[1000, 230, 240, 3]

卷积神经网络：卷积层：32个filter, 观察窗口的大小1*1, 3*3,5*5, 步长1, padding=“1”

tensorflow: padding=“SAME”　　[230, 240 ,32]

激活层：sigmoid,
使用relu

池化层： 2*2 ， 步长2　　池化层： 4*4， 3

4个目标值：”NZPP”　　26中可能性：ABCDEFGHIJKLMNOPQRSTUVWXYZ

5个目标值，数字，小写　　[0, 1, 2, 3, 4, 5, 6, 7, 8,……,25]

N　26种类别：ABCDEFGHIJKLMNOPQRSTU

Z　26种类别：ABCDEFGHIJKLMNOPQRSTU

P　26种类别：ABCDEFGHIJKLMNOPQRSTU

P　26种类别：ABCDEFGHIJKLMNOPQRSTU

手写数字：　输出10个值，softmax　10个类别的概率

[None, 10]

ABCDEFGHIJKLMNOPQRSTUVWXYZ

———>　[0.01,0.002……]

———>　[0.1,0.02……]

———>　[0.1,0.02……]

———>　[0.1,0.02……]

[None, 4 * 26]

One-hot

N　[0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0]

Z　[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1]

P　[0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0]

P　[0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0]

交叉熵损失计算：

N	Z	P	P

目标值：

13 25 15 15

[0,0,0,0,..1,0,0,0,0 0,0,0,0,..0,0,0,..0,1 0,0,0,0,..0,0,1,...0,0 0,0,0,0,..0,0,1,...0,0]

经过网络的输出 4*26=104

[1.2,23,34,10,9.....]

y_predict= [0.01,0.002...0.03 .....] 104 104

[0.12 [0

0.1 0

. .

. .

. .

. .

. .

. .

. .

. .

.] .]

某个样本s = - (y_true log(y_predict))

softmax: e^x/e^x+e^.....)

1*log()+ 1log()+ 1log()+ 1log() = 损失值

104个输出值　softmax 104个概率值

4 * 26

1、处理数据　　图片　　一一对应　　标签文件　　0, “NZPP”　　[13, 25, 15,15]

[20, 80, 3]　　1, “WKHK”　　[23, 16, 13,16]

tfrecords

二、识别验证码

1、从tfrecords读取　每一张图片image, label　　[[13, 25, 15, 15], [19, 23, 20, 16]….]

[100, 20, 80, 3]　　[100, 4]

2、建立模型，直接读取数据输入到模型当中　全连接层

x = [100, 20 * 80 * 3]　　y_predict = [100, 4* 26]

w = [20 * 80 * 3, 4 * 26]　　bias= [4 *26]

3、建立损失， softmax，交叉熵

（1）、先把[100, 4]zhuanhauncheng one_hot编码　—>[100, 4, 26]

3、梯度下降优化

准确率计算：

预测值

概率值

[

13 [[0,0,0,0,0,1,0,0,0.....],

22 [0,0,0,1,0,0,0,0,0.....],

15 [0,0,0,0,0,0,0,0,1.....],

15 [1,0,0,0,0,0,0,0,0.....]]

[[0,0,0,0,0,1,0,0,0.....],
[0,0,0,1,0,0,0,0,0.....],
[0,0,0,0,0,0,0,0,0.....],
[0,0,0,0,0,0,0,0,1.....]]

[[0,0,0,0,0,0,0,0,0.....],
[0,0,0,1,0,0,0,0,0.....],
[0,0,0,0,0,0,0,0,1.....],
[0,0,0,0,0,0,0,0,0.....]]

[[0,0,0,0,0,0,0,0,0.....],
[0,0,0,0,0,0,0,1,0.....],
[0,0,0,0,0,0,0,0,0.....],
[0,0,0,0,0,0,0,0,0.....]]

]

目标值

[

13 [[0,0,0,0,0,1,0,0,0.....],

25 [0,0,0,1,0,0,0,0,0.....],

15 [0,0,0,0,0,0,0,0,1.....],

15 [1,0,0,0,0,0,0,0,0.....]]

[[0,0,0,0,0,1,0,0,0.....],
[0,0,0,1,0,0,0,0,0.....],
[0,0,0,0,0,0,0,0,0.....],
[0,0,0,0,0,0,0,0,1.....]]

[[0,0,0,0,0,0,0,0,0.....],
[0,0,0,1,0,0,0,0,0.....],
[0,0,0,0,0,0,0,0,1.....],
[0,0,0,0,0,0,0,0,0.....]]

[[0,0,0,0,0,0,0,0,0.....],
[0,0,0,0,0,0,0,1,0.....],
[0,0,0,0,0,0,0,0,0.....],
[0,0,0,0,0,0,0,0,0.....]]

]

0 **0 1 2**

[100, 4, 26]

tf.argmax(预测值，2)

1

计算机网络七层模型

应用层　http协议
表示层
会话层
传输层　tcp/udp
网络　ip
数据链路层　ppp
物理层　rj45

RPC框架：远程过程调用

tcp/udp　http2.0

分布式系统：hadoop，　tensorflow
视频会议

对于底层协议的一个封装

解决一些传输错误，同步的问题

1、单机多卡：一台服务器上多台设备(GPU)　　单机：一批次去训练　　GPU > CPU

GPU：一批次的去训练

多机多卡

tensorflow:计算速度提高，设备多
怎么进行分布式?

ps

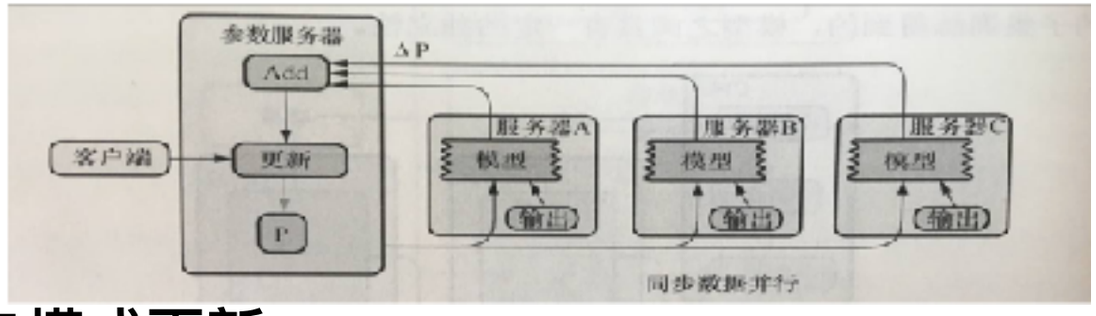
服务器1

参数服务器(parameter server)ps :更新参数，保存参数

工作服务器 worker：主要功能就是去计算

3、分布式更新参数的模式：协调存储，更新参数

（1）同步模式更新



2、集群

worker1

服务器2

gpu1

gpu2

所有worker默认指定
老大

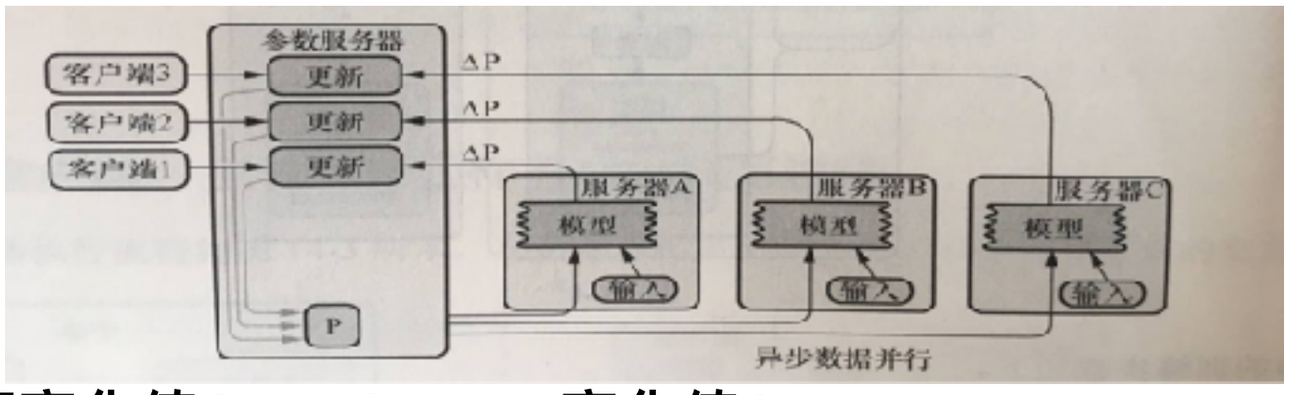
worker2

服务器3

gpu1

gpu2

（2）异步模式更新



创建会话，运行会话，创建文件

计算

worker1: 计算变化值1，　w’ = w - 变化值1

worker2: 计算变化值2，　w” = w - 变化值2　　$w_{-} =$

梯度下降

w’ = w - a(方向) 每批次得出变化量1

w” = w’- a(方向) 每批次得出变化量2

tensorflow:设备命名的规则

/job:ps/task:0
服务器类型 服务器第几台

/job:ps/task:1

参数服务器可以有多台

/job:worker/task:0/cpu:0
/job:worker/task:0/gpu:0
/job:worker/task:0/gpu:1

工作服务器可以有多台

/job:worker/task:1

1、对集群当中的一些ps,worker进行指定

2、创建对应的服务， ps:创建ps服务　　join()
worker创建worker服务　　运行模型，程序，初始化会话等等
指定一个默认的worker（老大）去做，

worker如何使用设备：　1、with tf.device(“/job:worker/task:0/gpu:0”):
计算等等

分布式使用方法：　2、with tf.device(tf.train.replica_device_setter(
worker_device=“/job:worker/task:0/gpu:0”,cluster=cluster)):
计算等等

3、会话　#tf.Session()不支持分布式

分布式会话函数 MonitoredTrainingSession(master=‘’,is_chief=True,checkpoint_dir=None,
hooks=None,config=None)

master:指定运行会话的老大　　task:0
is_chief: true, false　task:0　　task:1不去初始化话等等操作