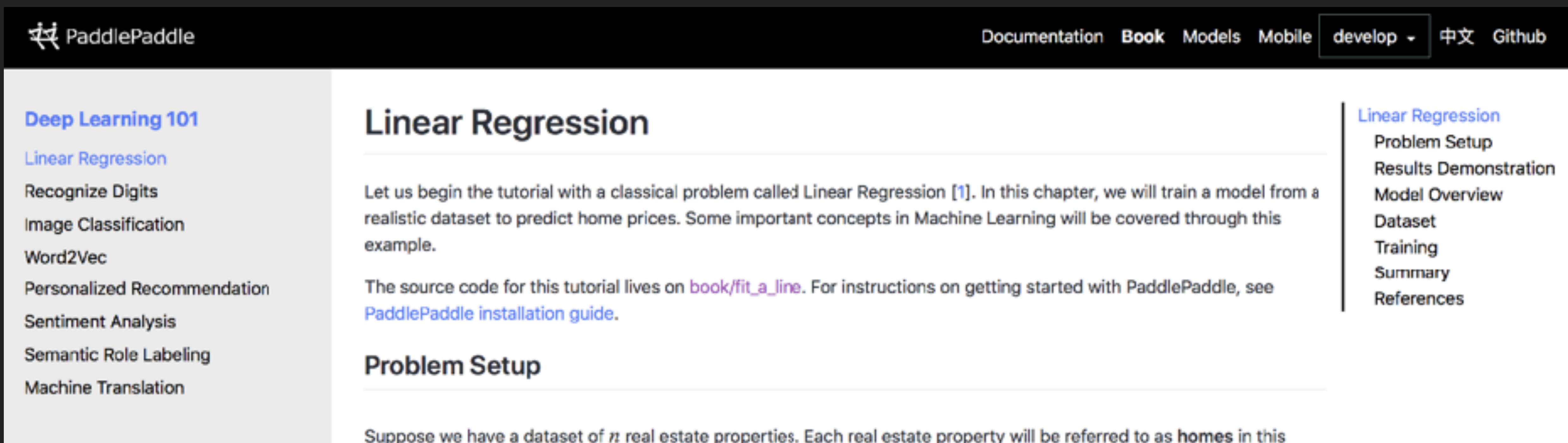
- Introduction
- PaddlePaddle Fluid
- Elastic Distributed Training
- Multi-GPU
- Sequence Model and LoDTensor
- Official Resources

OFFICIAL SITE AND DOCUMENTS

- http://www.paddlepaddle.org

PADDLEPADDLE BOOK

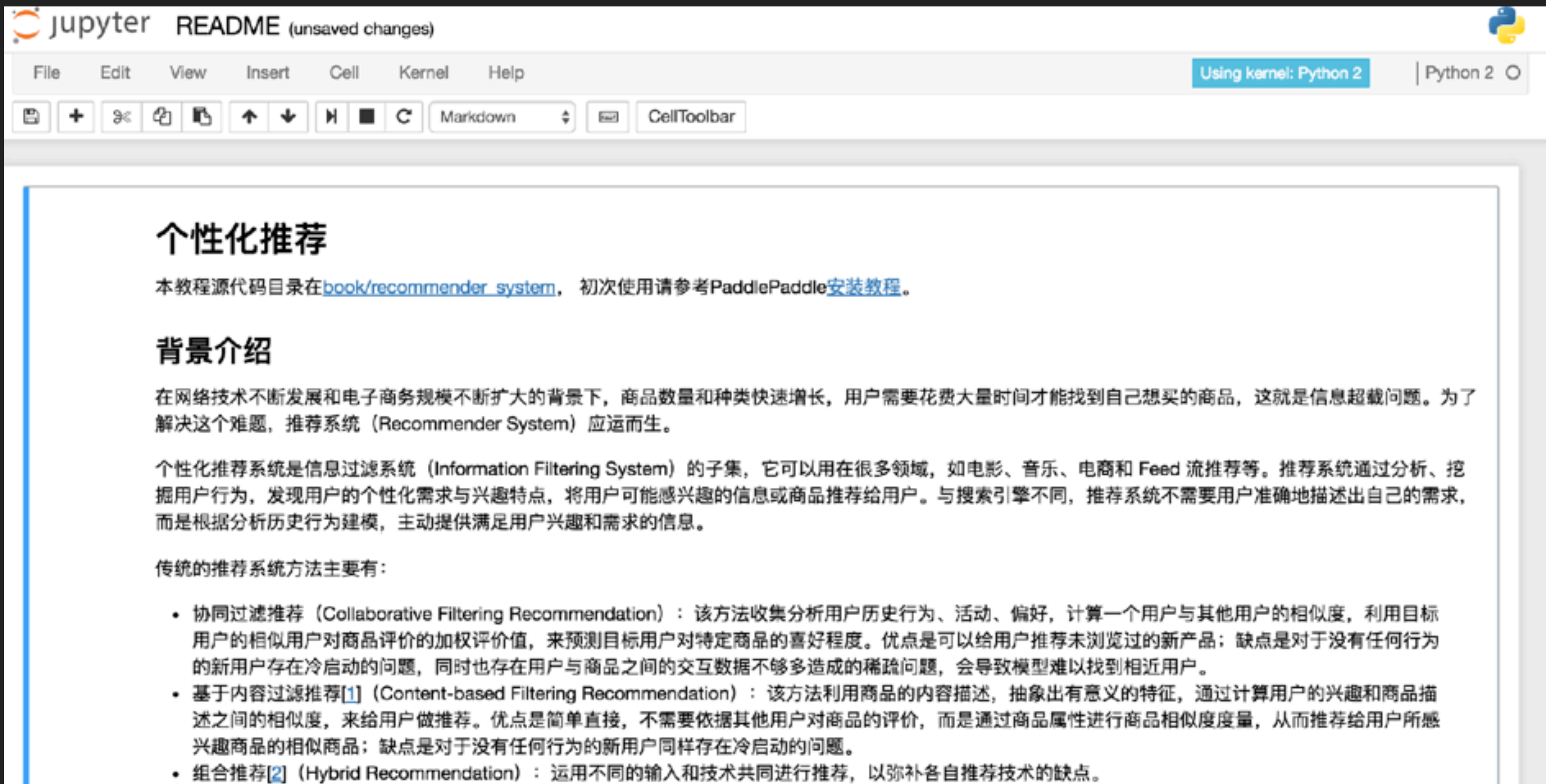
- http://www.paddlepaddle.org/docs/develop/book/01.fit_a_line/



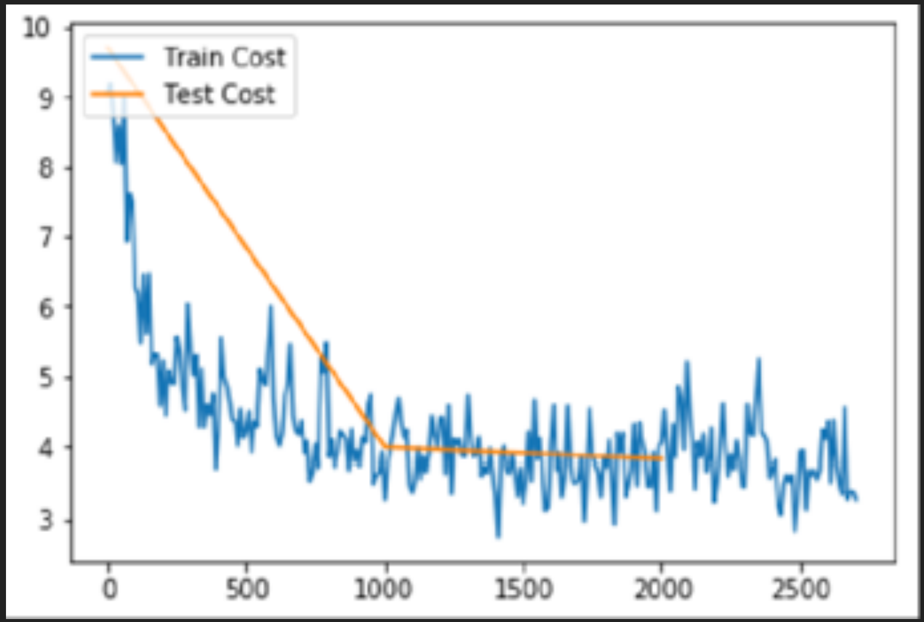
DOCKER BASED JUPYTER INTERACTIVE NOTEBOOK

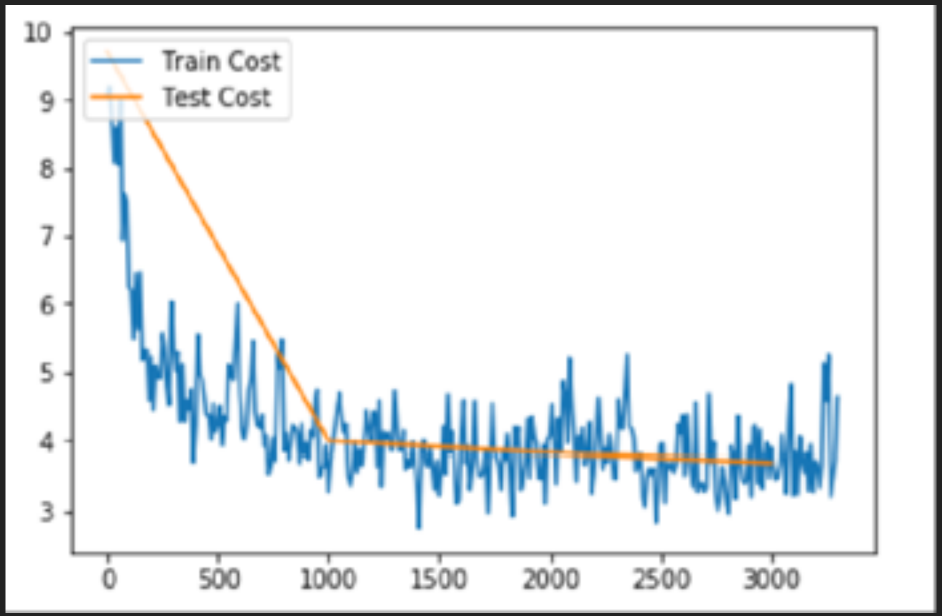
- Documents that contain live code, equations, visualizations and explanatory text in a single browser.
- 1. Pull and Run the book image:
  - DockerHub.com:
    - docker run -d -p 8888:8888 paddlepaddle/book
  - docker.paddlepaddle.org: (user in China)
    - docker run -d -p 8888:8888 docker.paddlepaddle.org/book
- 2. Local browser:
  - http://localhost:8888/

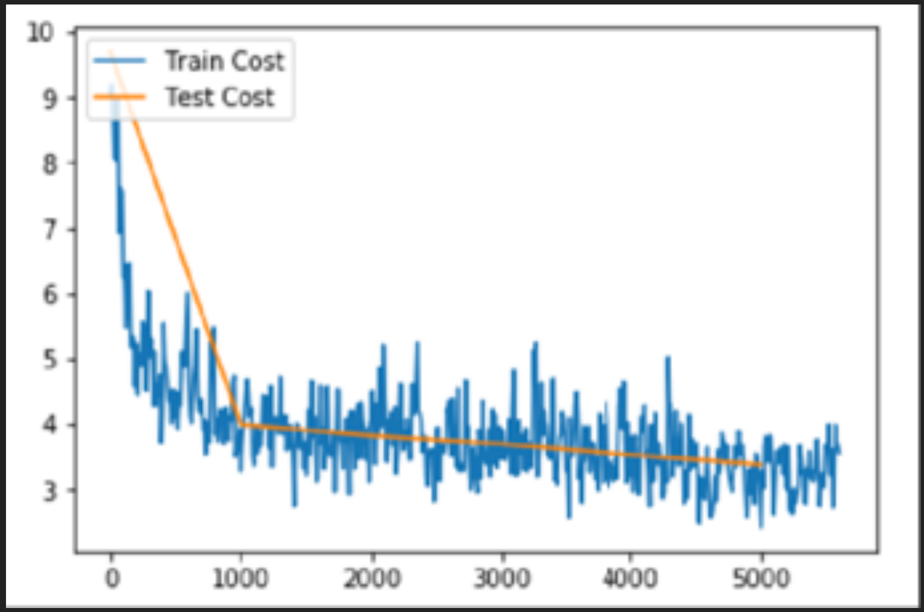
DOCKER BASED JUPYTER INTERACTIVE NOTEBOOK

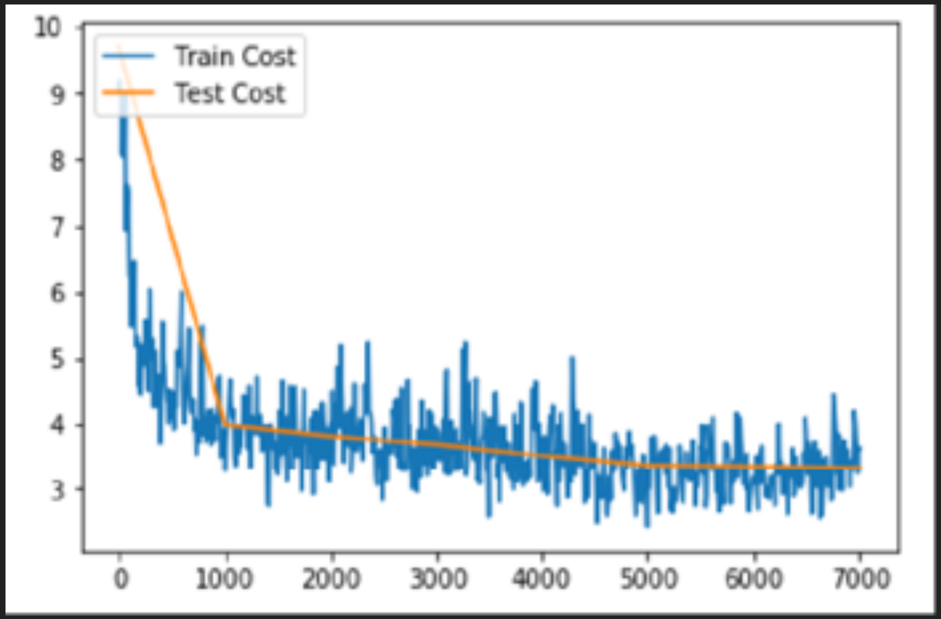


DOCKER BASED JUPYTER INTERACTIVE NOTEBOOK



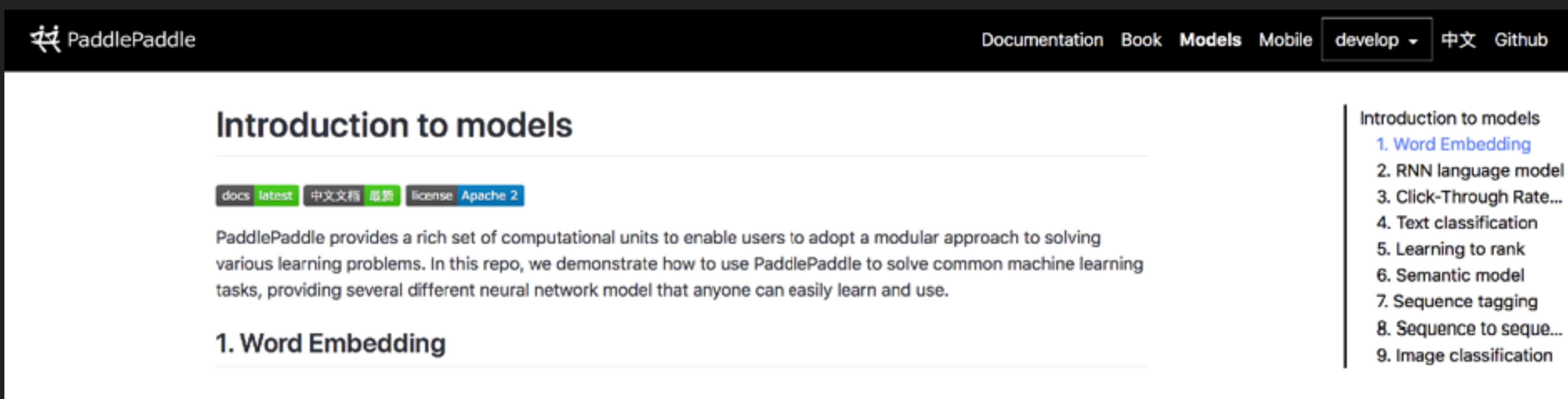






PRODUCTION LEVEL MODEL BANK

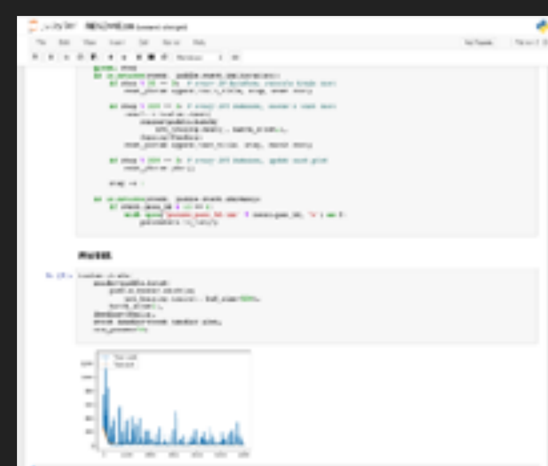
- http://www.paddlepaddle.org/docs/develop/models/README.html
- Pre-Trained models



PADDLEPADDLE CLOUD

- https://github.com/PaddlePaddle/cloud
- http://cloud.dlnel.org

深度学习开发方式的变革：开发、实验、分布式任务一键提交



OUTLINE

- Introduction
- PaddlePaddle Fluid
- Elastic Distributed Training
- Sequence Model and LoDTensor
- Official Resources

THANK YOU