

机器学习

算法 分类：神经网络（简单）
回归

深度学习

神经网络（深度）
图像：卷积神经网络
自然语言处理：循环神经网络

领域

图像等等等

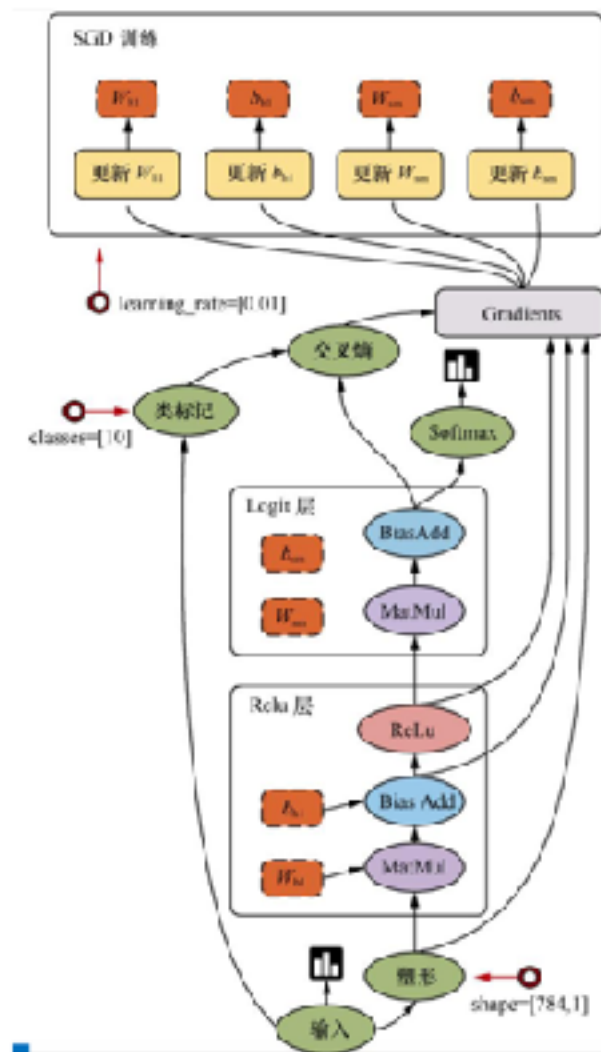
运行速度 数据量大：特征多 图片 计算 等待很长时间去优化 几小时，几天
算法：设计本身比较复杂

cpu: 运行操作系统，处理业务 计算能力不是特别突出 一个高中生

gpu: 专门为计算设计的 小学生*10000

非常简单：只需要一行代码

数据流图:



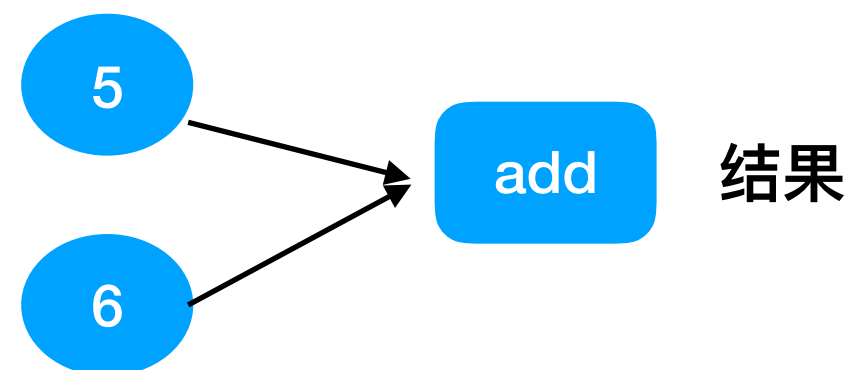
数组

变量

tensor: 张量

operation:(op): 专门运算的操作节点
所有操作都是一个op

图: graph:你的整个程序的结构



会话: 运算程序的图

程序图结构 — — —>

序列化文件

events 事件文件

web界面

tensorboard

summary: 摘要

计算密集型

IO密集型

框架

tensorflow

django, scrapy

cpu计算

http请求
磁盘操作等等

pythonAPI c++API RAPI

tensorflow:前端系统: 定义程序的图的机构
后端系统: 运算图结构

会话: 1、运行图的结构
2、分配资源计算
3、掌握资源 (变量的资源, 队列, 线程)

mac windows linux

cpu gpu

sess.run() 启动整个图

tensorflow->numpy

numpy:ndarray

数组

矩阵

张量

tensor类型

0

1

2维

2维

np.dot()



(10000,)

形状的改变

API

(100, 100)

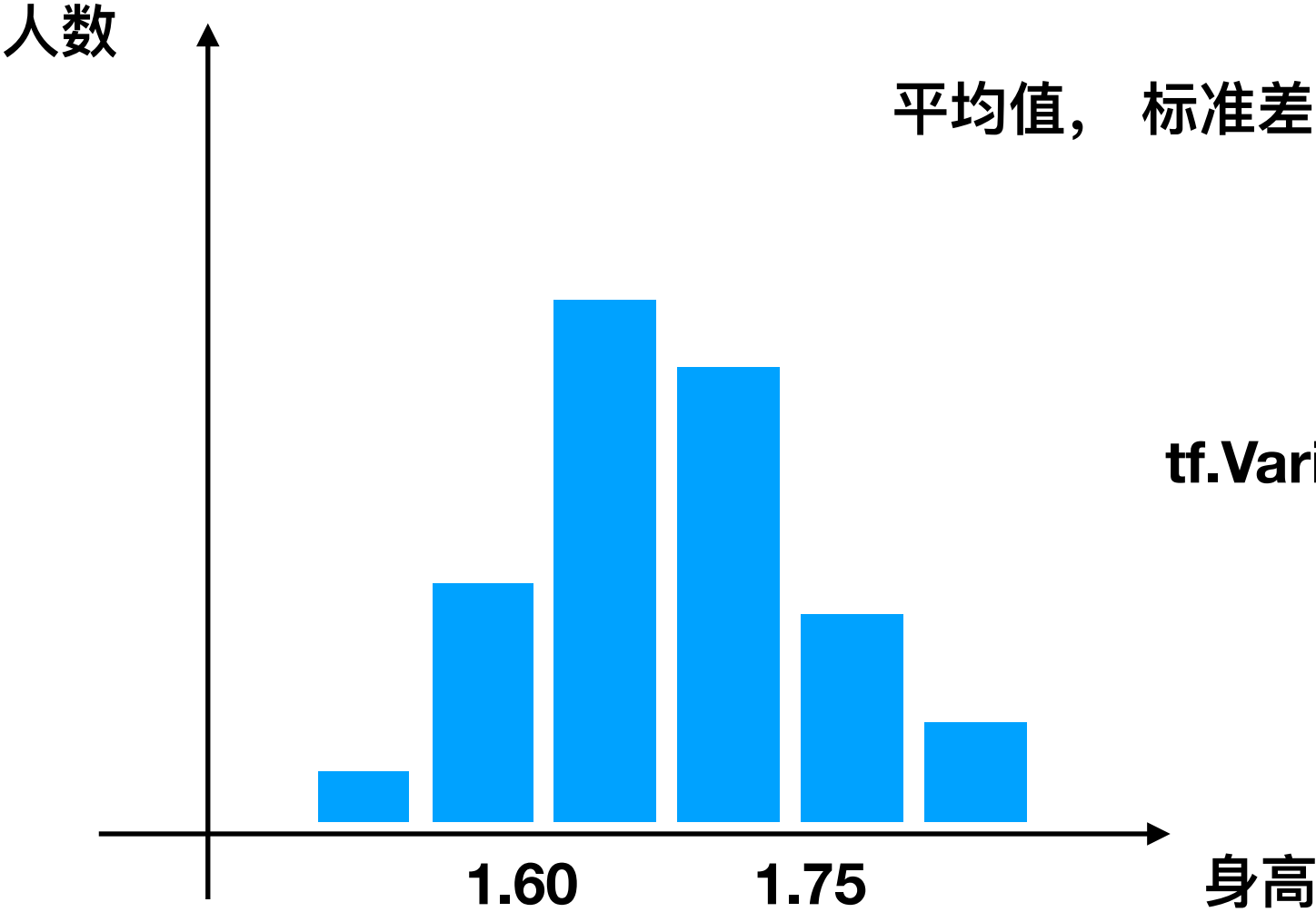
Numpy:reshape 把原来的数据通过直接修改

动态性状和静态形状： 在于有没有生成一个新的张量数据

正太分布（高斯分布）

平均值， 标准差

`tf.Variable()`



线性回归: $w_1x_1+w_2x_2+w_3x_3+.....w_nx_n +bias$

1、准备好1特征和1目标值 100 [100, 1]

$$y = x*0.7 + 0.8$$

2、建立模型 随机初始化准备一个权重w，一个偏置b

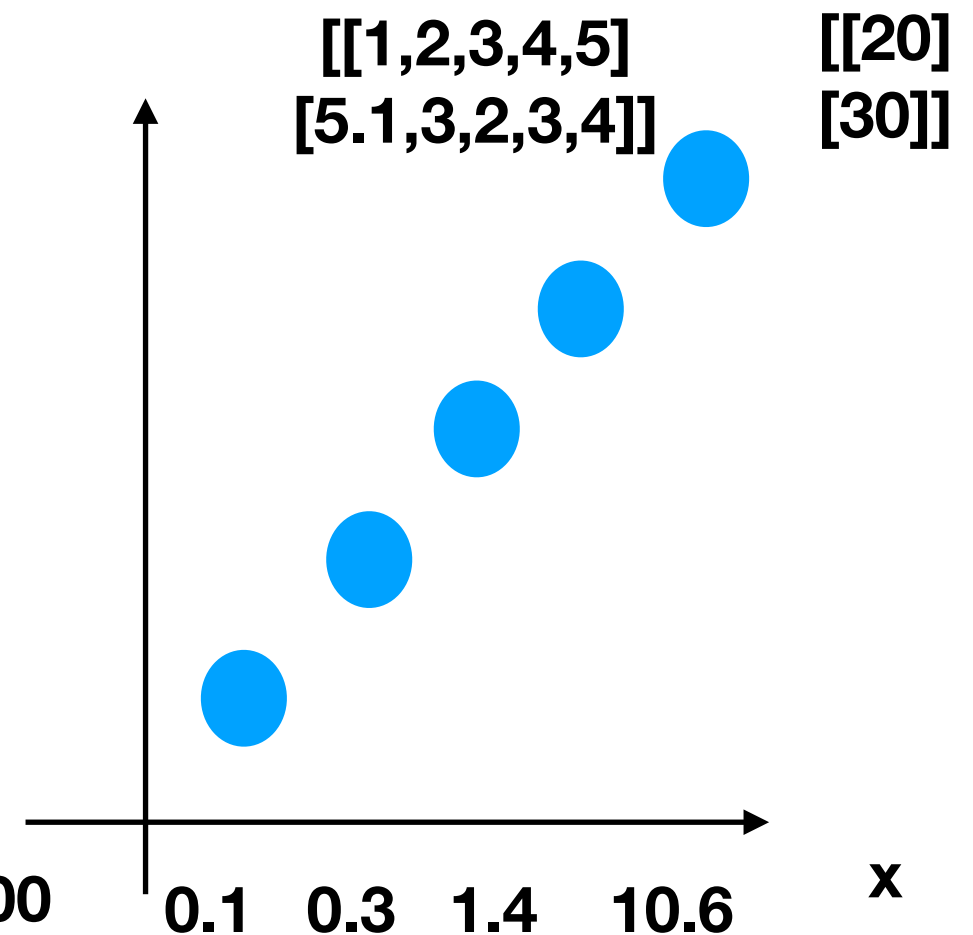
$$y_{predict} = x w + b \text{ (模型参数必须用变量定义)}$$

3、求损失函数，误差

$$\text{loss 均方误差 } (y_1-y_1')^2 ++ (y_{100} - y_{100}')^2/100$$

4、梯度下降去优化损失过程 指定学习率

矩阵相乘: (m行, n 行) * (n行, 1) (m行, 1) + bias



算法	策略	优化
线性回归	均方误差	梯度下降API 学习率

$2^2 = 4$ sumamry:events

保存模型的文件：checkpoint文件 检查点文件

100步	200步
权重，偏置	权重，偏置

model.data-00003-of-00004