如何使用机器学习算法优化分发链路

PP云 曾小伟

目录

1. 背景介绍
2. 分发质量的评价模型
3. 机器学习的使用
4. 未来展望
5. Q&A

点播视频源站介绍及分发链路优化的意义

点播视频源站是什么---观看点播的流程

点播视频：连续剧、电影、体育录像、自媒体制作、甚至包含短视频

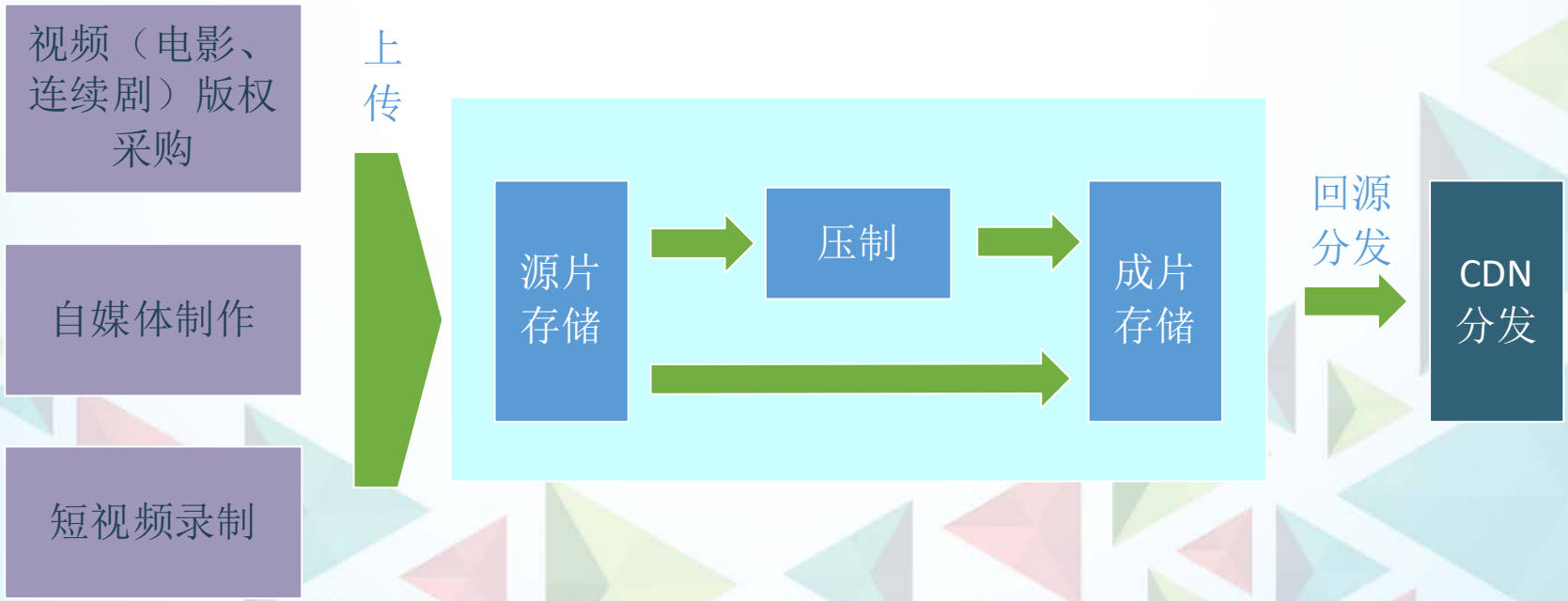
视频网站服务器提供点播观看资源 → 用户使用PC或移动客户端打开点播文件

点播视频源站介绍及分发链路优化的意义

点播视频源站是什么---源站的定义和点播视频处理流程

- 源站就是一个视频网站（或者视频内容提供方）存放自己视频资源，并和CDN分发网络对接的服务器(集群)
- 视频网站有内容（版权），而CDN擅长分发

点播（短视频）的处理流程



点播视频源站介绍及分发链路优化的意义

点播视频源站是什么---源站与CDN的网络拓扑

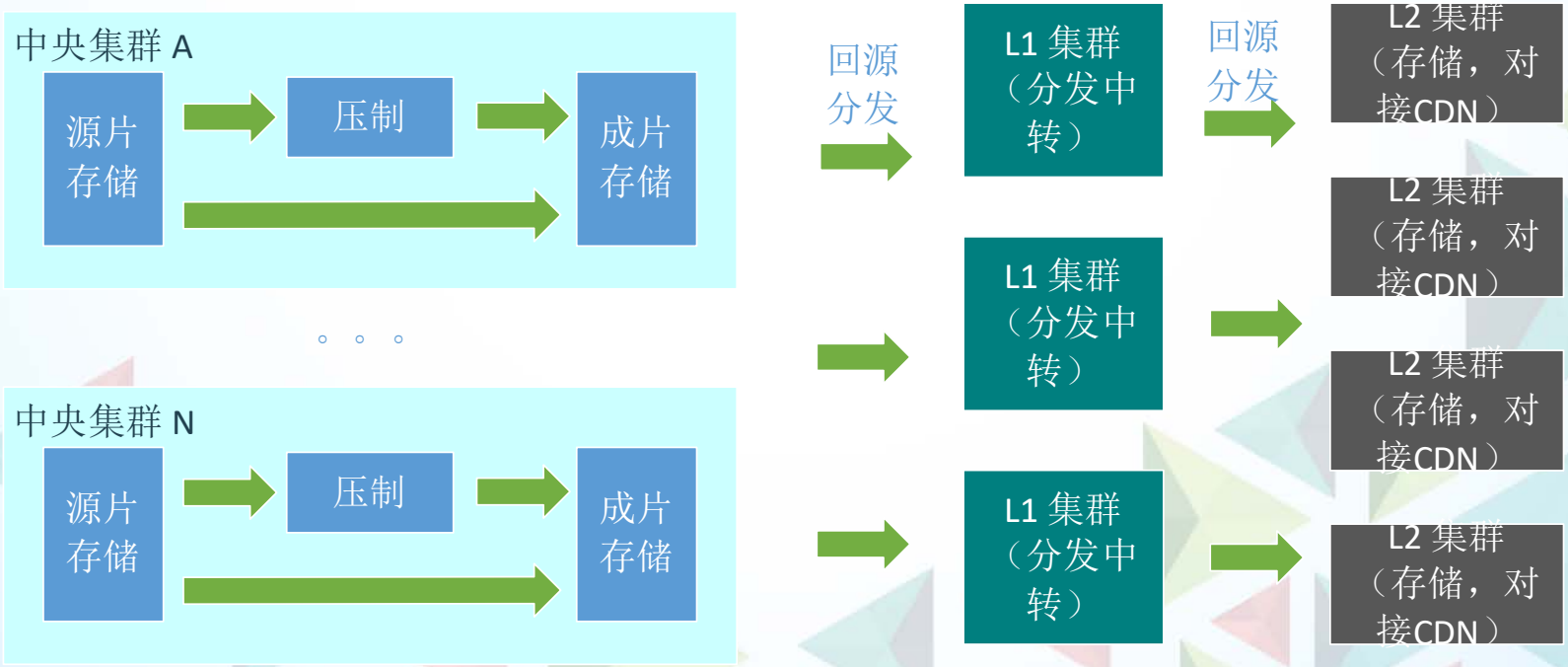
- 一个视频网站的(成片)视频，一般会存放在多地的多个机房
- 一个视频网站（如PPTV）会对接多个CDN，也可能自己建设CDN
- 每个CDN供应商对应的网络接入点位置、线路质量有区别

点播视频源站介绍及分发链路优化的意义

点播视频源站是什么---源站的多级结构和分发链路

- 视频网站的（成片）视频压制成功后，一般在全网只有一份，这一份数据文件需要发送到视频网站的多个机房（L2集群）中，这个过程，叫做源站的内部分发
- 视频源站与CDN的接入节点对接，这个叫做外部分发（不做重点介绍）

视频源站的多级结构



点播视频源站介绍及分发链路优化的意义

点播视频源站分发过程中存在的问题

- 树状分发链路建立的原因（经验、网络特性）、不同线路质量差异大
- 分发过程走的是互联网线路（专线太贵），互联网线路的稳定性不可预期
- 不同集群（机房）服务器的能力的差异，负载不均

济南蓝翔路多家公司网络出现故障,光缆主干线疑被挖断
凤凰网 2017年10月11日 07:26
原标题:济南蓝翔路多家公司网络出现故障,光缆主干线疑被挖断 国庆假期结束,很多人都回到单位开始了忙碌 原标题:济南蓝翔路多家公司网络出现故障,光缆主干线疑被...
查看更多相关新闻>> - 百度快照

小区通信光缆被挖断 居民表示没网生活不方便
网易河南站 2017年11月27日 14:33
对方表示小区网络故障是因为光缆被挖断了。"维修很快,关键是排查断点,排查需要一些时间。目前,维修师傅正在排查断点,争取尽快恢复。"该工作人员说。 本文来源:淇河... 百度快照

地铁施工挖断光缆 粤湘多地网络异常
网易 2017年03月31日 06:10
中国电信湖南分公司的官方微博"中国电信湖南客服"也在昨日14:11发布微博称,广州从化钟落潭挖地铁,把中国电信京汉广大长途干线光缆全挖断了。受故障影响,湖南省内...
查看更多相关新闻>> - 百度快照

广州地铁施工挖断光缆,多个地方无法上网电话不通
奥一网 2017年03月30日 22:13
广州从化钟落潭挖地铁,把中国电信京汉广大长途干线光缆全挖断。... "广州从化钟落潭挖地铁,把中国电信京汉广大长途干线光缆全挖断了,受故障影响,我省手机、宽带出... 百度快照

黑客武库升级DDoS电磁炮 威力已不止瘫痪美国半张互联网

凤凰科技 2018年03月03日 00:07
原标题:黑客武库升级DDoS电磁炮 威力已不止瘫痪美国半张互联网 DDoS攻击一直是黑客们钟爱的武器 原标题:黑客武库升级DDoS电磁炮 威力已不止瘫痪美国半张互联网 ... 查看更多相关新闻>> - 百度快照

平昌冬奥会是怎样被黑客入侵的?
中国教育 2018年04月10日 15:08
以史为镜,看看2018年平昌冬奥会的安全防线,是怎样被黑客攻破的?... 平昌冬奥会惨遭黑客攻击2018年平昌冬奥会终于落...比赛场馆附近网络瘫痪,观赛门票无法打印导致观众...
查看更多相关新闻>> - 百度快照

黑客攻击俄罗斯伊朗等多地网络 放话"不要干扰我们的选举"

新浪新闻 2018年04月09日 14:12
处7日发布通告称,不明黑客攻击了美国思科公司生产的交换机,导致互联网出现大...报道称,攻击导致数据中心整体瘫痪,许多著名网站无法打开,包括《丰坦卡报》、《共青团... 查看更多相关新闻>> - 百度快照

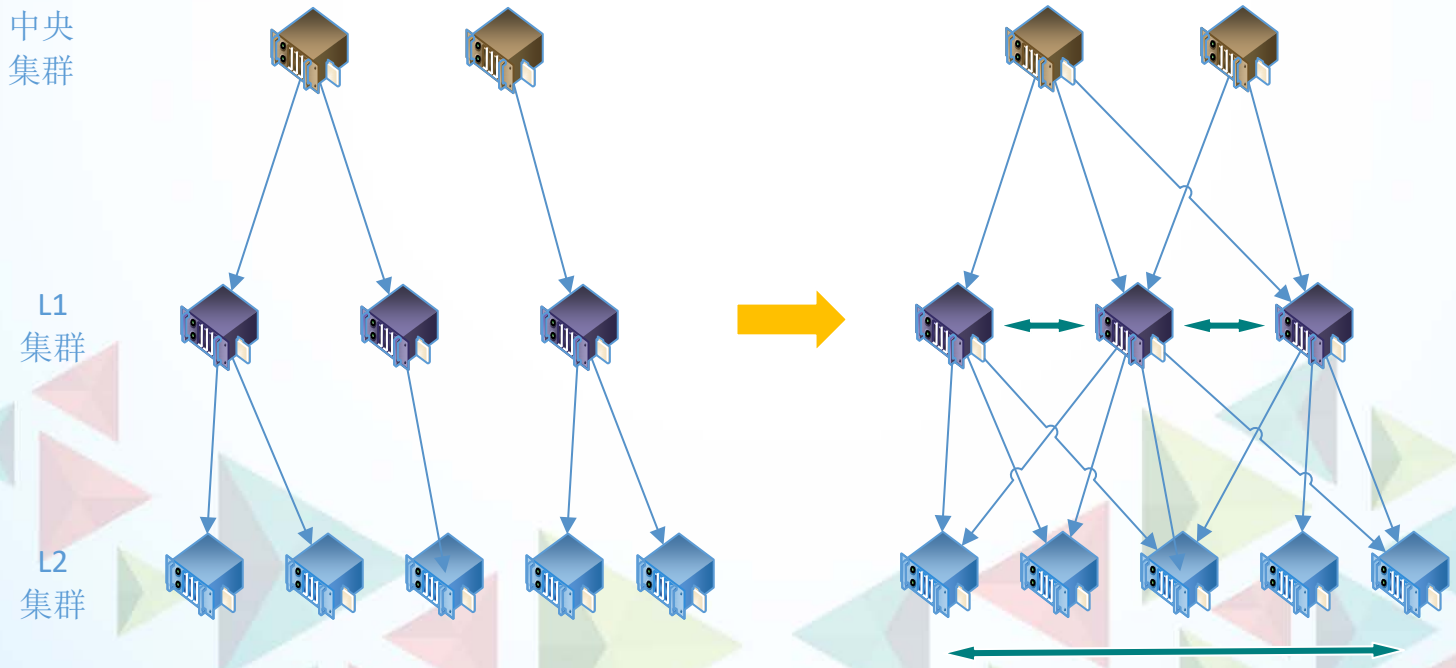
黑客攻击美国思科交换机 互联网出现大面积故障 这段代码高了

中华网 2018年04月10日 14:32
处7日发布通告称,不明黑客攻击了美国思科公司生产的交换机,导致互联网出现大...报道称,攻击导致数据中心整体瘫痪,许多著名网站无法打开,包括《丰坦卡报》、《共青团... 百度快照

点播视频源站介绍及分发链路优化的意义

点播视频源站分发链路优化的意义

- 如何解决上面说的问题呢，我们自然的会想到，如果每个文件的分发过程，都能自动选择一个最优秀的链路，而不是根据那个配置死的回源树，那么分发的过程应该会高效和稳定很多
- 同时，优化点播视频的分发链路，也将在互联网线路出现问题时，避免区域性的故障
- 最后，不同集群（机房）的负载也将更加合理和平均

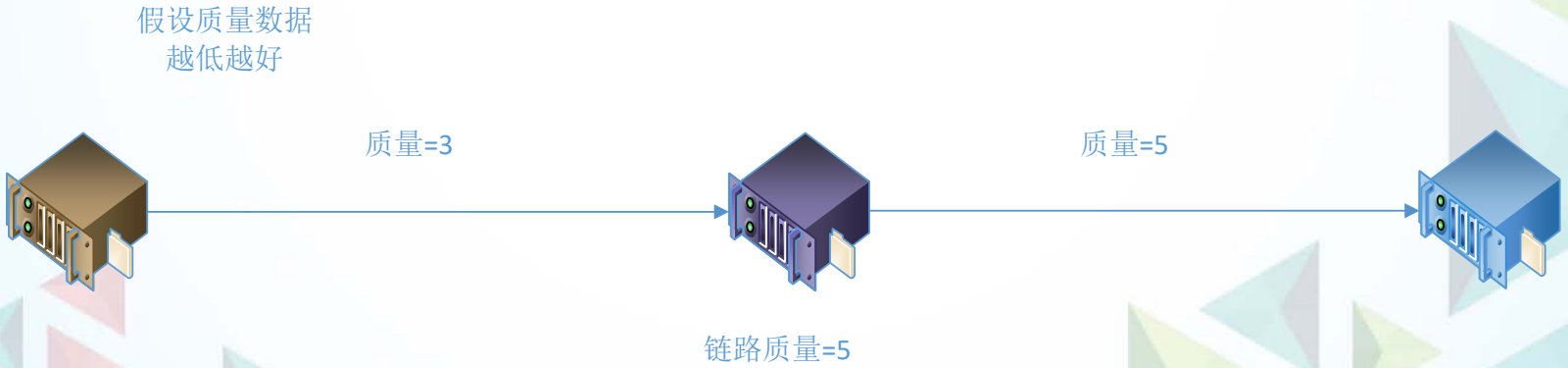


节点间数据分发质量的评估

前面提到了优化的办法是选择好的分发链路，那么，到底怎样分发链路才是一个好的链路呢？我们先来看以下事实：

- 链路是由数据通过的服务器节点构成的
- 链路中，相邻的服务器节点传输质量的最差的值，决定这条分发链路的质量上限

所以，我们首先研究两个服务器节点间传输质量的情况



节点间数据分发质量的评估

两个节点间数据传输质量的评估

用什么数据做质量指标呢，一个简单的想法就是文件的下载耗时

影响两个服务器节点间的传输质量（下载耗时）的因素有这些：

- 文件大小
- 服务器之间网络线路的情况，包括数据延迟、丢包率、跃点数等等
- 发送服务器（接收服务器）的当前负载，包括CPU负载，内存用量，IO负载，当前带宽，存储用量
- 当前时间

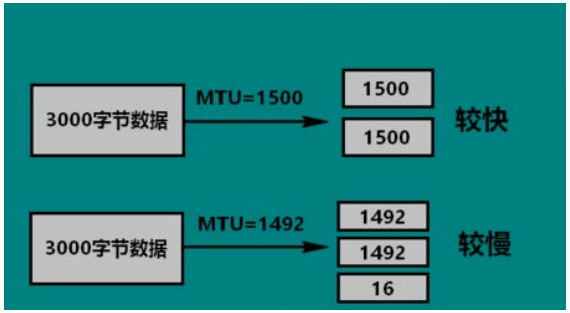
我们接下来会简单分析，这些因素会对下载有什么影响

节点间数据分发质量的评估

两个节点间数据传输质量的评估---影响下载时间的因素分析

- 文件大小：

文件越大，下载越慢，耗时越长

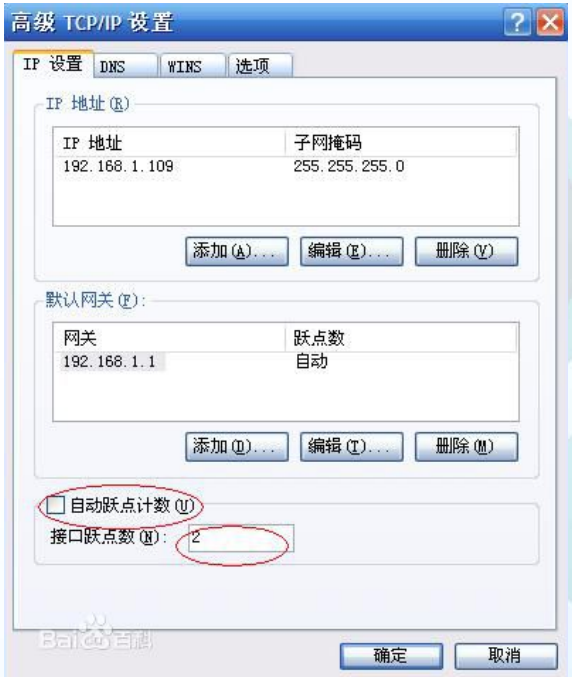


- 服务器之间网络线路的情况

两个服务器之间的网络情况，具体有这些指标

✓ 延迟
✓ MTU（Maximum Transmission Unit）
✓ 丢包率
✓ 跃点数
✓ 服务器的最大带宽

```
正在 Ping pptv.com [180.153.106.86] 具有 32 字节的数据:
来自 180.153.106.86 的回复: 字节=32 时间=6ms TTL=54
来自 180.153.106.86 的回复: 字节=32 时间=2ms TTL=54
来自 180.153.106.86 的回复: 字节=32 时间=3ms TTL=54
来自 180.153.106.86 的回复: 字节=32 时间=3ms TTL=54
来自 180.153.106.86 的回复: 字节=32 时间=2ms TTL=54
来自 180.153.106.86 的回复: 字节=32 时间=3ms TTL=54

180.153.106.86 的 Ping 统计信息:
    数据包: 已发送 = 6, 已接收 = 6, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 2ms, 最长 = 6ms, 平均 = 3ms
```



节点间数据分发质量的评估

两个节点间数据传输质量的评估---影响下载时间的因素分析

- 发送服务器（接收服务器）的当前负载情况

✓ CPU负载
✓ 内存用量
✓ IO负载
✓ 当前带宽使用情况

```
top - 15:46:04 up 407 days, 20:42,  1 user,  load average: 0.00, 0.01, 0.05
Tasks: 242 total,   1 running, 241 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.1 sy,  0.0 ni, 99.9 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  1874112 total,   690704 free,   176872 used,  1006536 buff/cache
KiB Swap:  2097148 total,  2097148 free,        0 used.  1406492 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
48860 root      20   0  146316   2156   1384 R   0.3  0.1   0:00.13 top
    1 root      20   0   44572   7052   2424 S   0.0  0.4   6:47.95 systemd
    2 root      20   0       0      0      0 S   0.0  0.0   0:00.47 kthreadd
    3 root      20   0       0      0      0 S   0.0  0.0   0:09.30 ksoftirqd/0
    5 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/0:0H
    7 root      rt   0       0      0      0 S   0.0  0.0   0:00.67 migration/0
```

- 当前时间：

以上提到的数据，都是随着时间抖动的

这里要提一个问题，正是由于相同尺寸、相同链路的下载速度，在不同时间点的表现差异巨大，才需要引入一个动态的预估机制。

节点间数据分发质量的评估

建立两个节点间数据传输质量的评估模型的设想

度量两个节点间的分发质量，我们可以用一个数值，下载时长（Download Time，缩写为DT）来表示，这个数据受到很多具体的、随时间呈现一定规律的因素（变量）影响，那么我们可以有一个美好的设想：用一个模型，或者说是一个函数，来描述这些因素与DT间的关系，后续新的文件下载时，使用这个模型，输入当前这些变量，预测文件下载的耗时。

假设函数如下：

DT = Func(file_size,current_time,defer,cpu_load,mem_load，io_load, …….)

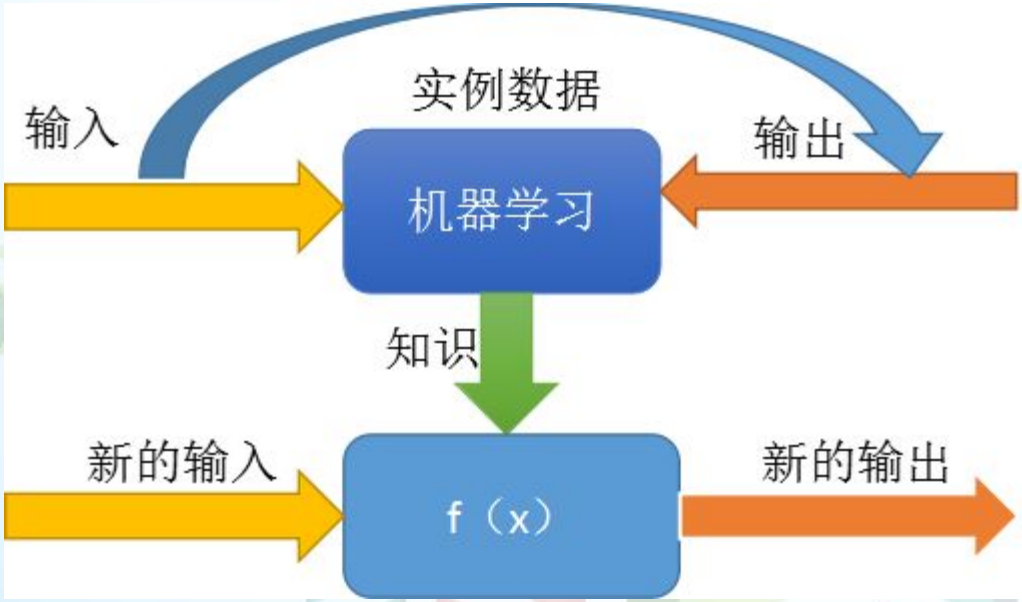
我们知道一个文件大小为100M，当前时间点已知，节点间这些变量已知，我们可以根据这个函数算出时长来。

机器学习算法的设计和实现

前面提出了建立一个模型（函数）的设想，来预测DT，这个模型如何建立呢

- 数学上，模型建立叫做曲线拟合，不同变量个数，不同维度有不同方法
- 在计算机科学的范畴内，AI研究的先驱者，提出了利用神经网络做机器学习的办法，来处理这个问题。

机器学习是什么呢，现在网上有很多解释，我们这里简单来说明下：



模拟或实现人类的学习行为，以获取新的知识或技能，重新组织已有的知识结构使之不断改善自身的性能

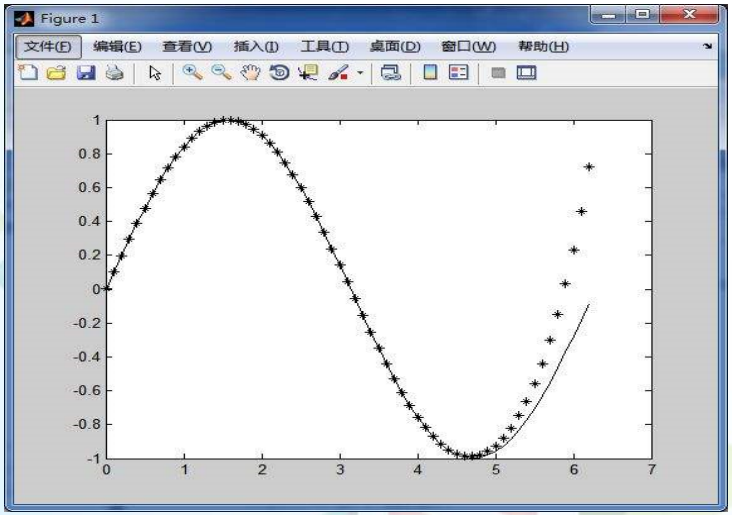
机器学习算法的设计和实现

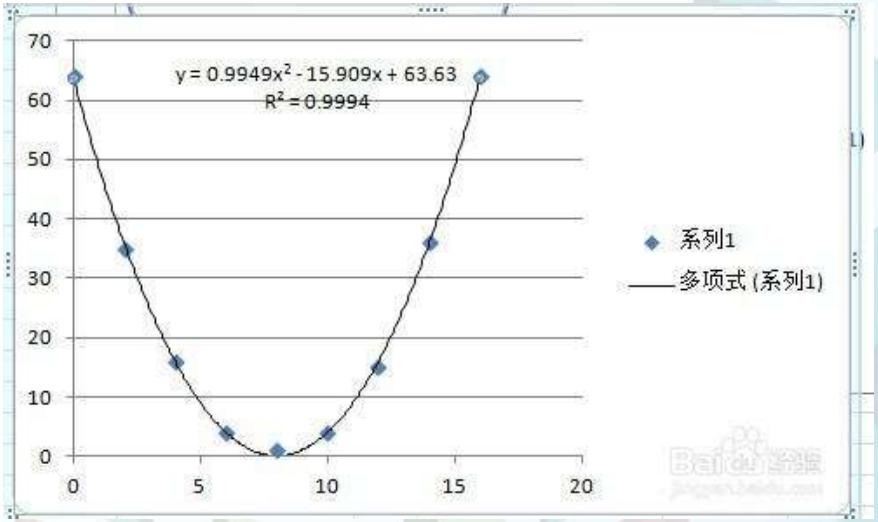
机器学习的实现

机器学习的本质就是让机器根据已有的数据，去分析出一个模型来表示隐藏在这些数据背后的规律（函数）。

我们先说简单的建立模型的办法：数学上拟合采用的函数：

- 选取拟合函数（幂函数，指数函数，对数函数，三角函数等）
- 设定参数
- 比较误差，调节参数
- 迭代





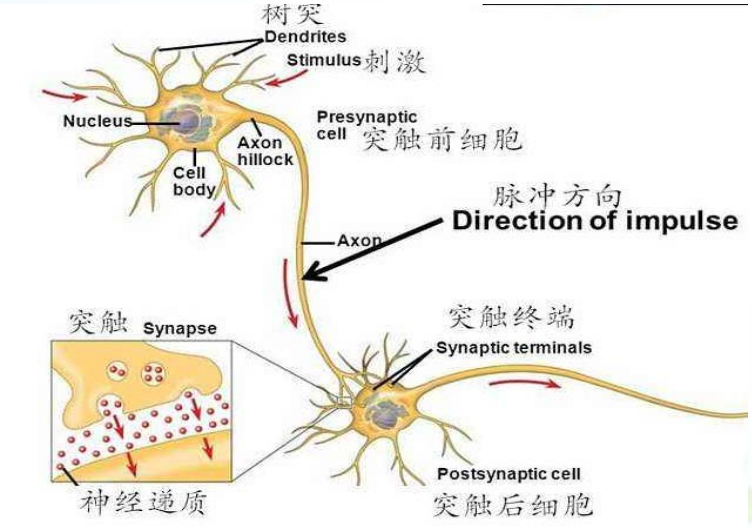
机器学习算法的设计和实现

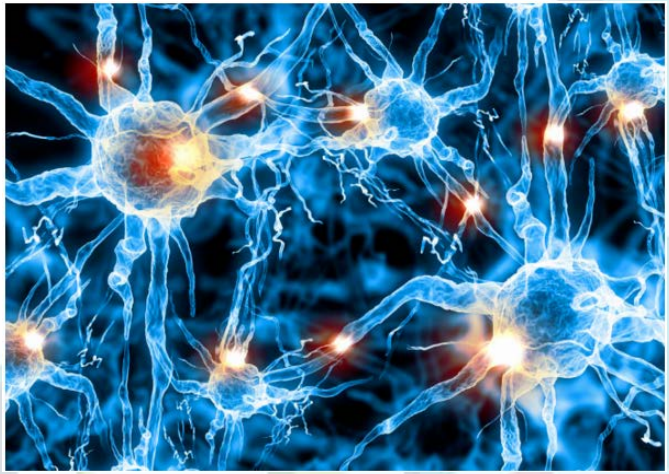
机器学习的实现 --- 神经网络的概念和引入

和数学领域一样，AI领域里，输入通过一个模型（函数）变成输出，这个模型（函数）靠什么确定呢，靠猜！

猜不可怕，问题是如何猜的更准？

AI的先驱提出了模拟人类大脑神经元的思路，来处理如何猜模型的问题。

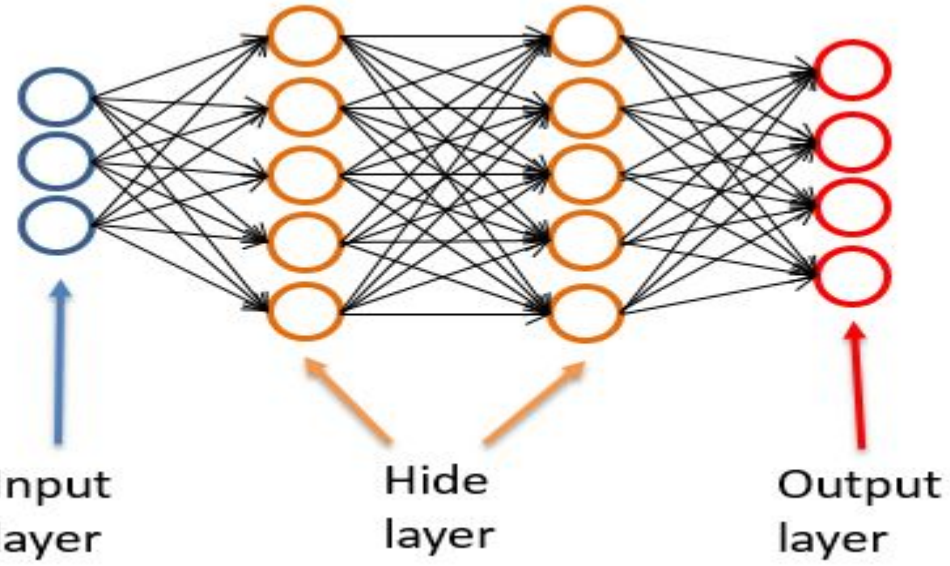




机器学习算法的设计和实现

机器学习的实现 --- 人工神经网络的概念

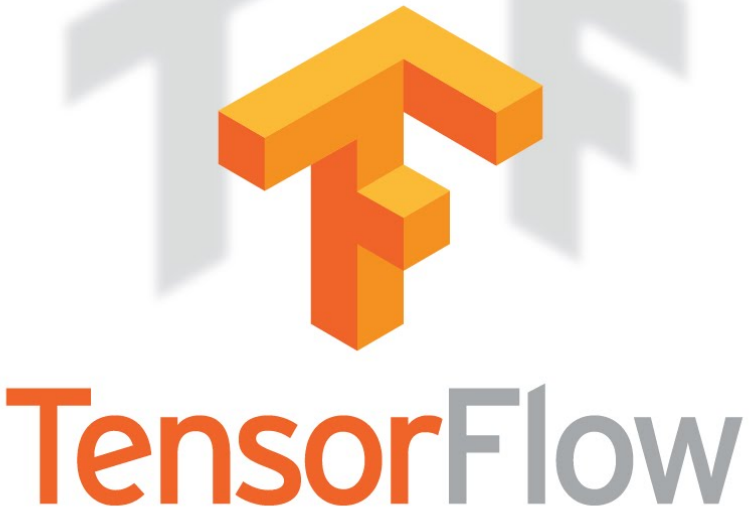
人工神经网络就是，人为的构造出一些处理节点（模拟脑神经元），每个节点有个函数，处理几个输入，生成若干输出，每个节点与其它节点组合，综合成一个模型（函数）。



机器学习算法的设计和实现

TensorFlow的介绍

Tensorflow是一个通过神经网络实现深度学习的平台，我们给它输入，输出，约定一些（猜测模型）的规则，让他去帮我们猜测具体模型。



机器学习算法的设计和实现

使用Tensorflow --- 训练数据的收集：

- 下载的文件大小：可直接记录
- 当前时间：使用unix timestamp
- 网络情况的统计：来自一些测速工具
- 发送（接收）服务器的负载情况：来自zabbix的记录
- 下载结果：下载记录，离散化处理

机器学习算法的设计和实现

Zabbix 的介绍：

- 基于WEB界面的提供分布式系统监视以及网络监视功能的企业级的开源解决方案
- zabbix能监视各种网络参数，保证服务器系统的安全运营；并提供灵活的通知机制以让系统管理员快速定位/解决存在的各种问题
- zabbix server与可选组件zabbix agent
- 支持Linux，Solaris，HP-UX，AIX，Free BSD，Open BSD，OS X

机器学习算法的设计和实现

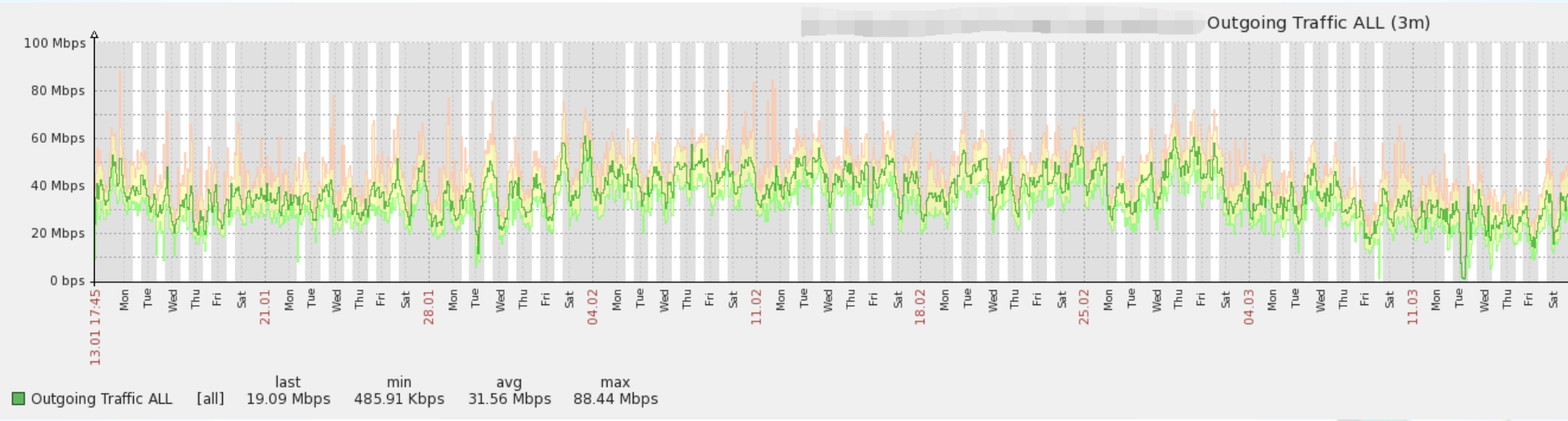
Zabbix 的介绍：

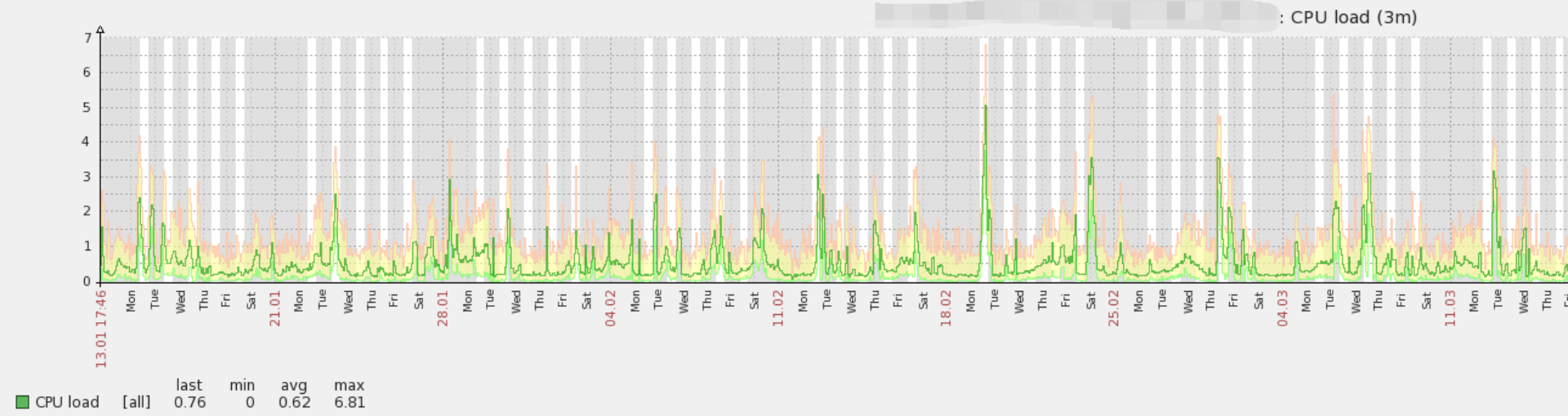
下图是zabbix对于当前带宽的统计情况

Incoming Packets eth0	13 Apr 2018 17:29:20	33	-3	Graph
Incoming Packets eth1	13 Apr 2018 17:29:19	2170	+7	Graph
Incoming Traffic ALL	13 Apr 2018 17:29:12	20.9 Mbps	-341.71 Kbps	Graph
Incoming Traffic eth0	13 Apr 2018 17:28:29	16.66 Kbps	-2.48 Kbps	Graph
Incoming Traffic eth1	13 Apr 2018 17:28:28	21.27 Mbps	+989.58 Kbps	Graph
Interface IP Address ALL	13 Apr 2018 16:22:56		-	History
Interface Mac Address ALL	12 Apr 2018 22:50:22	eth1@ ..	-	History
Interface Speed eth0	13 Apr 2018 17:05:33	1 Gbps	-	Graph
Interface Speed eth1	13 Apr 2018 17:05:32	1 Gbps	-	Graph
Network TCP Connection Count [estab]	13 Apr 2018 17:21:46	91	-	Graph
Outgoing Packets eth0	13 Apr 2018 17:28:31	1023	+45	Graph
Outgoing Packets eth1	13 Apr 2018 17:28:30	955	+88	Graph
Outgoing Traffic ALL	13 Apr 2018 17:29:13	20.04 Mbps	+236.42 Kbps	Graph
Outgoing Traffic eth0	13 Apr 2018 17:28:29	10.55 Mbps	+130.74 Kbps	Graph
Outgoing Traffic eth1	13 Apr 2018 17:28:28	9.7 Mbps	+1.32 Mbps	Graph

机器学习算法的设计和实现

Zabbix 的介绍（3个月的带宽负载与cpu负载）：





机器学习算法的设计和实现

网络数据的收集 --- Iperf

- 带宽
- 丢包率
- MSS/MTU
- TCP/UDP支持

```
------------------------------------------------------------
Server listening on 80
------------------------------------------------------------
Accepted connection from 117.135.159.30, port 11224
[  6] local 139.129.24.132 port 80 connected to 117.135.159.30 port 11225
[  8] local 139.129.24.132 port 80 connected to 117.135.159.30 port 11226
[ 10] local 139.129.24.132 port 80 connected to 117.135.159.30 port 11227
[ ID] Interval           Transfer     Bandwidth
[  6]   0.00-1.00   sec   301 KBytes  2.46 Mbits/sec
[  8]   0.00-1.00   sec   346 KBytes  2.84 Mbits/sec
[ 10]   0.00-1.00   sec   467 KBytes  3.82 Mbits/sec
[SUM]   0.00-1.00   sec  1.09 MBytes  9.12 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[  6]   1.00-2.00   sec   548 KBytes  4.49 Mbits/sec
[  8]   1.00-2.00   sec   174 KBytes  1.43 Mbits/sec
[ 10]   1.00-2.00   sec  75.5 KBytes   618 Kbits/sec
[SUM]   1.00-2.00   sec   798 KBytes  6.54 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[  6]   2.00-3.00   sec   230 KBytes  1.89 Mbits/sec
[  8]   2.00-3.00   sec   265 KBytes  2.17 Mbits/sec
[ 10]   2.00-3.00   sec  81.8 KBytes   670 Kbits/sec
[SUM]   2.00-3.00   sec   577 KBytes  4.73 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[  6]   3.00-4.00   sec   243 KBytes  1.99 Mbits/sec
[  8]   3.00-4.00   sec  52.8 KBytes   433 Kbits/sec
[ 10]   3.00-4.00   sec   278 KBytes  2.28 Mbits/sec
[SUM]   3.00-4.00   sec   574 KBytes  4.70 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[  6]   4.00-5.00   sec   194 KBytes  1.59 Mbits/sec
[  8]   4.00-5.00   sec   103 KBytes   845 Kbits/sec
[ 10]   4.00-5.00   sec   282 KBytes  2.31 Mbits/sec
[SUM]   4.00-5.00   sec   579 KBytes  4.74 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[  6]   5.00-6.00   sec   218 KBytes  1.78 Mbits/sec
[  8]   5.00-6.00   sec  75.5 KBytes   618 Kbits/sec
[ 10]   5.00-6.00   sec   284 KBytes  2.33 Mbits/sec
[SUM]   5.00-6.00   sec   577 KBytes  4.73 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[  6]   6.00-6.02   sec  5.03 KBytes  2.14 Mbits/sec
[  8]   6.00-6.02   sec  0.00 Bytes  0.00 bits/sec
[ 10]   6.00-6.02   sec  8.80 KBytes  3.75 Mbits/sec
[SUM]   6.00-6.02   sec  13.8 KBytes  5.89 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[  6]   0.00-6.02   sec  0.00 Bytes  0.00 bits/sec                  sender
[  6]   0.00-6.02   sec  1.70 MBytes  2.37 Mbits/sec                receiver
[  8]   0.00-6.02   sec  0.00 Bytes  0.00 bits/sec                  sender
[  8]   0.00-6.02   sec  1018 KBytes  1.38 Mbits/sec                receiver
[ 10]   0.00-6.02   sec  0.00 Bytes  0.00 bits/sec                  sender
[ 10]   0.00-6.02   sec  1.44 MBytes  2.01 Mbits/sec                receiver
[SUM]   0.00-6.02   sec  0.00 Bytes  0.00 bits/sec                  sender
[SUM]   0.00-6.02   sec  4.13 MBytes  5.76 Mbits/sec                receiver
```

机器学习算法的设计和实现

网络数据的收集 --- Ping&Traceroute

- 延迟
- 连通性

```
来自 10.200.47.117 的回复: 字节=32 时间=2ms TTL=59
来自 10.200.47.117 的回复: 字节=32 时间=3ms TTL=59
来自 10.200.47.117 的回复: 字节=32 时间=9ms TTL=59
来自 10.200.47.117 的回复: 字节=32 时间=2ms TTL=59
来自 10.200.47.117 的回复: 字节=32 时间=3ms TTL=59
来自 10.200.47.117 的回复: 字节=32 时间=4ms TTL=59
来自 10.200.47.117 的回复: 字节=32 时间=5ms TTL=59
来自 10.200.47.117 的回复: 字节=32 时间=6ms TTL=59
来自 10.200.47.117 的回复: 字节=32 时间=8ms TTL=59
来自 10.200.47.117 的回复: 字节=32 时间=13ms TTL=59
来自 10.200.47.117 的回复: 字节=32 时间=8ms TTL=59
来自 10.200.47.117 的回复: 字节=32 时间=6ms TTL=59
来自 10.200.47.117 的回复: 字节=32 时间=8ms TTL=59
来自 10.200.47.117 的回复: 字节=32 时间=3ms TTL=59
来自 10.200.47.117 的回复: 字节=32 时间=7ms TTL=59
来自 10.200.47.117 的回复: 字节=32 时间=13ms TTL=59
来自 10.200.47.117 的回复: 字节=32 时间=7ms TTL=59

10.200.47.117 的 Ping 统计信息:
    数据包: 已发送 = 48, 已接收 = 48, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 2ms, 最长 = 13ms, 平均 = 4ms
```

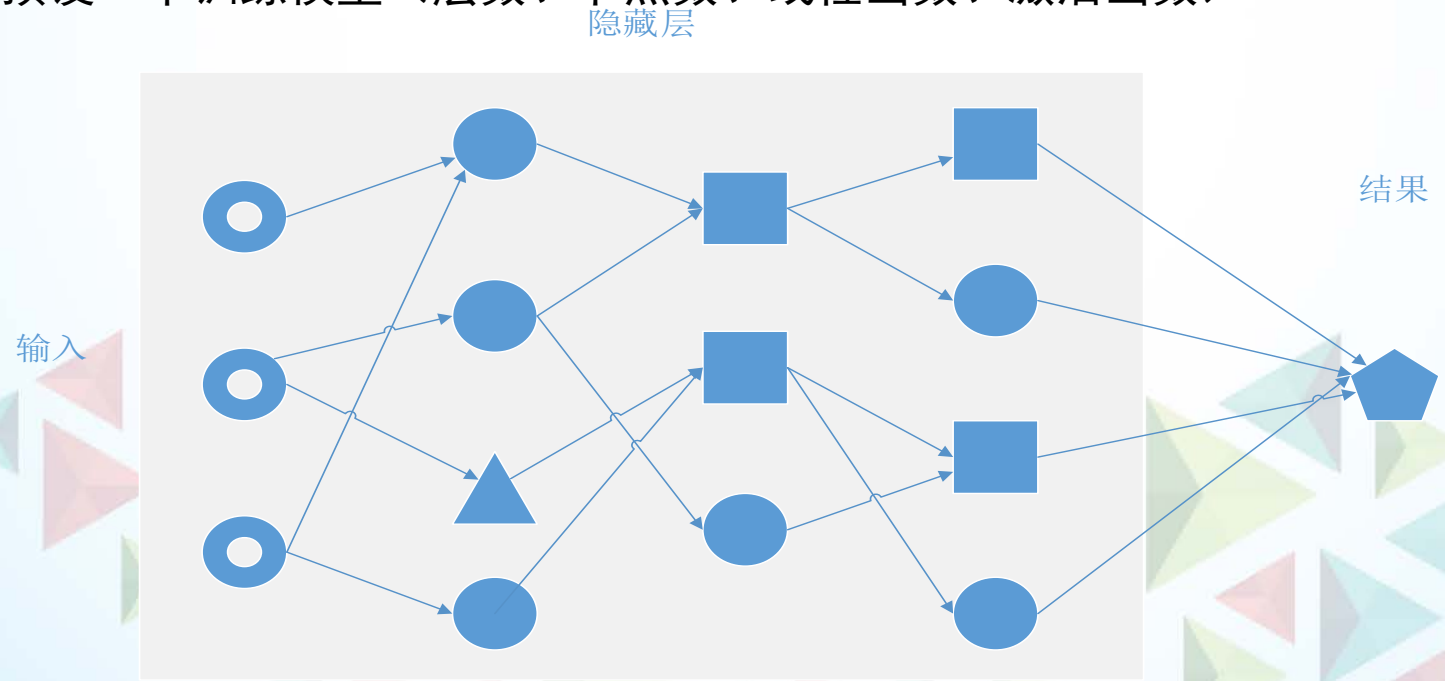
- 跃点数

```
c:\>tracert 10.200.47.117

通过最多 30 个跃点跟踪到 10.200.47.117 的路由

  1    <1 毫秒    <1 毫秒    <1 毫秒 10.200.120.1
  2     1 ms      1 ms       1 ms  10.200.206.1
  3    <1 毫秒    <1 毫秒    <1 毫秒 10.200.254.49
  4     3 ms      5 ms       3 ms  172.111.111.1
  5     5 ms      7 ms       7 ms  10.200.0.11
  6     3 ms      2 ms       2 ms  10.200.47.117

跟踪完成。
```

机器学习算法的设计和实现

数组的组合与准备

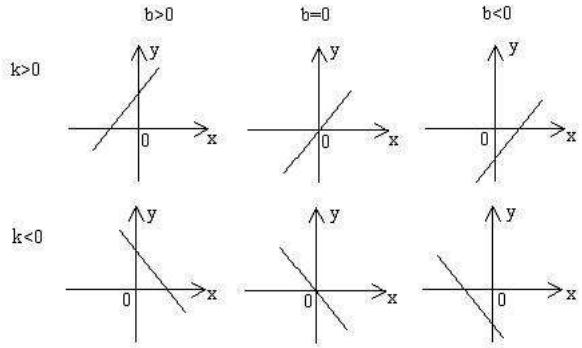
- 将网络数据，负载数据，文件大小，时间，格式化后组合成一条json
- 将下载耗时数据分段离散化（why？），作为训练的结果
- 预设一个训练模型（层数、节点数、线性函数、激活函数）

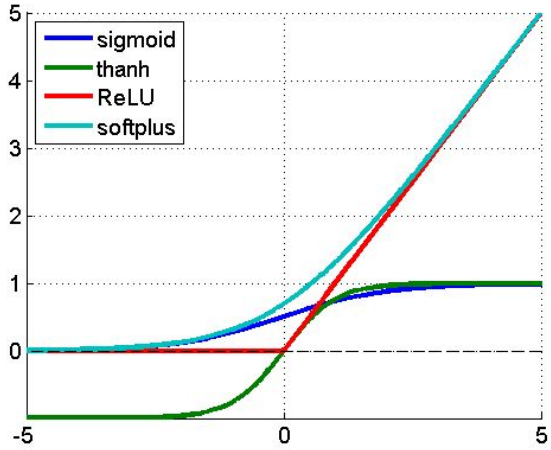


机器学习算法的设计和实现

神经网络参数的调节与优化：

- 初始学习率
- 学习率衰减率
- 隐藏层节点数量
- 迭代轮数
- 正则化系数
- 滑动平均衰减率
- 批训练数量
- 线性函数、激活函数设置

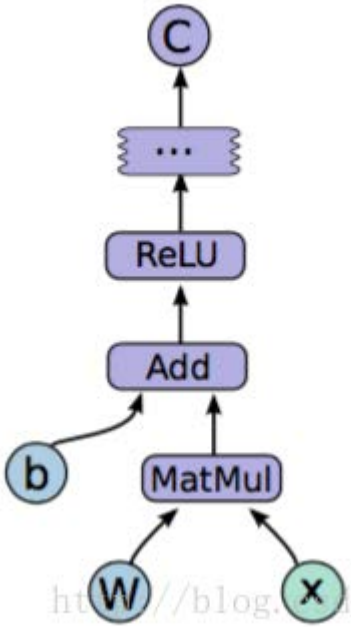




机器学习算法的设计和实现

TensorFlow的具体处理

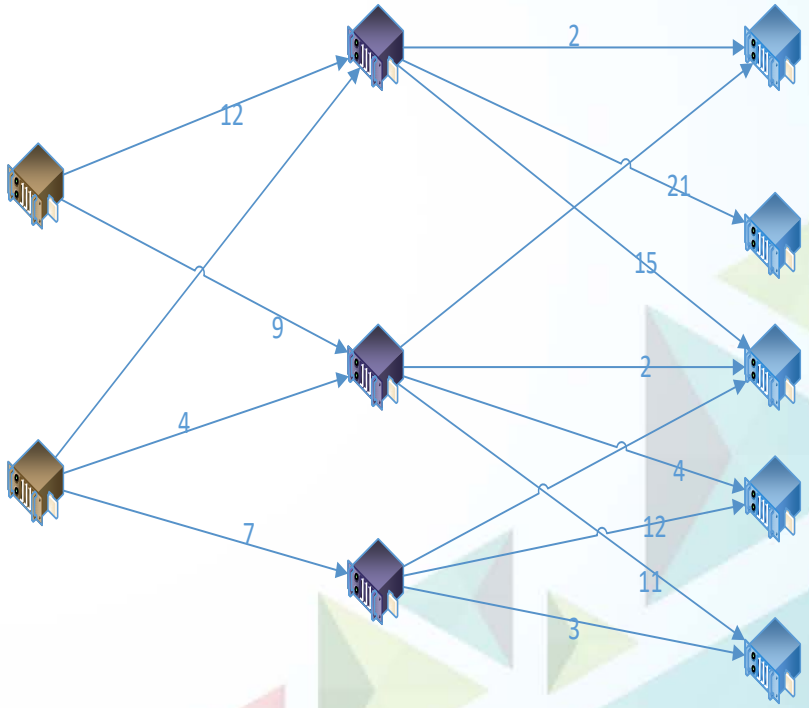
- 数据输入
- 训练
- 训练结果审查
- 调节参数
- 重新训练
- 训练结果审查
- ……
- 训练结果满意



机器学习算法的设计和实现

通过Tensorflow的结果，计算最优传输路径

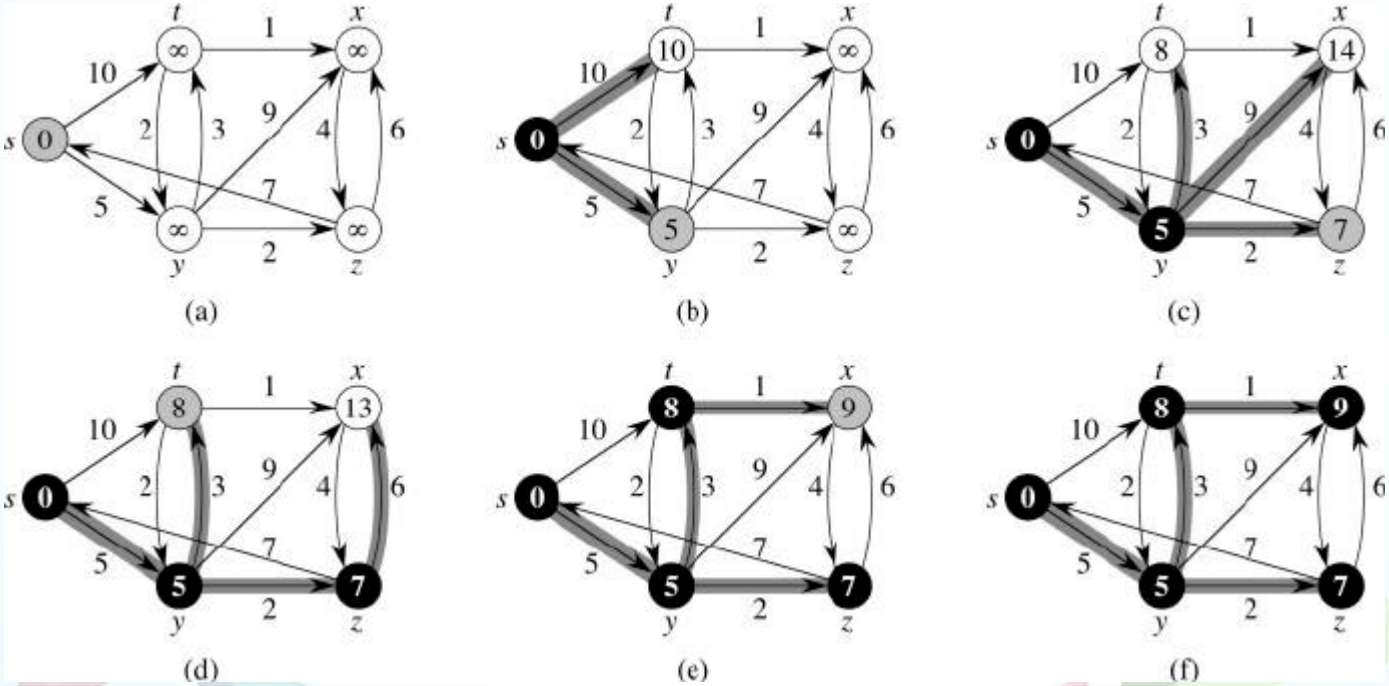
根据训练出来的模型，可以针对当前时间，特定大小的文件，计算出各个节点间的传输耗时



机器学习算法的设计和实现

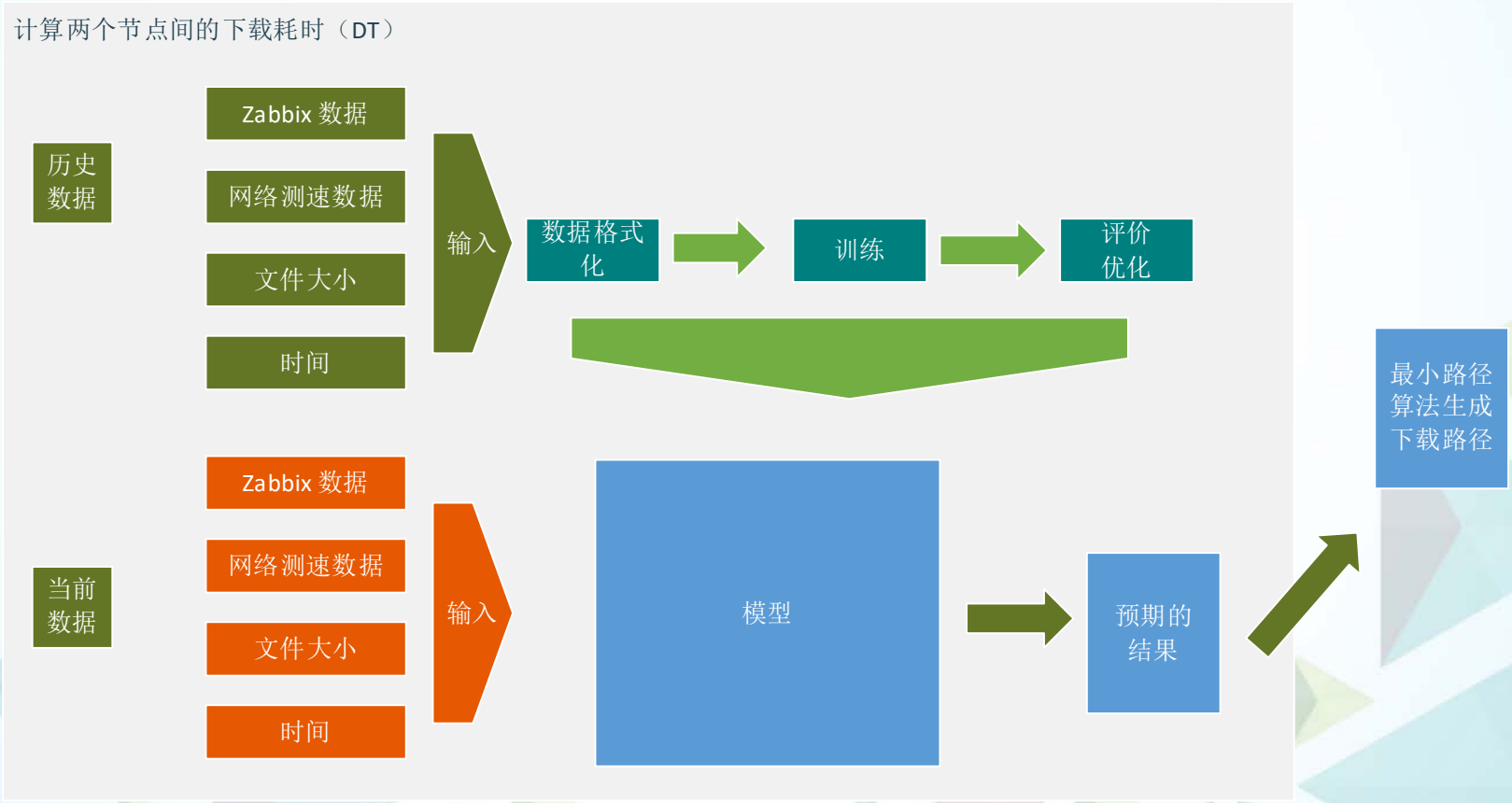
通过Tensorflow的结果，计算最优传输路径

使用最短路径算法（Dijkstra算法 翻译为 迪杰斯特拉算法），生成最优（次优）线路



机器学习算法的设计和实现

整体流程



未来发展的展望

- 将文件分片，使用最大流算法进行分片下载
  机房的带宽已经购买了，用满才划算
  EK 算法，Dinic算法等

- 最小费用最大流
  （在最大流有多组解时，给每条边在附上一个单位费用的量，问在满足最大流时的最小费用是多少）
  不同机房的建设时，成本不一致，可以在机房后期扩容时提供指导

- 对直播回源链路和点播分段实时回源的优化
  直播与点播的区别，使用方案时要注意的情况

Q&A

交流时间

Thank You