

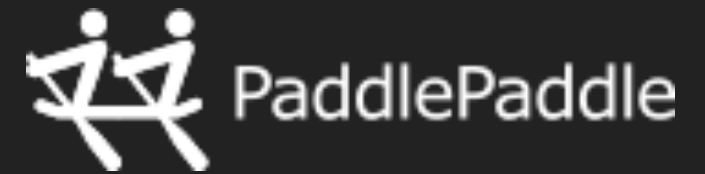
PARALLEL AND DISTRIBUTED DEEP LEARNING

PADDLEPADDLE

OUTLINE

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

INTRODUCTION



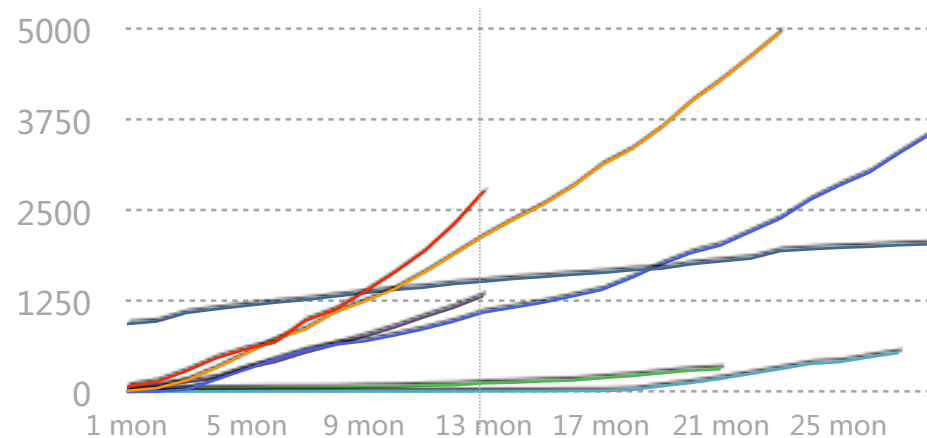
- ▶ Official Website: <http://paddlepaddle.org/>
- ▶ Github: <https://github.com/PaddlePaddle/Paddle>
- ▶ Deep Learning Platform for both Enterprise & Research
- ▶ Easy-to-use, Flexibility, Efficiency and Scalability
- ▶ Official Release: pypi and Docker image



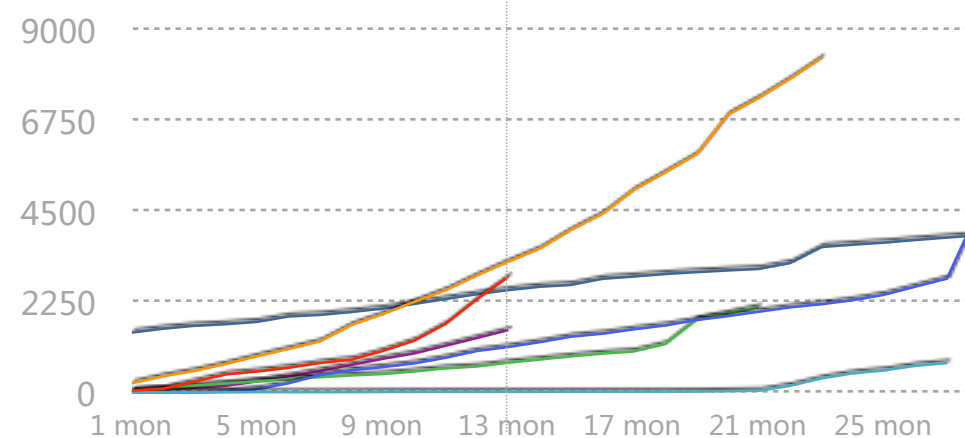
A BIT HISTORY

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

Github activities (Pull requests)



Github activities (Issues)



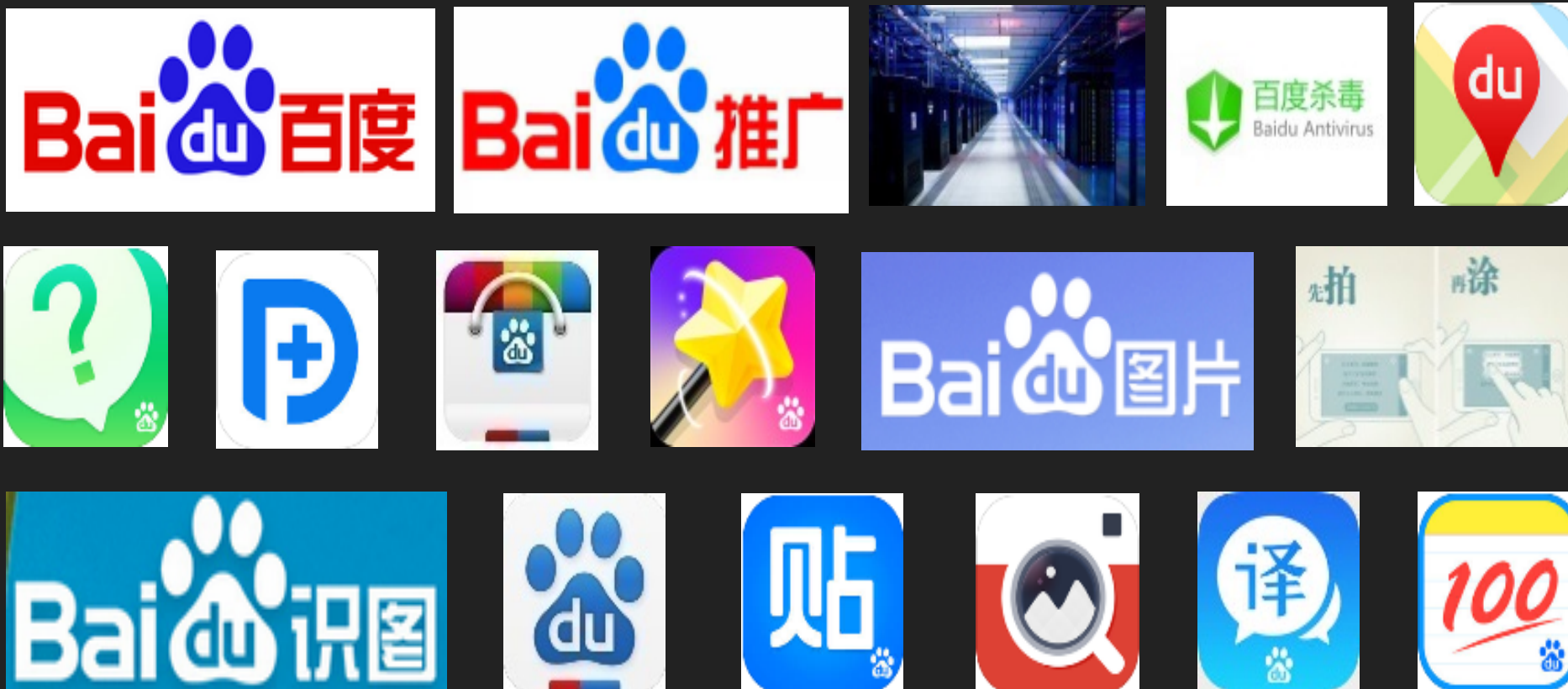
— PaddlePaddle
— TensorFlow
— Caffe
— MXNet
— PyTorch
— Caffe2
— CNTK

WHY WE START THE PADDLEPADDLE PROJECT ?

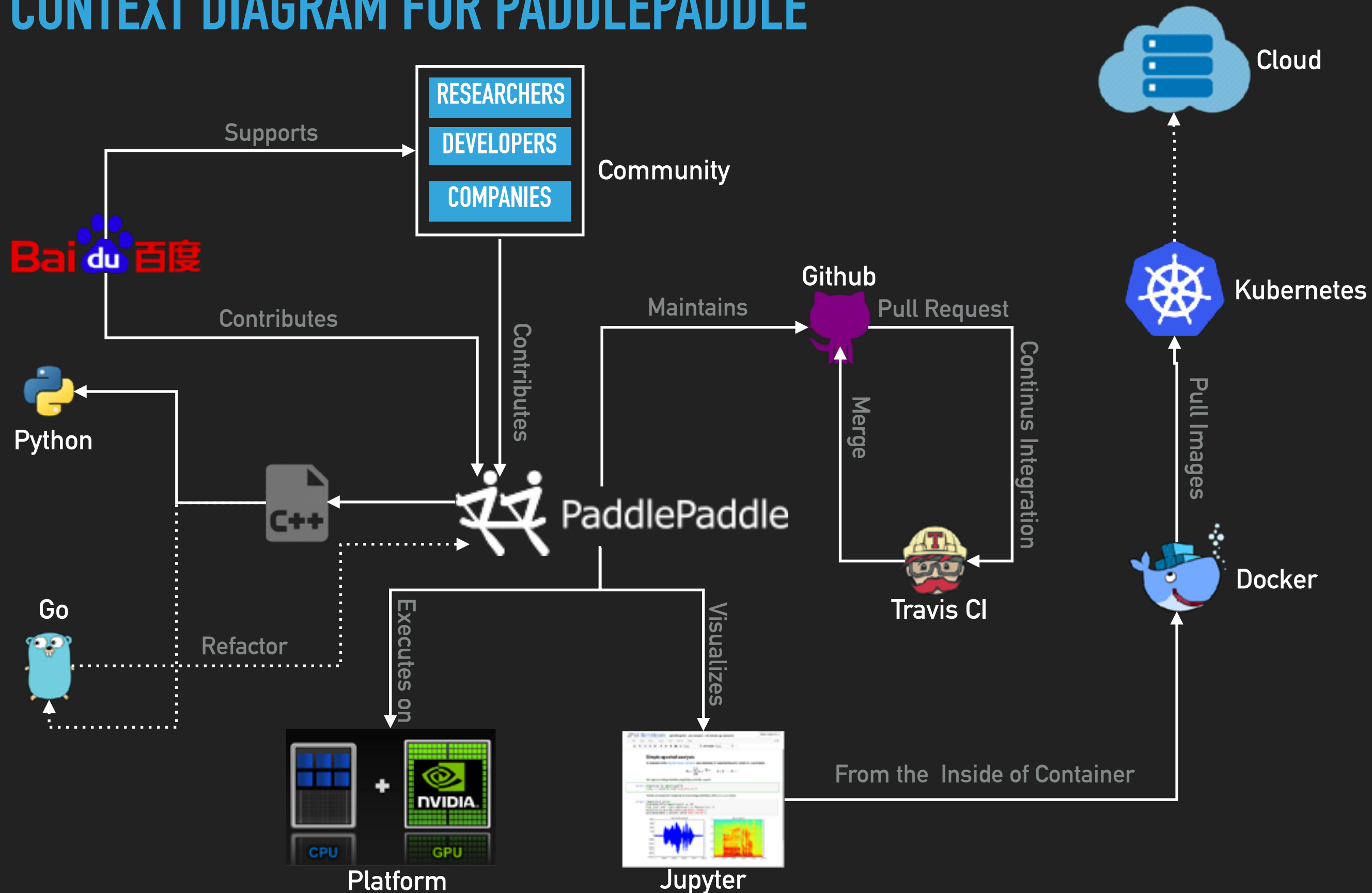
- ▶ **Make use of billion data**
- ▶ People do not have such open source platforms in 2013
- ▶ Toolkits on deep learning were about image recognition in 2013
- ▶ Requirement: Products around Our Search Engine
- ▶ Text based Problem (Natural Language Understanding)

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 = U * *hh;
    int s = fc_out + hidden_out;
    int act = sigmoid(sum);
    mem[i] = act;
    o1[i] = act; o2[i] = hidden_out;
}
```

运行效率高

TensorFlow

PaddlePaddle Fluid

PyTorch

编程便捷

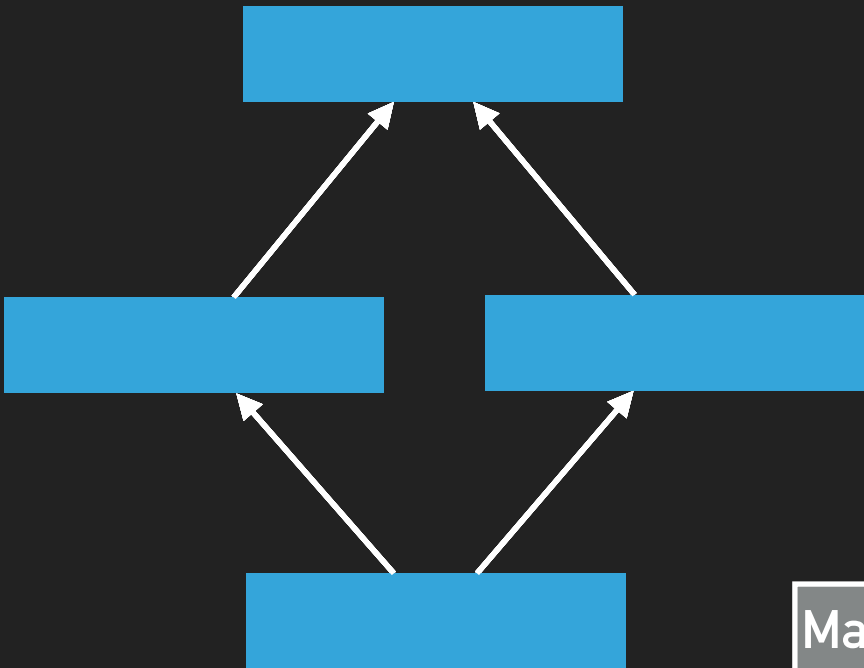
PADDLEPADDLE FLUID

- ▶ 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

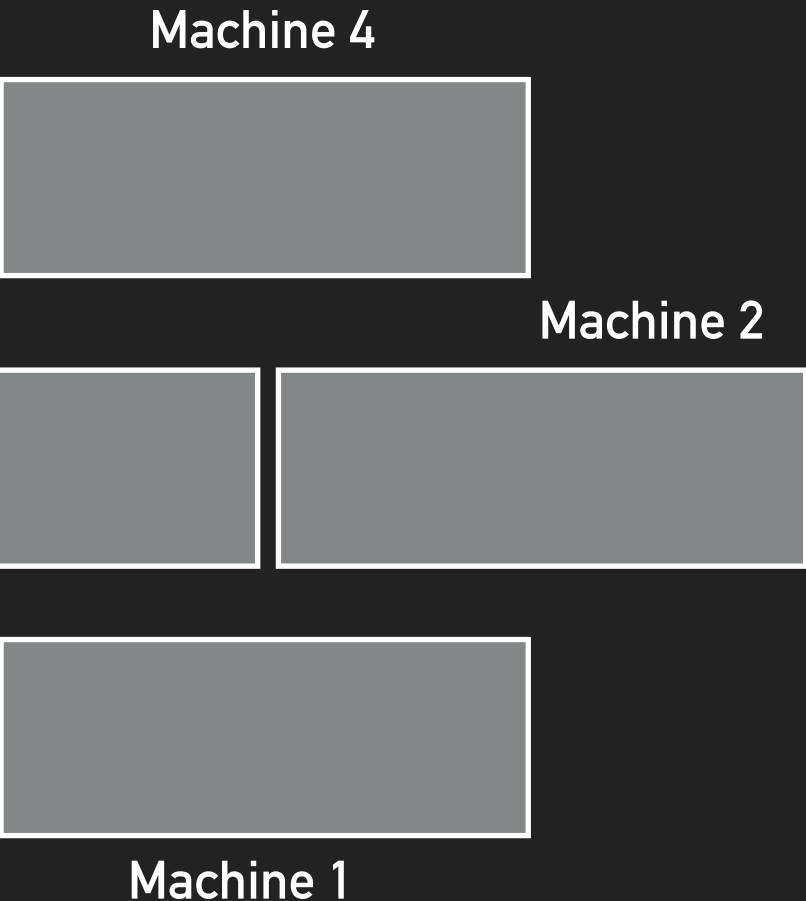
OUTLINE

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

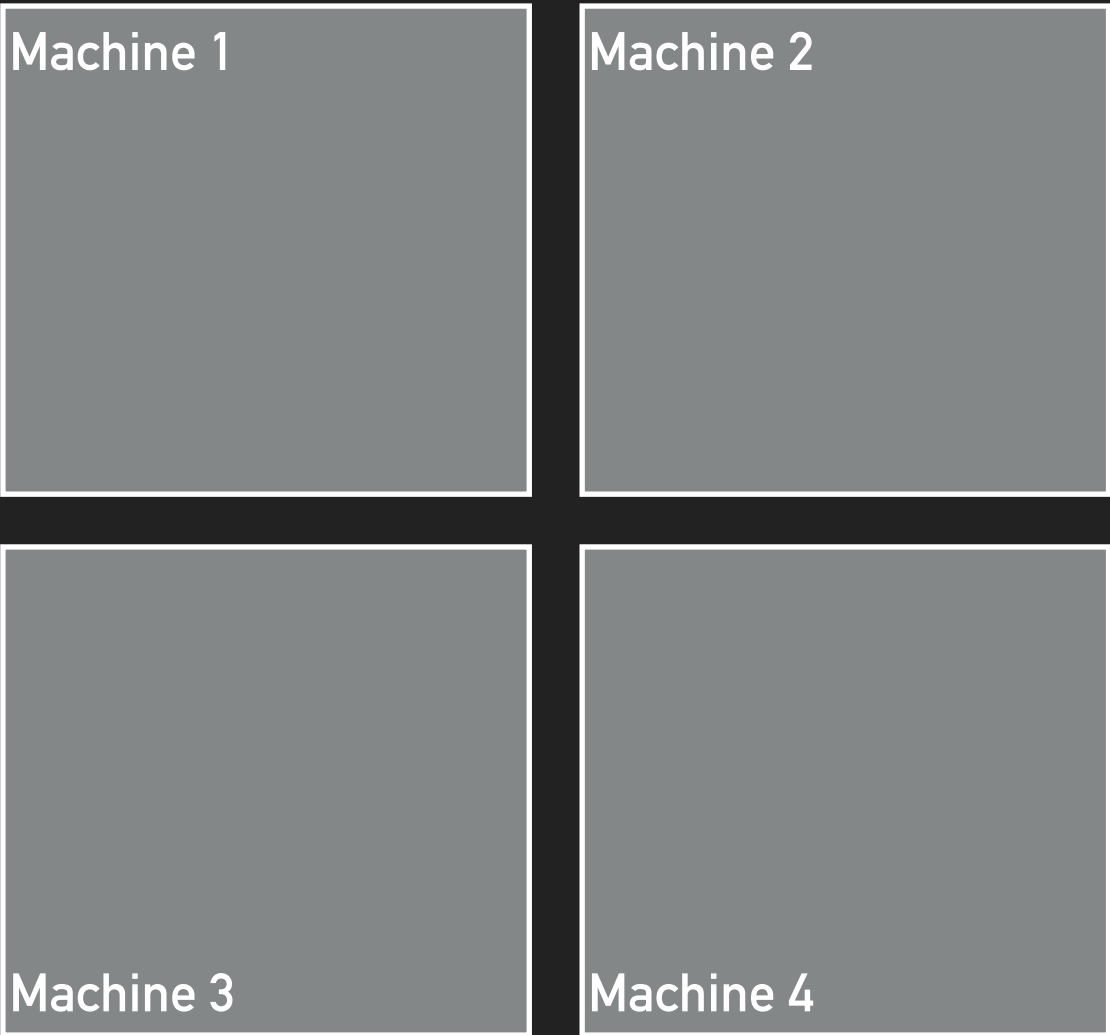
DISTRIBUTED TRAINING



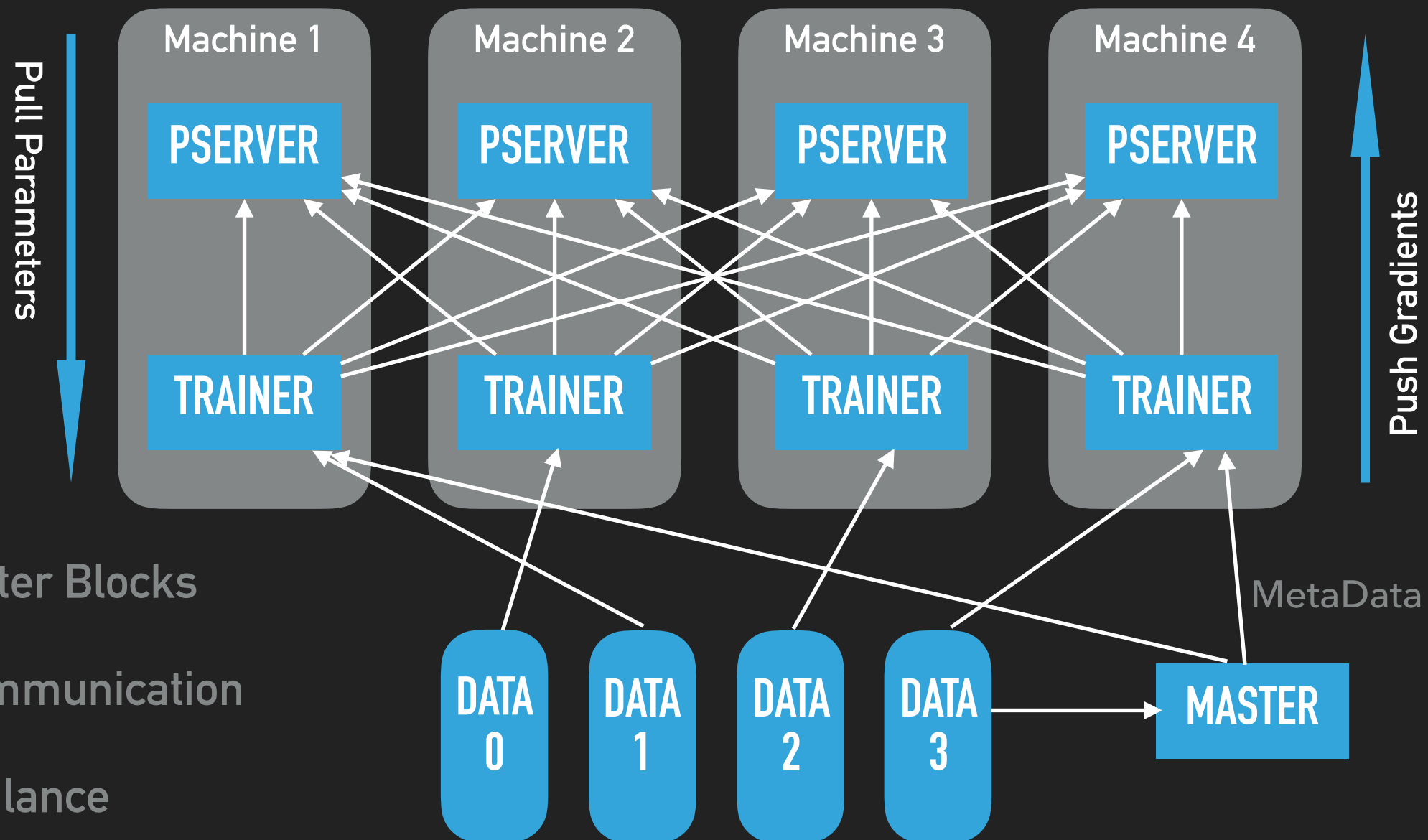
Model Parallelism



Data Parallelism



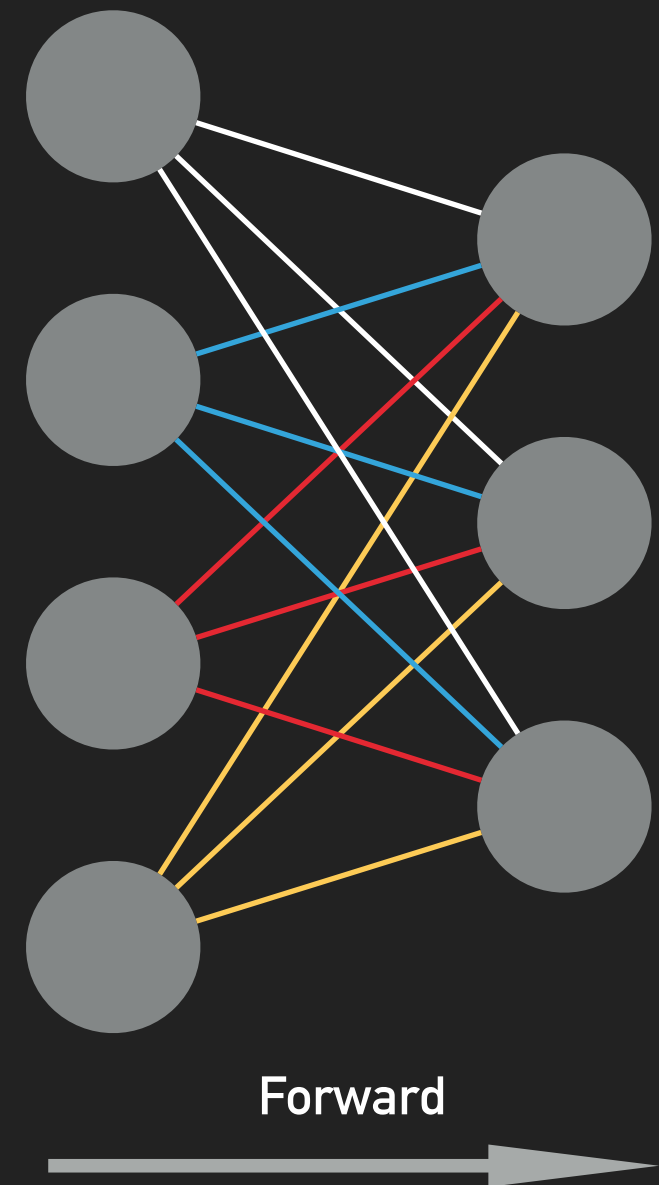
FAULT TOLERANT DISTRIBUTED TRAINING



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

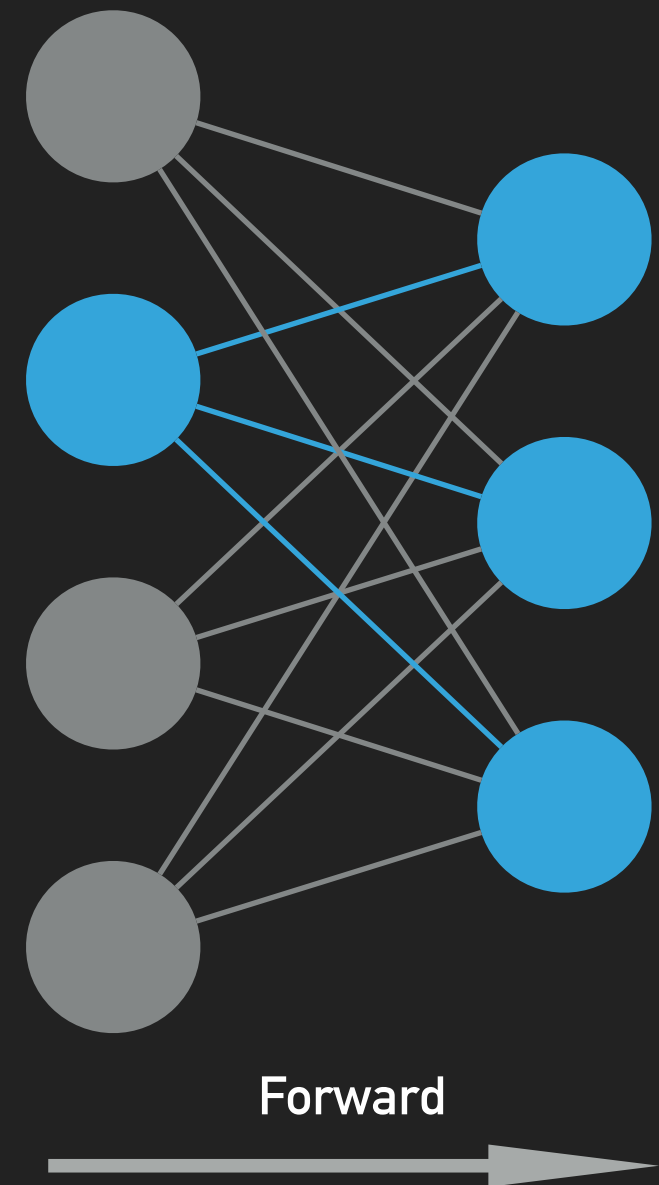
SPLIT PARAMETER BLOCKS

- ▶ 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



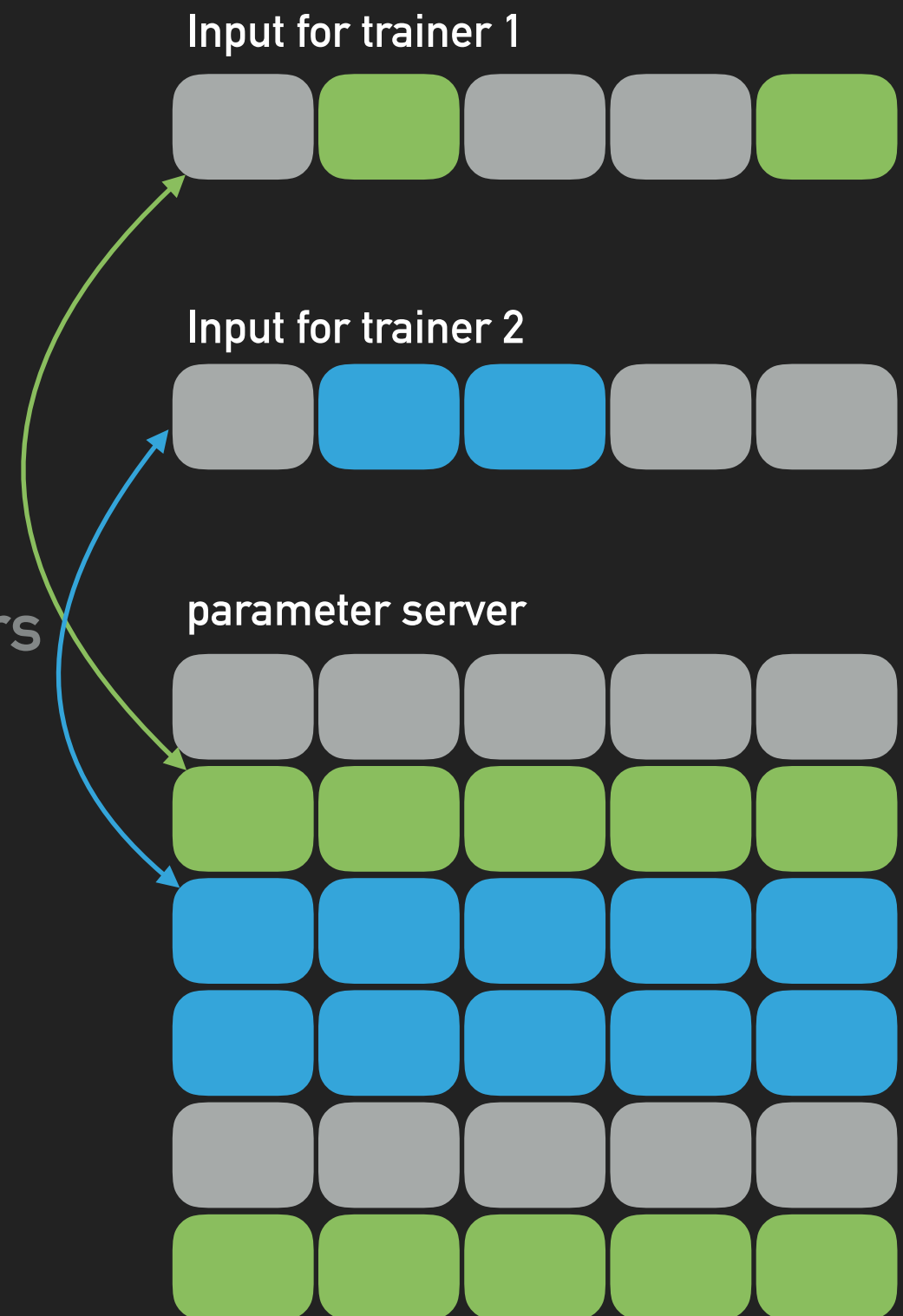
DISTRIBUTED SPARSE TRAINING

▶ 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

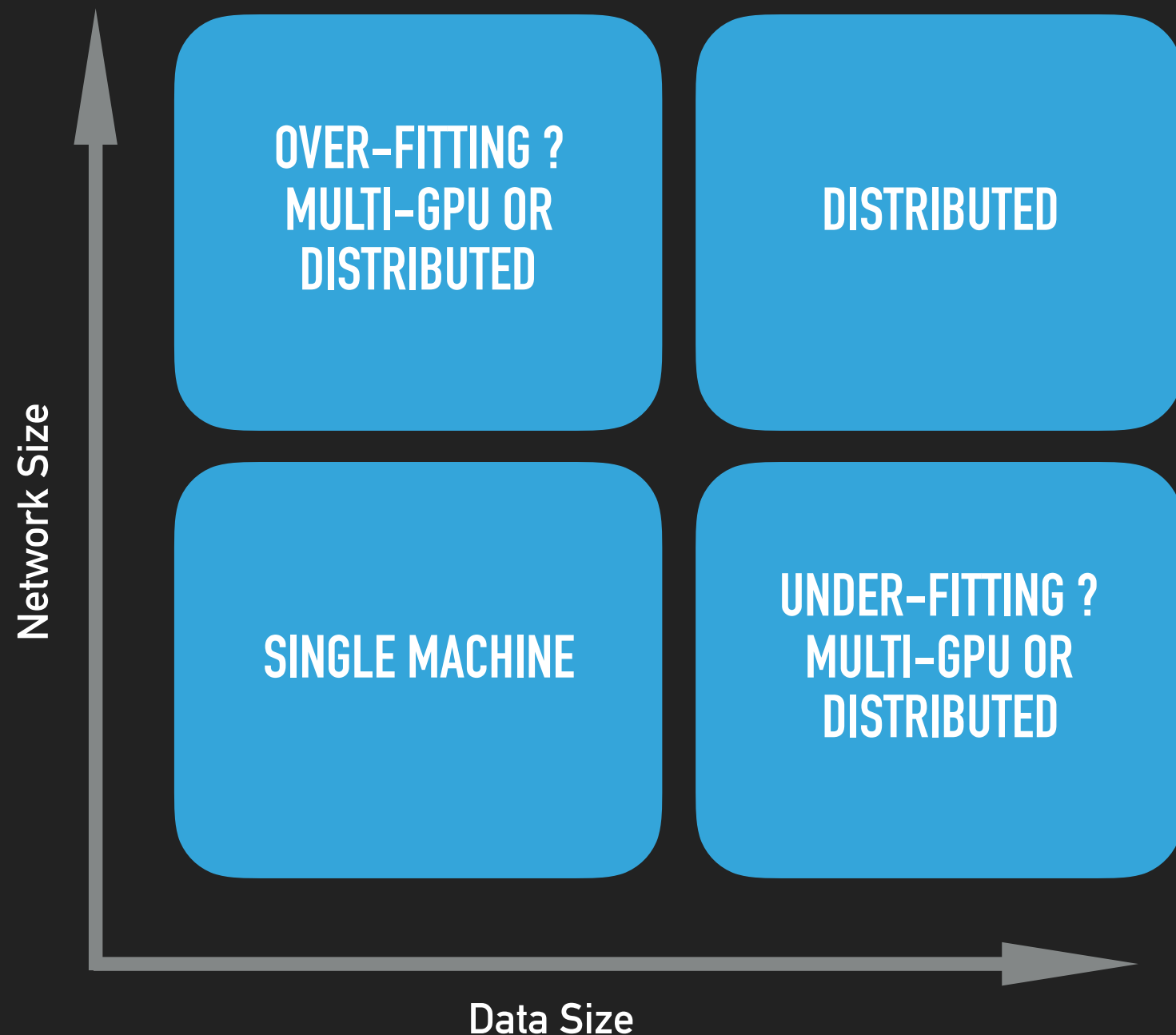


OUTLINE

- ▶ 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

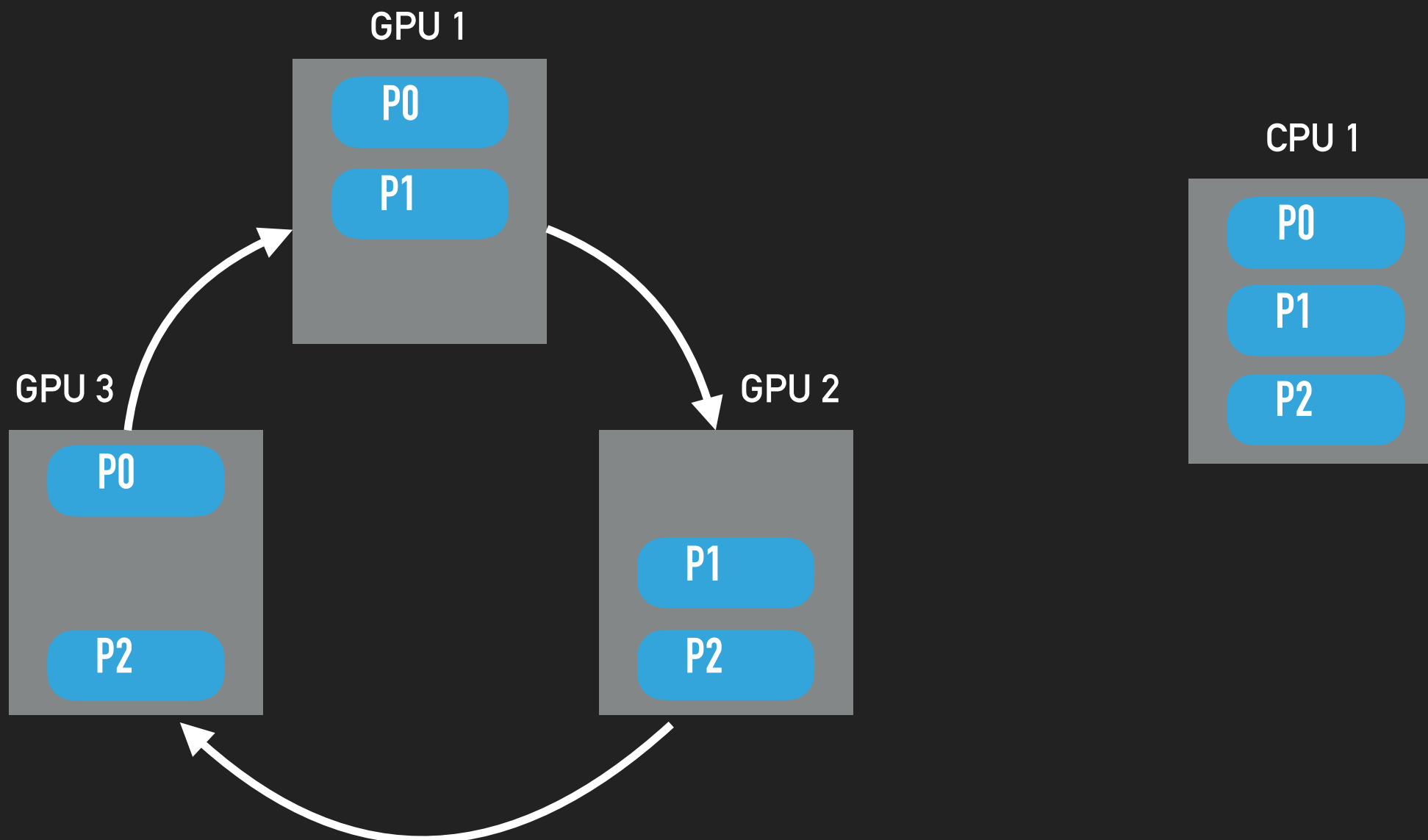


DISTRIBUTED VS SINGLE MACHINE

- ▶ 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

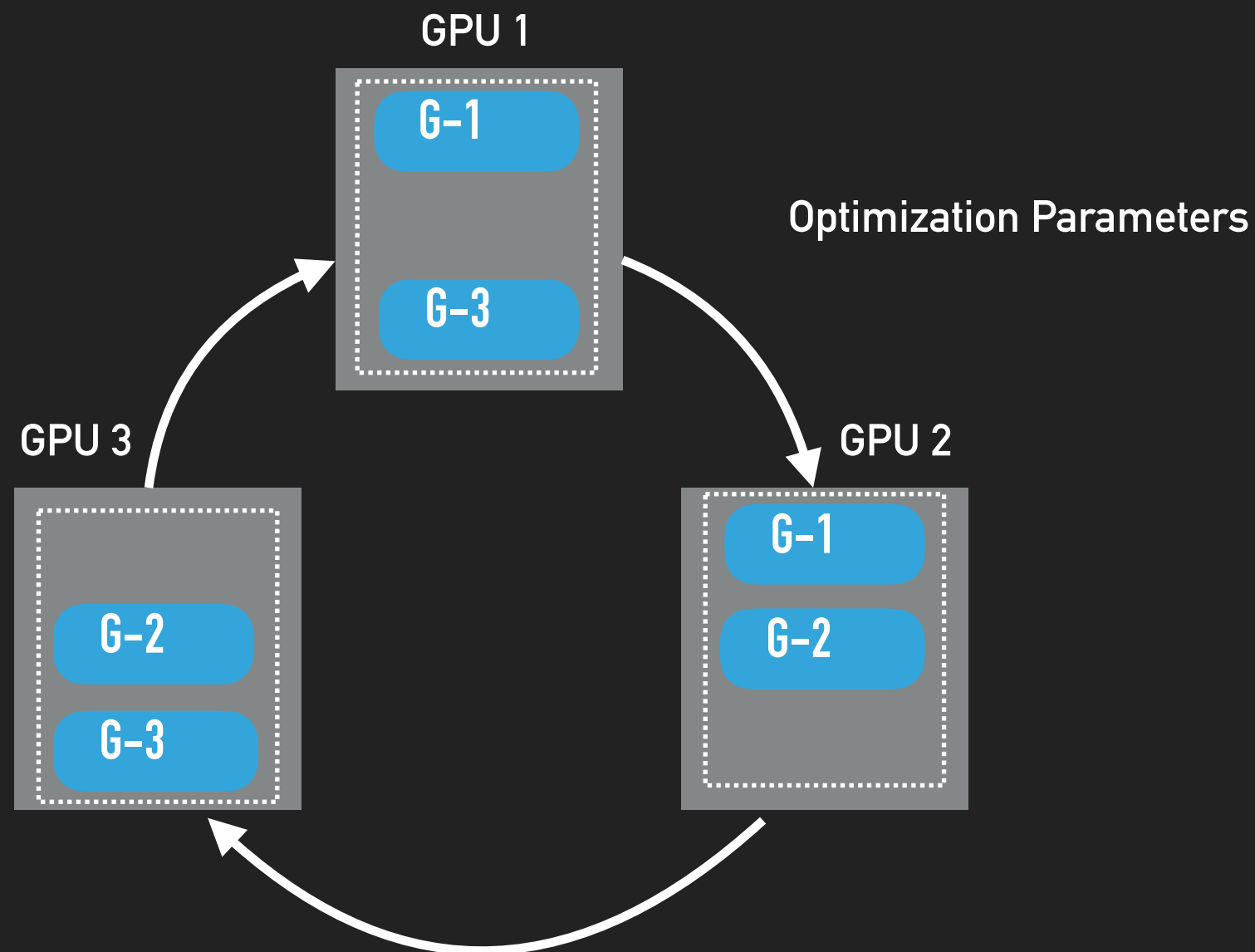
MULTI-GPU COMMUNICATION

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



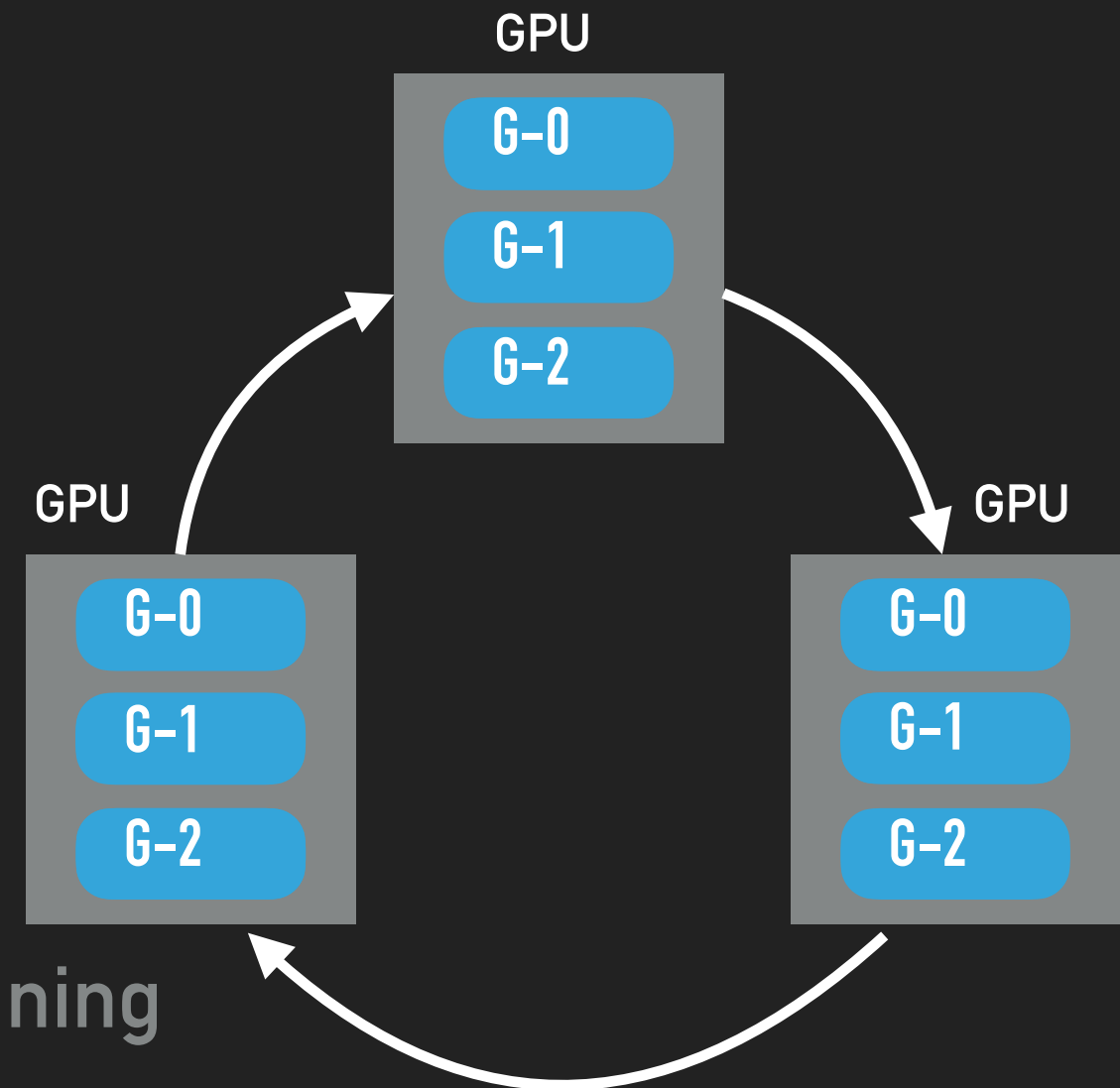
MULTI-GPU COMMUNICATION IN PADDLEPADDLE

► Ring-based network communication



MULTI-GPU COMMUNICATION IN PADDLEPADDLE

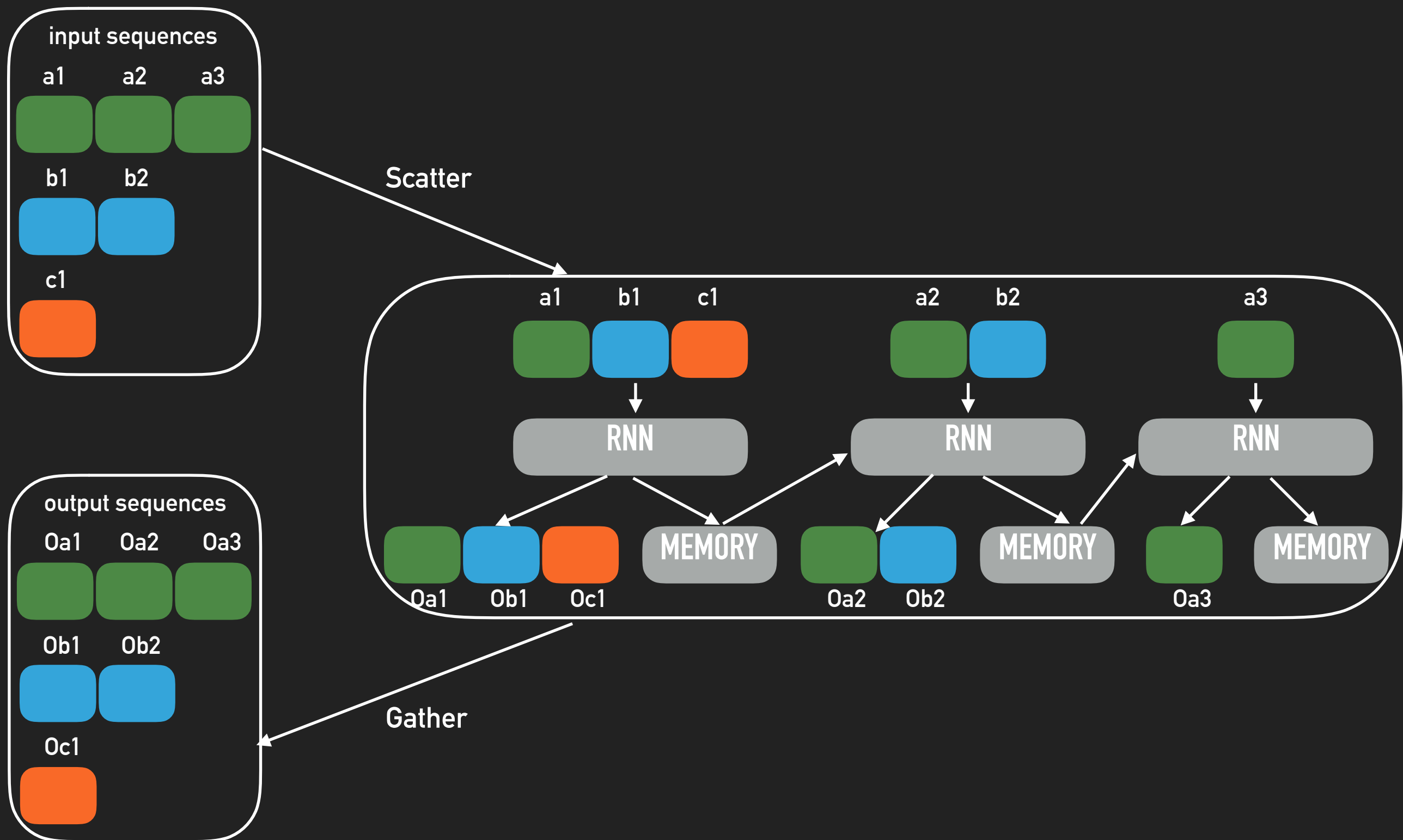
- ▶ Ring-based network communication
- ▶ Advantages:
 - ▶ SGD method is simple
 - ▶ network communication utilization
 - ▶ coarse-grained synchronization
- ▶ Disadvantages:
 - ▶ no Async SGD
 - ▶ network overhead when sparse training



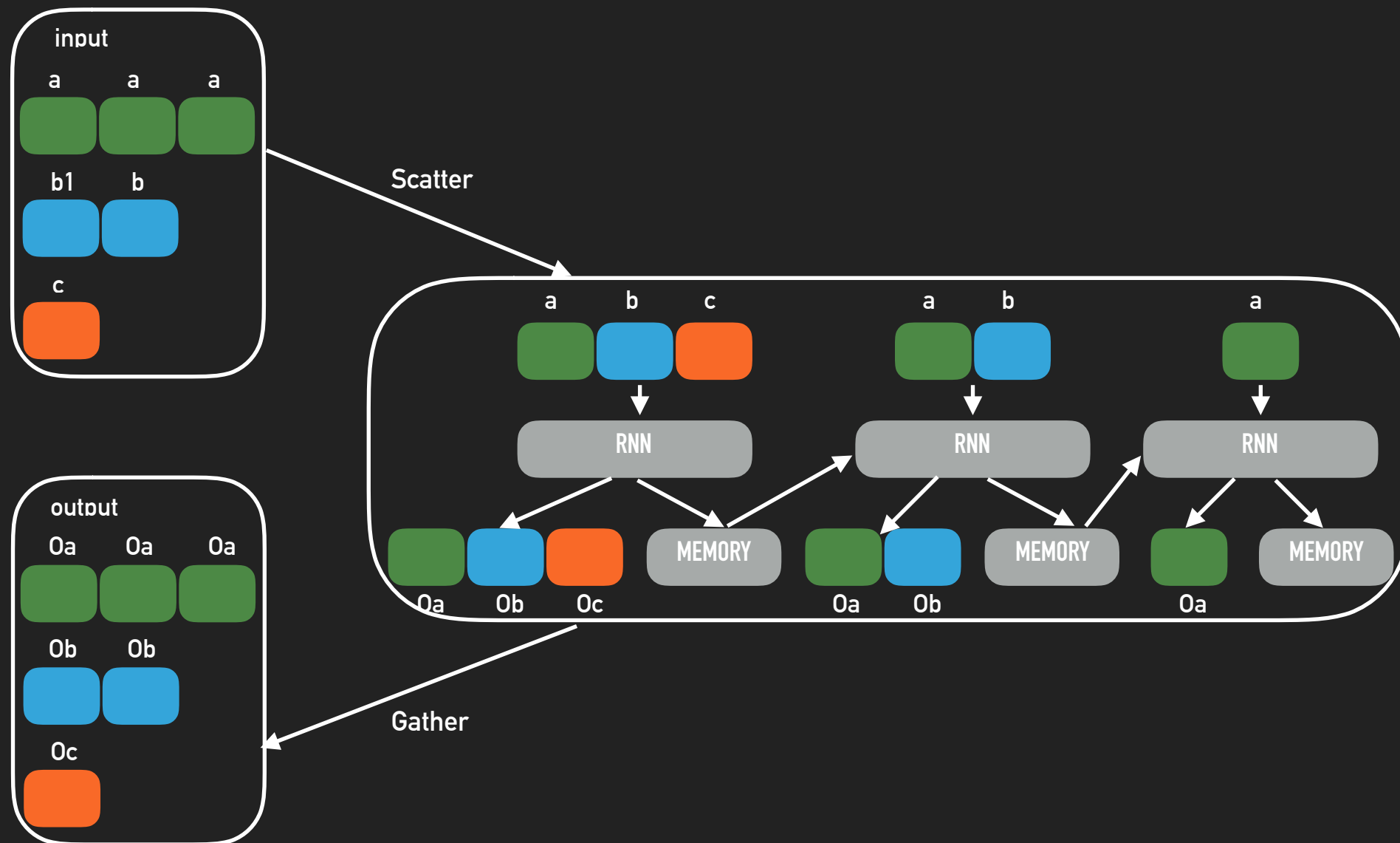
OUTLINE

- ▶ 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/

 PaddlePaddle

Documentation **Book** Models Mobile **develop** 中文 Github

Deep Learning 101
Linear Regression
Recognize Digits
Image Classification
Word2Vec
Personalized Recommendation
Sentiment Analysis
Semantic Role Labeling
Machine Translation

Linear Regression

Let us begin the tutorial with a classical problem called Linear Regression [1]. In this chapter, we will train a model from a realistic dataset to predict home prices. Some important concepts in Machine Learning will be covered through this example.

The source code for this tutorial lives on [book/fit_a_line](#). For instructions on getting started with PaddlePaddle, see [PaddlePaddle installation guide](#).

Problem Setup

Suppose we have a dataset of n real estate properties. Each real estate property will be referred to as **homes** in this

Linear Regression
Problem Setup
Results Demonstration
Model Overview
Dataset
Training
Summary
References

 PaddlePaddle

文档 教程 模型库 **develop** English Github

深度学习入门
新手入门
识别数字
图像分类
词向量
个性化推荐
情感分析
语义角色标注
机器翻译

线性回归

让我们从经典的线性回归（Linear Regression [1]）模型开始这份教程。在这一章里，你将使用真实的数据集建立起一个房价预测模型，并且了解到机器学习中的若干重要概念。

本教程源代码目录在[book/fit_a_line](#)，初次使用请参考PaddlePaddle[安装教程](#)，更多内容请参考本教程的[视频课堂](#)。

背景介绍

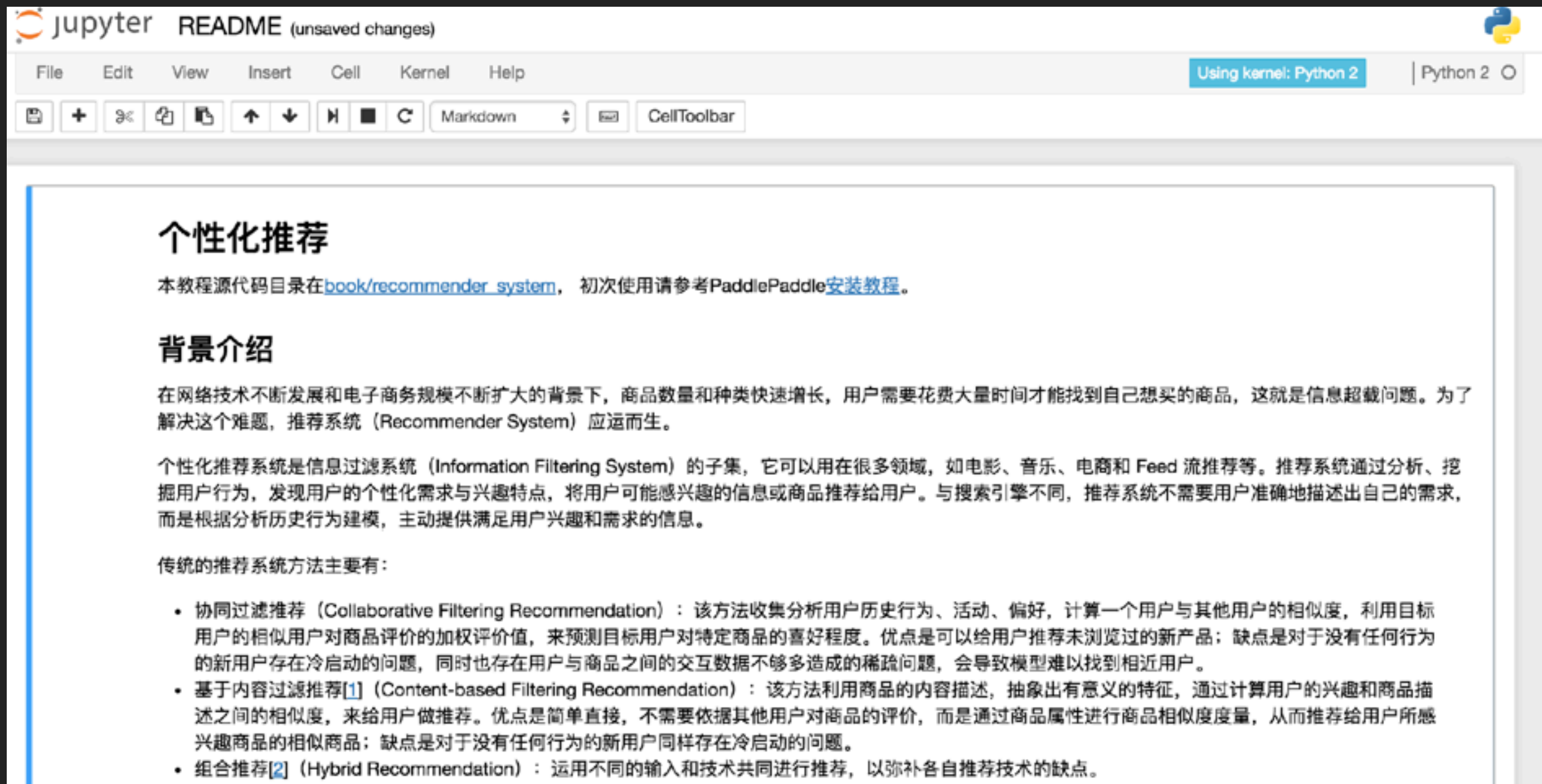
给定一个大小为 n 的数据集 $\{y_i, x_{i1}, \dots, x_{id}\}_{i=1}^n$ ，其中 x_{i1}, \dots, x_{id} 是第 i 个样本 d 个属性上的取值， y_i 是该样本待预测的目

线性回归
背景介绍
效果展示
模型概览
数据集
训练
总结
参考文献

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



The screenshot shows a Jupyter Notebook interface with a menu bar (File, Edit, View, Insert, Cell, Kernel, Help) and a toolbar. The notebook title is "jupyter README (unsaved changes)". The kernel is set to "Python 2". The main content area displays a README file with the following text:

个性化推荐

本教程源代码目录在[book/recommender_system](#)，初次使用请参考PaddlePaddle[安装教程](#)。

背景介绍

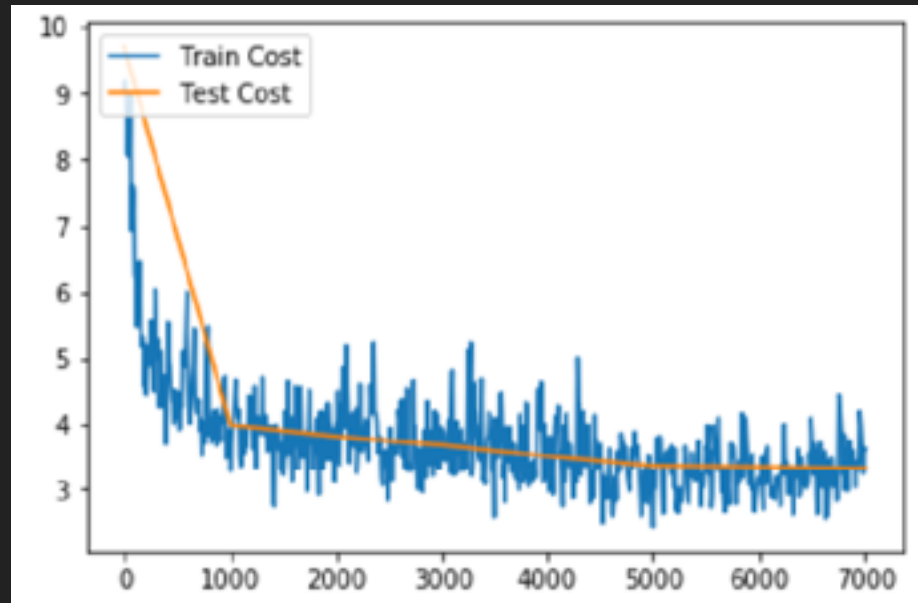
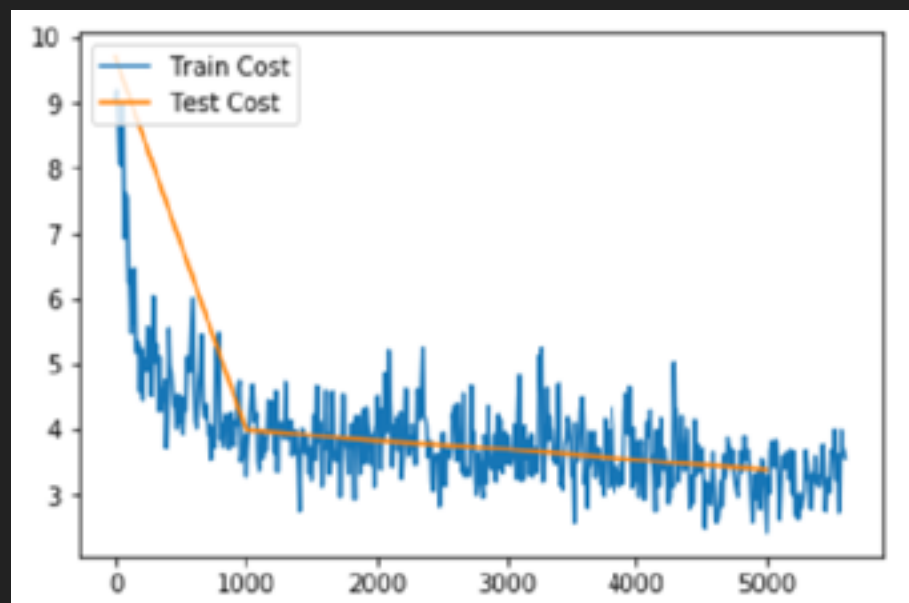
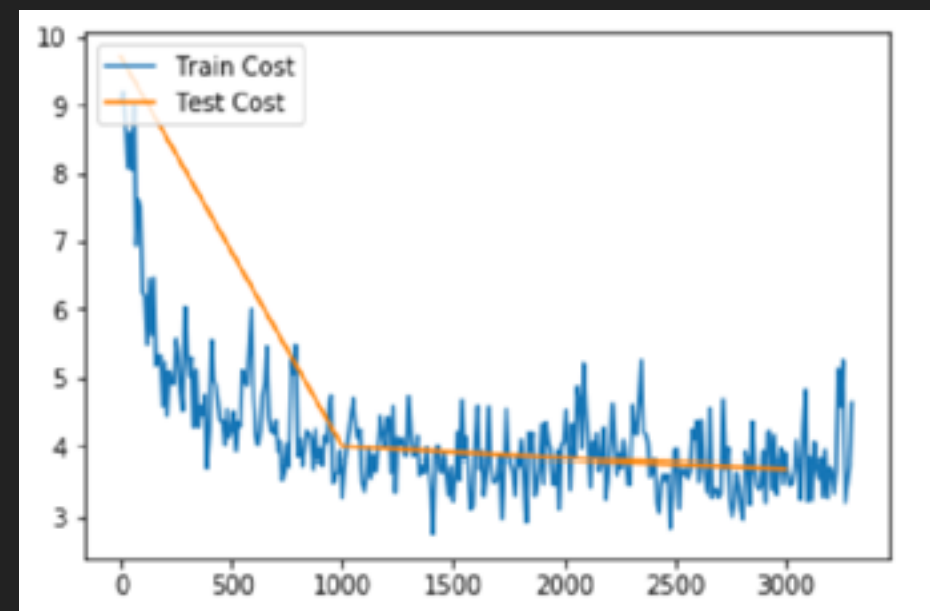
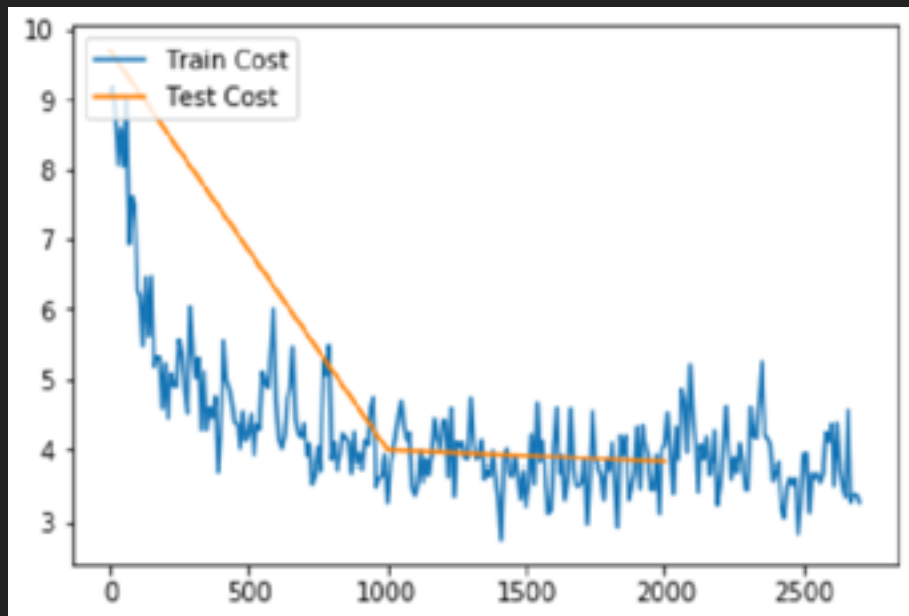
在网络技术不断发展和电子商务规模不断扩大的背景下，商品数量和种类快速增长，用户需要花费大量时间才能找到自己想买的商品，这就是信息超载问题。为了解决这个难题，推荐系统（Recommender System）应运而生。

个性化推荐系统是信息过滤系统（Information Filtering System）的子集，它可以用在很多领域，如电影、音乐、电商和 Feed 流推荐等。推荐系统通过分析、挖掘用户行为，发现用户的个性化需求与兴趣特点，将用户可能感兴趣的信息或商品推荐给用户。与搜索引擎不同，推荐系统不需要用户准确地描述出自己的需求，而是根据分析历史行为建模，主动提供满足用户兴趣和需求的信息。

传统的推荐系统方法主要有：

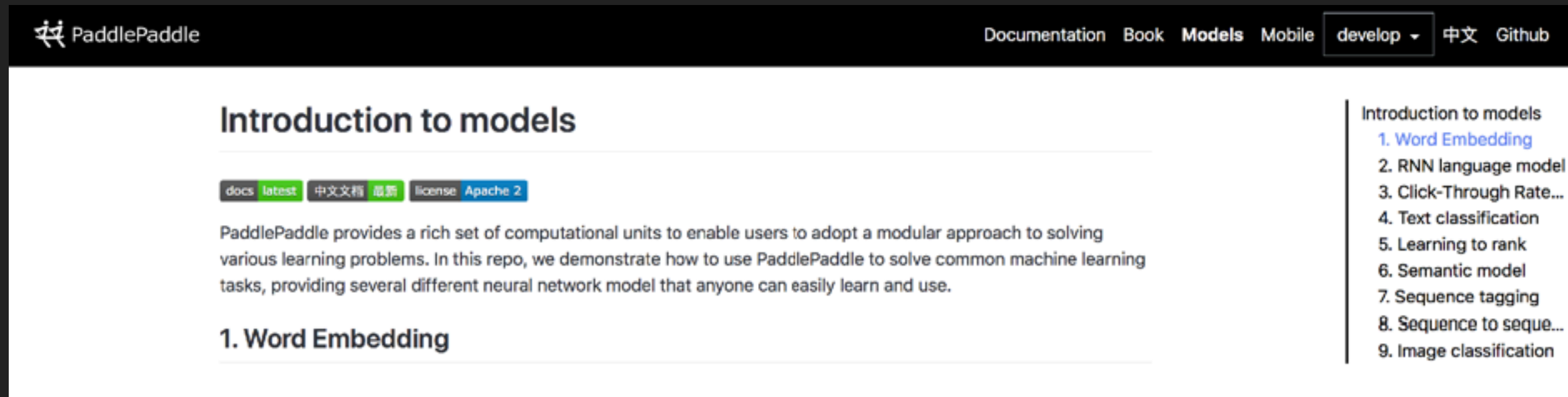
- 协同过滤推荐（Collaborative Filtering Recommendation）：该方法收集分析用户历史行为、活动、偏好，计算一个用户与其他用户的相似度，利用目标用户的相似用户对商品评价的加权评价价值，来预测目标用户对特定商品的喜好程度。优点是可以给用户推荐未浏览过的新产品；缺点是对于没有任何行为的新用户存在冷启动的问题，同时也存在用户与商品之间的交互数据不够多造成的稀疏问题，会导致模型难以找到相近用户。
- 基于内容过滤推荐^[1]（Content-based Filtering Recommendation）：该方法利用商品的内容描述，抽象出有意义的特征，通过计算用户的兴趣和商品描述之间的相似度，来给用户做推荐。优点是简单直接，不需要依据其他用户对商品的评价，而是通过商品属性进行商品相似度度量，从而推荐给用户所感兴趣商品的相似商品；缺点是对于没有任何行为的新用户同样存在冷启动的问题。
- 组合推荐^[2]（Hybrid Recommendation）：运用不同的输入和技术共同进行推荐，以弥补各自推荐技术的缺点。

DOCKER BASED JUPYTER INTERACTIVE NOTEBOOK



PRODUCTION LEVEL MODEL BANK

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



PaddlePaddle

Documentation Book Models Mobile develop 中文 Github

Introduction to models

docs latest 中文文档 最新 license: Apache 2

PaddlePaddle provides a rich set of computational units to enable users to adopt a modular approach to solving various learning problems. In this repo, we demonstrate how to use PaddlePaddle to solve common machine learning tasks, providing several different neural network model that anyone can easily learn and use.

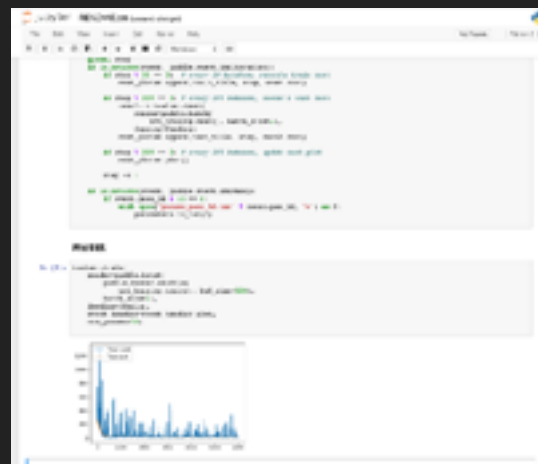
1. Word Embedding

- Introduction to models
- 1. Word Embedding
- 2. RNN language model
- 3. Click-Through Rate...
- 4. Text classification
- 5. Learning to rank
- 6. Semantic model
- 7. Sequence tagging
- 8. Sequence to seque...
- 9. Image classification

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



Name	Namespace	Status	Age
paddle-node-00	paddle	Running	0
paddle-node-01	paddle	Running	0
paddle-node-02	paddle	Running	0
paddle-node-03	paddle	Running	0
paddle-node-04	paddle	Running	0
paddle-node-05	paddle	Running	0
paddle-node-06	paddle	Running	0
paddle-node-07	paddle	Running	0
paddle-node-08	paddle	Running	0
paddle-node-09	paddle	Running	0
paddle-node-10	paddle	Running	0
paddle-node-11	paddle	Running	0
paddle-node-12	paddle	Running	0
paddle-node-13	paddle	Running	0
paddle-node-14	paddle	Running	0
paddle-node-15	paddle	Running	0
paddle-node-16	paddle	Running	0
paddle-node-17	paddle	Running	0
paddle-node-18	paddle	Running	0
paddle-node-19	paddle	Running	0
paddle-node-20	paddle	Running	0
paddle-node-21	paddle	Running	0
paddle-node-22	paddle	Running	0
paddle-node-23	paddle	Running	0
paddle-node-24	paddle	Running	0
paddle-node-25	paddle	Running	0
paddle-node-26	paddle	Running	0
paddle-node-27	paddle	Running	0
paddle-node-28	paddle	Running	0
paddle-node-29	paddle	Running	0
paddle-node-30	paddle	Running	0
paddle-node-31	paddle	Running	0
paddle-node-32	paddle	Running	0
paddle-node-33	paddle	Running	0
paddle-node-34	paddle	Running	0
paddle-node-35	paddle	Running	0
paddle-node-36	paddle	Running	0
paddle-node-37	paddle	Running	0
paddle-node-38	paddle	Running	0
paddle-node-39	paddle	Running	0
paddle-node-40	paddle	Running	0
paddle-node-41	paddle	Running	0
paddle-node-42	paddle	Running	0
paddle-node-43	paddle	Running	0
paddle-node-44	paddle	Running	0
paddle-node-45	paddle	Running	0
paddle-node-46	paddle	Running	0
paddle-node-47	paddle	Running	0
paddle-node-48	paddle	Running	0
paddle-node-49	paddle	Running	0
paddle-node-50	paddle	Running	0
paddle-node-51	paddle	Running	0
paddle-node-52	paddle	Running	0
paddle-node-53	paddle	Running	0
paddle-node-54	paddle	Running	0
paddle-node-55	paddle	Running	0
paddle-node-56	paddle	Running	0
paddle-node-57	paddle	Running	0
paddle-node-58	paddle	Running	0
paddle-node-59	paddle	Running	0
paddle-node-60	paddle	Running	0
paddle-node-61	paddle	Running	0
paddle-node-62	paddle	Running	0
paddle-node-63	paddle	Running	0
paddle-node-64	paddle	Running	0
paddle-node-65	paddle	Running	0
paddle-node-66	paddle	Running	0
paddle-node-67	paddle	Running	0
paddle-node-68	paddle	Running	0
paddle-node-69	paddle	Running	0
paddle-node-70	paddle	Running	0
paddle-node-71	paddle	Running	0
paddle-node-72	paddle	Running	0
paddle-node-73	paddle	Running	0
paddle-node-74	paddle	Running	0
paddle-node-75	paddle	Running	0
paddle-node-76	paddle	Running	0
paddle-node-77	paddle	Running	0
paddle-node-78	paddle	Running	0
paddle-node-79	paddle	Running	0
paddle-node-80	paddle	Running	0
paddle-node-81	paddle	Running	0
paddle-node-82	paddle	Running	0
paddle-node-83	paddle	Running	0
paddle-node-84	paddle	Running	0
paddle-node-85	paddle	Running	0
paddle-node-86	paddle	Running	0
paddle-node-87	paddle	Running	0
paddle-node-88	paddle	Running	0
paddle-node-89	paddle	Running	0
paddle-node-90	paddle	Running	0
paddle-node-91	paddle	Running	0
paddle-node-92	paddle	Running	0
paddle-node-93	paddle	Running	0
paddle-node-94	paddle	Running	0
paddle-node-95	paddle	Running	0
paddle-node-96	paddle	Running	0
paddle-node-97	paddle	Running	0
paddle-node-98	paddle	Running	0
paddle-node-99	paddle	Running	0

OUTLINE

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



THANK YOU