

我这样来设计 6 代监控

本文是在不直接泄露某些信息的条件下,做一次如何设计的技术回忆.请大家自己分析和反推.全文属实并兑现,不是吹嘘.

AI 可以是战略级技术吗?

只有让对手无从下手,或则需要很多时间才能复制,例如 5-10 年,这种技术才可以做战略技术.

AI 建模:钱管够,人类的心理可以变态到为技术而活,用 1-2 年时间可以复制任何先进建模技术.例如用 nnapi,opencl 来代替 cuda 建模,使用 opencl 或自定义框架来复制 openai 的 transform 流程.

大数据+GPU 算力:钱管够,可在市场中明码实价交易达成,且不需要时间缓冲,可在急需时用钱达成.

AI 算法:与不可战胜的专制化商业帝国竞争,弱势方选择开源,暴露商业帝国的技术秘密,降低商业帝国统治力.在微软 vs google 进程中,AI 作为新方向一直都是竞争焦点,世界因此而获益:大量开源算法,新的应用点被发掘出来,大量行业被重新设定运行模式.我使用法估价模型对比招聘信息来判断上市公司研究院的真实技术实力,以此判断目标 AI 公司专业程度.

资源引用型 AI 算法定价模型= $((\text{算法员数量}/\text{目标算法的人头普及程度}) * \text{算法员月收入} * 12 \text{ 月执行周期} + (\text{市场投放运行估价} * \text{算法含金占比} 0.0001)) * \text{风险系数} 0.1$

独一无二创造型 AI 算法定价模型= $(\text{算法员数量} * \text{算法员月收入} * 12 \text{ 月执行周期} + (\text{市场投放运行估价} * \text{算法含金占比} 0.0001)) * \text{风险系数} 0.1$ -新点子被 google 流派开源并造成的破坏估值(根据市场定律,目标领域存在热度会在 1 年内出现开源对手,算法几乎不具备复制时间门槛,多花心思都能想到 solve)

这是我个人设计的算法估值模型,用于估算上市公司研究院的真实实力.当我得到了目标领域对手的数据以后,能大致推算出复合项目+算法的战略价值.我会把这些东西,诚实的告诉甲方,他们会在项目执行中,自己惦记这些机会和风险:甲方工作从头到尾都非常努力.

AI 算法是不可以作为核心战略技术,在 6 代监控,它只能影响一定的技术战略价值权重.不能是因为 AI 有不确定性,就用别的领域技术来压制 AI,需要资金,市场,技术用一种淡定的心来平衡对待.

核心技术战略需要放在项目整体的推进能力上,局部的方案和技术,无法形成整体力量,而整体力量主要来自各方的坚守岗位.

流程数据化与流程脚本化孰轻孰重

6代监控在立项之初,有一个选择层面的问题,就是 AI 的识别处理流程,是用数据来控制,还是用脚本来控制.这会直接影响 6代在未来的 AI 流程走向.

以 100 套子程序为基数来描述问题,也可以说 100 个函数.

数据控流程:这种模型是程序机制化,例如,点击菜单,进入各项子程序.或则程序在运行中,遇上某个状态机,进入子程序.当处理子程序基数到 100 个,会需要设计数据结构,编辑修改数据结构,驱动数据结构.这种程序范式,多出现在,系统参数,系统配置,web 应用程序中琳琅满目的参数,许多服务器的后台系统参数,在这些应用模型中,假如我们在 6代这样干,也许会做出数十个配置界面来生成和管理 6代的数据运行支持.

脚本控流程:脚本含义的表达数据和表达流程,无论脚本是否包含 if,for,goto.脚本的载体可以自设计,也可以 lua,xml,json,yaml,ini,txt.无论怎么设计脚本,底层逻辑:人对程序控制的表达.而脚本使用深入以后,会有许多词法,并且沾有许多编译原理的技术,因为脚本意义如此.

在做出选择中,我们的工作流程是按效用分析,在假想条件下评估了数据流程和脚本流程所带来的效用:最终我们选择了脚本流程,这也代表未来 6代在集成环节的工作方式.做出这种选择前,我们考虑了非常多的因素,从前期部署,到中期研发迭代,再到交付前的各种优化体系.

例如 AI 流程的脚本,我们是通过 excel 生成出来的,只需要录入一个 ip 表+一个监控文本说明+一个 AI 工作方式代号.之后,直接放到服务器群集去运行,在运行中,会出现大量你根本无法预料的问题,例如交换机,路由器带宽不够,cpu 拉不动,gpu 拉不动,内存不够,显存不够,这时候,脚本就发挥作用了:带宽不够让脚本停 100 路,开 100 路,分批负载,cpu 拉不动就把 cpu 程序扔给 gpu,如果 gpu 还是拉不动就开优化模式,如果优化还是不行,那就降分辨率,降帧率.

400 路规模的高清监控,对于 AI 服务器的要求非常变态,需要精确控制各种硬件利用率,以及,解决各种 IDC 故障,我们给出了组合优化模型,通过搭配已经跑出上千种优化效果.最终,当我们解决好 AI 服务器以后,我们发现,单日数据流量竟然高达 1PB,这太恐怖了,idc 的网络和存储系统几乎被我们用完到接近宕机瘫痪.我记得,idc 经常给甲方发微信,网络风暴又出现了,赶快救火

给 6 代监控做 AI 建模支持

数据+算法+算力=AI,这是 6 代监控的关键功能之一,而实现 AI 有两个大项目级别的技术方向

- 用平台解决 AI 监控运行:这步是做系统级开发,等同于一个小型操作系统,前面几节描述的脚本模型,数据库模型,这些都是平台的一部分.属于 AI 项目的 RunTime+APP 环节.
- 解决建模问题:从底层逻辑来看,这是一种对人力的合理运用,它几乎脱离于 IDE 工具,甚至脱离整个编程工具的范围.AI 是大部分人类长期努力攻关的方向,它不会短期冒泡就没了.建模行动需要有很多人,这些人可以外包公司程序员,研究院科学家,系统集成工作中的业务员,以及,酱油,这是无数人共同的项目.驱动人力建模是一种工程,或则说,生产工艺.而生产的根本在于工具,需要发明工具,再使用工具.全部工具都会因需而生.**我眼中 AI 建模,就是做工具+使用工具.**

我们所看到的通用模型,例如人脸检测器,车牌检测器,行人检测器(流量),这些都是在工具辅助下完成的建模.在非开放的特殊领域,例如交通,教育,特殊化建模会对工具有更大的需求:更需要工具,专业领域的数据没有共建资源,很多数据都是独有的.

数据+算法+算力=AI,在从数据->算力的生产过程,工具是它们的桥梁,另一方面 delphi,c#,java,web ui,这些都很适合做工具.

通过工具生产模型的流程

1. 确保数据到位:我们在 6 代通过监控下载+采集,收集了数十 TB 的视频数据,同时也在使用 aigc/openai 这类工具进行生成,但生成非常慢,另一方面,通过关键字搜抖音快手油管这也是一种快捷的数据来源.我们的数据都来自视频,单图的规格太小.
2. 生产模型:模型对数据的要求各有不同,场景模型偏大数据,动辄可以到数百 GB,检测器和语义分割模型偏精细标注,一般高收入的公司,技术员工资>10 万/月,他们会外包出去标注,然后自己来训练.专业领域的数据都是不开放的,根本没有公共资源可以用,都是自己来生产模型.生产工具全部因需而生,并且要用生产的方式直接脱离程序员领域,程序员总是喜欢自己写工具自己用,总是喜欢用批处理,爬虫,在真正的模型生产中,被工具催生的人类劳动力可以单日上万个检测器标注,或则单日清理数十 tb 的场景分类数据,合理的工具+人力=优于爬虫和三方开源数据的模型生产效率.再说一次,delphi 做工具是极优选择.
3. 测试和应用:我们对于模型的测试并没有基于 loss 和 excel 散点统计,我们全部放到平台实际跑测试,在项目最初期间我们还没有建设完成 6 代监控的平台,这时候我们是用模拟平台方式做了一个仿真 6 代运行的工具来做测试,在中后期,模型都是直接放平台来跑,既完成测试也完成应用.我们并不追求看起来高端漂亮,而是项目切实落地,造福于社会,同时带来经济效益.

AI 建模不可以直接让技术员折腾,技术无法脱离技术思维,这做不了应用模型,AI 需要数据+算法+算力相结合,超出了技术范围,技术思维只会计较有没有源码,自己能不能玩转源码.并且技术员和市场+资金之间一种天然交流屏障,这让大部分技术员无法驱动外部资源运作.而建模需要从资源角度去结合人力进行开展,即使 1 个人建模,也需要需要明确工作,通用模型不具技术竞争能力,正经搞 AI 都需自己建模.做 AI 建模会方向性的长期走流程,不可忽视对劳动资源的运用.这就是我们对于建模的观点.

完.