

PHP

2017·北京
全球开发者大会

高可用的 PHP

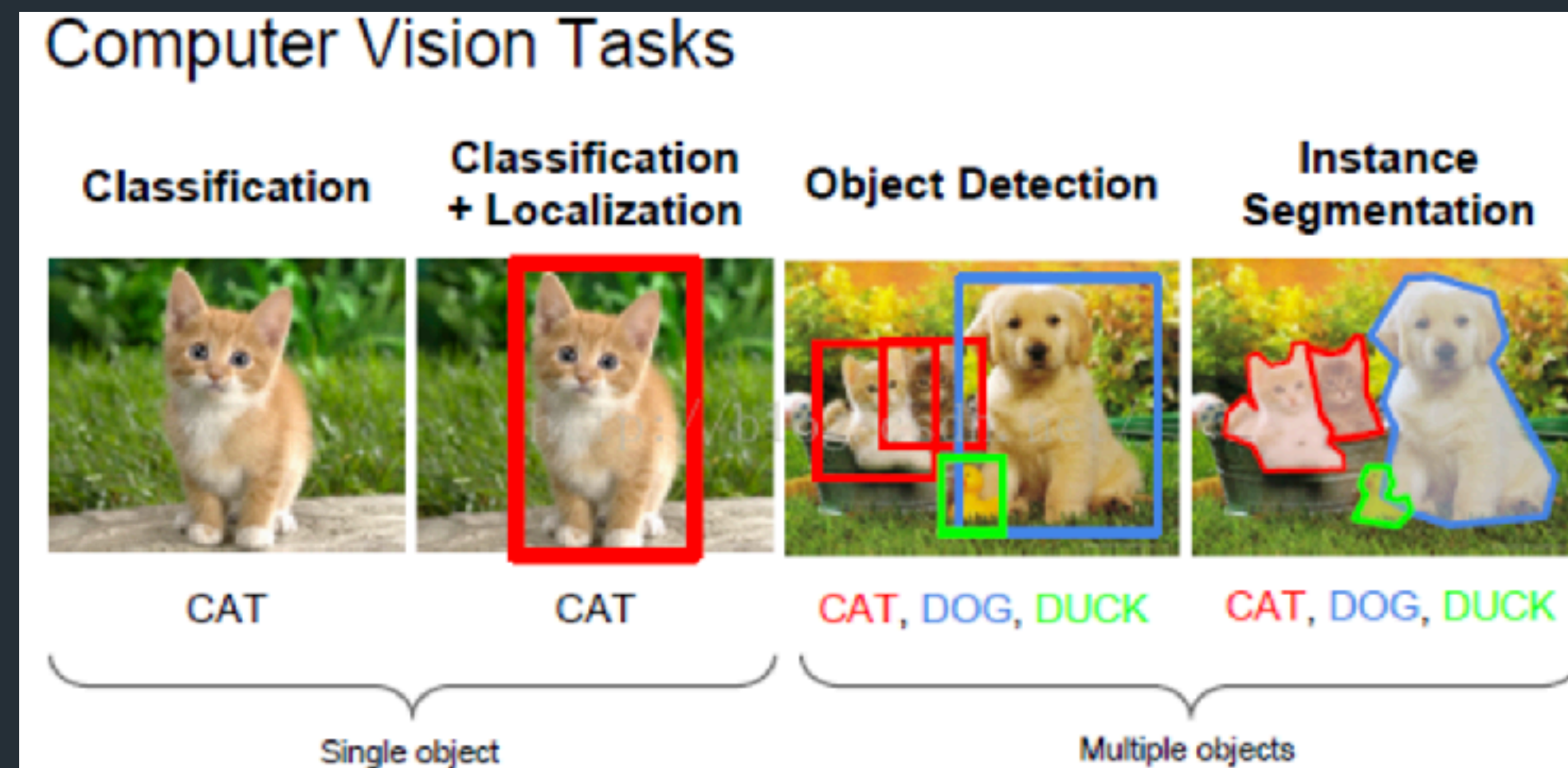
PHP在机器学习上的应用及云深度学习平台的架构设计与实现

《TensorFlow技术解析与实战》作者 TensorFlow技术社区创建者 李嘉璇

主题大纲

- 机器学习与深度学习的应用
- PHP对机器学习的支持及对神经网络的实现
- 云深度学习平台架构与设计
- 从Google Cloud Machine Learning借鉴的经验

ML/DL的应用



图像

图片分类

目标检测

图像分割

语音

语音识别

- 语音听写
- 语音转写
- 命令词识别

语音合成

声纹识别



文本

文本情感分析

文本分类

文本挖掘



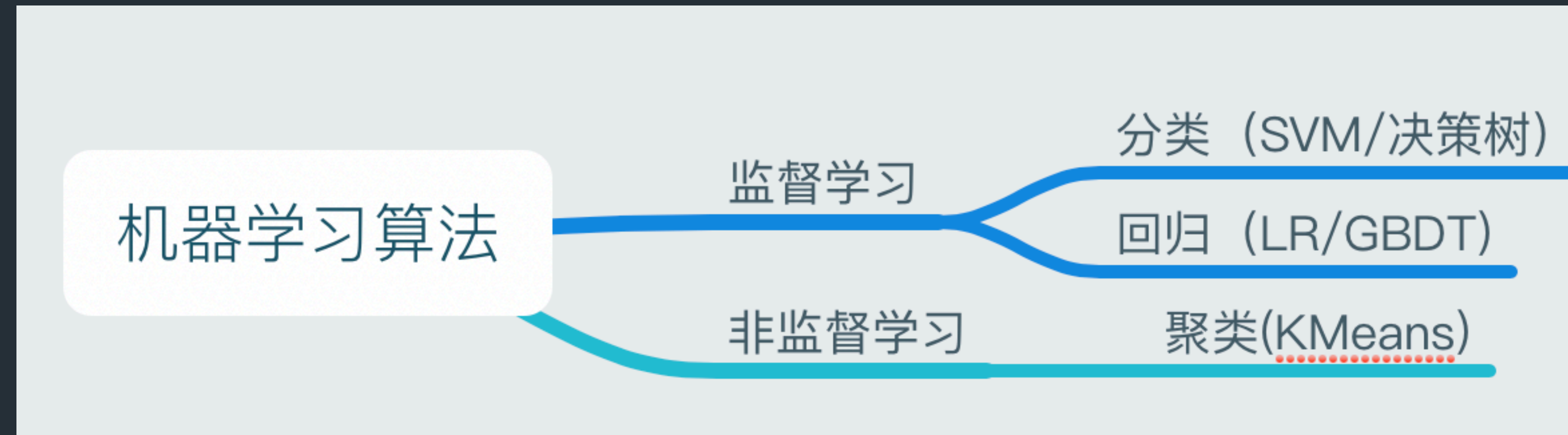


机器学习

机器学习

什么是机器学习？

- 从数据中抽取规律，用于提取有用信息、解释数据、预测未来



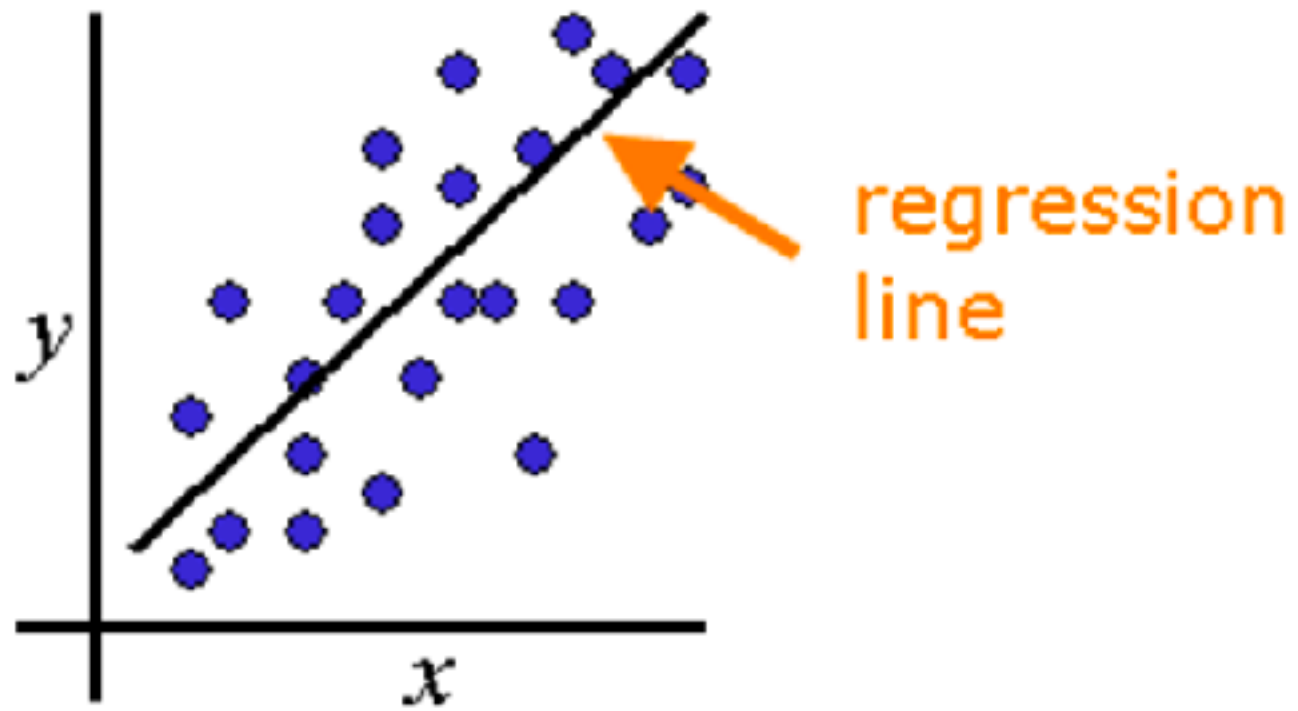
机器学习过程：

- 数据预处理（去噪/归一化）
- 训练模型
- 评估模型（MSE/F1Score/AUC）
- 应用模型

机器学习算法应用

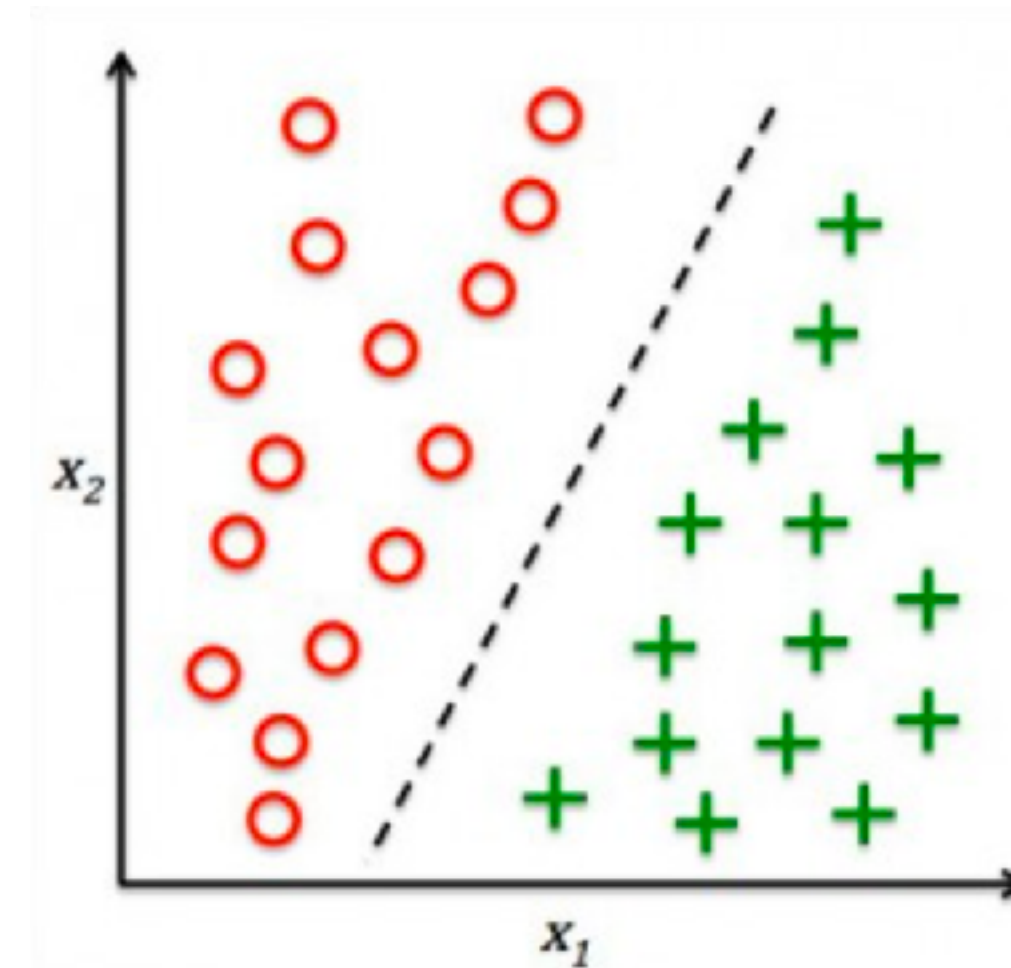
回归

房价预测/用户消费预测



分类

文本/图像/视频/垃圾邮件

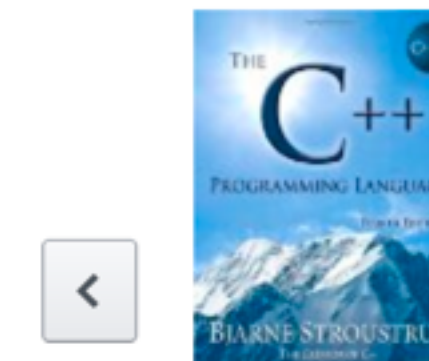


推荐

电商/视频推荐

Your recently viewed items and featured recommendations

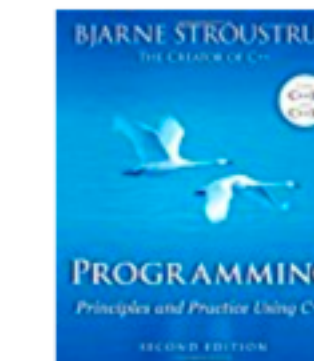
Inspired by your browsing history



The C++ Programming Language, 4th Edition
» Bjarne Stroustrup



C Programming Language, 2nd Edition
» Brian W. Kernighan



Programming: Principles and Practice Using C++...
» Bjarne Stroustrup

排序

文档相关性检索

A screenshot of a Google search results page for the query "php". The search bar at the top shows "php" and the Google logo. Below the search bar, there are tabs for "全部" (All), "图片" (Images), "新闻" (News), "图书" (Books), "视频" (Videos), and "更多" (More). The search results show approximately 9,300,000,000 results found in 0.42 seconds. The first result is "PHP: Hypertext Preprocessor" from php.net, with a link to "php.net" and a "翻译此页" (Translate this page) button. Below this, there is a snippet of text about PHP being a popular general-purpose scripting language. The second result is "PHP - 维基百科, 自由的百科全书" (PHP - Wikipedia, the free encyclopedia), with a link to the Wikipedia page and a "翻译此页" button. Below this, there is a snippet of text about PHP being a popular general-purpose scripting language. The third result is "Laravel - The PHP Framework For Web Artisans" from laravel.com, with a link to the website and a "翻译此页" button. Below this, there is a snippet of code and text about Laravel. The fourth result is "Newest 'php' Questions - Stack Overflow" from stackoverflow.com, with a link to the questions page and a "翻译此页" button. Below this, there is a snippet of text about a PHP error.

神经网络

结构

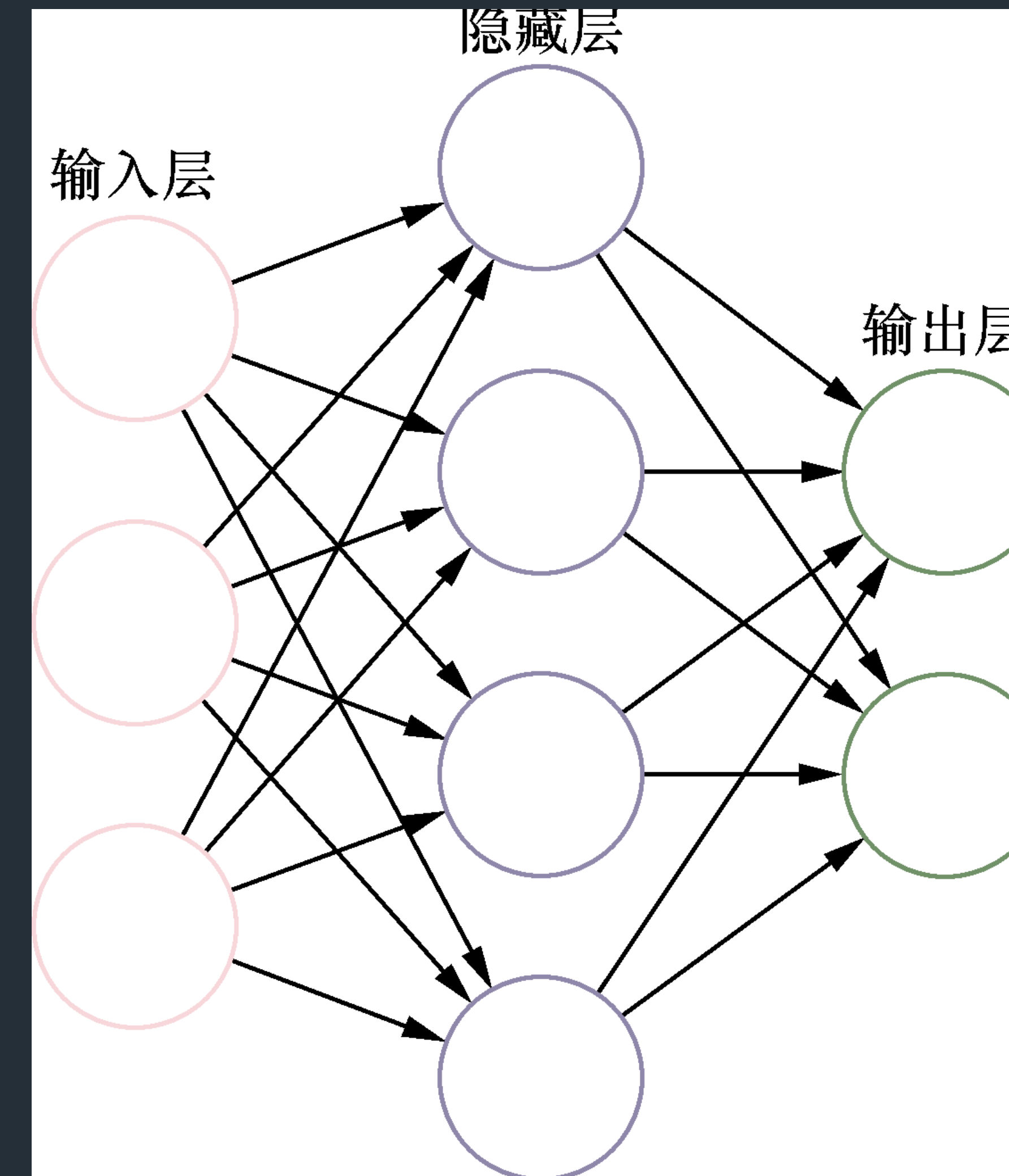
- 输入层：节点数固定
- 隐藏层：节点数可以指定
- 输出层：节点数固定

箭头：预测过程中数据流向

连接线：对应不同权重，需要训练

过程

- 前向传播
- 反向传播



机器学习框架对神经网络的实现——TensorFlow



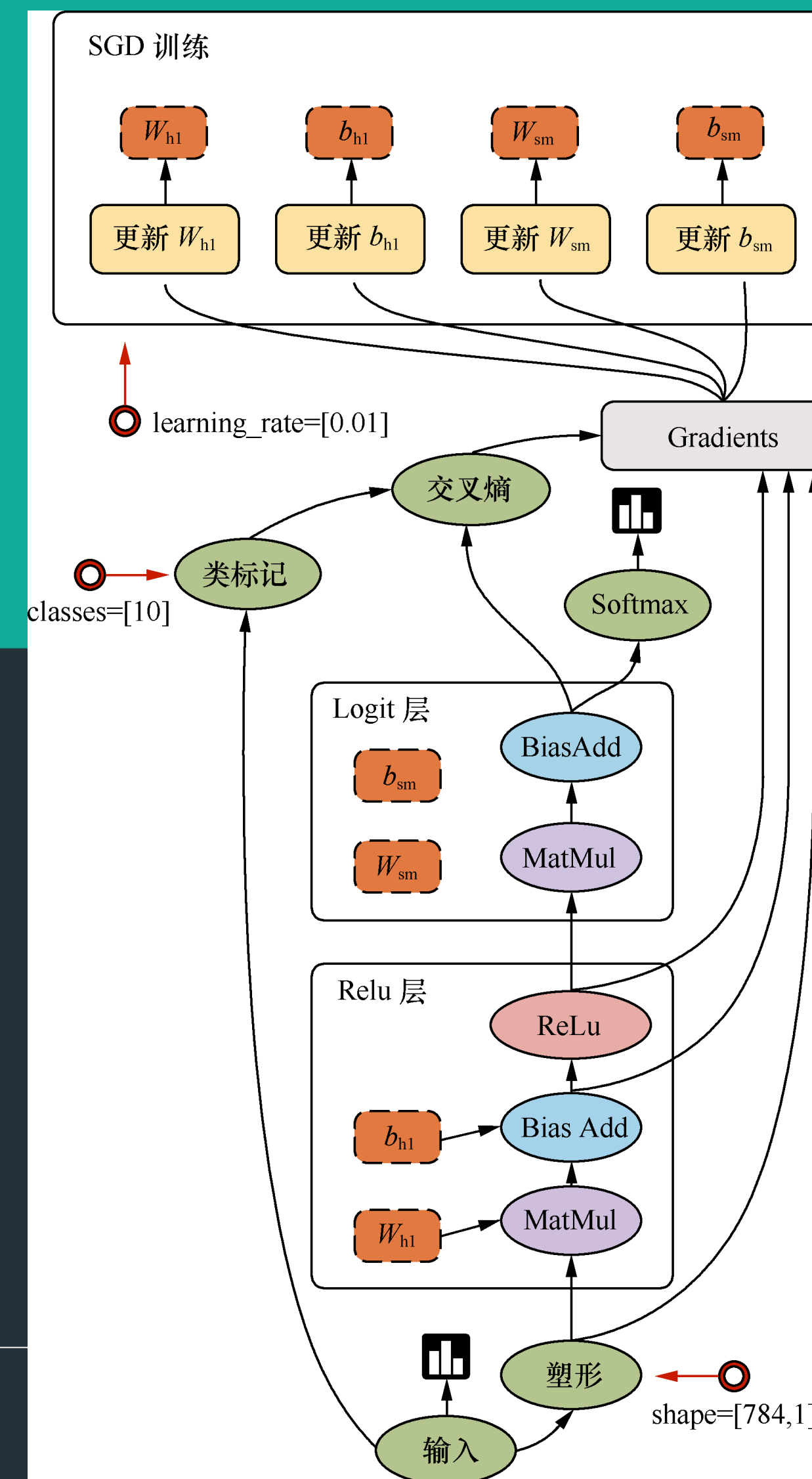
TensorFlow

- 机器学习/深度学习框架
- 异构设备分布式计算
- C++/Python/Java/Go编程接口

用神经网络实现训练的过程：

- 加载数据
- 定义超参数及网络模型
- 训练模型
- 评估模型
- 预测模型

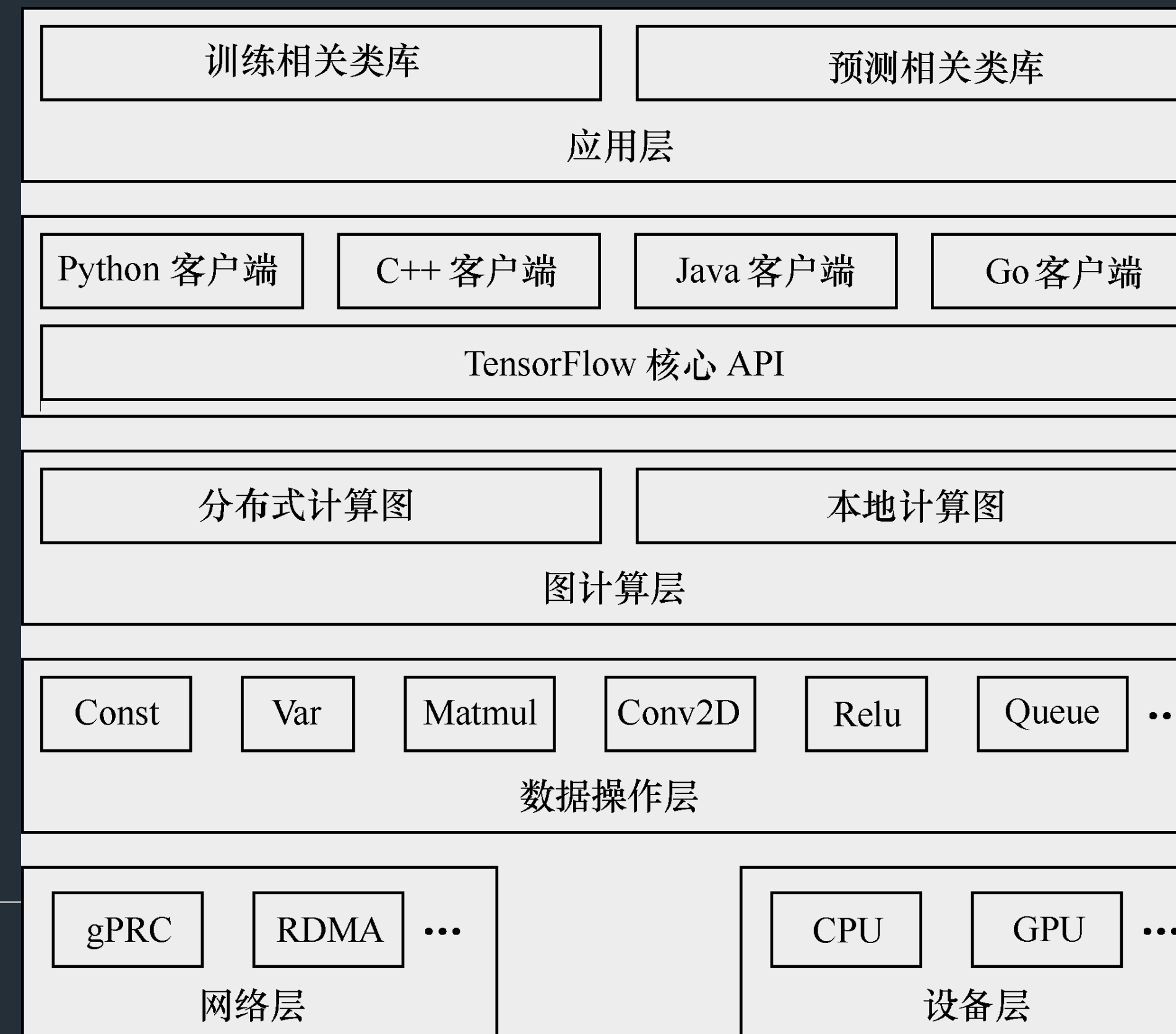
TensorFlow编程模型——数据流图



TensorFlow的架构

TensorFlow的系统结构以C API为界，将整个系统分为「前端」和「后端」两个子系统：

- 前端系统：提供编程模型，负责构造计算图；
- 后端系统：提供运行时环境，负责执行计算图。





PHP对机器学习的 实现——PHP-ML

PHP-ML



PHP的机器学习库：

- 基于PHP7
- 基于扩展形式安装
- 包含算法，交叉验证，神经网络，预处理，特征提取

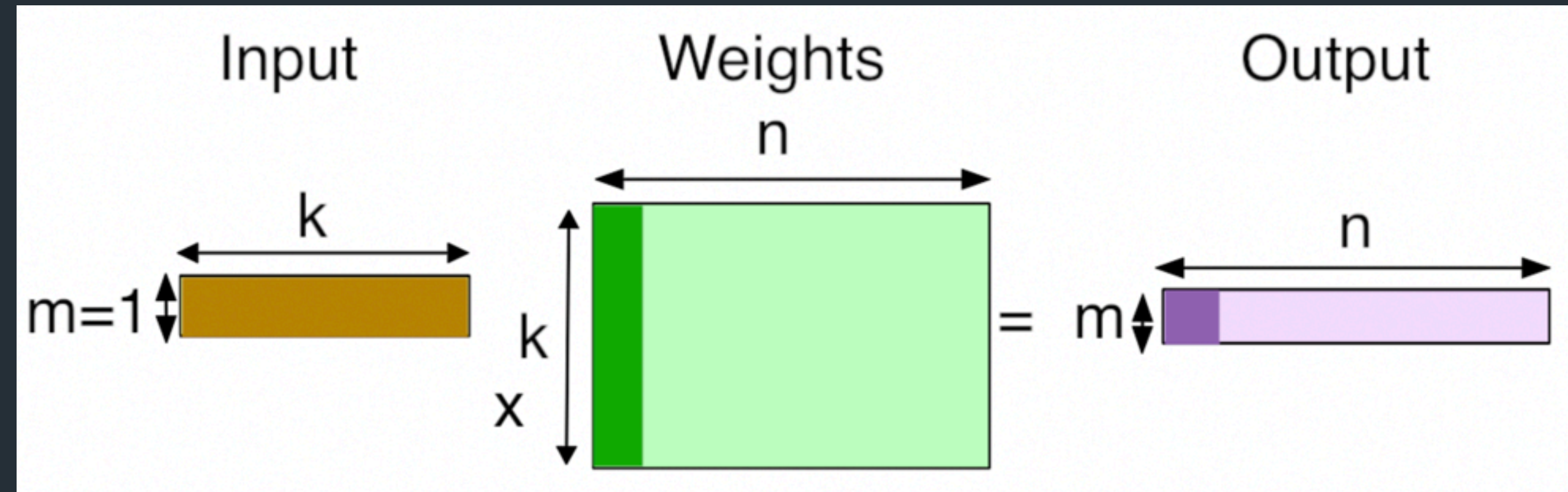
8种数据类型：

- 4种标量类型（Bool、Integer、Float、String）
- 2种复合类型（Array、Object）
- 2种特殊类型（Resource、NULL）

新增2种数据类型：

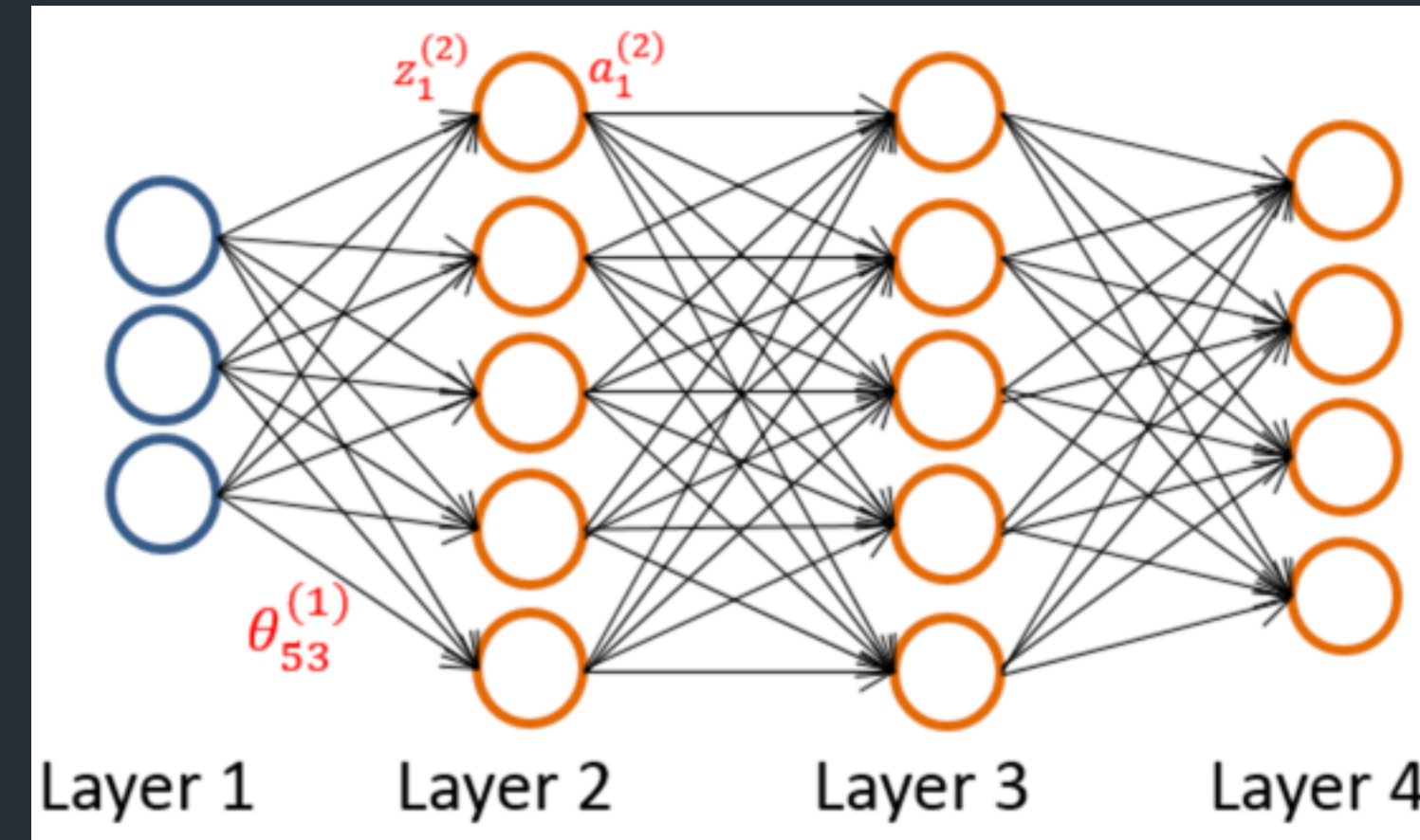
- Matrix类型（Array转换成 Matrix）
- Set类型(交、并、差集运算)

PHP做ML/DL的难点



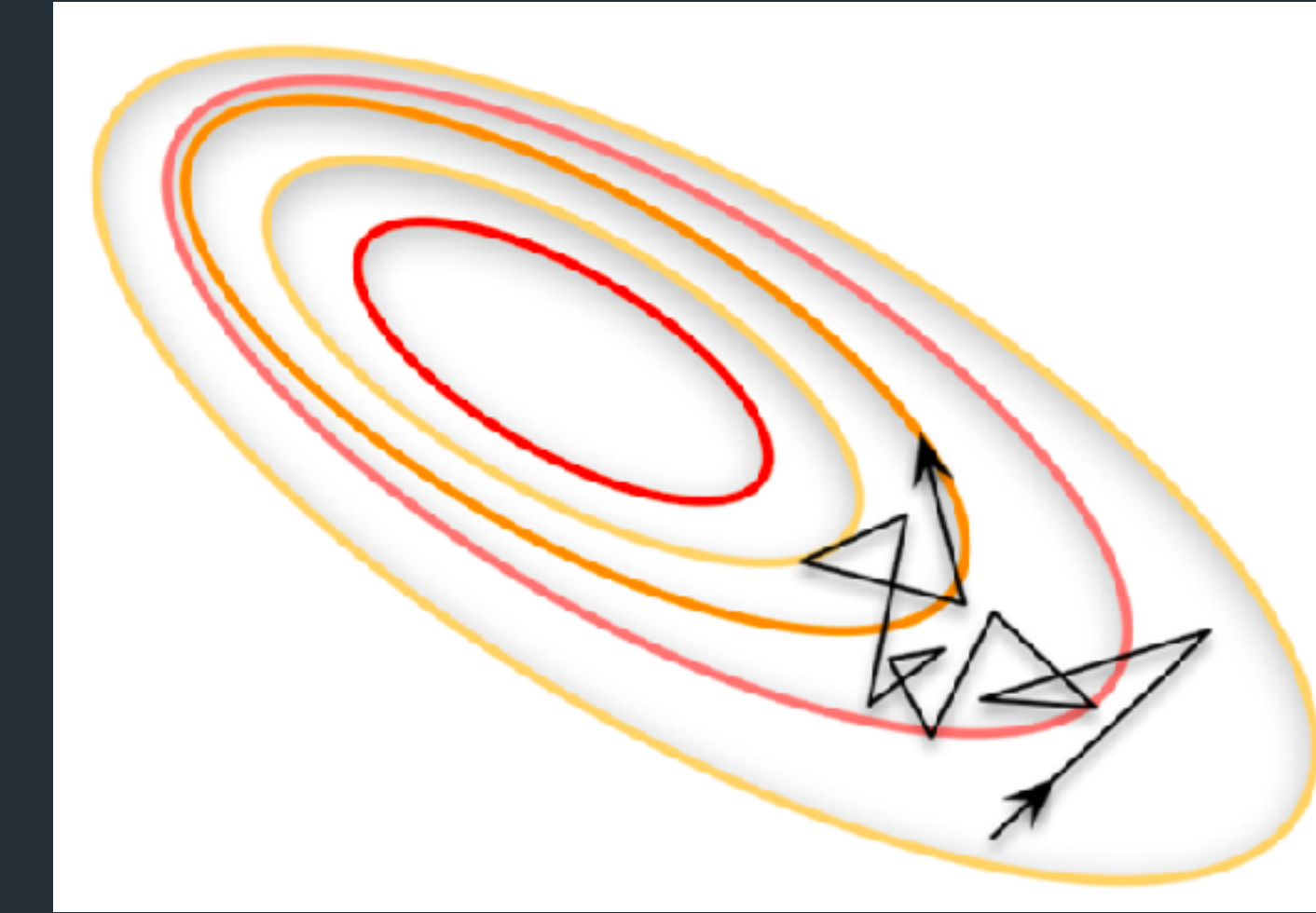
进行矩阵运算？

Array转为Matrix来实现。



构建网络模型？

网络的每一层神经元视为层数组；
相邻层之间的神经元连接视为突触数组；
神经元之间连接的值构成权重。



分批次 (batch) 进行训练？

全数据集 (Full Batch Learning) 和批
梯度下降法 (Mini-batches Learning)
训练方法分开

PHP-ML包含的算法

1 分类

SVC
KNN
贝叶斯

2 回归

最小二乘线性回归
支持向量回归

3 聚类

KMeans
基于密度的聚类
关联规则学习

4 神经网络

多层感知机 (MLP)

PHP对神经网络的实现

```
use Phpml\Classification\MLPClassifier;
$mlp = new MLPClassifier(4, [2], ['a', 'b', 'c']);

// 4 nodes in input layer, 2 nodes in first hidden layer and 3 possible labels.
```

Train

To train a MLP simply provide train samples and labels (as array). Example:

```
$mlp->train(
    $samples = [[1, 0, 0, 0], [0, 1, 1, 0], [1, 1, 1, 1], [0, 0, 0, 0]],
    $targets = ['a', 'a', 'b', 'c']
);
```

Use partialTrain method to train in batches. Example:

```
$mlp->partialTrain(
    $samples = [[1, 0, 0, 0], [0, 1, 1, 0]],
    $targets = ['a', 'a']
);
```

```
public function __construct(int $inputLayerFeatures, array $hiddenLayers, array $classes, int $iterations = 10000, ActivationFunction $activationFunction = null, int $theta = 1)
{
    if (empty($hiddenLayers)) {
        throw InvalidArgumentException::invalidLayersNumber();
    }

    if (count($classes) < 2) {
        throw InvalidArgumentException::invalidClassesNumber();
    }

    $this->classes = array_values($classes);
    $this->iterations = $iterations;
    $this->inputLayerFeatures = $inputLayerFeatures;
    $this->hiddenLayers = $hiddenLayers;
    $this->activationFunction = $activationFunction;
    $this->theta = $theta;

    $this->initNetwork();
}

/**
 * @return void
 */
private function initNetwork()
{
    $this->addInputLayer($this->inputLayerFeatures);
    $this->addNeuronLayers($this->hiddenLayers, $this->activationFunction);
    $this->addNeuronLayers([count($this->classes)], $this->activationFunction);

    $this->addBiasNodes();
    $this->generateSynapses();

    $this->backpropagation = new Backpropagation($this->theta);
}
```

对外接口



定义超参数，初始化网络FP、BP过程

2017 PHP 全球开发者大会



PHP对神经网络的实现

```
/**
 * @param int $nodes
 */
private function addInputLayer(int $nodes)
{
    $this->addLayer(new Layer($nodes, Input::class));
}
```

输入层



```
/**
 * @param array $layers
 * @param ActivationFunction|null $activationFunction
 */
private function addNeuronLayers(array $layers, ActivationFunction $activationFunction = null)
{
    foreach ($layers as $neurons) {
        $this->addLayer(new Layer($neurons, Neuron::class, $activationFunction));
    }
}
```

隐藏层



```
private function addBiasNodes()
{
    $biasLayers = count($this->layers) - 1;
    for ($i = 0; $i < $biasLayers; ++$i) {
        $this->layers[$i]->addNode(new Bias());
    }
}
```

加入偏置项

PHP对神经网络的实现

```
private function generateSynapses()
{
    $layersNumber = count($this->layers) - 1;
    for ($i = 0; $i < $layersNumber; ++$i) {
        $currentLayer = $this->layers[$i];
        $nextLayer = $this->layers[$i + 1];
        $this->generateLayerSynapses($nextLayer, $currentLayer);
    }
}
```



```
/**
 * @param Layer $nextLayer
 * @param Layer $currentLayer
 */
private function generateLayerSynapses(Layer $nextLayer, Layer $currentLayer)
{
    foreach ($nextLayer->getNodes() as $nextNeuron) {
        if ($nextNeuron instanceof Neuron) {
            $this->generateNeuronSynapses($currentLayer, $nextNeuron);
        }
    }
}

/**
 * @param Layer $currentLayer
 * @param Neuron $nextNeuron
 */
private function generateNeuronSynapses(Layer $currentLayer, Neuron $nextNeuron)
{
    foreach ($currentLayer->getNodes() as $currentNeuron) {
        $nextNeuron->addSynapse(new Synapse($currentNeuron));
    }
}
```



```
/**
 * @param Node $node
 * @param float|null $weight
 */
public function __construct(Node $node, float $weight = null)
{
    $this->node = $node;
    $this->weight = $weight ?: $this->generateRandomWeight();
}
```

产生神经突触，构造网络层之间的连接

相邻两层神经元之间加入连接

连接即为训练的权重参数

PHP对神经网络的实现

```
public function train(array $samples, array $targets)
{
    $this->reset();
    $this->initNetwork();
    $this->partialTrain($samples, $targets, $this->classes);
}

/**
 * @param array $samples
 * @param array $targets
 */
public function partialTrain(array $samples, array $targets, array $classes = [])
{
    if (!empty($classes) && array_values($classes) !== $this->classes) {
        // We require the list of classes in the constructor.
        throw InvalidArgumentException::inconsistentClasses();
    }

    for ($i = 0; $i < $this->iterations; ++$i) {
        $this->trainSamples($samples, $targets);
    }
}
```

进行训练



```
/**
 * @param array $sample
 * @param mixed $target
 */
protected function trainSample(array $sample, $target)
{
    // Feed-forward.
    $this->setInput($sample)->getOutput();

    // Back-propagate.
    $this->backpropagation->backpropagate($this->getLayers(), $this->getTargetClass($target));
}
```

批次训练, FP及BP过程



```
/**
 * @param array $layers
 * @param mixed $targetClass
 */
public function backpropagate(array $layers, $targetClass)
{
    $layersNumber = count($layers);

    // Backpropagation.
    for ($i = $layersNumber; $i > 1; --$i) {
        $this->sigmas = [];
        foreach ($layers[$i - 1]->getNodes() as $key => $neuron) {
            if ($neuron instanceof Neuron) {
                $sigma = $this->getSigma($neuron, $targetClass, $key, $i == $layersNumber);
                foreach ($neuron->getSynapses() as $synapse) {
                    $synapse->changeWeight($this->theta * $sigma * $synapse->getNode()->getOutput());
                }
            }
        }
        $this->prevSigmas = $this->sigmas;
    }

    // Clean some memory (also it helps make MLP persistency & children more maintainable).
    $this->sigmas = null;
    $this->prevSigmas = null;
}
```

BP过程更新权重

PHP-ML的缺点

- 支持多个全连接层，但还未实现卷积
- 优化方法较为单一：BGD（Batch Gradient Descent）
 - 考虑支持SGD、Mini-batch Gradient Descent、Momentum
- 未支持RNN相关模型

PHP实现卷积的例子：

imageconvolution

(PHP 5 >= 5.1.0, PHP 7)

imageconvolution — Apply a 3x3 convolution matrix, using coefficient and offset

Description

```
bool imageconvolution ( resource $image , array $matrix , float $div , float $offset )
```


落地应用场景

```
languages.csv
"sentence","language"
"Hello, do you know what time the movie is tonight?","english"
"I am calling to make reservations","english"
"I would like to know if it is at all possible to check in","english"
"What time does the swimming pool open?","english"
"Where is the games room?","english"
"We must call the police.","english"
"How many pupils are there in your school?","english"
"Twenty litres of unleaded, please.","english"
"Is it near here?","english"
"Do I have to change?","english"
"Which counter do I go to to change money?","english"
"I would like two postcards, please.","english"
"There was a big explosion.","english"
"Where is the television room?","english"
"Where's the nearest railway station?","english"
"The storms caused flooding.","english"
"Where is the duty free shop?","english"
"I witnessed it happening.","english"
"I would like two postcards, please.","english"
"How about going to the cinema?","english"
"Où est la boulangerie?","french"
"Je voudrais une boîte de chocolates.","french"
"Y a-t-il un autre hôtel près d'ici?","french"
"Vérifiez la batterie, s'il vous plaît.","french"
"La banque ouvre à quelle heure?","french"

(tf) → php-ml-examples git:(master) ✖ php classification/languageDetection.php
Accuracy: 0.8333333333333333
```

SVC模型进行句子语种分类

```
wine.csv
alcohol,malic acid,ash,alcalinity of ash,magnesium,total phenols,flavanoid
wines,proline,class
14.23,1.71,2.43,15.6,127,2.8,3.06,.28,2.29,5.64,1.04,3.92,1065,1
13.2,1.78,2.14,11.2,100,2.65,2.76,.26,1.28,4.38,1.05,3.4,1050,1
13.16,2.36,2.67,18.6,101,2.8,3.24,.3,2.81,5.68,1.03,3.17,1185,1
14.37,1.95,2.5,16.8,113,3.85,3.49,.24,2.18,7.8,.86,3.45,1480,1
13.24,2.59,2.87,21,118,2.8,2.69,.39,1.82,4.32,1.04,2.93,735,1
14.2,1.76,2.45,15.2,112,3.27,3.39,.34,1.97,6.75,1.05,2.85,1450,1
14.39,1.87,2.45,14.6,96,2.5,2.52,.3,1.98,5.25,1.02,3.58,1290,1
14.06,2.15,2.61,17.6,121,2.6,2.51,.31,1.25,5.05,1.06,3.58,1295,1
14.83,1.64,2.17,14,97,2.8,2.98,.29,1.98,5.2,1.08,2.85,1045,1
13.86,1.35,2.27,16,98,2.98,3.15,.22,1.85,7.22,1.01,3.55,1045,1
14.1,2.16,2.3,18,105,2.95,3.32,.22,2.38,5.75,1.25,3.17,1510,1
14.12,1.48,2.32,16.8,95,2.2,2.43,.26,1.57,5,1.17,2.82,1280,1
13.75,1.73,2.41,16,89,2.6,2.76,.29,1.81,5.6,1.15,2.9,1320,1
14.75,1.73,2.39,11.4,91,3.1,3.69,.43,2.81,5.4,1.25,2.73,1150,1
14.38,1.87,2.38,12,102,3.3,3.64,.29,2.96,7.5,1.2,3,1547,1
13.63,1.81,2.7,17.2,112,2.85,2.91,.3,1.46,7.3,1.28,2.88,1310,1
14.3,1.92,2.72,20,120,2.8,3.14,.33,1.97,6.2,1.07,2.65,1280,1
13.83,1.57,2.62,20,115,2.95,3.4,.4,1.72,6.6,1.13,2.57,1130,1
14.19,1.59,2.48,16.5,108,3.3,3.93,.32,1.86,8.7,1.23,2.82,1680,1
13.64,3.1,2.56,15.2,116,2.7,3.03,.17,1.66,5.1,.96,3.36,845,1
14.06,1.63,2.28,16,126,3,3.17,.24,2.1,5.65,1.09,3.71,780,1
12.93,3.8,2.65,18.6,102,2.41,2.41,.25,1.98,4.5,1.03,3.52,770,1
13.71,1.86,2.36,16.6,101,2.61,2.88,.27,1.69,3.8,1.11,4,1035,1
12.85,1.6,2.52,17.8,95,2.48,2.37,.26,1.46,3.93,1.09,3.63,1015,1
13.5,1.81,2.61,20,96,2.53,2.61,.28,1.66,3.52,1.12,3.82,845,1
13.05,2.05,3.22,25,124,2.63,2.68,.47,1.92,3.58,1.13,3.2,830,1

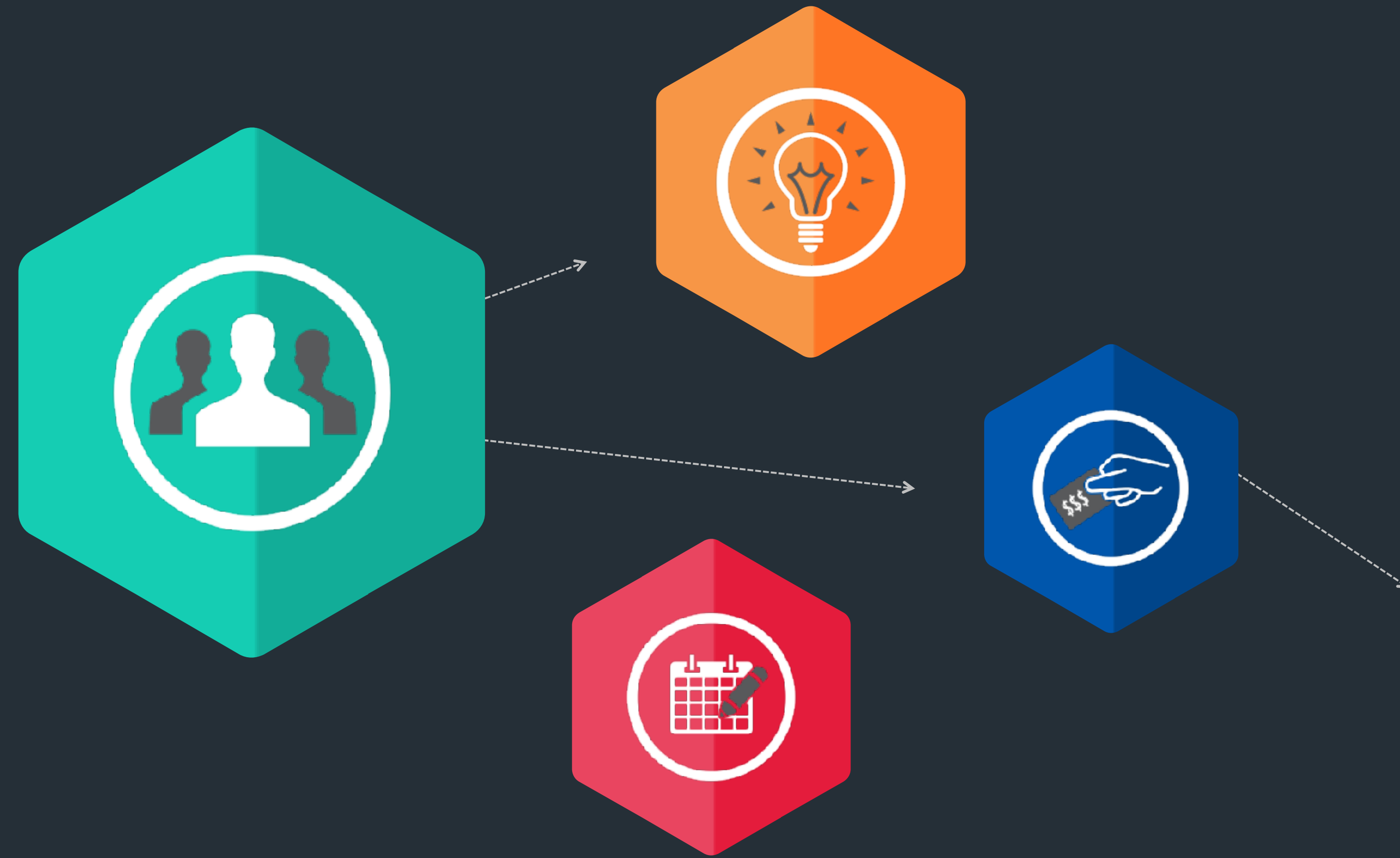
(tf) → php-ml-examples git:(master) ✖ php regression/wineQuality.php
Accuracy: 0.7272727272727273
```

SVR模型对酒的特征进行回归

云深度学习平台的 架构设计及实现



为什么需要云深度学习平台



- 大规模训练

- 数据集大 (GB-PB)
- 模型参数多 (几GB-几百GB)
- 训练周期长 (H-D-W)

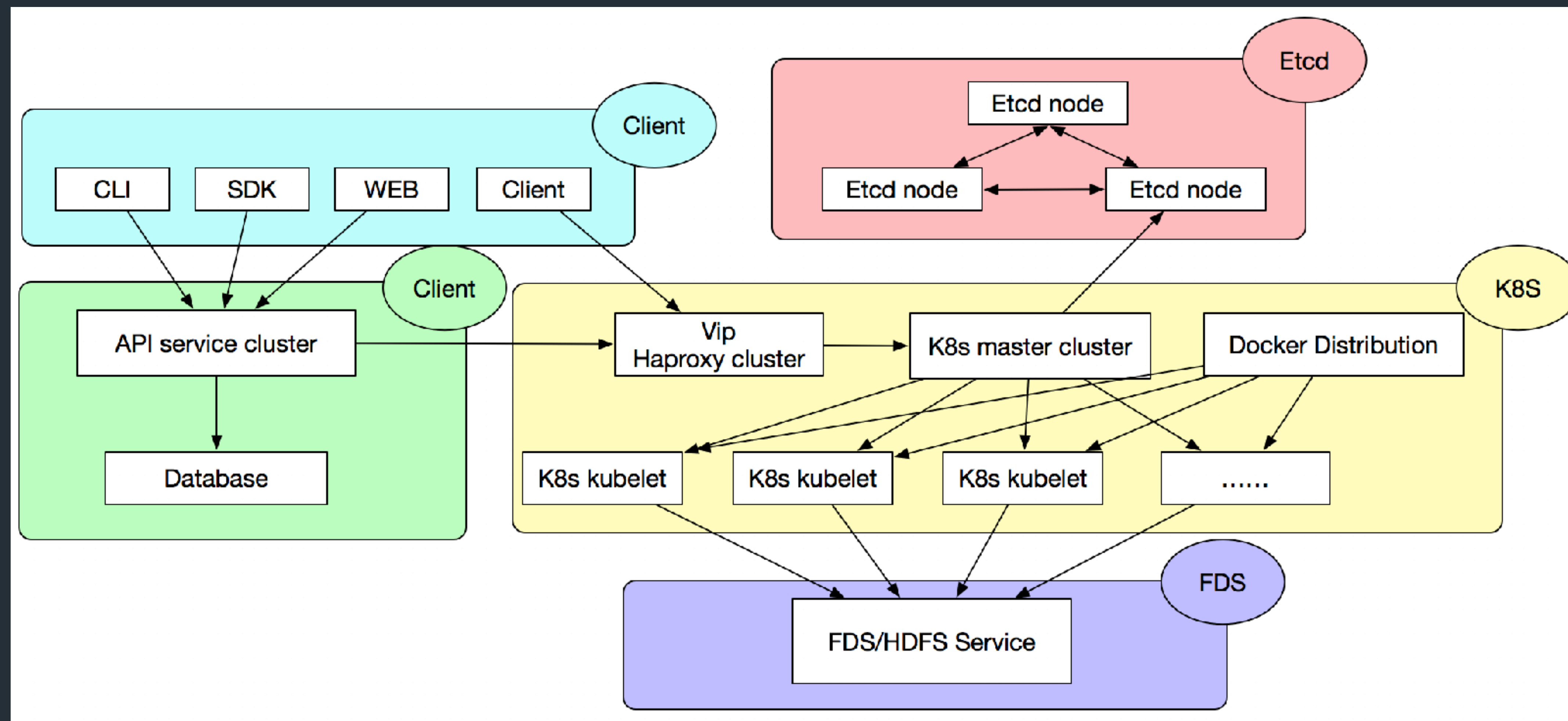
- 大规模分布式集群训练的递进式解决方案

- Distributed TensorFlow
- TensorFlow & Kubernetes & Prometheus
- TensorFlowOnSpark

云深度学习平台的设计宗旨

- 云端训练替代本机训练
- 提交训练任务替代ssh+脚本运行
- 集成模型开发、训练、调优、部署和预测一站式解决方案
- 训练任务的高可用性，失败重启
- 资源隔离及动态调度
- 训练和超参数自动调优同时进行

云深度学习平台的架构设计



调参经验



参数初始化

uniform均匀分布初始化

normal高斯分布初始化

svd初始化：对RNN有比较好的效果。



数据预处理方式

```
X -= np.mean(X, axis = 0) # zero-center
```

```
X /= np.std(X, axis = 0) # normalize
```



训练技巧

- 1.刚开始, 先上小规模数据, filter调大, 先奔着过拟合去。
- 2.Loss设计要合理: 分类就是Softmax, 回归就是L2的loss。
- 3.观察loss胜于观察准确率



确认分类网络学习充分

看Softmax输出的概率的分布。二分类, 刚开始的网络预测都是在0.5上下, 很模糊。随着学习过程, 网络预测会慢慢的移动到0,1这种极值附近



可视化

中间结果可视化 (水波纹或者噪点)

权重的可视化



Ensemble

同样的参数,不同的初始化方式

不同的参数,通过cross-validation,选取最好的几组

同样的参数,模型训练的不同阶段, 即不同迭代次数的模型。

不同的模型,进行线性融合. 例如RNN和传统模型。

其他经验参考: 《Neural Networks: Tricks of the Trade》

从Google Cloud Machine Learning Engine借鉴的经验

完成2件事情：

- 通过在云中运行TensorFlow训练应用程序，可以大规模训练机器学习模型。
- 在云中托管这些训练好的模型，以便可以使用它们来做新数据的预测。

满足的特性：

- 易用性
 - 云端训练模型
 - 一键部署基于GPU的开发环境，秒级启动分布式训练任务
 - 一键部署训练好的模型
 - 开放了API、SDK、命令行和Web控制台等多种访问方式
- 高性能
 - 高性能GPU计算，支持数据并行和模型并行、单机多卡和多机多卡的分布式训练。
- 安全性
 - 多租户认证授权机制（例如Access key/Secret key），在线动态调整用户Quota配额
- 完整性
 - 提供模型开发、训练、调优、部署和预测一站式解决方案。



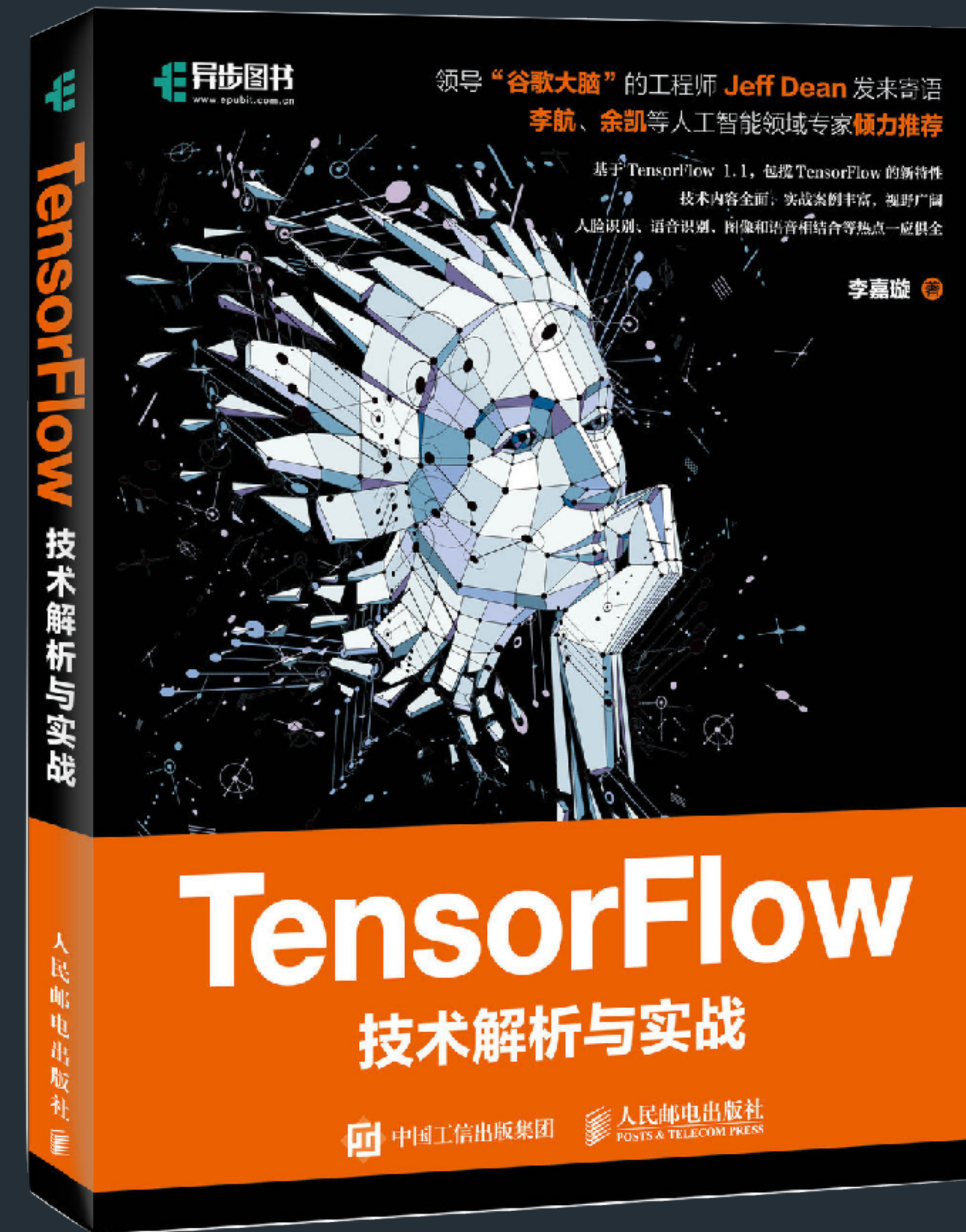
WeChat: ljxtensorflow

Email: qiyueli_2013@gmail.com

微博: 李嘉璇的微博



个人微信



签售

PHP 2017·北京

全球开发者大会

