



中国科学院大学

University of Chinese Academy of Sciences

Deep Learning

Development platform

Xinfeng Zhang (张新峰)

School of Computer Science and Technology

University of Chinese Academy of Sciences

Email: xfzhang@ucas.ac.cn



计算机科学与技术学院

SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY



提纲

- 深度学习平台概览
- Tensorflow
- PyTorch
- Keras
- PaddlePaddle
- Huawei平台



1

深度学习平台概览

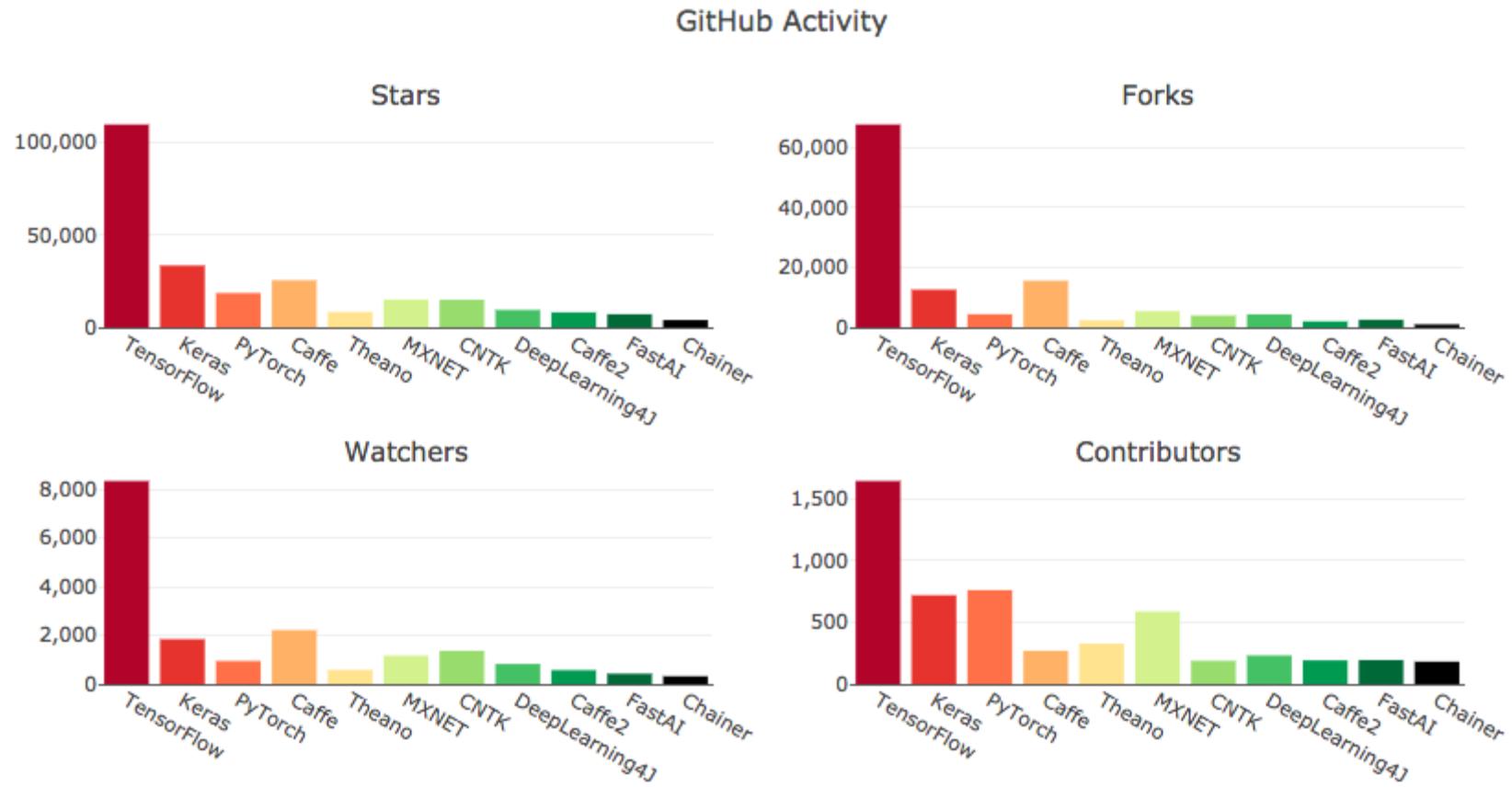
深度学习框架对照

工具	发布者	支持语言	支持系统
Theano	蒙特利尔大学	Python	Linux/Mac OS/Windows
Caffe	UC Berkeley	Python/C++/ Matlab	Linux/Mac OS/Windows
TensorFlow	Google	Python/C++/ Java/Go	Linux/Mac OS/Android/iOS/Windows
Torch	Facebook	C/C++/Lua/	Linux/Mac OS/ Windows/Android/iOS
Pytorch	Facebook	C/C++/Lua/ Python	Linux/Mac OS/ Windows/Android/iOS
MXNet	DMLC (分布式机器学习 社区)	Python/C++/Matlab/ Julia/Go/R/Scala	Linux/Mac OS/ Windows/Android/iOS
CNTK	Microsoft	Python/C++/C#/ BrainScript	Linux/Windows
Neon	Intel	Python	Linux
Deep Learning Toolbox	MATLAB	MATLAB	Linux/Mac OS/Windows



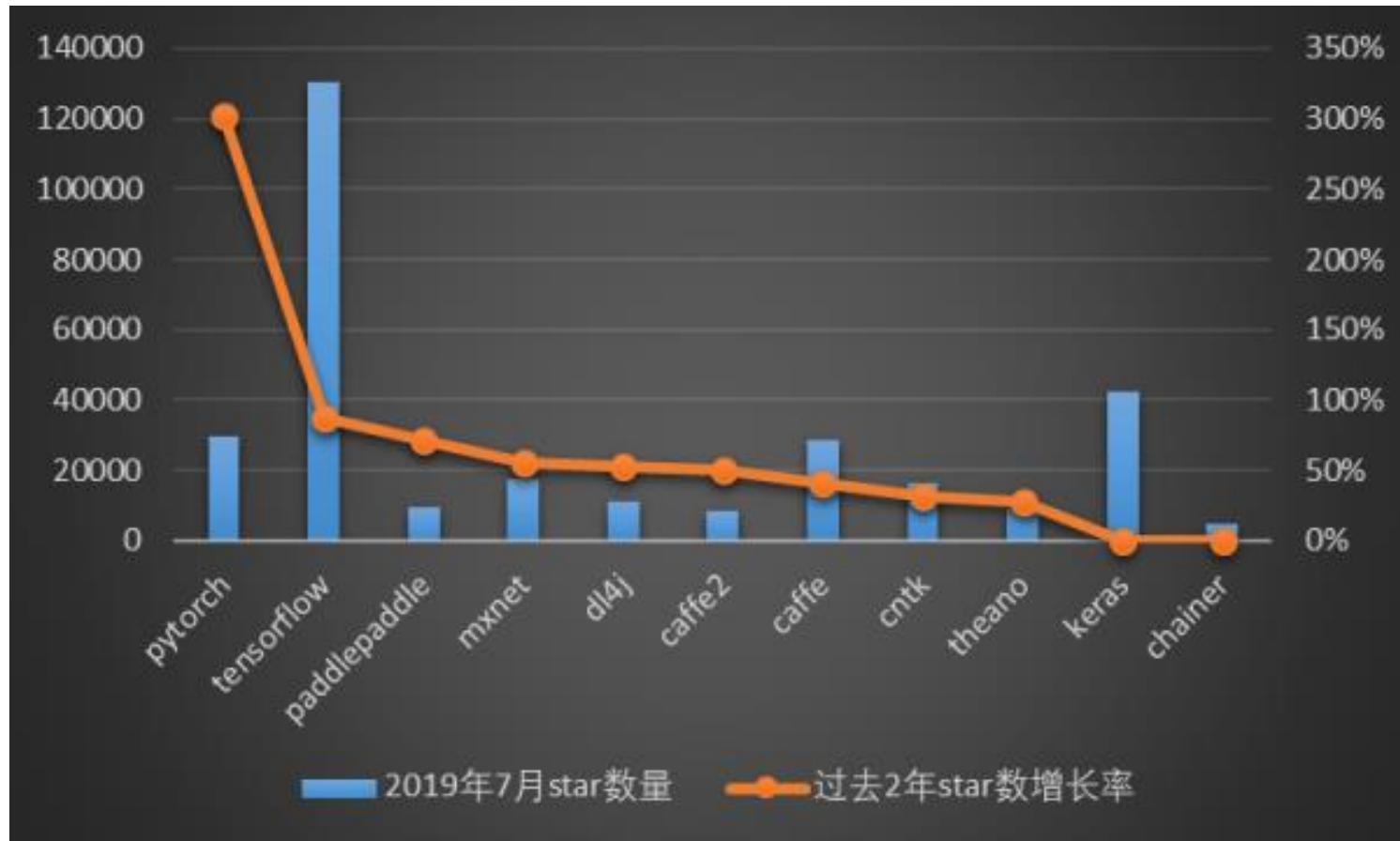
深度学习框架对照

□ 2018年GitHub统计



深度学习框架对照

□ 2019年GitHub统计





2

Tensorflow

TensorFlow的起源

- 2015年11月6日， Google公司宣布推出全新的机器学习开源工具TensorFlow
- TensorFlow最初由Google Brain团队开发， 基于Google开发的深度学习基础架构DistBelief构建起来的
- 官网： <https://tensorflow.google.cn/>
- Github： <https://github.com/tensorflow/tensorflow>

TensorFlow的起源

□ **TensorFlow**: 一个采用计算图的形式表述数值计算的编程框架

- 张量 (Tensor) : 计算图的基本数据结构，可以理解为多维数据
- 流 (Flow) : 张量之间通过计算互相转化的过程

TensorFlow中的张量

- 张量是一个物理量，对高维(维数 ≥ 2)的物理量进行“**量纲分析**”的一种工具
- 简单的可以理解为：
 - 一维数组称为矢量
 - 二维数组为二阶张量
 - 三维数组为三阶张量
 - ...

TensorFlow中的计算图

□ 计算图

- 用“节点”（Nodes）和“边”（Edges）的有向图来描述数学计算的图

□ 节点

- 表示施加的数学操作，但也可以表示数据输入（feed in）的起点/输出（push out）的终点，或者是读取/写入持久变量（persistent variable）的终点

□ 边

- 表示“节点”之间的输入/输出关系。“边”可以运输“大小可动态调整”的“张量”

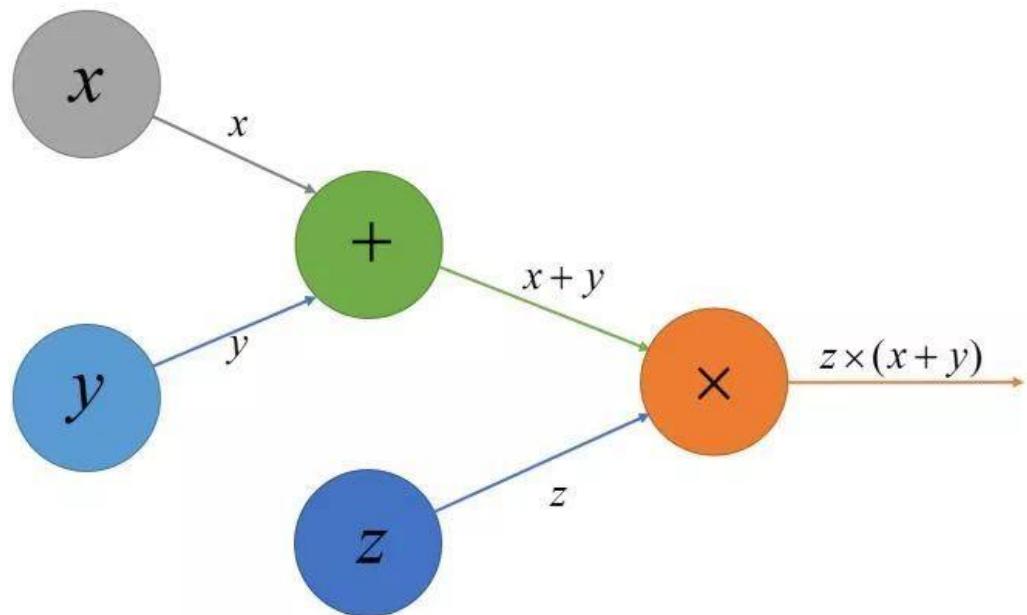
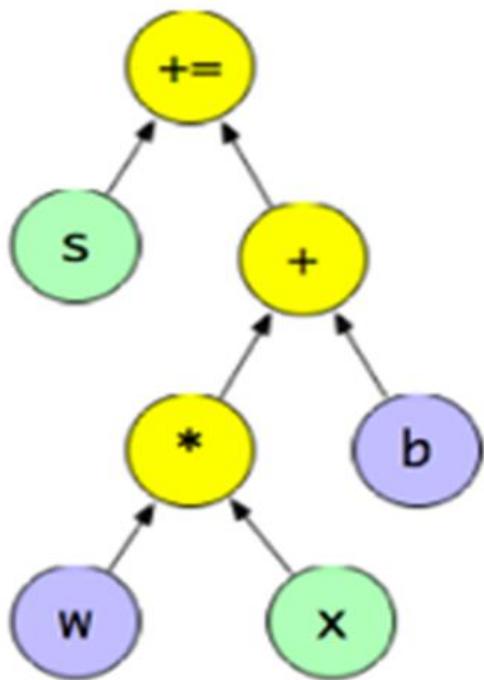
TensorFlow中的计算图

□ 节点类型

- Operation: 有一个或者两个输入节点的节点，执行某个操作
- Constant: 没有输入节点的节点，执行过程中节点的值不变
- Variable: 没有输入节点的节点，执行过程中节点中的值会变
- Placeholder: 没有输入节点的节点，执行过程中等待外部输入

TensorFlow中的计算图

□ 计算图示例

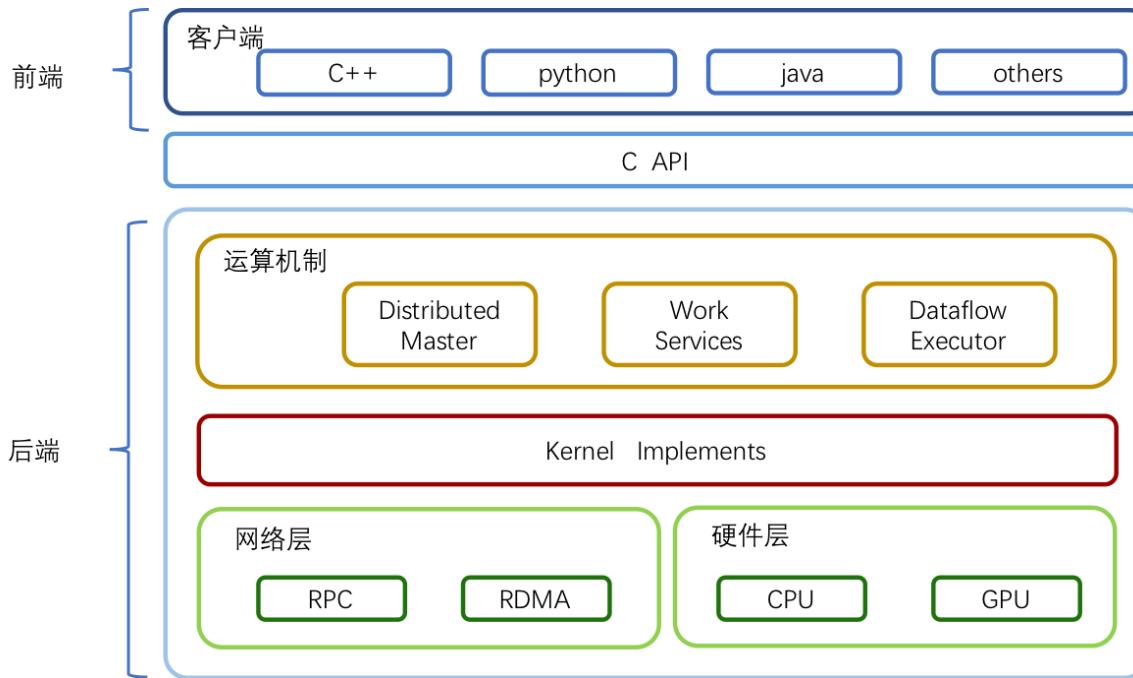


TensorFlow架构

□ 前端系统

- 提供多语言编程环境，提供统一的编程模型来构造计算图
- 通过Session的形式，连接TensorFlow后端的「运行时」系统，启动计算图的执行过程

□ 执行(后端)系统：提供运行时环境，负责执行计算图



TensorFlow的使用

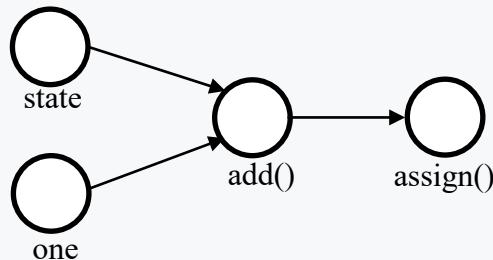
- 使用张量（Tensor） 表示数据
- 通过变量（Variable） 输入训练数据， 维护状态
- 使用计算图（Computational graph） 来表示计算任务
- 在会话（Session） 的上下文（Context） 中执行计算图

TensorFlow编程示例1： 创建计算图

```
1 import tensorflow as tf
2
3 #使用tensorflow在默认的图中创建节点，这个节点是一个变量，命名为 "counter"
4 state = tf.Variable(0, name="counter")
5
6 #创建一个常量
7 one = tf.constant(1)
8
9 #对常量与变量进行简单的加法操作，这点需要说明的是：在TensorFlow中，
10 #所有的操作op和变量都视为节点，tf.add() 的意思就是在tf的默认图中添加一个op，这个op是用来做加法操作的。
11 new_value = tf.add(state,one)
12
13 #赋值操作。将new_value的值赋值给update变量
14 update = tf.assign(state,new_value)
15
```

TensorFlow编程示例1： 创建计算图

```
16 #此处用于初始化变量。但是这句话仍然不会立即执行。需要通过sess来将数据流动起来。如果有Variable，一定需要写这句话
17 init = tf.initialize_all_variables()
18
19 #切记：所有的运算都应该在session中进行，即 with tf.Session() as sess
20 with tf.Session() as sess:
21     #对变量进行初始化，执行（run）init语句
22     sess.run(init)
23     #循环3次，并且打印输出。
24     for _ in range(3):
25         sess.run(update)
26         # 打印变量时也需要用sess.run
27         print(sess.run(state))
```

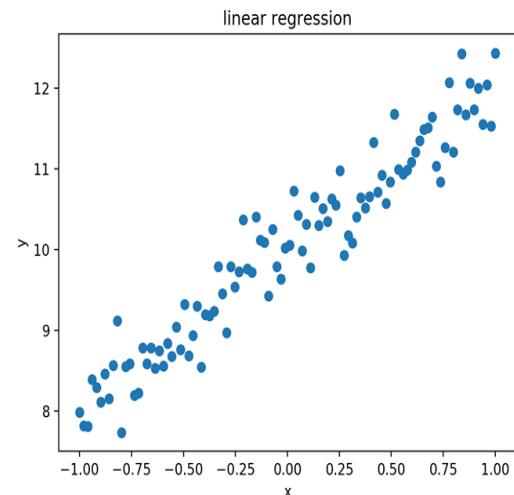


TensorFlow编程示例2：一元线性回归

```
# -*- coding: utf-8 -*-
import tensorflow as tf
import numpy as np

# 散点图数据, 理论 w=2, b=10
train_X = np.linspace(-1, 1, 100) # [-1, 1] 之间的 100 个数
train_Y = 2 * train_X + \
          np.random.randn(*train_X.shape) * 0.33 + 10

# 构建计算图
X = tf.placeholder(tf.float32, name="x")
Y = tf.placeholder(tf.float32, name="y")
w = tf.Variable(0.0, name="weight")
b = tf.Variable(0.0, name="bias")
loss = tf.square(Y - X * w - b) # 损失函数为离差平方和
train_op = tf.train.GradientDescentOptimizer(0.01).minimize(loss) # 梯度下降算法
```



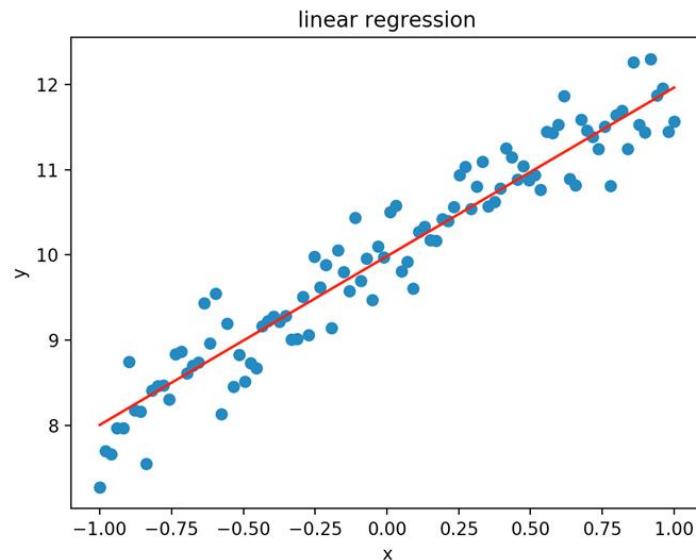
TensorFlow编程示例2：一元线性回归

```
# 运行计算图
with tf.Session() as sess:
    sess.run(tf.global_variables_initializer())

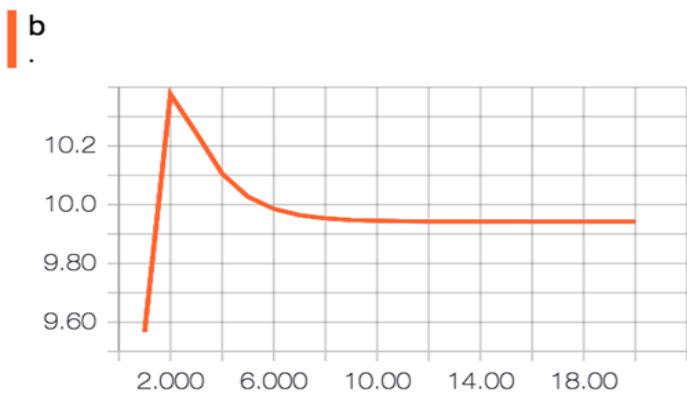
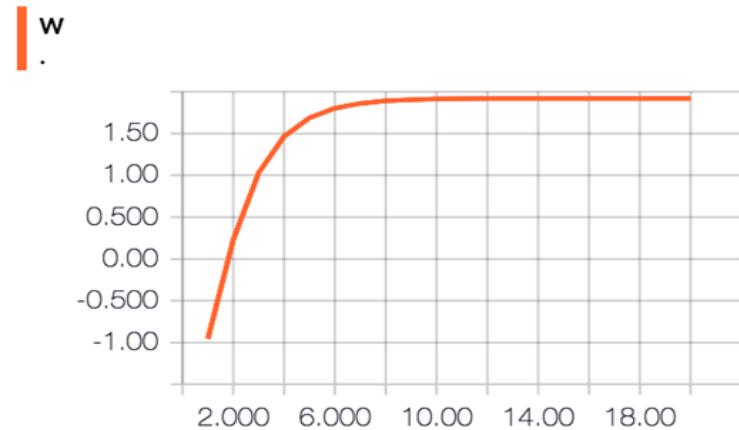
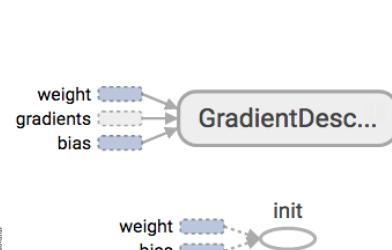
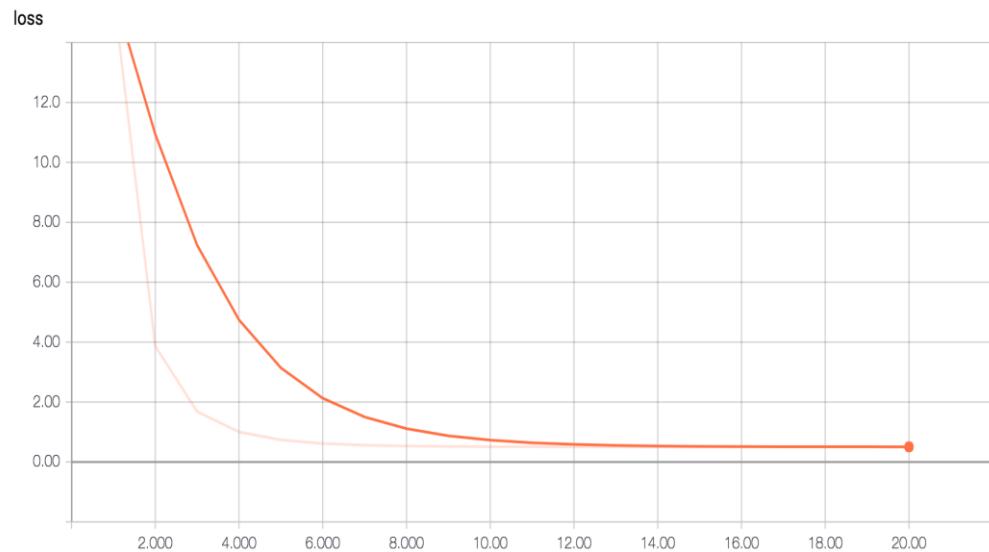
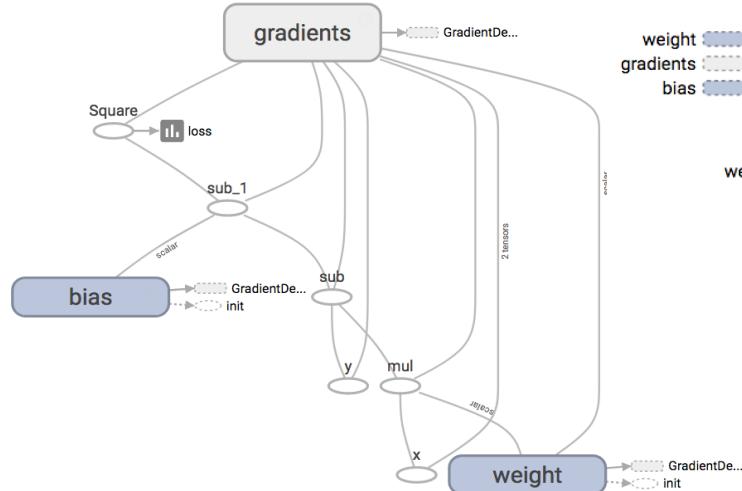
    for i in range(20):#20次迭代
        for (x, y) in zip(train_X, train_Y):
            op_result, w_result, b_result = \
                sess.run([train_op, w, b], feed_dict={X: x, Y: y})
        print("Epoch: {}, w: {}, b: {}".format(i+1, w_result, b_result))
```

```
Epoch: 14, w: 1.95590674877, b: 10.0173501968
Epoch: 15, w: 1.95616412163, b: 10.0172538757
Epoch: 16, w: 1.95629501343, b: 10.0172052383
Epoch: 17, w: 1.95636260509, b: 10.0171775818
Epoch: 18, w: 1.95639693737, b: 10.017162323
Epoch: 19, w: 1.95641410351, b: 10.0171556473
Epoch: 20, w: 1.95642268658, b: 10.0171527863
```

```
Process finished with exit code 0
```



TensorFlow编程示例2：一元线性回归



TensorFlow编程示例3：BP网络手写数字识别

□ 导入相关包及数据包

```
1 import tensorflow as tf  
2 import urllib  
3 from tensorflow.examples.tutorials.mnist import input_data  
4 import numpy as np  
5 import matplotlib.pyplot as plt  
6 from PIL import Image, ImageFilter  
7 mnist = input_data.read_data_sets("MNIST_data/", one_hot = True)
```

```
1 输出结果：  
2 Extracting MNIST_data/train-images-idx3-ubyte.gz  
3 Extracting MNIST_data/train-labels-idx1-ubyte.gz  
4 Extracting MNIST_data/t10k-images-idx3-ubyte.gz  
5 Extracting MNIST_data/t10k-labels-idx1-ubyte.gz
```

TensorFlow编程示例3：BP网络手写数字识别

- 建立BP网络模型

```
1 #建立BP神经网络模型
2 num_classes = 10#数据类型0-9
3 input_size = 784#28*28
4 hidden_units_size = 30#层节点数
5 batch_size = 100#
6 training_iterations = 50000#迭代次数
7 # 设置变量
8 X = tf.placeholder(tf.float32, shape = [None, input_size])
9 Y = tf.placeholder(tf.float32, shape = [None, num_classes])
10 W1 = tf.Variable (tf.random_normal ([input_size, hidden_units_size],
11                               stddev = 0.1))#hidden_units_size = 30#正态分布随机数
12 B1 = tf.Variable (tf.constant (0.1),
13                   [hidden_units_size])
14 W2 = tf.Variable (tf.random_normal ([hidden_units_size,
15                               num_classes], stddev = 0.1))#正态分布随机数
16 B2 = tf.Variable (tf.constant (0.1), [num_classes])
```

Tensorflow编程示例3：BP网络手写数字识别

• 建立BP网络模型

```
17 # 搭建计算网络 使用 relu 函数作为激励函数 这个函数就是  $y = \max(0, x)$  的一个类似线性函数 拟合程度还是不错的
18 # 使用交叉熵损失函数 这是分类问题例如： 神经网络 对率回归经常使用的一个损失函数
19 #第1层神经网络
20 hidden_opt = tf.matmul (X, W1) + B1#矩阵运算
21 hidden_opt = tf.nn.relu (hidden_opt)#激活函数
22 #第2层神经网络
23 final_opt = tf.matmul (hidden_opt, W2) + B2#矩阵运算
24 final_opt = tf.nn.relu (final_opt)#激活函数, 最终的输出结果
25 loss = tf.reduce_mean (
26     tf.nn.softmax_cross_entropy_with_logits (labels = Y, logits = final_opt))#损失函数, 交叉熵方法
27 opt = tf.train.GradientDescentOptimizer (0.1).minimize (loss)
28 init = tf.global_variables_initializer ()#全局变量初始化
29 correct_prediction = tf.equal (tf.argmax (Y, 1), tf.argmax (final_opt, 1))
30 accuracy = tf.reduce_mean (tf.cast (correct_prediction, 'float'))#将张量转化成float
```

Tensorflow编程示例3：BP网络手写数字识别

- 模型训练及结果

```
1 # 进行计算 打印正确率
2 sess = tf.Session ()#生成能进行TensorFlow计算的类
3 sess.run (init)
4 for i in range (training_iterations) :
5     batch = mnist.train.next_batch (batch_size)#每次迭代选用的样本数100
6     batch_input = batch[0]
7     batch_labels = batch[1]
8     training_loss = sess.run ([opt, loss], feed_dict = {X: batch_input, Y: batch_labels})
9     if (i+1) % 10000 == 0 :
10         train_accuracy = accuracy.eval (session = sess, feed_dict = {X: batch_input,Y: batch_labels})
11         print ("step : %d, training accuracy = %g " % (i+1, train_accuracy))
```

1	step : 10000, training accuracy = 0.98
2	step : 20000, training accuracy = 0.98
3	step : 30000, training accuracy = 1
4	step : 40000, training accuracy = 1
5	step : 50000, training accuracy = 1

Tensorflow编程示例3：BP网络手写数字识别

- 测试结果可视化

```
1 #####测试集输出结果可视化
2 def res_Visual(n):
3     #sess=tf.Session()
4     #sess.run(tf.global_variables_initializer())
5     final_opt_a=tf.argmax (final_opt, 1).eval(session=sess,feed_dict = {X: mnist.test.images,Y: mnist.test.labels})
6     fig, ax = plt.subplots(nrows=int(n/5),ncols=5 )
7     ax = ax.flatten()
8     print('前{}张图片预测结果为: '.format(n))
9     for i in range(n):
10         print(final_opt_a[i],end=',')
11         if int((i+1)%5) ==0:
12             print('\t')
13             #图片可视化展示
14             img = mnist.test.images[i].reshape((28,28))#读取每行数据, 格式为Ndarry
15             ax[i].imshow(img, cmap='Greys', interpolation='nearest')#可视化
16             print('测试集前{}张图片为: '.format(n))
17 res_Visual(20)
```

TensorFlow编程示例3：BP网络手写数字识别

- 测试结果

1 前20张图片预测结果为：

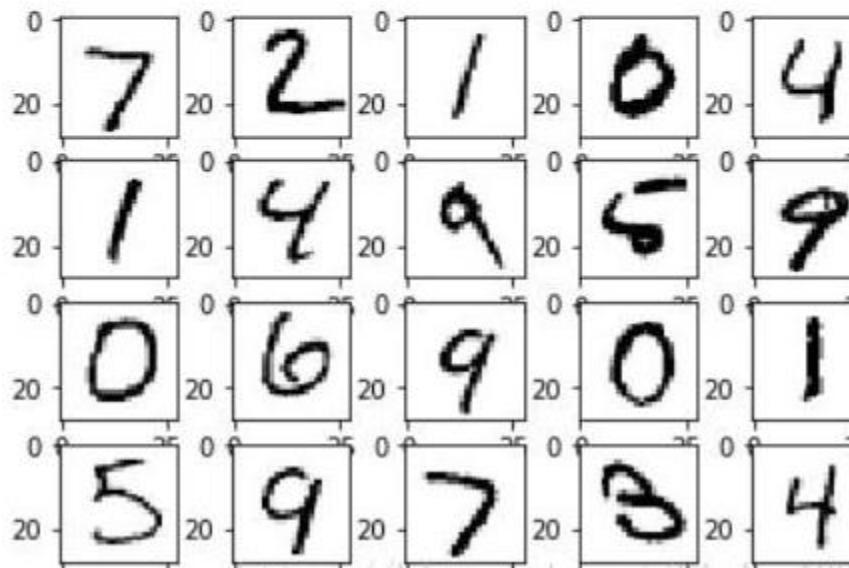
2 7,2,1,0,4,

3 1,4,9,6,9,

4 0,6,9,0,1,

5 5,9,7,3,4,

6 测试集前20张图片为：





中国科学院大学
University of Chinese Academy of Sciences

2

Pytorch

Pytorch的起源

- 2017年1月，Facebook人工智能研究院（FAIR）团队在Github上开源了Pytorch，并迅速占据Github热度榜榜首
- Pytorch的历史可以追溯到2002年就诞生的Torch。Torch使用了一种不是很大众的语言Lua作为接口
- 2017年，Torch的幕后团队对Tensor之上的所有模块进行了重构，推出了Pytorch



简洁直观的设计理念

- PyTorch 的设计思路是线性、直观且易于使用的。当你的代码出现Bug 的时候，不会因为错误的指向或者异步和不透明的引擎浪费太多的时间
- PyTorch 的代码相对于TensorFlow 而言，更加简洁直观，同时对于TensorFlow高度工业化的很难看懂的底层代码，PyTorch 的源代码就要友好得多，更容易看懂，更加吸引初学者学习

简洁直观的设计理念

□ PyTorch代码示例

- 构造初始化矩阵

```
x = torch.rand(5, 3)  
print(x)
```

输出：

```
tensor([[ 0.6291,  0.2581,  0.6414],  
       [ 0.9739,  0.8243,  0.2276],  
       [ 0.4184,  0.1815,  0.5131],  
       [ 0.5533,  0.5440,  0.0718],  
       [ 0.2908,  0.1850,  0.5297]])
```

- 函数计算

```
y = torch.rand(5, 3)  
print(x + y)
```

输出：

```
tensor([[-0.1859,  1.3970,  0.5236],  
       [ 2.3854,  0.0707,  2.1970],  
       [-0.3587,  1.2359,  1.8951],  
       [-0.1189, -0.1376,  0.4647],  
       [-1.8968,  2.0164,  0.1092]])
```

PyTorch使用动态计算图

□ TensorFlow 属于命令式的编程语言，而且是静态的，首先必须构建一个神经网络，然后一次又一次使用同样的结构，如果想要改变网络的结构，需要从头开始

- 静态计算这样的机制允许编译器进行更大程度的优化，但是这也意味着你所期望的程序与编译器实际执行之间存在着更多的差异，不便于debug
- 采用的诸如 scope 和 sessions 等不寻常的机制容易使人感到疑惑不解

PyTorch使用动态计算图

- PyTorch通过一种反向自动求导的技术，可以让你零延迟地任意改变神经网络的行为，能够为你奇思妙想的实现获得最高的速度和最佳的灵活性，这也是PyTorch 对比 **TensorFlow** 最大的优势
 - 比如在RNN网络中， PyTorch可以动态调整网络时间步(Time step)

PyTorch安装和使用

□ 网址：<https://pytorch.org/>

- 较好的移动端支持：Android和IOS
- 丰富的教程（包含中文）：<http://pytorch123.com/>

The screenshot shows the PyTorch build configuration interface. It includes dropdown menus for PyTorch Build (Stable (1.5) or Preview (Nightly)), Your OS (Linux, Mac, Windows), Package (Conda, Pip, LibTorch, Source), Language (Python, C++ / Java), and CUDA (9.2, 10.1, 10.2, None). Below the configuration is a command-line instruction: `conda install pytorch torchvision cudatoolkit=10.2 -c pytorch`.

PyTorch Build	Stable (1.5)	Preview (Nightly)		
Your OS	Linux	Mac	Windows	
Package	Conda	Pip	LibTorch	Source
Language	Python	C++ / Java		
CUDA	9.2	10.1	10.2	None

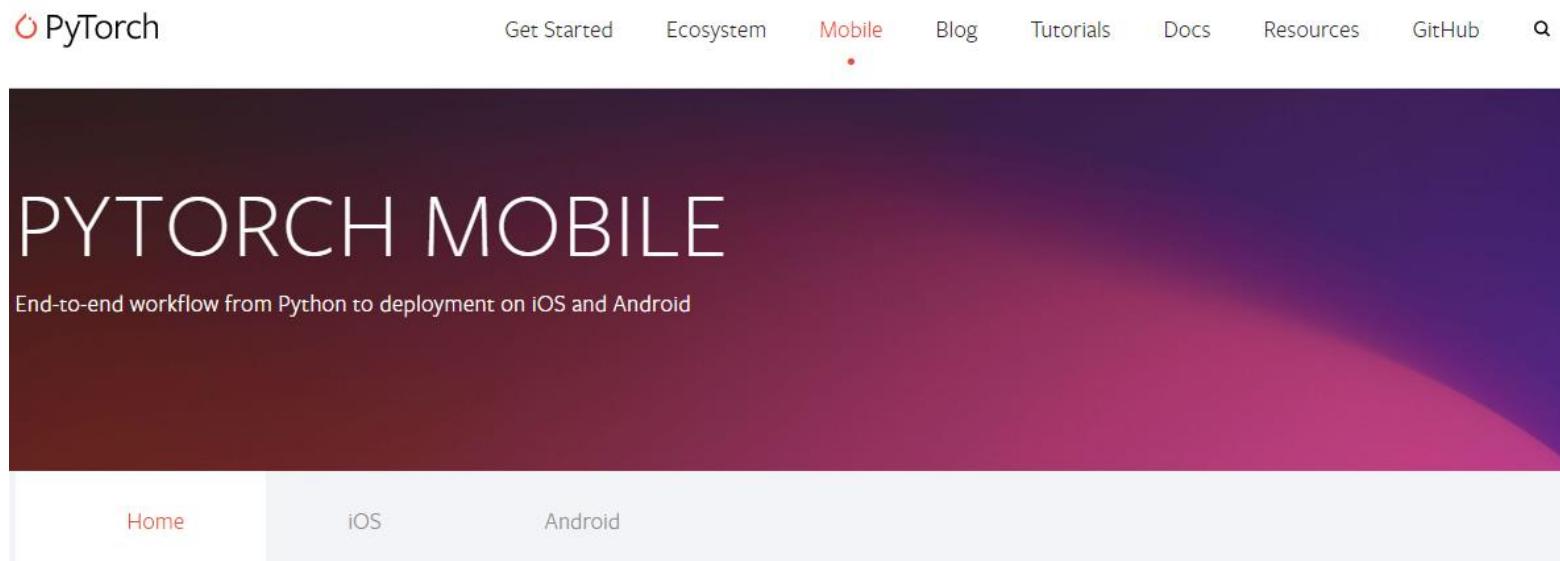
Run this Command:

```
conda install pytorch torchvision cudatoolkit=10.2 -c pytorch
```

PyTorch安装和使用

□ 网址: <https://pytorch.org/>

– 较好的移动端支持: Android和IOS



PyTorch Mobile

Running ML on edge devices is growing in importance as applications continue to demand lower latency. It is also a foundational element for privacy-preserving techniques such as federated learning. As of PyTorch 1.3,

PyTorch安装和使用

□ 通过定义类实现神经网络

```
import torch
import torch.nn as nn
import torch.nn.functional as F

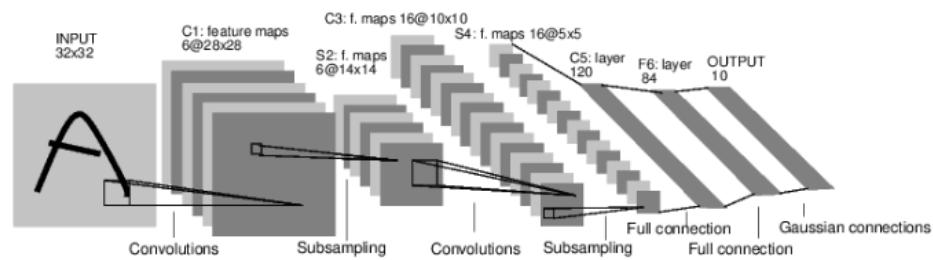
class Net(nn.Module):

    def __init__(self):
        super(Net, self).__init__()
        # 1 input image channel, 6 output channels, 5x5 square convolution
        # kernel
        self.conv1 = nn.Conv2d(1, 6, 5)
        self.conv2 = nn.Conv2d(6, 16, 5)
        # an affine operation: y = Wx + b
        self.fc1 = nn.Linear(16 * 5 * 5, 120)
        self.fc2 = nn.Linear(120, 84)
        self.fc3 = nn.Linear(84, 10)

    def forward(self, x):
        # Max pooling over a (2, 2) window
        x = F.max_pool2d(F.relu(self.conv1(x)), (2, 2))
        # If the size is a square you can only specify a single number
        x = F.max_pool2d(F.relu(self.conv2(x)), 2)
        x = x.view(-1, self.num_flat_features(x))
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = self.fc3(x)
        return x

    def num_flat_features(self, x):
        size = x.size()[1:] # all dimensions except the batch dimension
        num_features = 1
        for s in size:
            num_features *= s
        return num_features

net = Net()
print(net)
```



输出：

```
Net(
  (conv1): Conv2d(1, 6, kernel_size=(5, 5), stride=(1, 1))
  (conv2): Conv2d(6, 16, kernel_size=(5, 5), stride=(1, 1))
  (fc1): Linear(in_features=400, out_features=120, bias=True)
  (fc2): Linear(in_features=120, out_features=84, bias=True)
  (fc3): Linear(in_features=84, out_features=10, bias=True)
)
```

PyTorch安装和使用

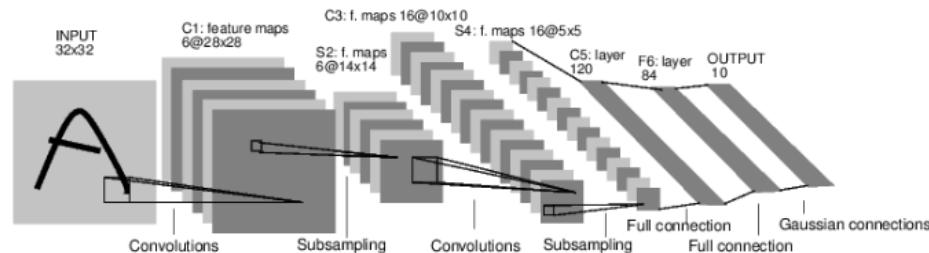
□ 通过定义类实现神经网络

训练：

```
import torch.optim as optim

# create your optimizer
optimizer = optim.SGD(net.parameters(), lr=0.01)

# in your training loop:
optimizer.zero_grad()    # zero the gradient buffers
output = net(input)
loss = criterion(output, target)
loss.backward()
optimizer.step()    # Does the update
```

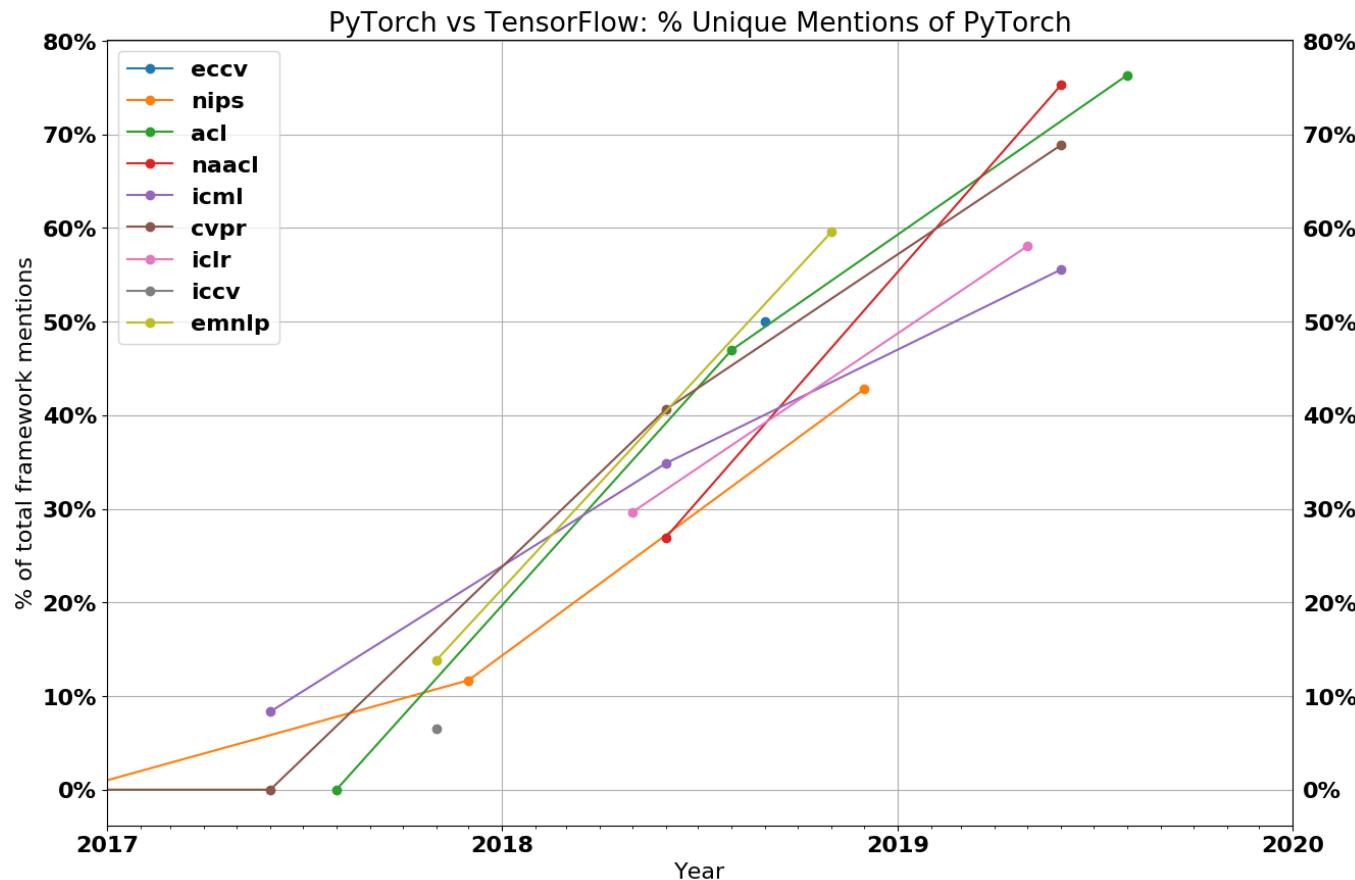


测试：

```
outputs = net(images)
```

PyTorch快速发展

□ 单独使用 PyTorch 的论文与 Tensorflow/PyTorch 论文之间的比率





3 Keras

Keras的起源

- Keras于2015年由Google人工智能研究员Francois Chollet等人提出
- Keras是一个高层的神经网络API，由纯Python编写而成并使用其他深度学习框架Tensorflow、Theano及CNTK作为后端



较低的使用门槛

- Keras为支持快速实验而生，能够把想法迅速转换为结果
- Keras应该是深度学习框架中最容易上手的一个，它提供了一致而简洁的API，能大大减少一般应用下用户的工作量，对于新用户非常友好

过度的封装

- 严格意义上来说，Keras并不能称为一个深度学习框架，它更像是一个高层的深度学习接口。Keras的缺点也很明显：过度的封装导致调试困难
- 由于同时支持三种不同的深度学习框架作为后端，Keras做了层层封装，导致用户在新增操作或是获取底层的数据信息时过于困难
- 许多Bug隐藏在层层封装之下，导致Keras的程序有些缓慢

Keras示例

□ 创建网络模型

```
from keras.models import Sequential
model = Sequential()
from keras.layers import Dense

#第一个卷积层，4个卷积核，每个卷积核大小5*5。1表示输入的图片的通道,灰度图为1通道。
model.add(Convolution2D(4, 1, 5, 5, border_mode='valid'))
model.add(Activation('tanh'))
#第二个卷积层，8个卷积核，每个卷积核大小3*3。4表示输入的特征图个数，等于上一层的卷积核个数
#激活函数用tanh
#采用maxpooling， poolsize为(2,2)
model.add(Convolution2D(8,4, 3, 3, border_mode='valid'))
model.add(Activation('tanh'))
model.add(MaxPooling2D(poolsize=(2, 2)))

#第三个卷积层，16个卷积核，每个卷积核大小3*3
#激活函数用tanh
#采用maxpooling， poolsize为(2,2)
model.add(Convolution2D(16, 8, 3, 3, border_mode='valid'))
model.add(Activation('tanh'))
model.add(MaxPooling2D(poolsize=(2, 2)))

#全连接层，先将前一层输出的二维特征图flatten为一维的。
#Dense就是隐藏层。16就是上一层输出的特征图个数。4是根据每个卷积层计算出来的：(28-5+1)得到24,(24-3+1)/2得到11，(11-3+1)/2得到4
#全连接有128个神经元节点,初始化方式为normal
model.add(Flatten())
model.add(Dense(16*4*4, 128, init='normal'))
model.add(Activation('tanh'))
#Softmax分类，输出是10类别
model.add(Dense(128, 10, init='normal'))
model.add(Activation('softmax'))
```

Keras示例

□ 创建网络模型

```
# 训练  
model.compile(loss=keras.losses.categorical_crossentropy,  
optimizer=keras.optimizers.SGD(lr=0.01, momentum=0.9, nesterov=True))  
model.fit(x_train, y_train, epochs=5, batch_size=32)  
loss_and_metrics = model.evaluate(x_test, y_test, batch_size=128)  
classes = model.predict(x_test, batch_size=128)
```



4

PaddlePaddle(飞桨)

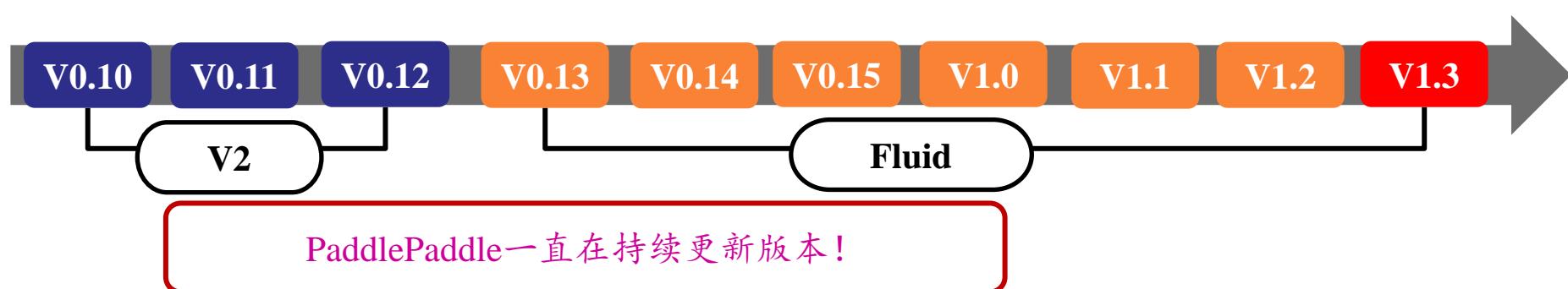
PaddlePaddle(飞桨)的发展历程

□ PaddlePaddle是百度提供的开源深度学习框架，致力于为开发者和企业提供最好的深度学习研发体验，框架本身具有易学、易用、安全、高效四大特性，是最适合中国开发者和企业的深度学习工具

— 网址：<http://www.paddlepaddle.org/zh>

PaddlePaddle前身是2013年百度自主研发的深度学习平台，一直为百度内部工程师使用。

2016年9月1日百度世界大会上，百度首席科学家Andrew Ng（吴恩达）首次宣布将百度深度学习平台对外开放，命名PaddlePaddle。



PaddlePaddle的功能特点与优势

功能特点

- 教程文档友好，提供jupyter文档

开源

- 开源，代码在github上公开

- 社区活跃，合作伙伴支持，有很多的贡献者

速度快、方便

- 底层C++编写，运行速度快，占用内存少

- 接口Python，使用方便，开发便捷

分布式系统

- 底层硬件同时支持CPU和GPU

- 分布式系统，支持单机运行和云上运行

部署方便

- 支持docker部署和原生包部署

- 支持多操作系统windows、macOS、Linux

优势分析

- ◆ 提供搭建私有云的全套解决方案
 - 帮助用户保护数据
 - 充分支持分布式系统
 - 模型并行和数据并行

- ◆ 灵活易用且性能高
 - 支持多种机器学习和优化算法，可定制
 - 全面支持多机、多GPU环境，优化的通信实现使高吞吐与高性能成为可能

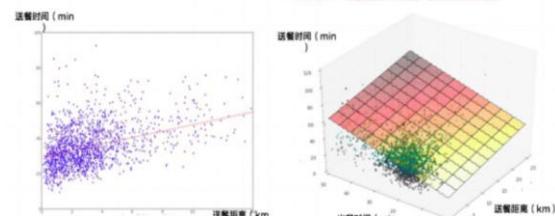
官网：<http://www.paddlepaddle.org/>

PaddlePaddle的功能特点与优势

□ PaddlePaddle已在百度30多项主要产品和服务之中发挥着巨大的作用

外卖预估出餐时间

如何更准确预估出餐时间



更多维度因子考虑进行机器学习拟合曲线寻找规律

精准推荐用户所需信息



PaddlePaddle_易学易用的全开源深度学习平台



百度深度学习平台PaddlePaddle是并行分布式全功能深度学习框架，易学易用，高效灵活，支持海量图像识别分类、机器翻译和自动驾驶等多个领域的业务需求全面开源！

• 用科技让复杂的世界更简单

技术服务: 人脸与人体识别

客户身份: 个人开发者

联系方式: 请输入您的邮箱

我要咨询

字符识别(OCR)

机器翻译

百度翻译手机版提供中英日韩等10多个语种的文本互译；使用OCR技术实现拍照整屏实时翻译、菜单翻译等功能，无需键盘输入，提升用户体验。



病毒和垃圾信息检测



百度卫士

更快，更安全



自动驾驶



海量图像识别分类



PaddlePaddle的模型库

□ PaddlePaddle 为开发者提供了丰富的模型库，从而可以快速构建强大的应用，满足各种场景的需求

◆ 计算机视觉

◆ **图像分类**是根据图像的语义信息对不同类别图像进行区分，是计算机视觉中重要的基础问题。主要应用于安防领域的人脸识别和智能视频分析等，交通领域的交通场景识别等。PaddlePaddle提供了常用模型：AlexNet、VGG、

GoogleNet、ResNet等，方便用户使用

◆ **目标检测**的目标是给定一张图像或是一个视频帧，让计算机找出其中所有目标的位置，并给出每个目标的具体类别。PaddlePaddle提供了Faster-RCNN、MobileNet-SSD、PyramidBox等模型，满足不同场景需求



PaddlePaddle的模型库

自然语言处理

◆ 阅读理解-DuReader模型

应用场景：语音助手、智能搜索、智能客服、智能机器人

◆机器翻译-Transformer模型

应用场景：机器翻译

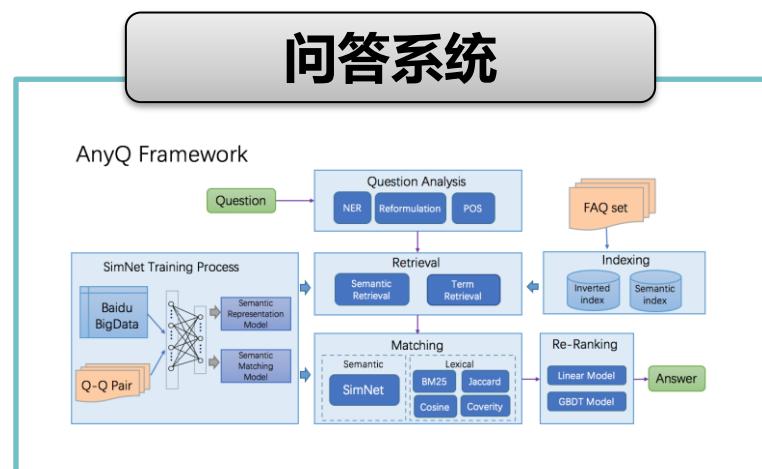
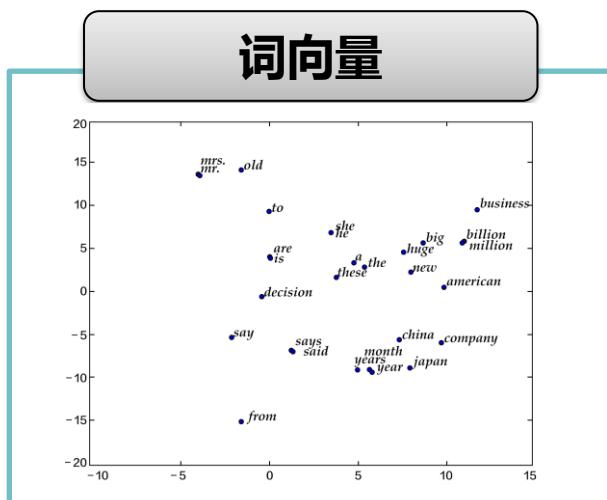
◆语义匹配-DAM模型

应用场景：问答系统、对话系统、智能客服等

◆情感分析-Senta模型

应用场景：舆情分析、热点话题分析

◆语言模型LSTM、GRU等



PaddlePaddle的模型库

推荐

以新闻推荐系统为例，存在多个可以使用深度学习技术的环节，例如新闻的自动化标注，个性化新闻召回，个性化匹配与排序等

PaddlePaddle对推荐算法的训练提供了完整的支持，并提供了多种模型配置供用户选择。如TagSpace、RRU4Rec、SSR、DeepCTR等模型

个性化推荐

个性化推荐挖掘用户的行为，将用户可能感兴趣的信息和商品推荐给用户，提高交易成功率和用户体验度



除此之外，PaddlePaddle还为开发者们提供了语音识别、强化学习等方面的一些模型库，可以满足各种场景需求

PaddlePaddle的本地环境搭建

目前PaddlePaddle支持以下环境：

- Ubuntu 14.04 / 16.04 / 18.04
- CentOS 7 / 6
- MacOS 10.11 / 10.12 / 10.13 / 10.14
- Windows7 / 8 / 10 (专业版/企业版)

- PaddlePaddle支持四种形式的安装，pip安装、conda安装、docker安装、源码安装，**本教程以Windows7下pip安装为例。**
- 若需要docker或源码安装，请参考：
http://paddlepaddle.org/documentation/docs/zh/1.3/beginners_guide/install/install_Docker.html

PaddlePaddle的本地环境搭建

□ 不同安装方式参见

<https://www.paddlepaddle.org.cn/install/quick>

[快速安装](#) [下一步做什么](#)

操作系统	Windows	macOS	Ubuntu	CentOS
安装方式	pip	conda	docker	源码编译
Python 版本	Python 3			Python 2
CUDA 版本	CUDA 10	CUDA 9	CPU版本	

百度AI Studio简介

□ 百度AI Studio一站式开发平台

- 囊括了AI教程、代码环境、算法算力、数据集，并提供免费的在线云计算的一体化编程环境。官网地址：<http://aistudio.baidu.com/>

AI Studio特性



本平台支持Chrome/Firefox/Safari等浏览器，在IE/360等其他浏览器上不能保证正常运行

百度AI Studio平台已经为使用者预置了Python语言环境，以及百度PaddlePaddle深度学习开发框架。同时用户可以在其中自行加载Scikit-Learn等机器学习库。

百度AIStudio简介

Baidu 大脑 | AI Studio

项目 数据集 比赛 课程 社区 教育合作 文档 访问飞桨官网 中 | En

 大脑 | AI Studio

AI学习地图

- ◆ 新手入门 秒懂强大的在线编程平台
 - ◆ 初级-高级课程 帮您轻松掌握AI开发

项目列表

- ◆ 公开项目 经典应用场景
 - ◆ 论文复现 帮助理解最新论文并比较
 - ◆ 个人项目 开发者项目列表

2000+优质公开项目，覆盖CV、NLP、推荐算法等众多AI热门领域，完美支持Notebook、脚本及图形化任务。

Designing Network Design Spaces 翻译笔记 2020-04-10

2020年何凯明组的论文，主旨是借助NAS（神经网络搜索）技术，探索出更加一般化的卷积网络设计准则。论文地址：<https://arxiv.org/>

百度AIStudio简介

◆ 创建你自己的项目

- 1) **创建项目**: 点击按钮创建项目
- 2) **项目环境**: Python2.7和Python3.7
- 3) **预加载项目框架**: PaddlePaddle-1.4.1和1.7.1
- 4) **配置资源**: Notebook、脚本任务
- 5) **项目名称**: 用来标识项目, 便于日后进行查找和管理, 创建后支持修改



创建项目

* 项目名称: 项目名称最多输入40字

* 项目标签: 请选择

* 配置资源: Notebook

* 预加载项目框架: PaddlePaddle 1.7.1

* 项目环境: python3.7

* 项目描述: python3.7
python2.7

数据集 + 添加数据集 ■ 创建数据集

创建 取消

万物皆可embedding之图数据embedding-GCN详解(下) 2020-04-12

万物皆可embedding之图数据embedding-GCN详解(下), 训练GCN用作半监督分类, 本文中将以空手道俱乐部为例, 选取2个节点...

xinxinxinxixi Notebook 初级 计算机视觉 深度学习 ... 137 16 708 8

基于Paddle预训练模型的迁移训练(Fix_ResNeXt101) 2020-04-09

在本项目中, 使用预训练模型 (Fix_ResNeXt101_32x48d_wsl), 对于21种鱼类图片进行实践讲解

李长安 Notebook 初级 计算机视觉 78 4 415 22

Designing Network Design Spaces 翻译笔记 2020-04-10

2020年何凯明组的论文, 主旨是借助NAS(神经网络搜索)技术, 探索出更加一般化的卷积网络设计准则。论文地址: [https://arxiv.org/...](https://arxiv.org/)

百度AIStudio简介

◆ 创建你自己的项目

- 1) **创建项目**: 点击按钮创建项目
- 2) **项目环境**: Python2.7和Python3.7
- 3) **预加载项目框架**: PaddlePaddle-1.4.1和1.7.1
- 4) **配置资源**: Notebook、脚本任务
- 5) **项目名称**: 用来标识项目, 便于日后进行查找和管理, 创建后支持修改



创建项目

* 项目名称: 项目名称最多输入40字

* 项目标签: 请选择

* 配置资源: Notebook

* 预加载项目框架: PaddlePaddle 1.7.1

* 项目环境: python3.7

* 项目描述: python3.7
python2.7

数据集 + 添加数据集 ■ 创建数据集

创建 取消

万物皆可embedding之图数据embedding-GCN详解（下） 2020-04-12

万物皆可embedding之图数据embedding-GCN详解（下），训练GCN用作半监督分类，本文中将以空手道俱乐部为例，选取2个节点...

xinxinxinxixi

Notebook

初级 计算机视觉 深度学习 ...

137

16

708

8

1000+开放数据集，种类多样，支持数据集预览、下载、上传，单次上传容量高达100G

基于Paddle预训练模型的迁移训练Fix_NeXt101_32x48d_wsl 2020-04-09

在本项目中，使用预训练模型（Fix_NeXt101_32x48d_wsl），对于21种鱼类图片进行实践讲解

编写 李长安

Notebook

初级 计算机视觉

78

4

415

22

Designing Network Design Spaces 翻译笔记 2020-04-10

2020年何凯明组的论文，主旨是借助NAS（神经网络搜索）技术，探索出更加一般化的卷积网络设计准则。论文地址：[https://arxiv.org/...](https://arxiv.org/)

百度AIStudio创建项目

◆ 创建你自己的项目

✓ 如果项目涉及到数据集，可以考虑直接使用系统预置的数据集，点击[添加数据集](#)按钮

注意：每个项目最多可以引入两个数据集，如无合适的数据集，用户也可以[自行上传创建新数据集](#)。

✓ 点击“创建”，“查看”，可进入项目详情页

项目已创建成功，是否查看项目详情？

[查看](#)

[取消](#)

公开/个人数据集切换

公开数据集 个人数据集 收藏数据集 搜索公开数据集

经	经典MNIST数据集 手写数字，训练集数量60000, 测试集数量10000. 参考: http://...
<input checked="" type="checkbox"/>	infer_3.png infer_3.png
<input type="checkbox"/>	infer_9.jpg 数字9
<input type="checkbox"/>	房价预测 房价预测
<input type="checkbox"/>	鸢尾花数据集 鸢尾花数据集

[添加\(1/2\)](#) 点击添加

< 1 2 3 4 5 ... 130 >

百度AIStudio运行项目

◆ 项目详情页

✓ 进入项目详情页

我的项目 > Test01

Test01

测试创建项目

Notebook 初级 计算机视觉 深度学习 数据分析 分类 私有 Python3 2020-04-23 21:33:24

版本内容 数据集 在线服务

请选择版本 ① 草稿 2020-04-23 21:51:50

In[1]

```
# 查看当前挂载的数据集目录, 该目录下的变更重启环境后会自动还原
# View dataset directory. This directory will be recovered automatically after resetting environment.
!ls /home/aistudio/data
```

In[2]

```
# 查看工作区文件, 该目录下的变更将会持久保存. 请及时清理不必要的文件, 避免加载过慢.
# View personal work directory. All changes under this directory will be kept even after reset. Please clean unnecessary files in time to speed
!ls /home/aistudio/work
```

启动环境 公开项目
删除 修改 设置为公开
停止 部署
✓点击进入，项目环境初始化
删除项目 修改项目名称、数据集等

百度AIStudio运行项目

◆ 项目详情页

✓ 进入项目详情页



百度AIStudio编辑代码

◆ 代码编辑页面

The screenshot shows the Baidu AIStudio interface with a red box highlighting the left sidebar and the top navigation bar.

左侧栏 (Test01):

- 文件夹图标: Test01
- 数据集 (Dataset):
 - 工作目录 (work)
 - 数据 (data)
 - 外部库 (external-libraries)
- 设置 (Settings) 图标

顶部栏:

- 文件 (File)、编辑 (Edit)、运行 (Run)、代码执行器 (Code Executor)、帮助 (Help) 菜单项，均被红色方框包围。
- 工具栏 (Toolbars): 包括 Notebook、终端-1、Code、Markdown、定位到当前运行Cell等按钮。
- 状态栏: 空闲中 (Idle)。

内容编辑区 (Content Editor):

显示了三个代码单元格 (Cells) 的内容：

```
[1] 1 # 查看当前挂载的数据集目录, 该目录下的变更重启环境后会自动还原
     2 # View dataset directory. This directory will be recovered automatically after resetting environment.
     3 !ls /home/aistudio/data

[2] 1 # 查看工作区文件, 该目录下的变更将会持久保存. 请及时清理不必要的文件, 避免加载过慢.
     2 # View personal work directory. All changes under this directory will be kept even after reset. Please clean unnecessary files.
     3 !ls /home/aistudio/work

[3] 1 # 如果需要进行持久化安装, 需要使用持久化路径, 如下方代码示例:
     2 # If a persistence installation is required, you need to use the persistence path as the following:
     3 !mkdir /home/aistudio/external-libraries
     4 !pip install beautifulsoup4 -t /home/aistudio/external-libraries

[→] Looking in indexes: https://pypi.mirrors.ustc.edu.cn/simple/
Collecting beautifulsoup4
  Downloading https://mirrors.tuna.tsinghua.edu.cn/pypi/web/packages/e8/b5/7bb03a696f2c9b7af792a8f51b82974e51c268f15e925fc834876
    
```

百度AIStudio编辑代码

◆ 新建块

✓ 选择块类型：Code或Markdown, 分别插入代码块或文字块

The screenshot shows the AIStudio Notebook interface. At the top, there's a toolbar with various icons. A red box highlights the '+ Code' button. Below the toolbar, a code cell is displayed with the following Python code:

```
# 如果需要进行持久化安装, 需要使用持久化路径, 如下方代码示例: (共4行)
[2]
1 import sys
2 sys.path.append('/home/aistudio/external-libraries')
3 # -*- coding: utf-8 -*-
4
5 print("Welcome to AIStudio")
```

Below the code cell, the output shows the message 'Welcome to AIStudio'. At the bottom, it says '运行时长: 3毫秒 结束时间: 2020-04-23 22:46:48'.

✓ 代码块，输入代码，点击运行
，在下方输出运行结果

快捷工具栏



- 运行 : 运行当前选中的code cell.
- 全部运行 : 运行全部code cell.
- 中断 : 中断运行某个code cell.
- 重启 : 重启代码执行器, 清空环境中的环境变量、缓存变量、输出结果.
- 保存 : 保存Notebook项目文件.

百度AIStudio数据集

◆ 创建数据集

创建数据集 X

* 数据集名称

* 数据集简介

* 上传数据集文件 [↑ 上传文件](#)
最多上传10个文件，单个文件不超过10GB，文件名长度<= 18

下一步 取消



5

Huawei 平台



背景介绍

□ 教育部-华为“智能基座”产教融合协同育人基地

- 将鲲鹏、昇腾知识融入课程中



鲲鹏服务器

□ 鲲鹏计算产业

- 基于Kunpeng处理器构建的全栈IT基础设施、行业应用及服务，包括PC、服务器、存储、操作系统、中间件、虚拟化、数据库、云服务、行业应用以及咨询管理服务等

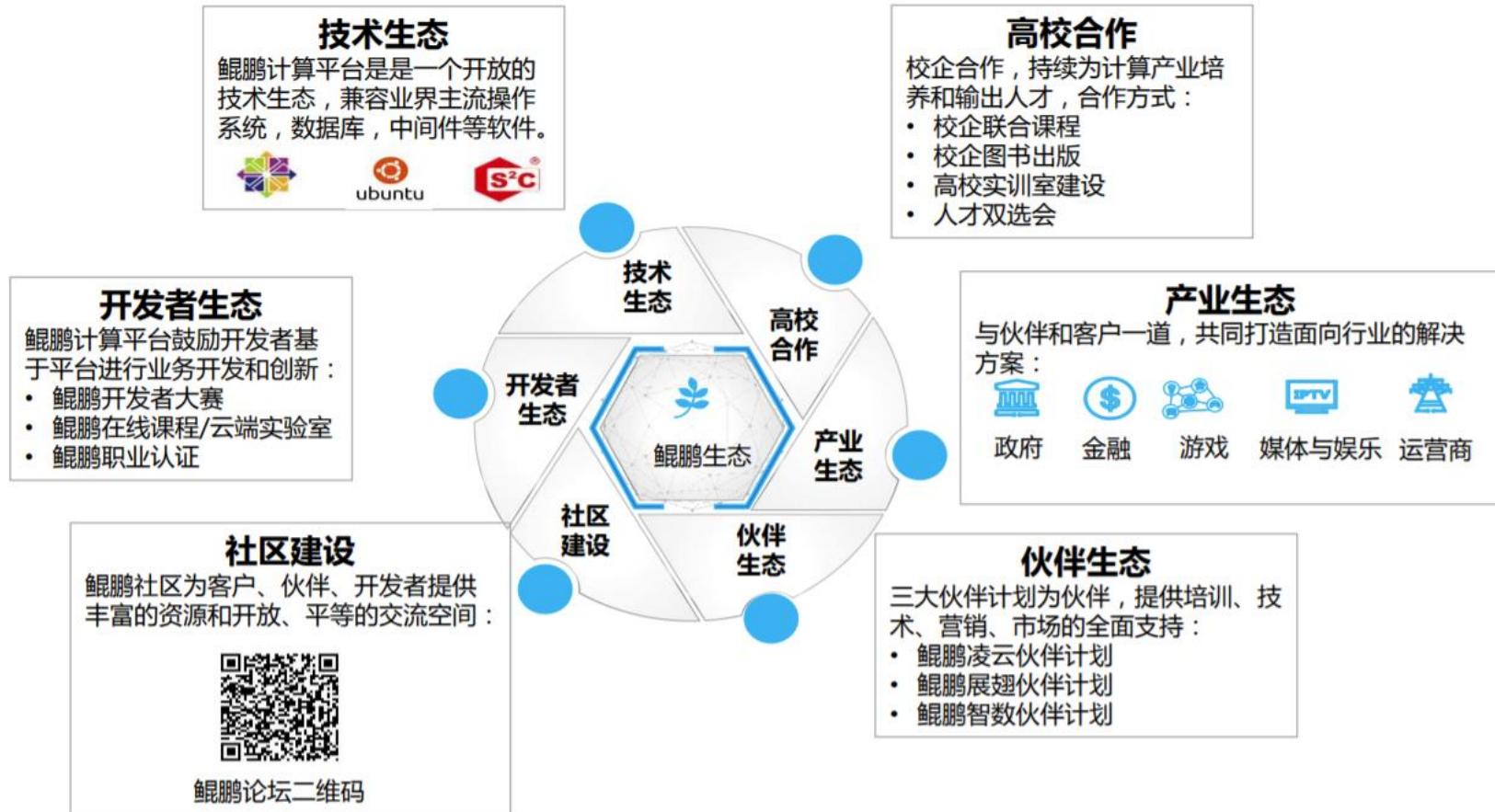


	社区发行版	商用发行版
国产OS	openEuler ...	KYLIN 中标麒麟 NeoKylin deepin ... 高斯数据库
国外OS	CentOS debian ...	SUSE ... ubuntu



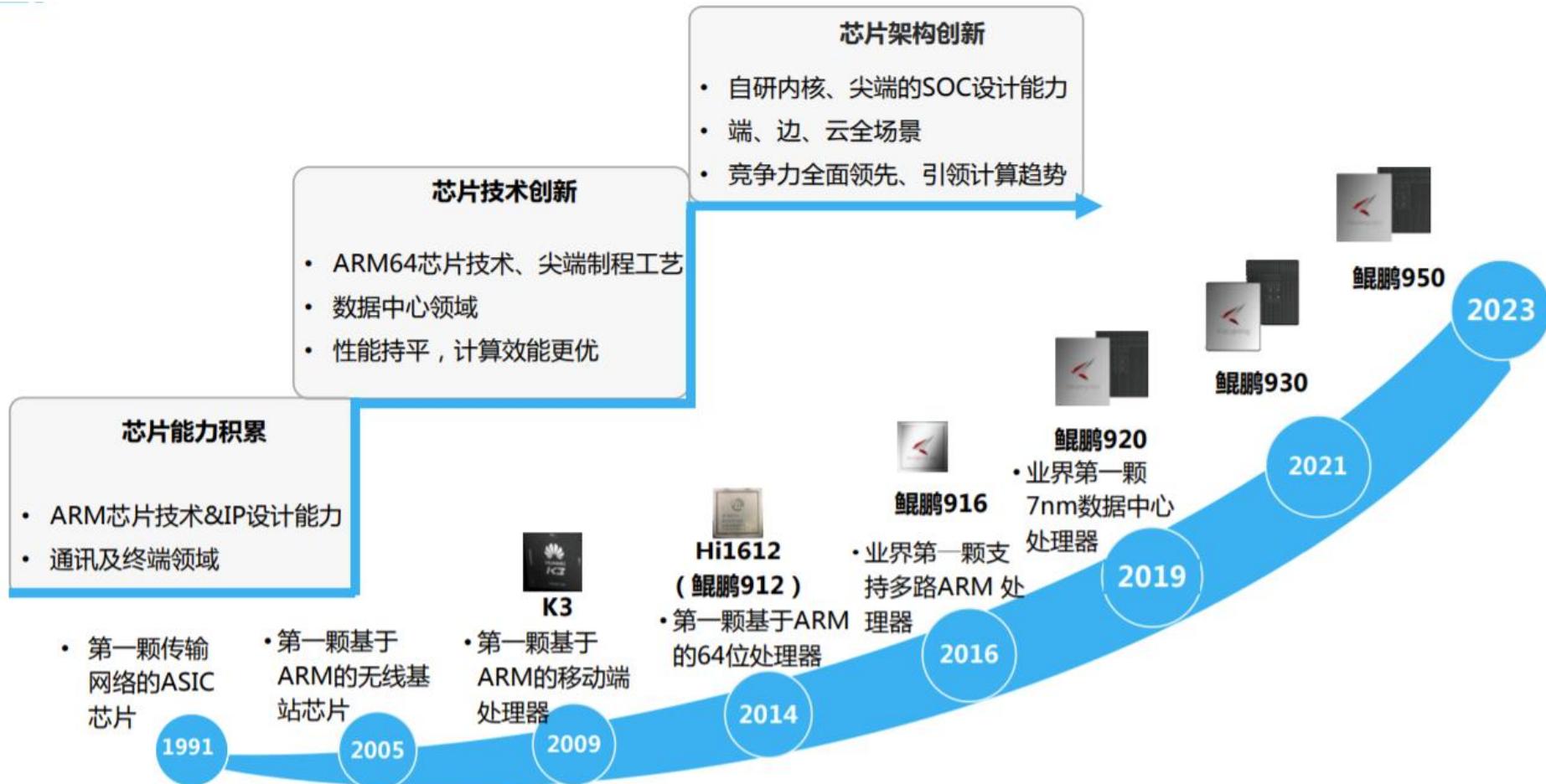
鲲鹏服务器

□ 鲲鹏计算产业生态全景



鲲鹏服务器

□ 鲲鹏的产业发展



□ 人工智能处理器

- 昇腾910和昇腾310处理器，采用华为自研的达芬奇架构NPU
 - 自创的3D Cube矩阵乘法单元
 - 昇腾910支持全场景人工智能应用，昇腾310主要用在边缘计算等低功耗的领域
- CPU/GPU/TPU (Tensor Processing Unit)/NPU (Neural network Processing Unit)



昇腾服务器

人工智能处理器

行业应用

能源、金融、公共、交通、电信、制造、教育等更多行业应用



ModelArts



HiAI Service



第三方平台

应用使能

MindX



MindX DL
深度学习使能



MindX Edge
智能边缘使能



ModelZoo
优选模型库



MindX SDK
行业SDK

AI框架

MindSpore / TensorFlow / PyTorch / ...

异构计算架构

CANN



Atlas 200 AI加速模块



Atlas 300I 推理卡



Atlas 300T 训练卡



Atlas 800 推理服务器



Atlas 900 PoD



Atlas 500 智能小站



Atlas 500 Pro 智能边缘服务器



Atlas 800 训练服务器



Atlas 900 AI集群



全流程开发工具链 Mindstudio



管理运维工具链 FusionDirector/SmartKit



昇腾社区 <https://www.hiascend.com/>



计算机科学与技术学院

SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY

昇腾服务器

□ 人工智能处理器

Atlas系列产品：提供AI训练、推理卡及训练服务器

CANN（异构计算架构）：芯片使能、驱动层（类似英伟达CUDA）

MindSpore（AI框架）：全场景AI框架（类似TensorFlow、Pytorch）

MindX SDK（昇腾SDK）：行业SDK和应用解决方案

ModelArts（AI开发平台）：华为云AI开发平台

MindStudio（全流程开发工具链）：AI全流程开发IDE

The diagram illustrates the Huawei Ascend AI ecosystem, organized into several sections:

- 行业应用 (Industry Applications):** Includes ModelArts, HiAI Service, and Third-party Platform.
- 应用使能 (Application Enablement):** Includes MindX DL (深度学习使能), MindX Edge (智能边缘使能), ModelZoo (优秀模型库), and MindX SDK (行业SDK).
- AI框架 (AI Framework):** Lists MindSpore / TensorFlow / PyTorch / ...
- 异构计算架构 (Heterogeneous Computing Architecture):** Lists CANN.
- Hardware Components:** Shows physical units including Atlas 200 AI 加速模块, Atlas 300I 推理卡, Atlas 300T 训练卡, Atlas 800 推理服务器, Atlas 900 AI 集群, Atlas 500 智能小站, and Atlas 500 Pro 智能边缘服务器.

On the right side, there are vertical icons and links:

- 管理运维工具链 (Management and Operation Tools Chain) - FusionDirector/SmartKit
- 全流程开发工具链 (Full-Flow Development Tools Chain) - Mindstudio
- 昇腾社区 (Ascend Community) - <https://www.huascend.com/>



昇腾服务器

□ 昇腾CANN

- AI场景的异构计算架构，通过提供多层次的编程接口，支持用户快速构建基于昇腾平台的AI应用和业务
- CANN 3.0已经覆盖10+操作系统和14+智能终端设备，一次编写代码，就能实现端边云全场景复用，极大提升了开发效率

The diagram illustrates the HiAI ecosystem and product portfolio, organized into several sections:

- 行业应用 (Industry Applications):** Includes ModelArts, HiAI Service, and Third-party Platform.
- 应用使能 (Application Enablement):** Includes MindX DL (深度学习使能), MindX Edge (智能边缘使能), ModelZoo (优选模型库), and MindX SDK (行业SDK).
- AI框架 (AI Framework):** Includes MindSpore / TensorFlow / PyTorch / ...
- 异构计算架构 (Heterogeneous Computing Architecture):** Labeled CANN (Concurrent Acceleration Network).
- 产品 (Products):** Shows various hardware components:
 - Atlas 200 AI 加速模块 (Atlas 200 AI Acceleration Module)
 - Atlas 300I 推理卡 (Atlas 300I Inference Card)
 - Atlas 300T 训练卡 (Atlas 300T Training Card)
 - Atlas 800 推理服务器 (Atlas 800 Inference Server)
 - Atlas 800 训练服务器 (Atlas 800 Training Server)
 - Atlas 900 AI 集群 (Atlas 900 AI Cluster)
 - Atlas 500 智能小站 (Atlas 500 Intelligent Station)
 - Atlas 500 Pro 智能边缘服务器 (Atlas 500 Pro Intelligent Edge Server)
 - Atlas 900 PoD (Atlas 900 PoD)
- 管理运维工具链 (Management and Operation Tools):** FusionDirector/SmartKit.
- 昇腾社区 (HiAI Community):** https://www.huawei.com/



昇腾服务器

□ 昇腾CANN

- 支持多种AI开发平台，华为自己开发了MindSpore的AI平台
- CANN 3.0上可以接各种深度学习编程框架的AI算法，下可以加速昇腾全系列AI芯片的训练和推理
- 用户可以用任何第三方框架调用常用的API库，而无需关心计算资源优化的问题



The diagram illustrates the Huawei Ascend AI ecosystem, organized into several vertical columns:

- 行业应用 (Industry Applications):** Includes ModelArts (Model Arts), HiAI Service, and Third-party Platform.
- 应用使能 (Application Enablement):** Includes MindX DL (MindX Deep Learning), MindX Edge (MindX Edge Computing), ModelZoo (Model Zoo), and MindX SDK (MindX Industry SDK).
- AI框架 (AI Frameworks):** Lists MindSpore / TensorFlow / PyTorch / ...
- 异构计算架构 (Heterogeneous Computing Architecture):** Lists CANN (Concurrent Acceleration Network).
- 硬件 (Hardware):** Shows various Ascend products:
 - Atlas 200 AI 加速模块 (Atlas 200 AI Acceleration Module)
 - Atlas 300I 推理卡 (Atlas 300I Inference Card)
 - Atlas 300T 训练卡 (Atlas 300T Training Card)
 - Atlas 800 推理服务器 (Atlas 800 Inference Server)
 - Atlas 900 PoD (Atlas 900 Pod)
 - Atlas 500 智能小站 (Atlas 500 Intelligent Station)
 - Atlas 500 Pro 智能边缘服务器 (Atlas 500 Pro Intelligent Edge Server)
 - Atlas 800 训练服务器 (Atlas 800 Training Server)
- 全流程开发工具链 (Full-Flow Development Tool Chain):** Lists FusionDirector and SmartKit.
- 管理运维工具 (Management and Operation Tools):** Lists Mindstudio.
- 昇腾社区 (Huawei Ascend Community):** Provides the URL <https://www.huawei.com/>.



昇腾服务器

□ 昇腾CANN

- 快速发展的社区和深度学习库
- <https://www.hiascend.com/zh/software/modelzoo>

The screenshot shows the HiAscend Model Zoo interface. On the left, there's a sidebar with icons for various categories: 全部 (All), 计算机视觉 (CV), 推荐 (Recommendation), 自然语言处理 (NLP), 语音 (Speech), 强化学习 (Reinforcement Learning), and 类别 (categorization). The main area has several filter sections: 应用级别 (Application Level) with options like Released, Benchmark, Official, and Research; 类型 (Type) with Training Script and Offline Model; 应用领域 (Application Domain) with OCR, Object Detection, Image Classification, Instance Segmentation, Recommendation, Image Synthesis, Natural Language Processing, Semantic Segmentation, and Machine Translation; and 框架 (Framework) with MindSpore (selected), PyTorch, and TensorFlow. Below these filters are four model cards: PSENet (PyTorch, OCR, Official), ATC PSENet (FP16) (PyTorch, Image Classification, Official), ATC YoloV4 (FP16) (PyTorch, Object Detection, Official), and ATC VGG19 (FP16) (PyTorch, Image Classification, Official). Each card provides a brief description of the model.



MindSpore

□ 华为开源自研AI框架MindSpore

- <https://www.mindspore.cn/>
- MindSpore提供了Python编程范式
- MindSpore提供了动态图和静态图统一的编码方式
- 能够用串行算法代码，自动实现分布式并行训练
- 具备训练过程静态执行和动态调试能力，开发者通过变更一行代码即可切换模式，快速在线定位问题
- 最佳匹配昇腾处理器，最大程度地发挥硬件能力，帮助开发者缩短训练时间，提升推理性能
- 支持云、边缘和手机上的快速部署，实现更好的资源利用和隐私保护



MindArts

□ 用户访问平台

— <https://www.huaweicloud.com/product/modelarts.html>

The screenshot shows the ModelArts user interface. The top navigation bar includes links for Control Panel, location (Beijing 4), search, and various service centers. The left sidebar lists categories like General Overview, Auto Learning, Data Management, Development Environment, Algorithm Management, Training Management, Model Management, Deployment, AI Gallery, Exclusive Resource Pool, and Global Configuration. The main content area features a banner for 'MindSpore - Huawei's self-developed full-scenario AI computing framework'. Below the banner, there are four buttons: 'Buy套餐包', 'My套餐包', 'Product Pricing', and 'Buy Promotional Package Hot!'. A decorative Easter egg icon is also present. The 'Newbie Guide' section contains three numbered steps: 1. Preparation Work (指导您完成OBS桶和文件夹的创建，确保与ModelArts在同一区域), 2. Acquire Data (指导您从OBS下载数据集并解压), and 3. Execution Guidance (指导您开始任务并完成预测) . Each step has a 'I've completed' button at the bottom.



MindArts

□ 开发环境

— <https://www.huaweicloud.com/product/modelarts.html>

The screenshot shows the Huawei Cloud ModelArts console. At the top, there is a navigation bar with the Huawei logo, user information (北京四), a search bar, and links for 费用中心 (Cost Center), 资源 (Resources), 工单 (Work Orders), 企业 (Enterprise), 备案 (Filing), 支持与服务 (Support & Services), and 中文 (简体) (Chinese (Simplified)).

The main area is titled "Notebook". It features a "Create" button and a message stating "您最多可以创建10个Notebook, 还可以创建10个Notebook." Below this is a table with columns: 名称 (Name), 状态 (Status), 工作环境 (Working Environment), 规格 (Specification), 描述 (Description), 创建时间 (Creation Time), 创建者 (Creator), and 操作 (Operations). A note at the bottom of the table says "暂无数据" (No data available).

The left sidebar is titled "ModelArts" and contains the following sections:

- 总览 (Overview)
- 自动学习 (Automatic Learning)
- 数据管理 (Data Management)
 - 开发环境 (Development Environment) (highlighted with a red box)
 - Notebook
- 算法管理 (Algorithm Management)
- 训练管理 (Training Management)
- 模型管理 (Model Management)
- 部署上线 (Deployment and Go-Live)
- AI Gallery
- 专属资源池 (Exclusive Resource Pool)
- 全局配置 (Global Configuration)



□ 可以自主选择多种计算资源和AI框架

- <https://www.huaweicloud.com/product/modelarts.html>

The screenshot shows the 'Create Notebook' wizard on the Huawei Cloud ModelArts platform. The current step is 'Service Selection' (1). The 'Billing Mode' is set to 'Pay-as-you-go'. The 'Name' field contains 'notebook-f38d'. The 'Description' field is empty. In the 'Working Environment' section, the 'Public Image' tab is selected, showing options like Multi-Engine 1.0 (Python3, Recommended), Multi-Engine 1.0 (Python2), Multi-Engine 2.0 (Python3), and Ascend-Powered-Engine 1.0 (Python3). The 'Resource Pool' section shows 'Public Resource Pool' selected. Under 'Type', 'CPU' is chosen. Under 'Specification', the 'Temporary Free' plan for 'CPU Version' is selected, showing 2 vCPUs and 8GB memory. A note below says it's suitable for code editing and testing. A warning box at the bottom left lists two points about temporary free usage and specifies that ModelArts free quota does not include OBS storage costs. The 'Storage Configuration' section at the bottom right shows 'Cloud Disk (EVS)' selected, with a note that EVS files will be deleted when the notebook stops.

□ 在线编程

— <https://www.huaweicloud.com/product/modelarts.html>

Notebook

PyCharm ToolKit | ModelArts SDK | 使用指南

温馨提示：
状态为“ 运行中”的Notebook正在产生费用，不使用时，请及时停止。若您已经设置了自动停止，请注意Notebook的剩余运行时间。

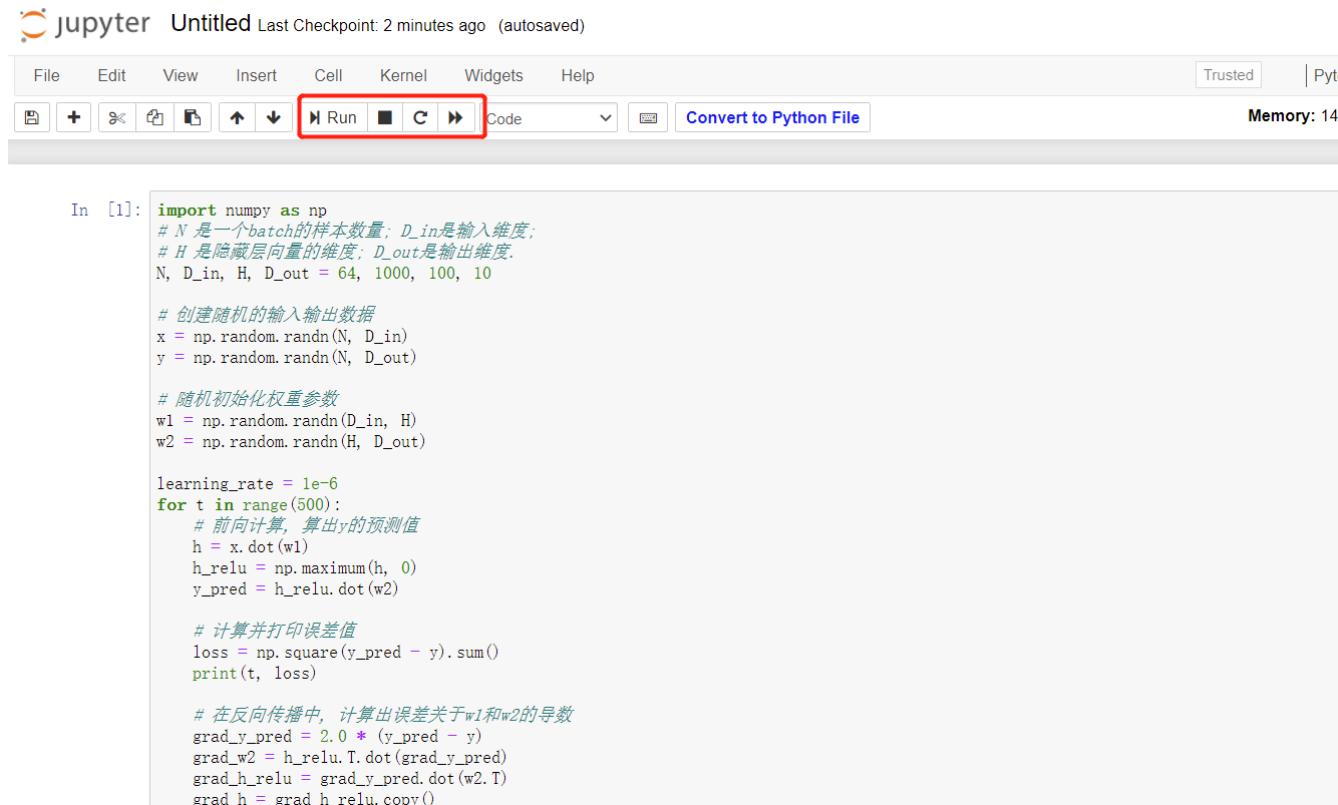
创建	您最多可以创建10个Notebook, 还可以创建9个Notebook。		只显示自己	全部状态	请输入名称查询		
名称	状态	工作环境	规格	描述	创建时间	创建者	操作
notebook-f38d	启动中	Multi-Engine 1.0 (py...)	2 核 4GiB	--	2021/05/11 09:55:05 GMT...	hw86190506	打开 打开JupyterLab 删除



MindSpore

□ 调试、运行等

- <https://www.huaweicloud.com/product/modelarts.html>



The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** jupyter Untitled Last Checkpoint: 2 minutes ago (autosaved)
- Toolbar:** File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Run, Stop, Cell, Run, Convert to Python File, Trusted, Pyto, Memory: 144.
- Code Cell:** In [1]:

```
import numpy as np
# N 是一个batch的样本数量; D_in是输入维度;
# H 是隐藏层向量的维度; D_out是输出维度.
N, D_in, H, D_out = 64, 1000, 100, 10

# 创建随机的输入输出数据
x = np.random.randn(N, D_in)
y = np.random.randn(N, D_out)

# 随机初始化权重参数
w1 = np.random.randn(D_in, H)
w2 = np.random.randn(H, D_out)

learning_rate = 1e-6
for t in range(500):
    # 前向计算, 算出y的预测值
    h = x.dot(w1)
    h_relu = np.maximum(h, 0)
    y_pred = h_relu.dot(w2)

    # 计算并打印误差值
    loss = np.square(y_pred - y).sum()
    print(t, loss)

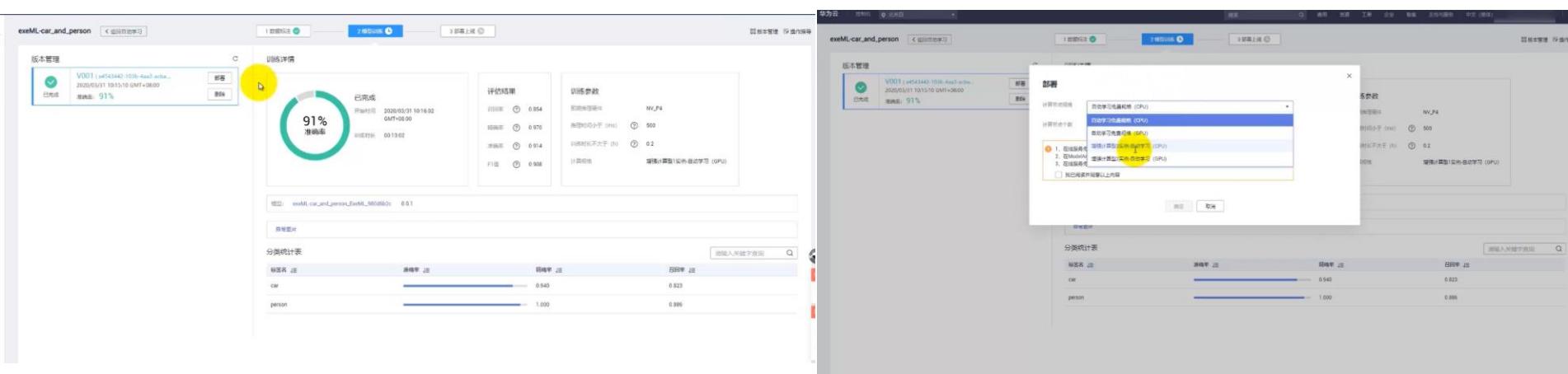
    # 在反向传播中, 计算出误差关于w1和w2的导数
    grad_y_pred = 2.0 * (y_pred - y)
    grad_w2 = h_relu.T.dot(grad_y_pred)
    grad_h_relu = grad_y_pred.dot(w2.T)
    grad_h = grad_h_relu.copy()
```




MindSpore

□ 可视化结果展示（训练和部署）

- <https://www.huaweicloud.com/product/modelarts.html>



华为云在线课程

□ 全面详细的课程培训体系

- <https://edu.huaweicloud.com/courses>

The screenshot displays the Huawei Cloud Education Platform's course catalog. At the top, there are filters for '技术领域' (Technology领域) including '全部', '计算', '存储', '网络', '云原生', '数据库', '人工智能' (selected), '鲲鹏', '昇腾', '大数据', '软件开发', '企业应用', '物联网', '视频', '应用中间件', '安全', '管理与监管', '解决方案', '华为云Stack', '迁移', and '通用技术'. Below these are filters for '用户角色' (User Role) with '全部' (selected), '初学者', '开发者', and '合作伙伴'. The sorting options are '综合排序', '最热', and '最新'. A search bar at the top right allows users to '搜索您感兴趣的课程' (Search for courses you are interested in). The main content area shows a grid of 8 course cards, each featuring the '人工智能' (AI) logo and a red '推荐' (Recommended) badge. The courses are:

- AI全栈成长计划-AI进阶篇 (15小时, 13.4k views, 华为云学院, 免费)
- ModelArts: 一站式AI开发平台 (2小时, 3k views, 华为云学院, 免费)
- AI全栈成长计划-AI应用篇 (15小时, 3.4k views, 华为云学院, 免费)
- 入门篇: 人工智能开启新纪元 (2小时, 1.7k views, 华为云学院, 免费)
- AI全栈成长计划-AI基础篇 (6小时, 10.4k views, 华为云学院, 免费)
- 文字识别服务 (1小时, 1.2k views, 华为云学院, 免费)
- 神经网络基础 (2小时, 409 views, 华为云学院, 免费)
- 人工智能概览 (1.5小时, 745 views, 华为云学院, 免费)



谢谢！



计算机科学与技术学院

SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY