**人工智能的多功能定向改造和低成本智慧工业调度系统的实现**

东莞中学初中部初三（18）班曾弘毅 谈世豪

1. **研究起因**

从水蒸汽、电子工程技术和IT技术的应用开始，工业社会从工业1.0时代跨入了工业3.0时代。工业4.0的概念一经提出，就在欧洲乃至全球制造业引起了极大的关注和认同，美国、德国、日本、中国都纷纷提出了自己的工业4.0战略。这反映了全球范围内存在一项共识：制造业的发展正处于关键的十字路口，各国都想通过模式之变来获得制造业新格局里的领导地位。在暑假期间，我回到了位于旧工业区的老家，怀揣着一个“技术男”对工业4.0时代的好奇，我走访参观了老家的几个知名大工厂。结果，发现，我国工业发展不平衡性十分严重，相当多的企业还处于2.0到3.0阶段。

典型的问题如：在当地还是普遍地使用着传统的按需调用系统，工厂存在调料难、原材料生产点分布较散、工厂机械使用效率不高的问题（由某工厂提供的开销和贸易交易数据得出结论，见附件①），更要命的是传统的货物流的统计分析方法不能适应现代类型的交易所产生的数据量，这就造成了统计和分析时，计算结果存在大的瑕疵，工厂市场会出现“货源跟不上”“等候延误时机”等状况发生。但是，如果加以改造，就必须升级换代当前的智能分析系统，以改善实际境遇。但鉴于当地的经济情况，目前所能利用的高成本的智能工业系统，可以胜任适用解决问题，但成本过高，几经权宜就成为了“凑合着”的状态。

“管理出效益”，经济学基本理论如是告诉我们。于是，我决定着手创造一款新的系统，改变这一状态，让“货源”“物流”“订单”“生产”等要素或环节更为优化地对接好，弥合不必要的时间浪费，和因速度慢而造成的损失，以达成提速增效的目的。

与此同时，新系统的问世，可以在“待改进系统的目标工厂群体”中，能广泛地应用，让企业能用，好用，敢用，用出价值；让它在提高我国基础类型人工智能的技术水平的路途上，留下做出自己在技术层面关于“侧供给改革”的努力和贡献。

1. **内容和目标**

（1）建构一个新型低成本的“神经网络”系统，达到智能再升级的目的。

（2）优化算法，降低运算量，以实现系统的优越性能的逐代提升。

1. **研发过程**

**3.1查找资料**

1. **硬件资料**

从网上查找的资料来看，工业级工作站大约在1000元左右，而一个工业节点需要2560元，远远高于工厂可承担的成本。而树莓派微型电脑系统只需要60~120元就可以达到相同效果，在收集到更多资料后，我们决定使用基于树莓派和现有计算资源来构建这套系统。

1. **软件基础理论和实现原理**

本次的人工智能的灵感和思路，基于发布于2018年1月的《科学》。在该论文中，他们提到了一种模拟大脑推理的方案，引起了我们的注意。

神经网络在运行过程中，会形成一种回环型回路，使得数据来回在推理结构中流转，尝试“可能性”，形成“策略和决断”，然后形成最优解，从而实现类似人类而代替人类思考的想象力。这种方法在计算量上稍微偏高。为此，我们整合了第一代设计（见附件②）的成果，降低了20%的计算量以来实现这一方案。

**3.2设计方案**

我们了解到，树莓派自带了40针的GPIO口，使用mirco-USB进行供电，而工业机械通常使用KB-264协议，于是我们在网上搜找了相关的资料，结果表明，GPIO口可以兼容普通的KB-264协议（数字端口）。

在树莓派到货后，我们进行了软件编写，根据设备兼容度和计算能源消耗比，我们选择了Tensorflow和MATLAB平台进行编写。

**(1)**

**开发软件和解决问题**

我们遇到的第一个问题便是树莓派的内存大小问题，树莓派自带了800MB的内存来运行程序。这对于大多数运行在ARM框架下的程序是足够的。但对于单机训练人工智能来讲，这个内存还是偏小。在翻找系统手册时，发现了系统内置的虚拟内存功能。通过命令行对储存卡分区和启用后，内存升至2GB，可以完成人工智能的训练计算。

另一个问题便是CPU计算频率不足，根据官方的说法，CPU的默认频率为800MHz（普通手机为850MHz ~ 1.2GHz），超频后CPU的功耗会上升。为了找到一个合适的频率，我们经过实验（具体方法为：研究每升1Hz，CPU升温多少度），最终选定1.1GHz为日常计算频率。同时配备金属外壳，提升散热能力水平，同时防止静电损坏电路板。

在以上问题解决后，我们开始在虚拟机上编写代码。

由于Python并不支持导出可执行文件，我们额外用了C++来引导程序生成。由于在Linux平台和Windows平台下的命令是不同的。我们分别编写了对应引导程序。

在编写时，考虑到工厂不稳定的环境，在每回合训练完人工智能后，会刷新记录文件，记录下当前进度，以便在意外断电时能恢复进度。同时编写了一套工业紧急反应系统，尽可能减小损失。

同时本次的开发也采用了开放源代码的形式开发，已在GitHub上开源，可以访问[www.github.com/ggPascal/Recoder-AI](http://www.github.com/ggPascal/Recoder-AI)以来获得源代码。

1. **实验和调试**

我们在虚拟的工业环境下，测试了人工智能和工业反应系统

首先进行的是人工智能的人体识别测试，通过手机摄像头来采集数据。

在第一次识别中，成功识别出路过的人，并圈出人的画面。

第二次识别中，在仅看手部的情况下，成功识别出尝试靠近正在工作的数控机床的人员，并在画面中圈出，给予警告。

在某一设备断线时，工业紧急系统及时将信息返回至中控服务器，并切换P2P连接线路（S2P协议）

在大量设备断线时，工业系统进入P2P（无服务器模式）连接，启动TCP/IP防拥堵功能，同时开启WIFI热点，允许有权限的设备接入。

1. **修复错误并完善功能**

在软件调试时，我们发现人工智能会将某些异物识别成其它物体，并不处理异物入侵事件。在研究之后，我们决定在下一个阶段增加样本的覆盖类型，以来识别这种情况并作出反应。

在进行网络传输测试时，我们发现在TCP/IP下的IPv4协议地址似乎会出现数量不足，对此，我们更换成了IPv6协议以来应对这种情况。同时保留IPv4协议以来满足小型工厂低成本的要求。

在进行网络安全测试时，树莓派的内置软件似乎会有崩溃的迹象。目前尚不清楚是否为系统的内置安全规则所致。

（相关测试报告请看附录的报告部分，见附件③）

1. **后续开发和期望**

通过本次的研究，我们已经基本解决工业领域的人工智能低成本问题。但有些软件方面的问题，我们限于知识水平方面，暂时无法解决。同时在开发时，我们发现这套系统也适用于智慧学校的解决方案，有待我们继续开发和创新！

另外限于篇幅有限，技术不断更新等问题，本论文并未添加一些详细的参数和具体的解释，还请各位老师访问在Github上的论文分支以来追溯新的论文！

地址：https://github.com/ggPascal/Recoder-AI/tree/artical

1. **创新点和理论一览创新点**

（一） 主要创新点

* + - 1. 类人型神经网络：

是使用了CNN神经网络和FDN神经网络的组合型神经网络，可以降低60%的样本量

* + - 1. P2P订阅者型分布式计算模式

基于IBM的P2P订阅者模式协议开发，可最大化利用节点的计算资源执行任务

1. 理论摘要
2. 神经网络理论

首先，根据神经网络解析式：∑(Wi+bi)（W表示权重，b表示隐藏值），可以分析得出，W和B有正比例关系

所以当i越大，任务计算量也会越大

而CNN神经网络为：∑(Wi+Bi)/R （R表示上一回合的输出值）

因此，这个表达式存在一个递归关系

FDDN神经网络表达式为：∑(W(i-1)+bi)/∑(Ri+S(i-1))（S表示上一回合的权值）

于是建立以下关系

A=∑(Wi+Bi)/R

C=∑(W(i-1)+ Ai)/∑(Ri+S(i-1))

（C为输出值）

鉴于人脑有回环式机制，所以A、C均为可循环参数

同时根据交替熵公式：∑(Wi\*Bi)设定结束参数

所以又可得∑(Wi\*Bi)≤ES（结束参数）

且0 < ES < 1.0

原因是在现实环境下无法达到最大精确度

同时由于A、C变量会不断流转，因此设置变量G来限制流转次数

但由于人脑的信息基本上是无限循环的

所以0<G且G≈∞

并且当i = G 时，保存信息至ED(i)并清空A、C

END.

1. P2P订阅者型分布式算法

首先建立Docker硬件分享平台，并且启用Google TCP/IP防拥堵算法

在CPU贡献模式下可以发现，CPU会有峰值访问

于是设置参数：--max-use=40

观察TCP/IP连接，发现默认连接数值不足，添加参数: --tcp-ip-connects-auto

将连接上传至IBM订阅型算法，并加入以下参数：--allow-UDP –allow-ipv6

完成搭建。

1. **附录**
2. 树莓派的基本参数

|  |  |  |
| --- | --- | --- |
| 参数项目 | 具体数值 | 可参照物体 |
| 工作电压 | 5V | 手机充电电压 |
| 工作电流 | 1.4A | 略低于手机充电电流 |
| CPU工作频率 | 默认800MHz 超频后可达1.4GHz | 大部分安卓手机的CPU频率 |
| 内存大小 | 800MB 可以使用交换空间增加内存 | 手机的运行内存 |

附件①

请见项目的GitHub仓库中的数据分支：

https://github.com/ggPascal/Recoder-AI/tree/data%EF%BC%88%E8%AE%AD%E7%BB%83%E5%8F%82%E8%80%83%E8%B5%84%E6%96%99%EF%BC%89

附件②

请访问GitHub中的‘OLD’分支：

https://github.com/ggPascal/Recoder-AI/blob/OLD%EF%BC%88%E7%AC%AC%E4%B8%80%E4%BB%A3%E8%AE%BE%E8%AE%A1%EF%BC%89

附件③

**网络安全测试报告**

**2018.3.10**

测试对象：1.基于P2P的网络测试

1. LAN网络标准测试
2. NetBIOS劫持测试
3. WIFI端口监听测试

使用平台：Kali linux

测试者：曾弘毅

测试过程：

树莓派开启WIFI以后，尝试使用aireplay-ng执行掉线测试，5分钟之后截获WAP握手包，使用reaver基于WPS连接协议进行欺骗。在测试时，发现有反向连接请求，已被拒绝，疑似为防火墙的反向攻击连接。在13分钟后，WIFI密码被破译。

在网络内发现了P2P协议下的防御信息，明文报文，发送端口为：1654

进入网络后，启动Nmap扫描设备基础信息，包括系统版本，运行的服务和所在的端口，系统类型并尝试破解最高权限用户密码。

14分钟后，扫描到以下信息

树莓派暴露80HTTP端口，无法进行暴力破解

暴露1447 MySQL端口，破解得到最高权限用户，用户名：test 密码：cc000sst1678@22

暴露telnet端口，为动态端口，无法确定确切端口

暴露100 SSH端口，有加密，无法暴力破解，等待监听

3分钟后尝试数据库破库，向数据库强制植入0x00f，无反应。

1分钟后检测到用户尝试登入系统，拦截下密码和用户名:

用户名：root 密码：145678@@qq.kil

至此登入成功，开始伪造IP和物理地址，销毁日志

同时发现一部安卓手机，为HUAWEI-P7，确定为登录发起者

通过扫描得知，开放了TELNET连接

通过断线操作，拦截了用户的密码和用户名，登记入数据库

至此WIFI端口监听测试结束

3分钟后，开始LAN网络测试

在锁定MAC地址为00-C5-\*\*-90-CD的路由器后，发送假数据包（以回合形式增加数据量，每回合加200KB）

15分钟完成测试，测得上传1.4MB/S 下载1.3MB/S 最大承载量：4.43MB（双工时模式）

P2P测试开始，发送递归查询包

15秒后返回设备列表

NETBIOS劫持开始，使用在WIFI端口监听测试的数据进行对照

使用WIFIITE进行WIFI劫持，成功。

发送假广播，伪装成为路由器，失败

发送假广播，伪装成订阅服务器，因加密连接无法建立，失败

自此，所有测试结束。